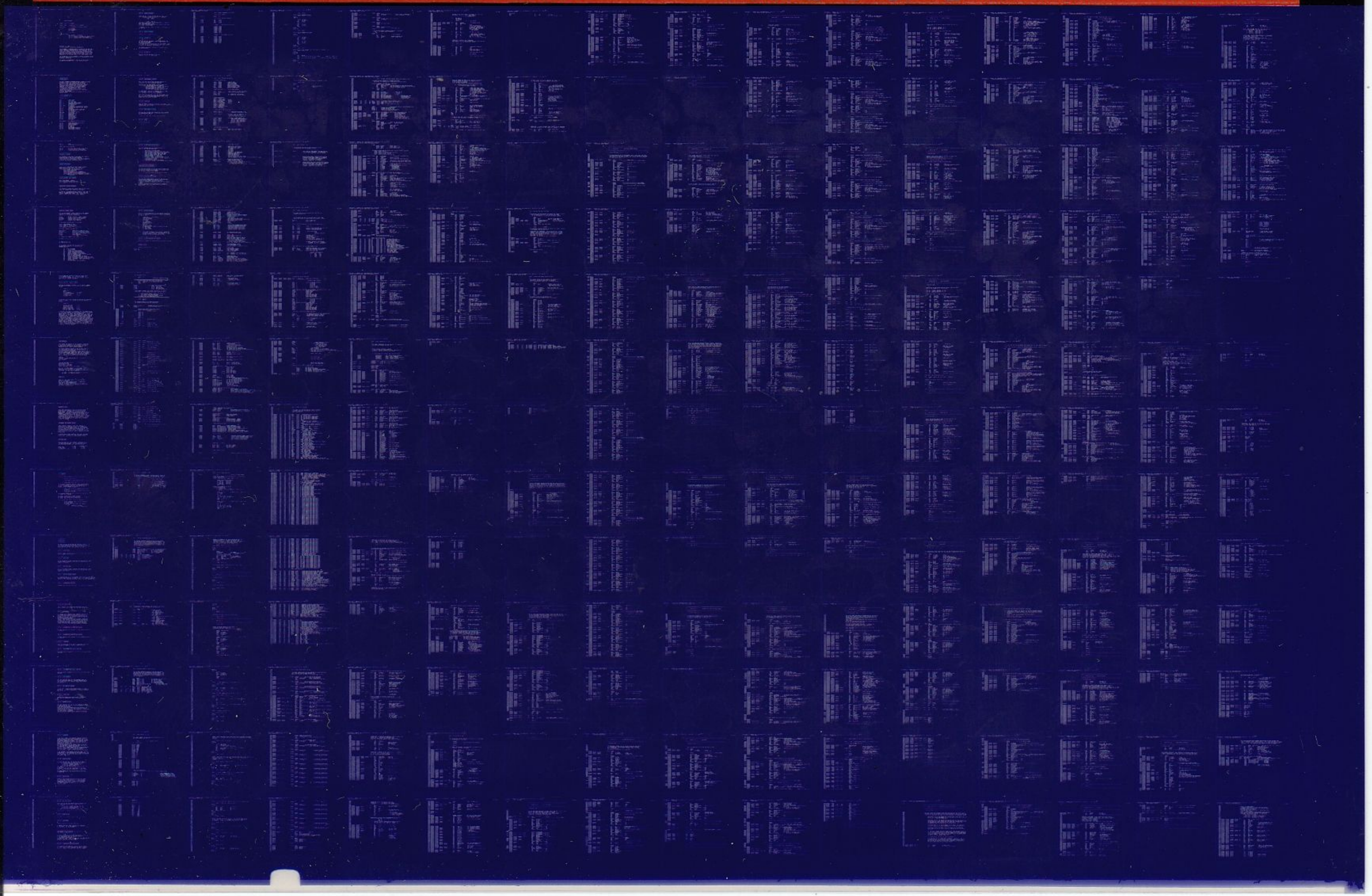
VSV11

VSV11 CSS DIAG
CVVSABO

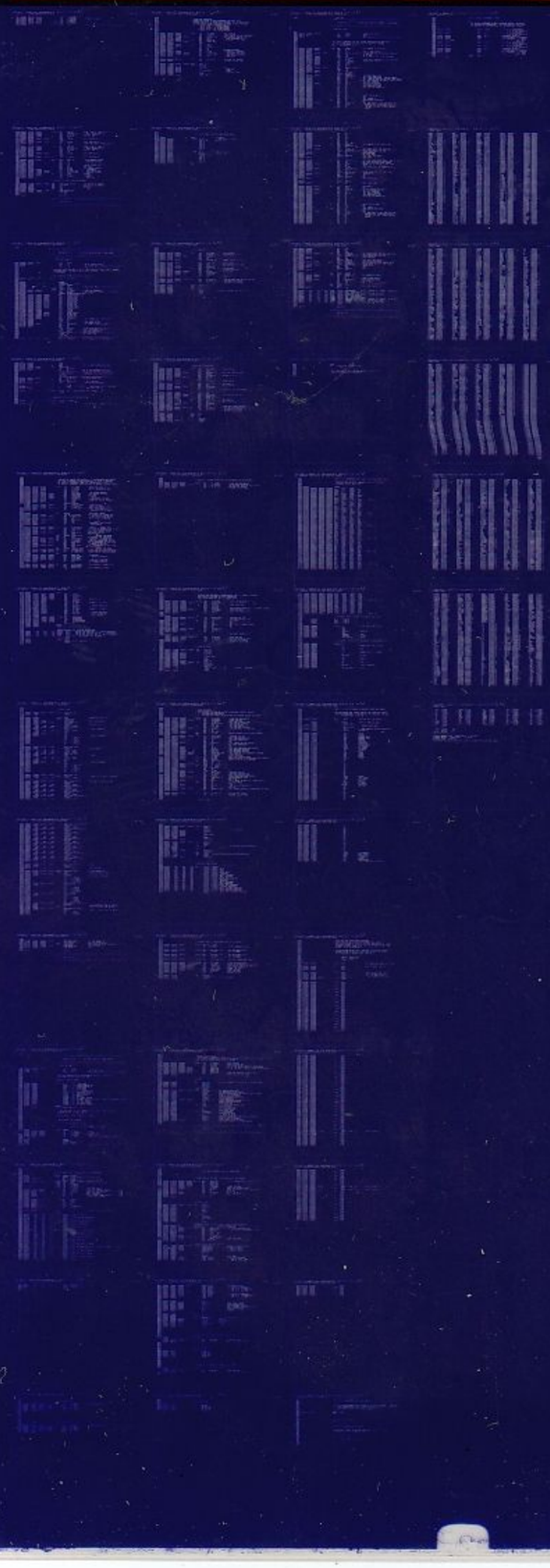
AH-E769B-MC
FICHE 1 OF 2



VSV11

VSV11 CSS DIAG
CVVSABO

AH-E769B-MC
FICHE 2 OF 2



.REM%

IDENTIFICATION

PRODUCT CODE: AC-E768B-MC

PRODUCT NAME: CVVSAB0 VSV11/VS11 DIAGNOSTIC

MAINTAINER: CSS PERIPHERALS & GRAPHICS GROUP

AUTHORS: DON MACOMBER
GUS PASQUANTONIO
BILL WEISKE

DATE: 9-JUL-1981

REVISION HISTORY:

BILL WEISKE 27-OCT-81
TEST 14 - BYPASS IF LSI AND .GT. 124K OF MEMORY.
TEST 17 - REPLACE CLRMEM WITH SETMEM = 0.
TEST 34 - INCREASE DELAY (WAS PROBLEM @50HZ)

1.0 PROGRAM ABSTRACT

THE VSV11 DIAGNOSTIC PROGRAM PROVIDES A SERIES OF TESTS
DESIGNED TO VERIFY THE INTEGRITY AND OPERABILITY OF THE
VSV11/VS11 Q-BUS VIDEO IMAGE PROCESSING SYSTEM.
SOME OF THE TEST SEQUENCES ARE VISUAL ONLY. HOWEVER ALL
TESTS WILL RUN WITH OR WITHOUT A DISPLAY MONITOR
CONNECTED TO THE SYSTEM. THE TESTS DESIGNATED
''DPU-ONLY'' WILL RUN WITHOUT THE ''DBUS'' CONNECTED
TO THE DISPLAY PROCESSOR MODULE (M7064).
THE PROGRAM WILL SUPPORT UP TO 16 VSV11 SYSTEMS.
HOWEVER, MULTI-UNIT CONFIGURATIONS ARE TESTED ONE AT
A TIME BY THE PROGRAM.
ANY LOGIC ERRORS ENCOUNTERED ARE REPORTED ON THE SYSTEM
CONSOLE DEVICE.

1.1 LIST OF HARDWARE TESTS

DPU ONLY TESTS:

TEST 1	STATIC RESET
TEST 2	RESET ONES
TEST 3	SOFT INIT ONES
TEST 4	REGISTERS UNIQUE ADDRESS
TEST 5	INCREMENTING REGISTERS
TEST 6	DPU START-STOP
TEST 7	DPU OPCODES
TEST 8	INCREMENTING HISTOGRAM BASE ADDRESS
TEST 9	INCREMENTING CHARACTER BASE ADDRESS
TEST 10	DJMS/DPOP
TEST 11	MAIN MEM MGT ACCESS (OVER 28K)
TEST 12	AUX MEM MGT ACCESS (OVER 28K)
TEST 13	STOP INTERUPT
TEST 14	DPU TIME-OUT INTERUPT
TEST 15	ERROR CODES
TEST 16	ADDRESS RELOCATE
TEST 17	CHARACTERS
TEST 18	ABSOLUTE POINTS
TEST 19	LONG VECTORS
TEST 20	RELATIVE POINTS
TEST 21	SHORT VECTORS
TEST 22	RUN-LENGTH

IMAGE MEMORY & SYNC GENERATOR TESTS & DISPLAYS:

TEST 23	CURSOR REGISTERS/SWITCH/MATCH
TEST 24	GRAPH-HISTOGRAM X
TEST 25	GRAPH-HISTOGRAM Y
TEST 26	BIT MAP (1)
TEST 27	BIT MAP (0)
TEST 28	IMAGE MEMORY CLEAR-SET
TEST 29	IMAGE MEMORY INTERLACE
TEST 30	IMAGE MEMORY PATTERNS

```
                        TEST 31         SPARE
                        TEST 32         SYSTEM VERIFICATION DISPLAY

                        STAND ALONE TESTS & ROUTINES:

                        TEST 33         JOY-STICK VERIFICATION (STAND ALONE)
                        TEST 34         SELECTED DISPLAYS (STAND ALONE)
                        TEST 35         SYSTEM CONFIGURATION (STAND ALONE)


                        --------------------
2.0                     STRUCTURE OF PROGRAM
                        --------------------

                        THIS PROGRAM IS STRUCTURED TO RUN UNDER THE DIAGNOSTIC
                        SUPERVISOR (REV D) AND THE XXDP+ MONITOR.
                        A BRIEF OVERVIEW OF OPERATING INSTRUCTIONS IS PROVIDED
                        IN SECTION 3.1 OF THIS DOCUMENT.  REFER TO CHAPTER 5 OF
                        THE XXDP+ USERS MANUAL FOR FURTHER DETAILS.


                        ----------------------
2.1                     HARDWARE REQUIREMENTS
                        ----------------------

                        PDP-11/LSI-11 PROCESSOR WITH 28K OR MORE OF MEMORY
                        CONSOLE DEVICE (LA30, LA36, VT50, VT100, ETC.)
                        XXDP+ LOAD DEVICE (RX, RP, RL, TM, DT, ETC.)
                        1 TO 8 VSV11 SYSTEMS, EACH CONSISTING OF:
                                M7064 DISPLAY PROCESSER
                                M7062 IMAGE MEMORY (1 TO 4 CHANNELS)
                                M7061 SYNC GENERATOR WITH CURSOR CONTROL
                                DISPLAY MONITOR, COLOR OR MONOCHROME (OPTIONAL).
                                DW11 UNIBUS TO LSI-11 BUS CONVERTER (OPTIONAL, VS11
                                        SYSTEMS ONLY).

                        --------------------------------
2.2                     RELATED DOCUMENTS AND STANDARDS
                        --------------------------------

                        XXDP+ USERS MANUAL ( CHQUSA )
                        VSV11 OPTION DESCRIPTION (YM-C183C)
                        VSV11 DIAGNOSTIC LISTING (SEC 7 OF THIS DOCUMENT).


                        --------------------------------
3.0                     LOADING AND STARTING PROCEDURES
                        --------------------------------

                        THIS PROGRAM IS LOADED AND STARTED FROM ANY XXDP+ MEDIA
                        USING THE STANDARD XXDP+ OPERATING PROCEDURES.

                        AT START UP, THE SUPERVISOR WILL IDENTIFY ITSELF AND THE
                        NAME OF THIS PROGRAM ON THE CONSOLE DEVICE, AND THEN
                        DISPLAY A COMMAND MODE PROMPT ( DR> ) WHICH INDICATES
                        READY TO ACCEPT ANY OF THE FOLLOWING COMMANDS.
```

3.1 SUPERVISOR COMMAND SUMMARY

THIS SECTION PRESENTS A BRIEF OVERVIEW OF THE COMMANDS
NECESSARY TO CONTROL THE OPERATION OF THIS DIAGNOSTIC
UNDER THE REV D DIAGNOSTIC SUPERVISOR.

STA(RT)	INITIAL START-UP -- BUILD P-TABLES.
RES(TART)	RESTART -- USE EXISTING P-TABLES.
CON(TINUE)	CONTINUE AFTER <^C> OR ERROR HALT.
PRO(CEED)	CONTINUE AFTER ERROR HALT.
EXI(T)	RETURN TO XXDP+ MONITOR.

THE FOLLOWING SWITCHES APPLY TO THE ABOVE:

/TESTS: <TEST NUMBERS TO RUN>
/PASS: <NUMBER OF PASSES TO RUN>
/FLAGS: <SEE FLAG LIST BELOW>
/EOP: <NUMBER OF PASSES 'TIL END-OF-PASS>

ADDITIONAL COMMANDS AVAILABLE ARE:

DRO(P)/UNIT:N	REMOVE UNIT N FROM TEST LIST
ADD/UNIT:N	ADD UNIT N (PREVIOUSLY DROPPED).
DIS(PLAY)/UNIT:N	PRINT UNITS P-TABLE AND STATUS.
PRI(NT)	PRINT STATISTICS (PER-UNIT STATUS).
ZFL(AGS)	CLEAR ALL FLAGS.
FLA(GS)	PRINT CURRENT FLAG SETTINGS.

THE GENERALIZED COMMAND STRING FORMAT IS:

COM(MAND)/SWITCH:VALUE/SWITCH:VALUE ... <CR>

3.2 RUN TIME OPTION FLAGS

THE FOLLOWING FLAGS ARE USED IN LIEU OF THE OLD
HARDWARE SWITCH REGISTER TO FURTHER DEFINE
PROGRAM BEHAVIOUR:

HOE	HALT ON ERROR
LOE	LOOP ON ERROR
IER	INHIBIT ALL ERROR REPORTS
IBE	INHIBIT BASIC ERROR REPORTS
IXR	INHIBIT EXTENDED ERROR REPORTS
PRI	SEND ALL REPORTS TO LINE PRINTER.
PNT	PRINT TEST NUMBERS (AND I.D'S) AS EXECUTED
BOE	GOOD OLD "BELL-ON-ERROR"
UAM	RUN IN "UNATTENDED" MODE (NO MANUAL)
ISR	INHIBIT STATISTICS (PER-UNIT STATUS EACH PASS)
IDU	INHIBIT "AUTO-DROP" (EXCEPT FOR NON-EXISTENT REG.)
ADR	EXECUTE "AUTO-DROP" (USER SUPPLIED CODE)
LOT	LOOP ON TEST
EVL	EVALUATE ERRORS (NOT IMPLEMENTED)

FLAG SETTINGS ARE ALTERED BY USING THE /FLAGS: SWITCH
IN ANY COMMAND STRING. I.E. STA/FLAG:IER:BOE <CR>
WILL START THE PROGRAM, RUN ALL TEST IN ORDER, INHIBIT
ERROR TYPEOUT, RING BELL ON ERROR.

3.3 INITIAL START-UP -- BUILD P-TABLES

WHENEVER THE PROGRAM IS STARTED, VIA THE STA(RT) COMMAND,
THE SUPERVISOR REQUESTS THE FOLLOWING P-TABLES PARAMETER
CHANGES:

```
        CHANGE HW (L) ?

        #UNITS (D)  ?
        UNIT 0
        DEVICE ADDRESS          (O) 172010 ?
        1ST INTERUPT VECTOR     (O) 320 ?
        INTERRUPT PRIORITY      (O) 4 ?
        LUT INSTALLED           (L) N ?
        FREQUENCY = 50HZ        (L) N ?
```

IN ADDITION, ON A START, RESTART OR CONTINUE THE SUPERVISOR
REQUESTS CHANGES TO THE SOFTWARE OPERATING PARAMETERS, AS
FOLLOWS:

```
        CHANGE SW?

        RUN DPU TESTS ONLY      (L) N ?
        LONG MEMORY TESTS       (L) N ?
        INHIBIT ITERATIONS      (L) N ?
        MANUFACTURING MODE      (L) N ?
        PER TEST ERROR LIMIT    (D) 25 ?
        PER UNIT ERROR LIMIT    (D) 200 ?
```

IF "RUN DPU TESTS ONLY?" IS ANSWERED "YES", NO COMMUNICATION
WITH THE IMAGE MEMORIES OR SYNC CHANNELS IS MADE; THE PROGRAM
CAN BE RUN WITH THE "DBUS" CABLE DISCONNECTED. IF "LONG
MEMORY TESTS" IS SELECTED, 15 DIFFERENT PIXEL DATA VALUES ARE
USED DURING TEST 28 (IMAGE MEMORY CLEAR-SET) RATHER THAN THE
STANDARD 5 VALUES. INHIBITING ITERATIONS CAUSES THE PROGRAM
TO RUN FASTER, EQUIVALENT TO THE FIRST PASS AFTER STARTUP.
IF "MANUFACTURING MODE" IS SELECTED, PREDEFINED PARAMETERS ARE
USED FOR THE IMAGE MEMORY AND SYNC CHANNEL CONFIGURATION DATA.
THE ERROR LIMITS DEFINE THE NUMBER OF ERRORS ALLOWED TO OCCUR
BEFORE A UNIT IS DROPPED (UNLESS THE /FLAG:IDU FLAG IS SET TO
INHIBIT DROPPING OF UNITS). FOR EXAMPLE, IF 25 ERRORS OCCUR
IN TEST 3 WHILE TESTING UNIT 2, UNIT 2 WILL BE DROPPED. OR,
IF UNIT 2 ACCUMULATES 200 ERRORS OVER THE COURSE OF SEVERAL TESTS
OR PASSES, IT WILL BE DROPPED.

4.0 ERROR REPORTING

LOGIC ERRORS ARE REPORTED BY THE DIAGNOSTIC SUPERVISOR
AS THEY OCCUR, HOWEVER, IT MUST BE NOTED THAT CAREFUL
OBSERVATION OF THE VARIOUS VISUALS FRAMES IS ALSO
REQUIRED TO VERIFY TOTAL SYSTEM PERFORMANCE.

EACH ERROR SIGNATURE CONSISTS OF ONE OR MORE LINES OF
TEXT, DESCRIBING THE ERROR, AND INCLUDES EXPECTED VS
RECEIVED DATA WHERE APPLICABLE.
THE FIRST LINE IS THE SUPERVISORS HEADER (BASIC), WHICH
INCLUDES PROGRAM NAME, ERROR TYPE (HRD, SFT, DEV FATAL,
OR SYS FATAL), ERROR NUMBER, TEST AND SUB-TEST NUMBERS,
AND PC. THE HEADER WILL NORMALLY BE FOLLOWED BY ONE OR
MORE (EXTENDED) LINES FURTHER IDENTIFYING THE ERROR.

EXAMPLES:

```
CVVSA DEV FTL ERR 00002 ON UNIT 00 TST 000 SUB 001 PC:_____
''BUS-INIT'' DIDN'T INITIALIZE DPU
UNIT 0 DROPPED
DR>

CVVSA HRD ERR 00801 ON UNIT 00 TST 008 SUB 001 PC:______
LONG VECTOR FAILURE
DXR,DYR EXP'D: 1776,0000
DXR,DYR REC'D: 0000,0000
ORIGIN: 0000,0000  DX,DY:041776,000000
```

NOTE THAT THE ERROR NUMBER IS IN THE FORMAT ''TTTEE''
WHERE EE IS THE E'TH ERROR CALL WITHIN TEST TTT.
WHEN ERRORS OCCUR OUTSIDE THE CONFINES OF A GIVEN TEST
(INITIALIZE, OR SOME GLOBAL SUBROUTINE), TTT = 0.

I.E. ERR 00002 = 2ND ERROR IN PROGRAM INITIALIZATION.
      ERR 00801 = 1ST ERROR IN TEST 8.

4.1 ERROR HALTS

ERROR HALTS ARE CONDITIONED BY THE HALT-ON-ERROR
SWITCH (/FLAG:HOE).
HALT IN THIS CONTEXT MEANS RETURN TO COMMAND MODE.
THERE ARE NO OTHER PROGRAM HALTS.
PRO(CEED) TO RESUME FROM THE POINT OF THE ERROR CALL.
CON(TINUE) TO RESTART AT THE BEGINING OF THE TEST IN
IN WHICH THE ERROR OCCURED.

4.2 DROPPING OF UNITS

UNDER SOME CONDITIONS OF FATAL OR EXCESSIVE ERRORS, A UNIT WILL BE "DROPPED" FROM THE LIST OF DEVICES BEING TESTED. FOR EXAMPLE, IF A DEVICE REGISTER IS NON-EXISTENT, THE UNIT IS DROPPED BEFORE ACTUAL TESTING COMMENCES. IN ADDITION, A UNIT WILL BE DROPPED IF 25 OR MORE ERRORS OCCUR WITHIN ANY ONE TEST (UNLESS THE /FLAGS:IDU SWITCH IS SET). THE DROPPED UNIT IS REPORTED AND APPEARS IN THE PER-UNIT STATISTICS REPORT. IN SOME CASES, THE ERROR COUNT IS NOT TESTED UNTIL THE END OF A TEST LOOP, SO MORE THAN 25 ERRORS CAN OCCUR BEFORE THE UNIT IS DROPPED.

5.0 PERFORMANCE AND PROGRESS REPORTS

NORMAL PROGRESS THROUGH A GIVEN TEST SEQUENCE IS INDICATED BY THE PRINTING OF EACH TEST'S NUMBER AND TITLE ON THE SYSTEM CONSOLE DEVICE (IF THE /FLAGS:PNT SWITCH IS USED), FOLLOWED BY AN END-PASS WHEN ALL TESTS HAVE BEEN EXECUTED ON ALL UNITS. IN ADDITION, IF THE /FLAGS:ISR SWITCH IS NOT SET, AND IF MULTIPLE UNITS ARE BEING TESTED, THE PER-UNIT STATUS IS PRINTED AT THE END OF EACH PASS.

IF RUNNING WITH ERRORS INHIBITED (/FLA:IER), AN ERROR COUNT IS KEPT FOR EACH TEST, AND DISPLAYED AT THE END OF THAT TEST. THERE ARE NO OTHER PROGRESS REPORTS.

5.1 EXECUTION TIMES

EXECUTION TIMES WILL VARY SOMEWHAT, DEPENDING ON CPU TYPE AND OPERATING MODE. THE FOLLOWING ARE TYPICAL EXECUTION TIMES OBSERVED ON A PDP-11/34 SYSTEM:

	1ST PASS	SUBSEQUENT PASSES
NORMAL MODE	4.5 MIN	8.6 MIN
DPU ONLY MODE	0.4 MIN	2.3 MIN
LONG MEMORY TEST MODE	5.6 MIN	9.7 MIN

6.0 TEST SUMMARIES

THERE ARE THREE CATAGORIES OF TESTS, AS FOLLOWS:

1. DPU ONLY TESTS (SEC. 6.1)
2. IMAGE MEMORY & SYNC GENERATOR TESTS & DISPLAYS (SEC. 6.2)
3. STAND-ALONE TESTS (SEC. 6.3)

THE FOLLOWING PARAGRAPHS PROVIDE A BRIEF DESCRIPTION OF THE TESTING SEQUENCE. REFER TO THE PROGRAM LISTING FOR DETAILS. DURING PROGRAM INITIALIZATION, CERTAIN MINIMAL TESTING IS DONE IN ORDER TO:

A. VERIFY THAT THE DEVICE ADDRESS FOR THIS UNIT IS VALID.
B. VERIFY THAT THIS UNIT IS PROPERLY INITIALIZED FOLLOWING A BUS-RESET.

THE FOLLOWING ERRORS MAY OCCUR OUTSIDE THE CONFINES OF THE NORMAL TEST SEQUENCE:

ERR 00000 UNEXPECTED DPU ERROR INTERRUPT.
ERR 00001 NON-EXISTANT DEVICE REGISTER. ABORT.
ERR 00002 BUS-INIT DIDN'T INITIALIZE DPU.

THE FOLLOWING NOTES APPLY TO ALL TEST SEQUENCES:

1. UNLESS OTHERWISE NOTED, TESTING CONTINUES AFTER ANY ERROR.
2. ALL UNSOLICITED KBD INPUTS ARE IGNORED EXCEPT FOR:
   - <^O> SUPPRESS TTY OUTPUT
   - <^C> HALT (RETURN TO COMMAND MODE)

6.1 THE DPU TESTS

THE TESTS IN THIS SECTION ARE THE "DPU ONLY" TESTS.
THEY ARE RUN WHENEVER THE PROGRAM IS RUN AND THE TEST
SWITCH DOES NOT EXCLUDE THEM. WHEN IN THE "DPU ONLY" MODE
THESE ARE THE ONLY TESTS THAT MAY (WILL) BE RUN.

6.1.1 TEST 1. STATIC RESET

EXECUTE A RESET INSTRUCTION AND CHECK EACH APPLICABLE REGISTER
FOR ITS CORRECT STATIC STATE.

6.1.2 TEST 2. RESET ONES

SET ALL REGISTERS TO ONES, CHECKING THAT ONES GOT SET, THEN
EXECUTE A RESET INSTRUCTION AND CHECK EACH REGISTER FOR ITS
CORRECT STATIC STATE.

6.1.3 TEST 3. SOFT INIT ONES

SET ALL REGISTERS TO ONES, CHECKING THAT ONES GOT SET, THEN
EXECUTE A SOFTWARE INITIALIZE AND CHECK EACH REGISTER FOR ITS
CORRECT STATIC STATE.

6.1.4 TEST 4. REGISTER UNIQUE ADDRESS

ALL REGISTERS ARE SET TO THEIR NORMAL STATIC STATE VIA SOFT INIT.
EACH APPLICABLE REGISTER, IN TURN, IS SET TO ITS CAPACITY FOR ONES.
THEN ALL OF THE OTHER REGISTERS ARE CHECKED THAT THEY REMAINED STATIC.

6.1.5 TEST 5. INCREMENTING REGISTERS

INCREMENT AND CHECK EACH REGISTER TO THE LIMIT OF ITS CAPABILITY.

6.1.6 TEST 6. DPU START-STOP

TEST 1 VERIFIES THAT THE DISPLAY PROCESSER (DPU) CAN
EXECUTE START, STOP, RESUME, NOP, AND JUMP INSTRUCTIONS.

6.1.7 TEST 7. DPU OPCODES

ALL OPCODES WHICH TRANSFER PIXEL INTENSITY DATA TO THE
DPU (100000 THRU 130000) ARE EXECUTED AND THE DSR IS
TESTED TO VERIFY THAT THE PIXEL DATA WAS RECEIVED AND
RETURNED BY THE DPU. ALL POSSIBLE VALUES OF PIXEL DATA
ARE PASSED WITH EACH OPCODE.
THEN ALL REMAINING OPCODES EXCEPT BIT MAPS (134000 AND
136000), JUMP (160000), AND STOP (172000) ARE EXECUTED
WITH BITS <9:0> OF THE OPCODE WORD SET TO ZERO AND WITH
NO FOLLOWING DATA WORD. IN THIS CONFIGURATION, EACH
OPCODE (INCLUDING THOSE UN-DEFINED) SHOULD APPEAR AS
A "NOP" AS FAR AS FINAL REGISTER CONTENT IS CONCERNED.

6.1.8 TEST 8. INCREMENTING HISTOGRAM BASE ADDRESS

SET BASE HISTOGRAM AND INCREMENT THE BASE ADDRESS REGISTER TO
ITS CAPACITY.

6.1.9 TEST 9. INCREMENTING CHARACTER BASE ADDRESS

SET BASE CHARACTER AND INCREMENT THE BASE ADDRESS REGISTER TO
ITS CAPACITY.

6.1.10 TEST 10. DJMS/DPOP

CHECK THE OPERATION OF THE JUMP TO SUBROUTINE INSTUCTION,
INCLUDING THE ENABLE BIT AND THE PCSAVE REGISTER.

6.1.11 TEST 11. MAIN MEMORY MGT ACCESS (OVER 28K)

CHECK MAIN MEMORY MANAGEMENT ACCESS FOR ALL AVAILABLE MEMORY
OVER 28K.

6.1.12 TEST 12. AUX MEMORY MGT ACCESS (OVER 28K)

CHECK AUX MEMORY MANAGEMENT ACCESS FOR ALL AVAILABLE MEMORY
OVER 28K.

6.1.13 TEST 13. STOP INTERRUPT

THIS TEST VERIFIES THAT THE STOP INTERRUPT LOGIC IN
THE DPU FUNCTIONS CORRECTLY. VARIOUS COMBINATIONS OF
STOP INSTRUCTIONS WITH INTERRUPT ENABLED AND DISABLED
ARE EXECUTED.

6.1.14 TEST 14. DPU TIME-OUT INTERUPT

IN THIS TEST, A NON-EXISTANT MEMORY ADDRESS IS PASSED TO
THE DPU PC. IT SHOULD RESPOND WITH A TIME-OUT INTERRUPT
BACK TO THE CPU.

6.1.15 TEST 15. ERROR CODES

GENERATE EACH ERROR CONDITION POSSIBLE (TO SOFTWARE)
AND CHECK FOR APPROPRIATE ERROR CODE GENERATION.

6.1.16 TEST 16. ADDRESS RELOCATE

THIS TEST VERIFIES THAT THE DPU CAN CALCULATE AN 18 BIT
PC FROM AN INITIAL 16 BIT PC, PLUS THE CONTENTS OF THE
12 BIT RELOCATE REGISTER.
SINCE THE 2 HIGH ORDER ADDRESS BITS ARE INVISIBLE, THE
TEST RELIES ON THE FACT THAT ANY CALCULATED PC WHICH
EXCEEDS 18 BITS IN LENGTH WILL BE TRUNCATED TO 18 BITS
(I.E. WRAP AROUND TO 000000).
THE TEST USES ALL COMBINATIONS OF RELOCATE FACTOR AND
INITIAL PC THAT YIELD A FINAL PC WITHIN THE TEST BODY.

6.1.17 TEST 17. CHARACTERS

IN THE VSV11, THERE IS NO HARDWARE CHARACTER GENERATOR.
INSTEAD, CHARACTER MODE CAUSES A SUBROUTINE CALL TO A
SUBPICTURE DISPLAY FILE WHICH CONTAINS THE CODE TO DRAW
THE DESIRED CHARACTER OR SYMBOL. THE CHARACTER SUB-PIX
ADDRESS IS OBTAINED BY USING THE ASCII CHAR CODE TO
INDEX INTO AN ADDRESS TABLE WHICH CONTAINS THE STARTING
ADDRESSES OF THE CHAR SET.
THE STARTING ADDRESS OF THE ADDRESS TABLE IS SET WITH
THE "SET CHAR BASE" INSTRUCTION, AND EACH CHAR SUB-PIX
IS TERMINATED WITH A "DPOP" INSTRUCTION.
THIS TEST USES 1 COMMON SUBROUTINE AS A PSUEDO-CHARACTER
AND SETS A UNIQUE BASE ADDRESS FOR EACH CHARACTER CODE
SUCH THAT: BASE + 2(CODE) = SUBROUTINE ADDRESS.
THE FULL ASCII SET (000 - 177) IS TESTED IN THIS MANNER.

IF THE PROGRAM IS NOT RUNNING IN DPU-ONLY MODE, THE FULL
CHARACTER SET IS THEN DISPLAYED ON THE SCREEN TWICE: IN
THE TOP DISPLAY, THE CHARACTERS ARE SITUATED ON EVEN-NUM-
BERED SCAN LINES, WHILE IN THE BOTTOM DISPLAY THEY ARE ON
ODD-NUMBERED SCAN LINES.

6.1.18 TEST 18. ABSOLUTE POINTS

THIS TEST VERIFIES THAT THE DPU CAN EXECUTE ABSOLUTE
POINT MODE INSTRUCTIONS UTILIZING THE FULL RANGE OF X/Y.
        (1) PLOT ALL X POINTS, HOLDING Y AT 0.
        (2) PLOT ALL Y POINTS, HOLDING X AT 0.
        (3) PLOT ALL POINTS WHERE X = Y.
THE X AND Y POSITION REGISTERS ARE TESTED AS EACH POINT
IS EXECUTED.
BY DEFAULT, PIXEL DATA ARE TRANSFERRED TO THE IMAGE
MEMORY, AND DISPLAYED ON THE MONITOR (IF THERE IS ONE).

THE IMAGE MEMORY AND VIDEO DISPLAY MAY BE INHIBITED
BY RESPONDING <YES> TO THE "RUN DPU TESTS ONLY" QUERY
AT START TIME.

6.1.19 TEST 19. LONG VECTORS

GENERATES LONG VECTORS WITH POSITIVE DELTA X/Y FROM
X,Y = 0,0 TO ALL POINTS WHERE X = Y (FANS DIAGONALLY
FROM BOTTOM TO LEFT EDGE). THEN A SIMILAR PATTERN
USING NEGATVE DELTA X/Y STARTING AT X,Y = MAX,MAX.
THE X AND Y POSITION REGISTERS ARE TESTED AS EACH
VECTOR IS DRAWN.

VIDEO DISPLAY IS INHIBITED IF "DPU ONLY" WAS SPECIFIED.

643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696

6.1.20 TEST 20. RELATIVE POINTS

VERIFY THAT THE DPU CAN GENERATE RELATIVE POINTS FROM
X,Y = 1000,1000 USING THE FULL RANGE OF DELTA X/Y
(-76 TO +76)

(1) PLOT ALL X POINTS, HOLDING Y AT -76.
(2) PLOT ALL Y POINTS, HOLDING X AT -76.
(3) PLOT ALL POINTS WHERE X = Y.

THE X AND Y POSITION IS VERIFIED AS EACH POINT IS DRAWN.

VIDEO DISPLAY IS INHIBITED IN "DPU ONLY" MODE AS BEFORE.

6.1.21 TEST 21. SHORT VECTORS

SIMILAR TO TEST 10 ABOVE, EXCEPT USING SHORT VECTORS.

6.1.22 TEST 22. RUN LENGTH

CHECK ALL ASPECTS OF THE RUN LENGTH MODE.

IF RUNNING "DPU ONLY" MODE, YOU WILL SEE "END-PASS"
AT THIS POINT. ALL OF THE REMAINING TESTS REQUIRE
THE SERVICES OF THE IMAGE MEMORY.

6.2 IMAGE MEMORY & SYNC GENERATOR TESTS & DISPLAYS

THE TESTS & DISPLAYS IN THIS SECTION REQUIRE THE SERVICES
OF THE IMAGE MEMORY AND SYNC GENERATOR MODULES.
IF THE PROGRAM IS BEING RUN IN
THE "DPU ONLY" MODE, THESE TESTS MAY NOT (WILL NOT) BE RUN.
OTHERWISE, THEY WILL BE RUN IN SEQUENCE AS LONG AS THEY
ARE NOT INHIBITED BY THE TEST SWITCH.

6.1.23 TEST 23. CURSOR REGISTERS/SWITCH/MATCH

THE CURSOR POSITION REGISTERS, JOYSTICK STATUS ENABLES,
THE "SOFT" SWITCH, SWITCH INTERRUPT AND MATCH INTERRUPT
ARE TESTED FOR EACH AVAILABLE SYNC GENERATOR CHANNEL.

6.2.1 TEST 24. GRAPH/HISTOGRAM X

GRAPH PLOT ALL X POINTS BETWEEN 200 AND 1600 (IN 30
UNIT INCREMENTS) USING ALL VALUES OF Y INCREMENT.
I.E. X 200, Y INCR 2
X 230, Y INCR 4
:
:
X 1600, Y INCR 76
THEN REPLOT EACH POINT IN HISTOGRAM MODE TO A RELATIVE
BASE LINE WHERE BASE = X-20.
THE X AND Y POSITION REGISTERS ARE TESTED AS EACH POINT
IS PLOTTED (IN BOTH MODES).

IF RUNNING "DPU ONLY" MODE, THIS AND ALL FOLLOWING TESTS
ARE NOT RUN.

6.2.2 TEST 25. GRAPH/HISTOGRAM Y

SAME AS TEST 11, EXCEPT PLOT IN Y.

6.2.3 TEST 26. BIT MAP (1)

VERIFIES THAT BIT MAP MODE 1 FUNCTIONS CORRECTLY FOR
ALL POSSIBLE RUN LENGTHS (0 TO 511.) AND IN BOTH 4 BIT
AND 8 BIT PIXEL MODES.
A 256 WORD BUFFER IS FILLED WITH THE VALUE 000377, WHICH
WILL YIELD A 2 ON, 2 OFF PATTERN OF 4 BIT PIXELS AND/OR
A 1 ON, 1 OFF PATTERN OF 8 BIT PIXELS.

THE X AND Y POSITION IS VERIFIED AT THE END OF EACH
BIT MAP OPERATION.

6.2.4 TEST 27. BIT MAP (0)

VERIFIES THAT BIT MAP MODE 0 FUNCTIONS
CORRECTLY IN ALL IT'S VARIATIONS.

6.2.5 TEST 28. IMAGE MEMORY CLEAR/SET

NOTE: THIS TEST AND THE ONE FOLLOWING REQUIRE THE SERVICES OF THE PIXEL-READ-BACK FUNCTION.

VERIFIES THAT THE IMAGE MEMORY "CLEAR TO ZERO" AND "SET TO DATA" INSTRUCTIONS FUNCTION CORRECTLY.

(1) THE IMAGE MEMORY IS "CLEARED" AND SELECTED PIXELS ARE READ-BACK AND VERIFIED.
(2) THE IMAGE MEMORY IS "SET TO ALL 1'S" AND SELECTED PIXELS READ-BACK AND VERIFIED.

IN "SHORT MODE", TEST ALL X ADDRESSES FOR Y = 0 AND 2 (EVEN VS ODD), THEN TEST ALL Y ADDRESSES FOR X = 0.
IN "LONG MODE", TEST ALL PIXEL ADDRESSES.

NOTE: PART 2 OF THIS TEST FILLS THE IMAGE MEMORY WITH ALL 1'S. THE VIEWING AREA SOULD BE ALL WHITE -- ANY BLANK SPOTS ARE EITHER BAD IMAGE MEMORY LOCATIONS OR BURNED SPOTS ON THE CRT PHOSPHOR.

6.2.6 TEST 29. INTERLACE

CHECK FOR MEMORY INTERLACE BY WRITING A HORIZONTAL, EVEN NUMBERED LINE, AND CHECKING THAT BACK THE "LINE" ABOVE THE ONE WRITTEN READS BACK AS ZERO.

6.2.7 TEST 30. IMAGE MEMORY PATTERNS

VERIFIES THAT A GIVEN PIXEL CAN BE SET TO ALL POSSIBLE BIT COMBINATIONS, AND THAT THE READ-BACK FUNCTION CAN CORRECTLY READ THAT DATA.

"SHORT" AND "LONG" MODE OPTIONS APPLY AS BEFORE.

6.2.8 TEST 31. SPARE

RUNS AS A BLANK TEST

6.2.9 TEST 32. SYSTEM VERIFICATION DISPLAY

THIS TEST PROVIDES ONE TEST PATTERN IN WHICH ALL
GRAPHICS FUNCTIONS ARE EXERCISED:

1. CHARACTERS AT BOTTOM CENTER.
2. EDGE CLIPPING AT LOWER LEFT AND UPPER RIGHT.
3. THREE OCTAGONS USING LONG VECTORS (LARGE ONE), SHORT VECTORS (2 SMALL ONES), AND SOME RELATIVE POINTS INSIDE THE SMALL ONES.
4. GRAPH-HISTOGRAM Y AT UPPER LEFT.
5. GRAPH-HISTOGRAM X AT LOWER RIGHT.
6. BIT MAPS -- MODE 1 AT BOTTOM, MODE 0 AT TOP.
7. A CIRCULAR VECTOR SWEEP AT CENTER, SHOWING 16 COLOR/INTENSITY LEVELS.
8. A PERIMETER OUTLINE OF THE DISPLAY.
9. STATIC FOR 2 SECONDS, THEN BLINKING FOR 2 SECONDS.

6.3 STAND-ALONE TESTS & ROUTINES

THE TESTS AND ROUTINES IN THIS SECTION ARE STAND-ALONE.
THEY ARE NOT INCLUDED IN THE NORMAL TEST SEQUENCE AND
MUST BE RUN BY UNIQUE START COMMAND

IE. START/TEST:NN<CR> OR RESTART/TEST:NN<CR>

6.3.1 TEST 33. JOY-STICK VERIFICATION

THIS IS A TEST FRAME FOR THE JOY-STICK, AND INCLUDES
3 NESTED BOXES(CO-ORDINATES LABELED), AND AN IN-LINE
READ-OUT OF THE CURRENT JOY-STICK POSITION.
A ''SWITCH'' INTERRUPT IS MARKED BY AN ''X'' AT THE CURRENT
CO-ORDS.
A ''MATCH'' INTERRUPT (ON THE BOXES ONLY) IS MARKED BY
AN X AS ABOVE, AND BY THE WORD 'MATCH'' DISPLAYED AT
UPPER RIGHT.

```
844
845                                6.3.2   TEST 34.  SELECTED DISPLAYS
846
847
848                                        THIS IS A PATTERN GENERATOR FOR COLOR MONITOR CONVERGENCE
849                                        SET-UP.  IT INCLUDES THE FOLLOWING PATTERNS SELECTABLE BY
850                                        THE USER AT WILL.
851
852                                        0       "TYPE THIS"     (ON CONSOLE)
853                                        1       TURN ON BLINKING
854                                        2       TURN OFF BLINKING
855                                        3       COLOR BARS
856                                        4       7 X 7 DOTS
857                                        5       7X7 CROSSHATCH
858                                        6       PERIMETER OUTLINE
859                                        7       BASIC COLOR ID
860                                        8       GUNS ID
861                                        9       ALL WHITE SCREEN
862                                        10      ALTERNATING WHITE SCREEN & PERIMETER OUTLINE
863                                        11      ALL RED SCREEN
864                                        12      ALL BLUE SCREEN
865                                        13      ALL GREEN SCREEN
866
867
868                                        NOTE:   SELECTIONS 4 THROUGH 6 ALSO INCLUDE A PERIMETER
869                                                OUTLINE OF THE SCREEN, IN WHICH EACH SIDE OF THE
870                                                OUTLINE INCLUDES 3 'DOTS".  SELECTIONS 3, 7 AND 8
871                                                INCLUDE A SOLID WHITE PERIMETER OUTLINE.
872
873
874
875                                6.3.3   TEST 35.   SYSTEM CONFIGURATION TYPE-OUT
876
877
878                                        PRINT ON THE CONSOLE THE CONFIGURATION OF THE VSV11 SYSTEM.
879
880
881
882                                7.0     PROGRAM LISTING
883
884
885                                        PROGRAM LISTING FOLLOWS:
886                                                                                              %
900                                .NLIST  BEX
901                                .DSABL  GBL
```

```
                               .SBTTL  PROGRAM HEADER

                                       .MCALL   SVC
 000000                                SVC                          ; INITIALIZE SUPERVISOR MACROS

                               ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
                               ; IF STRUCTURED MACROS ARE TO BE USED, YOU MUST ADD
                               ;       .REQ     SPMAC.SML (OR THE CURRENT EQUIVALENT)
                               ;       .MCALL   STRUCT
                               ;       STRUCT
                               ;
          000001               SVCGBL= 1                  ; LIST GLOBAL TAGS AT RIGHT MARGIN.
          000001               SVCTST= 1                  ;       DITTO   TEST TAGS.
          000001               SVCSUB= 1                  ;       DITTO   SUBTEST TAGS.
          000001               SVCTAG= 1                  ;       DITTO   ANY OTHER TAGS.
          000001               SVCINS= 1                  ;       DITTO   INSTRUCTIONS AND DATA.
                               ;
                               ; THESE SYMBOLS CONTROL THE LISTING FIELD OF ALL SVC MACRO
                               ; EXPANSIONS.  YOU MAY CHANGE THEM AT ANY TIME OR PLACE.
                               ;
                               ;        1 = RIGHT-JUSTIFY (MAKES IT EASY TO DISTINGUISH
                               ;            SVC'S MACRO CODE FROM YOUR OWN).
                               ;        0 = LEFT-JUSTIFY (ALIGN IN A NORMAL FASHION).
                               ;       -1 = DON'T LIST THE EXPANSIONS AT ALL.
                               ;%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

                                       .ENABL   ABS,AMA
          002000                       .=       2000
                               ;++
                               ; THE PROGRAM HEADER IS THE INTERFACE BETWEEN
                               ; THE DIAGNOSTIC PROGRAM AND THE SUPERVISOR.
                               ;--

          174561               PRGSIZ= <L$LAST>/16.       ; PROGRAM SIZE IN 1/8 K UNITS (OCTAL).
          000000               SVCGBL= 0                  ; ALIGN THE HEADER STUFF.
          000000               SVCINS= 0

 002000                                POINTER BGNSW,BGNSFT,BGNAU,BGNDU,BGNRPT
 002000                                HEADER  CVVSA,A,0,655.,0
 002000                        L$NAME::                   ;DIAGNOSTIC NAME
 002000      103                       .ASCII /C/
 002001      126                       .ASCII /V/
 002002      126                       .ASCII /V/
 002003      123                       .ASCII /S/
 002004      101                       .ASCII /A/
 002005      000                       .BYTE    0
 002006      000                       .BYTE    0
 002007      000                       .BYTE    0
 002010                        L$REV::                    ;REVISION LEVEL
 002010      101                       .ASCII   /A/
 002011                        L$DEPO::                   ;0
 002011      060                       .ASCII   /0/
 002012                        L$UNIT::                   ;NUMBER OF UNITS
 002012   000000                       .WORD    0
 002014                        L$TIML::                   ;LONGEST TEST TIME
 002014   001217                       .WORD    655.
 002016                        L$HPCP::                   ;POINTER TO H.W. QUES.
```

```
002016  002240                  .WORD   L$HARD
002020                  L$SPCP::                ;POINTER TO S.W. QUES.
002020  002522                  .WORD   L$SOFT
002022                  L$HPTP::                ;PTR. TO DEF. H.W. PTABLE
002022  002224                  .WORD   L$HW
002024                  L$SPTP::                ;PTR. TO S.W. PTABLE
002024  002502                  .WORD   L$SW
002026                  L$LADP::                ;DIAG. END ADDRESS
002026  113404                  .WORD   L$LAST
002030                  L$STA::                 ;RESERVED FOR APT STATS
002030  000000                  .WORD   0
002032                  L$CO::
002032  000000                  .WORD   0
002034                  L$DTYP::                ;DIAGNOSTIC TYPE
002034  000000                  .WORD   0
002036                  L$APT::                 ;APT EXPANSION
002036  000000                  .WORD   0
002040                  L$DTP::                 ;PTR. TO DISPATCH TABLE
002040  107114                  .WORD   L$DISPATCH
002042                  L$PRIO::                ;DIAGNOSTIC RUN PRIORITY
002042  000000                  .WORD   0
002044                  L$ENVI::                ;FLAGS DESCRIBE HOW IT WAS SETUP
002044  000000                  .WORD   0
002046                  L$EXP1::                ;EXPANSION WORD
002046  000000                  .WORD   0
002050                  L$MREV::                ;SVC REV AND EDIT #
002050     003                  .BYTE   C$REVISION
002051     003                  .BYTE   C$EDIT
002052                  L$EF::                  ;DIAG. EVENT FLAGS
002052  000000                  .WORD   0
002054  000000                  .WORD   0
002056                  L$SPC::
002056  000000                  .WORD   0
002060                  L$DEVP::                ; POINTER TO DEVICE TYPE LIST
002060  002162                  .WORD   L$DVTYP
002062                  L$REPP::                ;PTR. TO REPORT CODE
002062  033272                  .WORD   L$RPT
002064                  L$EXP4::
002064  000000                  .WORD   0
002066                  L$EXP5::
002066  000000                  .WORD   0
002070                  L$AUT::                 ;PTR. TO ADD UNIT CODE
002070  033012                  .WORD   L$AU
002072                  L$DUT::                 ;PTR. TO DROP UNIT CODE
002072  033110                  .WORD   L$DU
002074                  L$LUN::                 ;LUN FOR EXERCISERS TO FILL
002074  000000                  .WORD   0
002076                  L$DESP::                ;POINTER TO DIAG. DESCRIPTION
002076  002122                  .WORD   L$DESC
002100                  L$LOAD::                ;GENERATE SPECIAL AUTOLOAD EMT
002100  104035                  EMT     E$LOAD
002102                  L$ETP::                 ;POINTER TO ERRTBL
002102  000000                  .WORD   0
002104                  L$ICP::                 ;PTR. TO INIT CODE
002104  032014                  .WORD   L$INIT
002106                  L$CCP::                 ;PTR. TO CLEAN-UP CODE
002106  033220                  .WORD   L$CLEAN
```

```
        002110                          L$ACP::                 ;PTR. TO AUTO CODE
        002110  033216                         .WORD   L$AUTO
        002112                          L$PRT::                 ;PTR. TO PROTECT TABLE
        002112  032004                         .WORD   L$PROT
        002114                          L$TEST::                ;TEST NUMBER
        002114  000000                         .WORD   0
        002116                          L$DLY::                 ;DELAY COUNT
        002116  000000                         .WORD   0
        002120                          L$HIME::                ;PTR. TO HIGH MEM
        002120  000000                         .WORD   0
    942 002122                                  DESCRIPT <**** VSV11-VS11 DIAGNOSTIC ****>
        002122                          L$DESC::
        002122    052     052     052           .ASCIZ  /**** VSV11-VS11 DIAGNOSTIC ****/
                                                .EVEN
    943 002162                                  DEVTYP  <VSV11-VS11, AND DISPLAY MONITOR>
        002162                          L$DVTYP::
        002162    126     123     126           .ASCIZ  /VSV11-VS11, AND DISPLAY MONITOR/
                                                .EVEN
    944
    945         000001                  SVCGBL= 1               ; SHOVE EVERYTHING BACK TO THE RIGHT.
    946         000001                  SVCINS= 1
    947         177777                  SVCGBL=-1               ; KILL ALL SVC STUFF.
    948         177777                  SVCTST=-1
    949         177777                  SVCSUB=-1
    950         177777                  SVCTAG=-1
    951         177777                  SVCINS=-1
```

```
                                .SBTTL  DEFAULT HARDWARE P-TABLE

                                ;++
                                ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
                                ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
                                ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
                                ;--
002222                                  BGNHW   DFPTBL          ;DEFAULT HARD-P-TABLE

002224  172010                          .WORD   172010          ; 1ST (OF 4) REGISTER(S).
                                                                ;       ( 7 IF LUT INSTALLED ).
002226  000320                          .WORD   320             ; 1ST (OF 4) VECTOR(S).
                                                                ;       ( 5 IF LUT INSTALLED ).
002230  000200                          .WORD   PRI04           ; INTERRUPT PRIORITY.
002232  000000                          .WORD   0               ; LUT AVAILABLE (IF NONZERO)
002234  000000                          .WORD   0               ; SYNC FREQUENCY: 0 = 60HZ
                                                                ;                 NZ = 50HZ
002236                                  ENDHW
```

```
                                .SBTTL  HARDWARE PARAMETER CODING SECTION

                                ;++
                                ; THE HARDWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--
002236                                  BGNHRD

                                GPRMA   HPM1,0,0,160000,177776,YES ;GET 1ST REGISTER ADDRESS.
002240                          GPRMA   HPM2,2,0,0,776,YES      ;GET 1ST VECTOR ADDRESS.
002250                          GPRMD   HPM3,4,0,340,0,7,YES    ;GET INTERRUPT PRIORITY.
002260                          GPRML   HPM4,6,-1,YES           ;ASK IF LUT AVAILABLE
002272                          GPRML   HPM5,10,-1,YES          ;ASK IF FREQUENCY = 50 HZ.
002300                          EXIT HRD
002306  104  105  126  HPM1:   .ASCIZ  /DEVICE ADDRESS         /
002310  061  123  124  HPM2:   .ASCIZ  /1ST INTERRUPT VECTOR   /
002340  111  116  124  HPM3:   .ASCIZ  /INTERRUPT PRIORITY     /
002370  114  125  124  HPM4:   .ASCIZ  /LUT INSTALLED          /
002420  106  122  105  HPM5:   .ASCIZ  /FREQUENCY = 50HZ       /
                                .EVEN
002450                          ENDHRD
```

```
                                .SBTTL  SOFTWARE P-TABLE

                                ;++
                                ; THE SOFTWARE P-TABLE CONTAINS THE VALUES OF THE PROGRAM
                                ; PARAMETERS THAT CAN BE CHANGED BY THE OPERATOR.
                                ;--
002500                                  BGNSW   SFPTBL
002502  000000                DPUMOD:: .WORD  0               ; DPU/MEM TEST MODE...
                                                                ;... 0 = RUN ALL...
                                                                ;...NZ = RUN DPU ONLY.
002504  000000                IMTMOD:: .WORD  0               ; IMAGE MEMORY TEST MODE...
                                                                ;... 0 = SHORT TEST...
                                                                ;...NZ = LONG TEST.
002506  000000                TIDFLG:: .WORD  0               ; TEST ID'S...
                                                                ; ... 0 = TYPE TEST ID AT EACH TTET START...
                                                                ; ...NZ = DON'T.
002510  000000                NOITS:: .WORD   0               ; INHIBIT ITERATION OPTION.
                                                                ; ... 0 = ITERATE.
                                                                ; ...NZ = INHIBIT ITERATE.
002512  000000                MFGFLG:: .WORD  0               ; MANUFACTURING TEST MODE...
                                                                ;... 0 = USER/FIELD SVC...
                                                                ;...NZ = MANUFACTURING.
002514  000031                LERRMAX::       .WORD   25.     ; LOCAL (PER TEST) ERROR LIMIT
002516  000310                GERRMAX::       .WORD   200.    ; GLOBAL (PER UNIT) ERROR LIMIT
002520                                  ENDSW
```

```
                                .SBTTL  SOFTWARE PARAMETER CODING SECTION

                                ;++
                                ; THE SOFTWARE PARAMETER CODING SECTION CONTAINS MACROS
                                ; THAT ARE USED BY THE SUPERVISOR TO BUILD P-TABLES.  THE
                                ; MACROS ARE NOT EXECUTED AS MACHINE INSTRUCTIONS BUT ARE
                                ; INTERPRETED BY THE SUPERVISOR AS DATA STRUCTURES.  THE
                                ; MACROS ALLOW THE SUPERVISOR TO ESTABLISH COMMUNICATIONS
                                ; WITH THE OPERATOR.
                                ;--
002520                                  BGNSFT
002522                          GPRML   SPM1,0,-1,YES           ; GET OPTIONAL DPU ONLY FLAG.
002530                          GPRML   SPM2,2,-1,YES           ; GET MEM TEST MODE.
                                ;GPRML  SPM3,4,-1,YES           ; GET TEST ID CONTROL.
002536                          GPRML   SPM4,6,-1,YES           ; GET ITERATION CONTROL.
002544                          GPRML   SPM5,10,-1,YES          ; GET MANUFACTURING MODE.
002552                          GPRMD   SPM6,12,D,7777,0,7777,YES       ; GET LOCAL ERROR LIMIT
002564                          GPRMD   SPM7,14,D,7777,0,7777,YES       ; GET GLOBAL ERROR LIMIT
002576                                  EXIT SFT
002600  122  125  116  SPM1:   .ASCIZ  /RUN DPU TESTS ONLY     /
002630  114  117  116  SPM2:   .ASCIZ  /LONG MEMORY TESTS      /
                        ;SPM3:  .ASCIZ  /INHIBIT TEST ID'S      /
002660  111  116  110  SPM4:   .ASCIZ  /INHIBIT ITERATIONS     /
002710  115  101  116  SPM5:   .ASCIZ  /MANUFACTURING MODE     /
002740  120  105  122  SPM6:   .ASCIZ  /PER TEST ERROR LIMIT   /
002770  120  105  122  SPM7:   .ASCIZ  /PER UNIT ERROR LIMIT   /
003020                          ENDSFT          ; UNUSED.
```

```
                                .SBTTL  GLOBAL EQUATES SECTION

                                ;++
                                ; THE GLOBAL EQUATES SECTION CONTAINS PROGRAM EQUATES THAT
                                ; ARE USED IN MORE THAN ONE TEST.
                                ;--

003020                                  EQUALS          ; GET STANDARD EQUATES.
                                ;
                                ; BIT DIFINITIONS
                                ;
100000                          BIT15== 100000
040000                          BIT14== 40000
020000                          BIT13== 20000
010000                          BIT12== 10000
004000                          BIT11== 4000
002000                          BIT10== 2000
001000                          BIT09== 1000
000400                          BIT08== 400
000200                          BIT07== 200
000100                          BIT06== 100
000040                          BIT05== 40
000020                          BIT04== 20
000010                          BIT03== 10
000004                          BIT02== 4
000002                          BIT01== 2
000001                          BIT00== 1
                                ;
001000                          BIT9==  BIT09
000400                          BIT8==  BIT08
000200                          BIT7==  BIT07
000100                          BIT6==  BIT06
000040                          BIT5==  BIT05
000020                          BIT4==  BIT04
000010                          BIT3==  BIT03
000004                          BIT2==  BIT02
000002                          BIT1==  BIT01
000001                          BIT0==  BIT00
                                ;
                                ; EVENT FLAG DEFINITIONS
                                ;   EF32:EF17 RESERVED FOR SUPERVISOR TO PROGRAM COMMUNICATION
                                ;
000040                          EF.START==      32.             ; START COMMAND WAS ISSUED
000037                          EF.RESTART==    31.             ; RESTART COMMAND WAS ISSUED
000036                          EF.CONTINUE==   30.             ; CONTINUE COMMAND WAS ISSUED
000035                          EF.NEW==        29.             ; A NEW PASS HAS BEEN STARTED
000034                          EF.PWR==        28.             ; A POWER-FAIL/POWER-UP OCCURRED
                                ;
                                ;
                                ; PRIORITY LEVEL DEFINITIONS
                                ;
000340                          PRI07== 340
000300                          PRI06== 300
000240                          PRI05== 240
000200                          PRI04== 200
000140                          PRI03== 140
000100                          PRI02== 100
```

GLOBAL EQUATES SECTION

```
                000040                  PRI01== 40
                000000                  PRI00== 0
                                        ;
                                        ;OPERATOR FLAG BITS
                                        ;
                000004                  EVL==         4
                000010                  LOT==        10
                000020                  ADR==        20
                000040                  IDU==        40
                000100                  ISR==       100
                000200                  UAM==       200
                000400                  BOE==       400
                001000                  PNT==      1000
                002000                  PRI==      2000
                004000                  IXE==      4000
                010000                  IBE==     10000
                020000                  IER==     20000
                040000                  LOE==     40000
                100000                  HOE==    100000
1062
```

```
                                .SBTTL  MEMORY MANAGEMENT DEFINITIONS

                                ;*KT11 VECTOR ADDRESS

000250                          MMVEC=  250

                                ;*KT11 STATUS REGISTER ADDRESSES

177572                          SR0=    177572
177574                          SR1=    177574
177576                          SR2=    177576
172516                          SR3=    172516

                                ;*KERNEL "I" PAGE DESCRIPTOR REGISTERS

172300                          KIPDR0= 172300
172302                          KIPDR1= 172302
172304                          KIPDR2= 172304
172306                          KIPDR3= 172306
172310                          KIPDR4= 172310
172312                          KIPDR5= 172312
172314                          KIPDR6= 172314
172316                          KIPDR7= 172316

                                ;*KERNEL "I" PAGE ADDRESS REGISTERS

172340                          KIPAR0= 172340
172342                          KIPAR1= 172342
172344                          KIPAR2= 172344
172346                          KIPAR3= 172346
172350                          KIPAR4= 172350
172352                          KIPAR5= 172352
172354                          KIPAR6= 172354
172356                          KIPAR7= 172356
```

```
                        .SBTTL  VSV-11 INSTRUCTION EQUATES.
                        ;
                        ;PLOTTING INSTRUCTIONS.
                        ;
100000                  CHAR==  100000          ;CHARACTER MODE.
104000                  SVEC==  104000          ;SHORT VECTOR MODE.
110000                  LVEC==  110000          ;LONG VECTOR MODE.
114000                  APNT==  114000          ;ABSOLUTE POINT MODE.
120000                  GHX==   120000          ;GRAPH/HISTOGRAM X MODE.
124000                  GHY==   124000          ;GRAPH/HISTOGRAM Y MODE.
130000                  RPNT==  130000          ;RELATIVE POINT MODE.
                        ;
                        ;PIXEL DATA FIELD <10:2> FOR THE ABOVE OPCODES.
                        ;
002000                  L0==    BIT10           ; LEVEL 0 (BLACK) IS JUST THE ENABLE.
003774                  L255==  L0!<255.*BIT2>  ; LEVEL 255 IS THE MAX POSSIBLE.
003774                  ALL==   L255
003774                  ALLB==  3774            ; ALL INTENSITIES (WITH BLINK).
003374                  ALLNB== 3374            ; ALL INTENSITIES (WITHOUT BLINK).
                        ;
                        ;COLOR DEFINITIONS:
003774                  WHITE == ^O3774 ; ALL
002400                  GRN1 == ^O<400!2000>    ; BIT8!L0
003000                  GRN2 == ^O<1000!2000>   ; BIT9!L0
003400                  GRN3 == GRN1!GRN2       ; BIT8!BIT9
002200                  REDG == ^O<200!2000>    ; BIT7!L0
002100                  BLUEG == ^O<100!2000>   ; BIT6!L0
003600                  YELLOW == GRN3!REDG
003500                  EGGBL == GRN3!BLUEG
003300                  VIOLET == GRN2!REDG!BLUEG
003200                  GOLD == GRN2!REDG
002300                  MAGEN == REDG!BLUEG
                        ; COLOR LOOK-UP-RAM DEFINITIONS.
                        ;
010000                  LREAD== 1*BIT12         ; CSR -- READ DATA FROM ADDR <7:0>.
002000                  LDI==   1*BIT10         ;       INTERRUPT ON READ/WRITE DONE.
                        ;
010000                  LSET==  1*BIT12         ; DATA REG -- SET ALL ADDRESSES TO...
                                                ;...DATA PATTERN IN <11:0>...
                                                ;...BLU<11:8>, GRN<7:4>, RED<3:0>.
                        ;
000100                  GCOFF== 1*BIT6          ; MAINT REG -- GAMMA CORRECT OFF.
000040                  CVC==   1*BIT5          ;       COMP VIDEO > MDAC (READ-ONLY).
000020                  REDC==  1*BIT4          ;       RED DAC > MDAC (READ-ONLY).
000010                  GRNC==  1*BIT3          ;       GRN DAC > MDAC (READ-ONLY).
000004                  BLUC==  1*BIT2          ;       BLU DAC > MDAC (READ-ONLY).
000000                  MDV0==  0               ; SET MAINT DAC TO 0.0 V.
000001                  MDV1==  1               ;                  0.5 V.
000002                  MDV2==  2               ;                  1.0 V.
000003                  MDV3==  3               ;                  1.35 V.
                        ;
                        ; BIT MAP MODE DEFINITIONS.
                        ;
136000                  BM14==  136000          ; MODE 1 WITH 4 BIT PIXELS.
137000                  BM18==  137000          ; MODE 1 WITH 8 BIT PIXELS.
```

```
                                                        ; PIXEL COUNT IN <8:0>.
                
134000                  BM04==  134000                  ; MODE 0 WITH 4 BIT PIXELS.
135000                  BM08==  135000                  ; MODE 0 WITH 8 BIT PIXELS.
000000                  M32==   0                       ;  32 SQUARE PIXEL ARRAY.
000001                  M64==   1                       ;  64 SQUARE.
000002                  M128==  2                       ; 128 SQUARE.
000003                  M256==  3                       ; 256 SQUARE.
000010                  EX2==   10                      ; EXPAND BY 2, NO SMOOTHING.
000020                  EX4==   20                      ; EXPAND BY 4, NO SMOOTHING.
000050                  EXSM2== 50                      ; EXPAND AND SMOOTH BY 2.
000060                  EXSM4== 60                      ; EXPAND AND SMOOTH BY 4.
000100                  R256==  100                     ; 256 X 256 RESOLUTION.
000400                  RND8==  400                     ; ROUND FOR 8-BIT-WIDE IMAGE MEMORY
000200                  ODDSKP== 200                    ; SKIP ODD-NUMBERED LINES ON SCREEN (FOR NON-INTERLACED) ;@@@I
                
144000                  RNLN==  144000                  ;OP-CODE FOR RUN-LENGTH MODE
000100                  DBLPIX==        BIT6            ; DOUBLE THE PIXEL COUNT
000040                  YDOWN== BIT5                    ; DISPLAY LINES TOP-TO-BOTTOM
000400                  RLSKIP== BIT8                   ;SKIP LINES, WITHOUT THROWING THE DATA AWAY
                                                        ;   (FOR RUN-LENGTH MODE ONLY)
                                                        ;   SKIPS EVEN LINES IF START ON EVEN, & VICE-VERSA
162000                  IMREAD== 162000
```

```
1180                            ;CONTROL AND STATUS INSTRUCTIONS.
1181                            ;
1182        146040              CUOFF== 146040          ;CURSOR (JOY-STICK) VIDEO OFF.
1183        146060              CUON==  146060          ;CURSOR VIDEO ON.
1184        146100              CURD==  146100          ;READ JSX,JSY => DXR,DYR.
1185        146016              CUIM==  146016          ;INTERRUPT ON MATCH (SWITCH DISABLED).
1186        146013              CUIS==  146013          ;INTERRUPT ON SWITCH (MATCH DISABLED).
1187        146012              CUIOFF== 146012         ;INTERRUPTS (BOTH) OFF.
1188        175000              CUWT==  175000          ; CURSOR WRITE.
1189        175000              LDCP==  175000          ; LOAD CURSOR POSITION (=CUWT)
1190        146000              LDJSS== 146000          ;LOAD JOYSTICK STATUS DISPLAY INSTRUCTION
1191        175400              LDECC== 175400          ;LOAD EXTENDED CURSOR CONTROL
1192        000002              SWCHON== BIT1           ; SOFT SWITCH ON.
1193        175401              BLINK== 175401          ; BLINK.
1194        175400              NBLINK==175400          ; NO BLINK.
1195
1196        000002              JSWE == BIT1            ; JOYSTICK SWITCH-ENABLE WRITE ENABLE
1197        000001              JSWD == BIT0            ; JOYSTICK SWITCH-ENABLE WRITE DATA
1198        000010              JMWE == BIT3            ; CURSOR MATCH-ENABLE WRITE ENABLE
1199        000004              JMWD == BIT2            ; CURSOR MATCH-ENABLE WRITE DATA
1200        000040              JCWE == BIT5            ; CROSSHAIR-INTENSITY WRITE-ENABLE
1201        000020              JCWD == BIT4            ; CROSSHAIR-INTENSITY WRITE DATA
1202
1203
1204        150000              SETHB== 150000          ;SET GRAPH/HISTO BASE LINE.
1205        152000              SETCB== 152000          ;SET CHARACTER BASE ADDRESS.
1206
1207        160000              DJMP==  160000          ;DISPLAY JUMP.
1208        160001              DJMS==  160001          ;DISPLAY JUMP-TO-SUBROUTINE
1209        164000              DNOP==  164000          ;DISPLAY NO-OP.
1210        165000              DPOP==  165000          ;DPOP = RTS FROM CHARACTER SUB-PIX.
1211        164001              SYNC==  DNOP+1          ;PROCEED IN SYNC = DNOP + N <8:0>.
1212                                                    ;WAITS FOR N OCCURRENCES OF VERTICAL...
1213                                                    ;...SYNC START, THEN PROCEEDS...
1214                                                    ;...(16.6MS @ 60HZ OR 20MS @ 50HZ).
1215        000001              AUXSEG== BIT0           ;SPECIFY AUX. SEGMENT IN DISPATCH ADDRESSES.    @@@I
1216
1217        170200              SWTCH== 170200          ;SWITCH READ/WRITE STATUS.
1218        170140              CLRMEM== 170140         ;CLEAR IMAGE MEMORY TO 0'S.
1219        170100              SETMEM== 170100         ;SET IMAGE MEMORY TO CURRENT PIX DATA.
1220
1221        172000              STOP==  172000          ;DISPLAY STOP.
1222        171000              SIOFF== 171000          ;STOP INTERRUPT DISABLE.
1223        171400              SION==  171400          ;STOP INTERRUPT ENABLE.
1224        173000              STOPN== STOP!SIOFF      ;STOP, DO NOT INTERRUPT.
1225        173400              STOPI== STOP!SION       ;STOP AND INTERRUPT.
1226
1227        174100              GXI==   174100          ;SET X INCREMENT <5:0> FOR GRAPH/HST Y.
1228        174100              GYI==   GXI             ;SET Y INCREMENT <5:0> FOR GRAPH/HST X.
1229
1230        000001              HCPY==  BIT0            ;ENABLE HARD-COPY.
1231        000010              SWE==   BIT3            ;ENABLE SWITCH RD/WRT.
1232        176000              PROTEC== 176000         ;PROTECT MEMORY (OFF).
1233        176050              READ==  176040!SWE      ;READ MEM (DISPLAY), SWITCH ENABLED.
1234        176034              WRT==   176024!SWE      ;WRITE MEM (1'S AND 0'S), SWITCH ENABLED
1235        176036              WRT1==  176026!SWE      ;WRITE MEM (1'S ONLY), SWITCH ENABLED.
1236        176074              RDWRT== READ!WRT        ;READ, WRITE ALL DURING RETRACE.
```

```
1237          176076                  RDWRT1== READ!WRT1      ;READ, WRITE 1'S DURING RETRACE.
1238          176051                  HCOPY== READ!HCPY       ;READ MEM => HARD COPY DEVICE.
1239
1240          000000                  CH0==   0               ;CHANNEL SELECT BITS...
1241          000400                  CH1==   BIT8            ;...FOR MEMORY CONTROL...
1242          001000                  CH2==   BIT9            ;...INSTRUCTIONS (176XXX).
1243          001400                  CH3==   BIT8!BIT9
1244
1245          000000                  JS0==   CH0             ;JOY-STICK UNIT SELECT BITS...
1246          000400                  JS1==   CH1             ;...FOR CURSOR CONTROL...
1247          001000                  JS2==   CH2             ;...INSTRUCTIONS (146XXX).
1248          001400                  JS3==   CH3
```

```
                                ;OTHER USEFUL DEFINITIONS.
                                ;
040000                          I==     BIT14           ;INTENSIFY VECTOR OR POINT.
020000                          MSX==   BIT13           ;MINUS SIGNS FOR...
000100                          MSY==   BIT6            ;...SHORT FORM VECTOR DATA.
020000                          MXY==   BIT13           ;MINUS SIGN FOR LONG FORM DATA.
020000                          M==     BIT13           ;ANOTHER MINUS SIGN
040000                          HST==   BIT14           ;USE GHX/GHY DATA AS HISTOGRAM.
020000                          CGRPH== BIT13           ;USE GHX/GHY DATA AS CONNECTED GRAPH
060000                          FHST==  HST!BIT13       ;USE GHX/GHY DATA AS FILLED HISTOGRAM (BAR-GRAPH)
010000                          GHIINH== BIT12          ; INHIBIT ERROR DURING PLOT.

001776                          MAXX==  511.*2          ;MAXIMUM X ADDRESS (512 X 512 X 2).
001776                          MAXY==  MAXX            ;SAME FOR Y.
000776                          HAFX==  <MAXX/2>-1      ;HALF MAX X (CENTER SCREEN).
000776                          HAFY==  HAFX            ;SAME FOR Y.

001776                          MAXY50== 511.*2         ;MAX VISIBLE Y ON 50HZ SYSTEM.
000776                          HAFY50== <MAXY50/2>-1   ;HALF VISIBLE Y.

001676                          MAXY60== <511.-32.>*2   ;MAX VISIBLE Y ON 60HZ SYSTEM.
000736                          HAFY60== <MAXY60/2>-1   ;HALF VISIBLE Y.

000002                          DX==    2               ;DELTA X (1 PIXEL UNIT).
000002                          DYI==   DX              ;DELTA Y, INTERLACED MODE.
000004                          DYNI==  DX*2            ;DELTA Y, NON-INTERLACED MODE.

                                ;DSR REGISTER SELECT CODES:
000000                          SELDSR==        0       ;SELECT REAL DSR
000001                          SELPCS==        1       ;SELECT PCSAVE
000002                          SELFLG==        2       ;SELECT FLAGS
000003                          SELCSR==        3       ;SELECT CSR
000004                          SELMRR==        4       ;SELECT MAIN RELOC
000004                          SELMPM==        SELMRR  ;SELECT MAIN PROTECT
000006                          SELXRR==        6       ;SELECT AUX. RELOC
000006                          SELXPM==        SELXRR  ;SELECT AUX. PROTECT

000005                          SELHBA==        5       ;SELECT H-BASE
000007                          SELCBA==        7       ;SELECT C-BASE
000010                          WE==    BIT3            ;DSR WRITE-ENABLE
000014                          SETMRR==SELMRR!WE       ;SET MAIN-SEG RELOCATION REGISTER TO 0...
                                                        ;...OR SETMR!<ADDR/2> TO RELOCATE.
000015                          SETMPM==15              ;SET MAIN-SEG PROTECTION MASK
000016                          SETXRR==SELXRR!WE       ;SET AUX.-SEG RELOCATION REGISTER
000017                          SETXPM==17              ;SET AUX.-SEG PROTECTION MASK
000013                          WRTCSR==SELCSR!WE       ;WRITE THE CSR REGISTER
000012                          CLFLGS==SELFLG!WE       ;CLEAR THE FLAGS REGISTER
000010                          SETDSR==SELDSR!WE       ; SET REAL DSR.
000013                          SETCSR==WRTCSR          ; SET THE CSR REGISTER.
000012                          SETFLG==CLFLGS          ; CLEAR THE FLAGS REGISTER.

                                ;"WRITE" FUNCTIONS FOR DXR (BITS 15-14):
000000                          PIXRBK== 0*BIT14                ;PIXEL READBACK -- <15:14>=00
042000                          WRTJSS== 1*BIT14!BIT10          ;WRITE JOYSTICK STATUS REGISTER -- <15:14>=01
100000                          RSTPOS== 2*BIT14                ;RESTORE TRUE X-Y POSITION TO DXR,DYR -- <15:14>=10 ;aaaI
140000                          WRTCPX== 3*BIT14                ;WRITE CURSOR POSITION X-COORDINATE -- <15:14>=11
040000                          GTJSSW== BIT14                  ;GET JOYSTICK SWITCH INTERRUPT COORDINATES
```

```
1307
1308                            ;'WRITE" FUNCTIONS FOR DYR (BITS 15-14):
1309                            ;  PIXEL READBACK IS THE SAME - 00
1310    040000                  WRTMSR== 1*BIT14                ;WRITE MEMORY STATUS REGISTER BA -- <15:14>=01
1311    100000                  SINIT== 2*BIT14                 ;PERFORM "SOFT-INIT" -- <15:14>=10
1312    140000                  WRTCPY== 3*BIT14                ;WRITE CURSOR POSITION Y-COORDINATE -- <15:14>=11
1313
1314                            ;CSR BIT DEFINITIONS:
1315    000040                  FORCSI== BIT5           ;FORCE STOP INTERRUPT
1316    000100                  ENECHK== BIT6           ;ENABLE ERROR CHECKING
1317    000200                  CHPROT== BIT7           ;CHANNEL PROTECT
1318                            ;@@@I
1319                            ;FLAGS REGISTER BIT POSITIONS:
1320    000001                  PINTRA== BIT0           ;PENDING INTERRUPT, CHANNEL A
1321    000002                  PINTRB== BIT1           ;PENDING INTERRUPT, CHANNEL B
1322    000004                  PVEC2== BIT2            ;VECTOR BIT 2 FOR INTERRUPT
1323    000010                  JSLCKO== BIT3           ;JOYSTICK INTERRUPT LOCKOUT
1324
1325                            ; ERROR CODE DEFINITIONS.
1326
1327    100000                  ERR==   100000          ; ERROR (COMPOSITE).
1328    104003                  NXME==  ERR!4000!SELCSR ; NONEXISTANT MEMORY ERROR.
1329    114003                  MPE==   ERR!14000!SELCSR ; MEMORY PROTECTION ERROR.
1330    124003                  RSVDOP== ERR!24000!SELCSR ; RESERVED OPCODE, RESERVED OPERATION.
1331    120003                  SEQERR== ERR!20000!SELCSR ; SEQUENCE ERROR.
1332    140003                  SYNCTO== ERR!40000!SELCSR ; SYNC TIMEOUT FROM IMAGE MEMORY.
1333    144003                  DAVTO== ERR!44000!SELCSR ; DATA AVAILABLE TIMEOUT (PIXEL READBACK).
1334    150003                  DRDYTO== ERR!50000!SELCSR ; DATA READY TIMEOUT (JOYSTICK STATUS).
1335
1336                            ;DXR, DYR BITS
1337
1338    010000                  CHIE    ==BIT12                 ; CROSSHAIR INTENSITY ENABLE STATUS (1=ON)
1339    010000                  JSMIES  ==BIT12                 ; JOYSTICK MATCH INTERRUPT ENABLE STATE
1340    004000                  JSSIES  ==BIT11                 ; JOYSTICK SWITCH INTERRUPT ENABLE STATE
1341    042000                  WJSS    ==BIT14!BIT10           ; WRITE JOYSTICK STATUS REGISTER
1342    160000                  WECC    ==160000                ; WRITE EXTENDED CURSOR CONTROL
1343
1344
1345                            ;
1346                            ;SOME HANDY -11 OPDEF'S.
1347                            ;
1348    000403                  SKP3=   BR+3            ;SKIP NEXT 3 WORDS.
1349    000402                  SKP2=   BR+2            ;SKIP NEXT 2 WORDS.
1350    000401                  SKP1=   BR+1            ;SKIP NEXT WORD.
1351    000400                  SKP0=   BR+0            ;SAME AS A NOP.
1352
1353                            ;
1354                            ; SOME GENERAL EQUATES.
1355                            ;
1356
1357    000004                  ERRVEC==4                       ; POINTER TO ERROR VECTOR FOR BUS TIME OUT.
```

```
                                .SBTTL  SPECIAL MACROS AND OPDEFS.

000000                                  EN=0                    ; INITIALIZE ERROR NUMBER

                                ;
                                ; MACRO TO XCT MEMORY SETUP FUNCTIONS (STAT C INSTRUCTIONS)
                                ; VAILD FUNCTIONS ARE:
                                ; PROTEC, READ, WRT, WRT1, RDWRT, RDWRT1, HCOPY.
                                ;
                                        .MACRO  SETUP FUNC,N1,N2,N3,N4
                                        .IIF NB <N1>,   FUNC!CH'N1
                                        .IIF NB <N2>,   FUNC!CH'N2
                                        .IIF NB <N3>,   FUNC!CH'N3
                                        .IIF NB <N4>,   FUNC!CH'N4
                                        .ENDM   SETUP
                                ;
                                ; MACROS FOR ASSEMBLING VECTOR DATA.
                                ; CALLING ARGUMENTS ARE:
                                ;       A = INTENSIFY BIT, "I" OR "U".
                                ;       B = DELTA X, +/-0 THRU 1776 (76 IF SHORT TYPE).
                                ;       C = DELTA Y,            DITTO
                                ;
                                        .MACRO  SXY A,B,C       ;ASSEMBLE SHORT (1 WORD) DATA.
                                        .I = 40000
                                        .DX = B*200
                                        .DY = C
                                        .IIF IDN <U> <A>, .I = 0
                                        .IIF LT B, .DX = <-B*200>!20000
                                        .IIF LT C, .DY = <-C>!100
                                        .WORD   .I!.DX!.DY
                                        .ENDM   SXY

                                        .MACRO  LXY A,B,C       ;ASSEMBLE LONG (2 WORDS) DATA.
                                        .I = 40000
                                        .DX = B
                                        .DY = C
                                        .IIF IDN <U> <A>, .I = 0
                                        .IIF LT B, .DX = <-B>!20000
                                        .IIF LT C, .DY = <-C>!20000
                                        .WORD   .I!.DX, .DY
                                        .ENDM   LXY
```

```
1401                            ;
1402                            ; MACROS TO STANDARDIZE THE INITIALIZATION OF EACH TEST.
1403                            ; PROVIDES AN ASCIZ TITLE STRING, AND POINTER FOR
1404                            ; SUBSEQUENT PRINTING (IF REQ'D) VIA "PRINTF TNAM,TNUM" CALL.
1405                            ; ALSO INITS A SEQUENTIAL ERROR NUMBER SEQUENCE, AND
1406                            ; SETS AN ITERATION COUNT FOR THE CURRENT TEST.
1407                            ;
1408                                    .MACRO  BEGIN.TEST TEXT,LK,?TAG1,?TAG2
1409                            .RADIX 10
1410                                    TSTITLE \TN+1,<TEXT>,<LK>
1411                            .RADIX
1412                            .NLIST
1413                            .LIST MC
1414                            .LIST
1415                                    BGNTST
1416                                    TN=T$TESTNUM            ;TEST NUMBER
1417                                    EN=TN*100.              ;INIT ERROR NUMBER SEQUENCE.
1418                                    .IIF B <LK>, MOV #1,LOOPK       ;DEFAULT = 1.
1419                                    .IIF NB <LK>, MOV #LK,LOOPK
1420                                    MOV     #TN,TNUM        ;SET TEST NUMBER...
1421                                    MOV     #TAG1,TNAM      ;...AND NAME POINTER.
1422                                    BR      TAG2            ;SKIP OVER THE ASCII.
1423                            ;TAG1:  .ASCIZ  \%N%ATEST %D2%A: TEXT\
1424                            TAG1:   .ASCIZ  \%A  TEXT\
1425                                    .EVEN
1426                            TAG2:
1427                                    .ENDM   BEGIN.TEST
1428
1429                            ;
1430                            ; THIS MACRO IS USED BY MACRO "BEGIN.TEST" MACRO, ABOVE.
1431                            ;
1432                                    .MACRO  TSTITLE A,ASCI,I
1433                            .SBTTL  *** TEST 'A'    ASCI
1434                            .NLIST
1435                            .LIST ME
1436                            .LIST
1437
1438                            ;*****************************************************************************
1439                            ;*
1440                            ;*      BEGIN TEST 'A'  -   ASCI
1441                            ;*
1442                            ;*****************************************************************************
1443
1444                            ; (ITERATION COUNT = I)
1445
1446                            .NLIST
1447                            .NLIST ME
1448                            .LIST
1449                                    .ENDM   TSTITLE
1450
1451                            ;
1452                            ; MACROS TO STANDARDIZE THE END OF EACH TEST.
1453                            ;
1454
1455                                    .MACRO  END.TEST
1456                            .NLIST
1457                            .RADIX 10
```

```
                        .NLIST MC
                        .LIST ME
                        .LIST
                                ENDTN   \TN
                        .NLIST
                        .RADIX
                        .NLIST ME
                        .LIST
                                ENDTST
                                .ENDM   END.TEST

                        ;
                        ; THIS MACRO IS USED BY "END.TEST" MACRO, ABOVE.
                        ;
                                .MACRO  ENDTN   A

                        ;*******************************************************************************
                        ;*
                        ;*      END TEST 'A'
                        ;*
                        ;*******************************************************************************

                                .ENDM   ENDTN

                        ;
                        ; MACROS TO DEFINE SEQUENTIAL ERROR NUMBERS FOR
                        ; SUBSEQUENT SUPERVISOR ERROR CALLS.
                        ; REQUIRES PRIOR USE OF "BEGIN.TEST" MACRO.
                        ;
                                .MACRO  SFERR ADDR,PNTR         ; SYSTEM FATAL.
                                EN=EN+1
                                JSR     PC,INCERK
                                ERRSF   EN,ADDR,PNTR
                                JSR     PC,CKDROP
                                .ENDM   SFERR

                                .MACRO  DFERR ADDR,PNTR,CK              ; DEVICE FATAL.
                                EN=EN+1
                                JSR     PC,INCERK
                                ERRDF   EN,ADDR,PNTR
                                JSR     PC,CKDROP
                                .IF     NB,CK
                                CKLOOP
                                .ENDC
                                .ENDM   DFERR

                                .MACRO  SFTERR ADDR,PNTR,CK     ; SOFT ERROR.
                                EN=EN+1
                                JSR     PC,INCERK
                                ERRSOFT EN,ADDR,PNTR
                                JSR     PC,CKEMAX
                                .IF     NB,CK
                                CKLOOP
                                .ENDC
                                .ENDM   SFTERR

                                .MACRO  HRDERR ADDR,PNTR,CK     ; HARD ERROR.
```

```
1515                            .NLIST
1516                            .NLIST ME
1517                            .LIST
1518                                    EN=EN+1
1519                                    JSR     PC,INCERK
1520                                    ERRHRD  EN,ADDR,PNTR
1521                                    JSR     PC,CKEMAX
1522                                    .IF     NB,CK
1523                                    CKLOOP
1524                                    .ENDC
1525                                    .ENDM   HRDERR
1526
1527
1528                            ;
1529                            ; MACRO - DO SOME COMMON STUFF AT THE BEGINNING OF A TEST.
1530                            ;
1531                                    .MACRO  COMBEG
1532                            .NLIST
1533                            .LIST ME
1534                            .LIST
1535                                    JSR     PC,TSTGO        ; TITLE.
1536                                    JSR     PC,DPRESET      ; DO SOFT INIT.
1537                            .NLIST ME
1538                                    .ENDM   COMBEG
1539
1540                            ;
1541                            ; MACRO - DO SOME COMMON STUFF AT THE END OF A TEST.
1542                            ;
1543                                    .MACRO  COMEND  TAG,?L,?X
1544                            .NLIST
1545                            .LIST ME
1546                            .LIST
1547                                    JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
1548                                    BCS     L
1549                                    BR      X
1550
1551                            L:      JMP     TAG
1552
1553                            X:      JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
1554                            .NLIST
1555                            .NLIST ME
1556                            .LIST
1557                                    .ENDM   COMEND
1558
1559                            ;
1560                            ; MACRO TO CONTROL ITERATION LOOPS.
1561                            ;
1562                                    .MACRO  LOOPTO  TAG
1563                            .NLIST
1564                            .LIST ME
1565                            .LIST
1566                                    JSR     PC,LOOP
1567                                    BCS     TAG
1568                            .NLIST
1569                            .NLIST ME
1570                            .LIST
1571                                    .ENDM   LOOPTO
```

```
1572
1573                                    ;
1574                                    ; MACRO - WRITE FROM SPEC'D DATA SOURCE INTO SPEC'D DSR ACCESS REGISTER.
1575                                    ; ONLY DSR, FLG, CSR, MRR, MPM, XRR, XPM ARE VALID ARGUMENTS FOR 'DSRREG'.
1576                                    ; R0=WORK.
1577                                    ;
1578                                            .MACRO  WTDSRA  DATSRC,DSRREG
1579                                    .NLIST
1580                                    .LIST MEB
1581                                    .LIST
1582                                            .NTYPE  .NT,DATSRC
1583                                            .IF     EQ,.NT&^O77-^O27
1584                                            MOV     DATSRC&^C17!SET'DSRREG,@DSR
1585                                            .IFF
1586                                            MOV     #SET'DSRREG',R0 ; WRITE DATSRC INTO DSRREG .
1587                                            BIS     DATSRC,R0
1588                                            MOV     R0,@DSR
1589                                            .ENDC
1590                                    .NLIST
1591                                    .NLIST MEB
1592                                    .LIST
1593                                            .ENDM   WTDSRA
1594
1595                                    ;
1596                                    ; MACRO - READ FROM SPEC'D DSR ACCESS REGISTER INTO SPEC'D DATA DESTINATION.
1597                                    ; ONLY DSR, PCS, FLG, CSR, MRR, MPM, XRR, XPM, HBA, CBA ARE VALID
1598                                    ; 'DSRREG'' ARGUMENTS.
1599                                    ; R0=WORK.
1600                                    ;
1601                                            .MACRO  RDDSRA  DSRREG,DATDST
1602                                    .NLIST
1603                                    .LIST ME
1604                                    .LIST
1605                                            MOV     #SEL'DSRREG',@DSR       ; READ DSRREG INTO DATDST .
1606                                            MOV     @DSR,DATDST
1607                                    .NLIST
1608                                    .NLIST ME
1609                                    .LIST
1610                                            .ENDM   RDDSRA
1611
1612                                    ;
1613                                    ; MACRO - IF R1=R2, BRANCH TO OK, ELSE CALL HARDWARE ERROR.
1614                                    ;
1615                                            .MACRO  IFERROR ARG1,ARG2,?OK$,CK
1616                                    .NLIST
1617                                    .LIST ME
1618                                    .LIST
1619                                            CMP     R1,R2           ; OK?
1620                                            BEQ     OK$             ; YES.
1621                                            HRDERR  ARG1,ARG2,CK    ; NO.
1622                                    .NLIST
1623                                    .LIST ME
1624                                    .LIST
1625                                    OK$:
1626                                    .NLIST ME
1627                                            .ENDM   IFERROR
1628
```

```
1629                                    ;
1630                                    ; MACRO - IF R1=R2 AND R3=R4, BRANCH TO OK, ELSE CALL HARDWARE ERROR.
1631                                    ;
1632                                            .MACRO  IFERRX2 ARG1,ARG2,?OK$,?ERR,CK
1633                                    .NLIST
1634                                    .LIST ME
1635                                    .LIST
1636                                            CMP     R1,R2           ; 1ST ITEM OK?
1637                                            BNE     ERR             ; NO.
1638                                            CMP     R3,R4           ; 2ND ITEM OK?
1639                                            BEQ     OK$             ; YES.
1640                                    ERR:    HRDERR  ARG1,ARG2,CK    ; NO.
1641                                    .NLIST
1642                                    .LIST ME
1643                                    .LIST
1644                                    OK$:
1645                                    .NLIST ME
1646                                            .ENDM   IFERRX2
1647
1648                                    ;
1649                                    ; MACRO - SOFTWARE INITIALIZE THE DPU.  THIS DIFFERS FROM "DPINIT"
1650                                    ; IN TWO WAYS.  FIRST, IT IS ASSUMED THAT THE DPU IS STOPPED AND
1651                                    ; NOT HUNG UP.  SECOND, NO CHECKING IS DONE BY THIS CODE.
1652                                    ;
1653                                            .MACRO  INITDP
1654                                    .NLIST
1655                                    .LIST ME
1656                                    .LIST
1657                                            JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
1658                                    .NLIST
1659                                    .NLIST ME
1660                                    .LIST
1661                                            .ENDM   INITDP
1662
1663
1664                                            .MACRO  DPSTART ADRS
1665                                    .NLIST
1666                                    .LIST ME
1667                                    .LIST
1668                                            MOV     ADRS,@DPC       ; START THE DPU.
1669                                            JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
1670                                    .NLIST
1671                                    .NLIST ME
1672                                    .LIST
1673                                            .ENDM   DPSTART
1674
1675
1676
1677                                            .MACRO  DPCONT
1678                                    .NLIST
1679                                    .LIST ME
1680                                    .LIST
1681                                            BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
1682                                            JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
1683                                    .NLIST
1684                                    .NLIST ME
1685                                    .LIST
```

```
                        .ENDM   DPCONT

                ;MACRO TO DO A 3-OPERAND BIT-SET:
                        .MACRO  BISW3   S1,S2,DST
                .NLIST
                .LIST MEB
                .LIST
                        .NTYPE  .NT,DST
                        .IF     EQ,.NT&^O70
                        MOV     S2,DST
                        BIS     S1,DST
                        .IFF
                        MOV     S2,-(SP)
                        BIS     S1,(SP)
                        MOV     (SP)+,DST
                        .ENDC
                .NLIST
                .NLIST MEB
                .LIST
                        .ENDM   BISW3

                ;MACRO TO DO A 3-OPERAND BIT-CLEAR:
                        .MACRO  BICW3   S1,S2,DST
                .NLIST
                .LIST MEB
                .LIST
                        .NTYPE  .NT,DST
                        .IF     EQ,.NT&^O70
                        MOV     S2,DST
                        BIC     S1,DST
                        .IFF
                        MOV     S2,-(SP)
                        BIC     S1,(SP)
                        MOV     (SP)+,DST
                        .ENDC
                .NLIST
                .NLIST MEB
                .LIST
                        .ENDM   BICW3

                ;MACRO TO SET UP THE INTERRUPT MASK AND PRIME THE INTERRUPT FLAG:
                        .MACRO  INTSET  IEX
                .NLIST
                .LIST MEB
                .NLIST ME
                .LIST
                        .IF     NB,IEX
                        MOV     #^O177400!IOKCKIN!IOK'IEX,INTMASK       ; PRIME FLAG, EXPECT IEX
                        .IFF
                        MOV     #^O177400!IOKCKIN,INTMASK       ; PRIME FLAG, NO INTR. EXPECTED.
                        .ENDC
                .NLIST
                .NLIST MEB
                .LIST
                        .ENDM   INTSET
```

```
1743
1744                                    ;
1745                                    ; MACRO TO FORCE IN-LINE RETURN FROM REAL OR IMAGINED INTERUPT.
1746                                    ;
1747                                            .MACRO  FORRTI
1748                                    .NLIST
1749                                    .LIST ME
1750                                    .LIST
1751                                            MOV     #.+6,(SP)       ; FORCE RTI RETURN TO -
1752                                            RTI                     ; - NEXT LOC.
1753                                    .NLIST
1754                                    .NLIST ME
1755                                    .LIST
1756                                            .ENDM   FORRTI
```

```
                                .SBTTL  USER PATCHABLE FLAGS
                                ;
                                ; THE FOLLOWING LOCATIONS MAY BE PATCHED BY THE USER
                                ; TO OBTAIN THE RESULTS DESCRIBED FOR EACH.
                                ;
003020  000000                  FORCER: 0       ; FORCE TYPE ALL HARD ERRORS (THE ONES CALLED -
                                                ; - BY THE MACRO "IFERROR").  AN ERROR NEED NOT -
                                                ; - EXIST, JUST ASSUME AND TYPE THE MESSAGE.


003022  001700                  MFGM0:  1700    ; MANUFACTURING MEMORY 0 MASTER SPECIFICATION.
                                                ; IN MANUFACTURING MODE, MEMORY 0 IS COMPARED
                                                ; WITH THIS VALUE WHEN SYSTEM CONFIGURATION IS
                                                ; EXECUTED.  DISCREPANCY IS REPORTED.
                                                ; (CURRENTLY SHOWN FOR 4 BIT MEMORY.)

003024  120000                  MFGS0:  120000  ; MANUFACTURING SYNC CHAN 0 MASTER SPECIFICATION.
                                                ; IN MANUFACTURING MODE, SYNC CHAN 0 IS COMPARED
                                                ; WITH THIS VALUE WHEN SYSTEM CONFIGURATION IS
                                                ; EXECUTED.  DISCREPANCY IS REPORTED.
                                                ; (CURRENTLY SHOWN FOR INTERLACED SYNC CHAN.)
```

```
                                .SBTTL  GLOBAL DATA SECTION

                                ;++
                                ; THE GLOBAL DATA SECTION CONTAINS DATA THAT ARE USED
                                ; IN MORE THAN ONE TEST.
                                ;--

                                ;
                                ; THE FOLLOWING DATA ARE SET FOR EACH UNIT AT INIT TIME.
                                ; SINGLE UNIT DEFAULTS (LISTED) ARE IN THE DEFAULT P-TABLE.
                                ;
003026  000000                  UNITN:: 0               ;UNIT # UNDER TEST.
003030  000000                  QVP::   0               ;QUICK VERIFY FLAG.

                                ;
                                ;DEVICE REGISTER ADDRESSES *** KEEP IN THE ORDER GIVEN ***
                                ;
003032  172010                  DPC::   172010          ;DISPLAY PC
003034                          DRR::
003034  172012                  DSR::   172012          ;DISPLAY STATUS REGISTER (ALSO REL REG).
003036  172014                  DXR::   172014          ;DISPLAY X POSITION REGISTER.
003040  172016                  DYR::   172016          ;DISPLAY Y POSITION REGISTER.

003042  172020                  LSR::   172020          ; LOOK-UP-RAM STATUS.
003044  172022                  LDR::   172022          ;               DATA.
003046  172024                  LMR::   172024          ;               AND MAINT REGISTERS.

003050  000200                  DPRI::  PRI04           ;INTERRUPT PRIORITY.
003052  000320                  STPV::  320             ;STOP VECTOR
003054  000324                  JSMV::  324             ;JOY-STICK MATCH VECTOR.
003056  000330                  TOTV::  330             ;TIME-OUT VECTOR
003060  000334                  JSSV::  334             ;JOY-STICK SWITCH (DEMAND) VECTOR.
003062  000340                  LUTV::  340             ; LUT READ/WRITE DONE VECTOR.

003064  040000                  OPTI::  I               ;OPTIONAL "INTENSIFY" BIT FOR DPU...
                                                        ;...IF DPUMOD = 0, OPTI = I(NTENSIFY).
                                                        ;...OTHERWISE, OPTI = 0 (UNINTENSIFY).

                                ;
                                ; AND THESE ARE SET BY THE IMAGE MEMORY SIZER.
                                ;
003066  000004                  PDBITS:: 4              ;NUMBER OF PIXEL BITS (2, 4, 6, OR 8).
003070  000100                  PDI::   1*BIT6          ;PIXEL DATA INCREMENT (MIN INTENSITY).
003072  001700                  PDF::   17*BIT6         ;PIXEL DATA FIELD (MAX INTENSITY).
003074  176077                  PDM::   ^C<17*BIT6>     ;PIXEL DATA FIELD MASK.
                                                        ;
                                                        ;       STANDARD CONFIGURATIONS
                                                        ;       BITS    PDI     PDF
                                                        ;         2     0400    1400
                                                        ;         4     0100    1700
                                                        ;         6     0020    1760
                                                        ;         8     0004    1774
```

```
1836                                    ;
1837                                    ; STORAGE FOR DEVICE REGISTERS
1838                                    ;
1839 003076  000000  100000  000000     DUMMY:  0,100000,0,0    ;DUMMY DEVICE REGISTERS...
1840 003106  000000  000000  000000             0,0,0,0,0,0,0,0
1841                                                            ;...FOR MULTI-UNIT CHECKOUT.
1842
1843 003126  000000                     IDPC::  0               ;SAVED DPC ON ANY INTERRUPT.
1844 003130  000000                     IDSR::  0               ;       DITTO DSR
1845 003132  000000                     IDXR::  0               ;       DITTO DXR
1846 003134  000000                     IDYR::  0               ;       DITTO DYR
1847 003136  000000                     ILSR::  0               ;       DITTO LUT CSR
1848 003140  000000                     ILDR::  0               ;       DITTO LUT DATA
1849 003142  000000                     ILMR::  0               ;       DITTO LUT MAINT.
1850
1851                                    ;SAVED "INTERNAL" REGISTERS ON AN INTERRUPT (IN ORDER OF SELECT CODE):
1852 003144  000000                     ISDSR:: 0               ;SAVED "SELECTED" DSR
1853 003146  000000                     IPCSAV::        0       ;SAVED PCSAVE
1854 003150  000000                     IFLAGS::        0       ;SAVED FLAGS
1855 003152  000000                     ICSR::  0               ;SAVED CSR
1856 003154  000000                     IMAIN:: 0               ;SAVED MAIN SEG. MMGT.
1857 003156  000000                     IHBASE::        0       ;SAVE HISTOGRAM BASE
1858 003160  000000                     IAUX::  0               ;SAVED AUX. SEG MMGT.
1859 003162  000000                     ICBASE::        0       ;SAVED CHARACTER BASE
1860
1861 003164  000000                     SDPC::  0               ;SAVED DPC AT ANY OTHER TIME.
1862 003166  000000                     SDSR::  0               ;       DITTO DSR
1863 003170  000000                     SDXR::  0               ;       DITTO DXR
1864 003172  000000                     SDYR::  0               ;       DITTO DYR
1865                                    ;SAVED INTERNAL REGISTERS AT OTHER TIMES
1866 003174  000000                     SSDSR:: 0               ;SAVED "SELECTED" DSR
1867 003176  000000                     SPCSAV::        0       ;SAVED PCSAVE
1868 003200  000000                     SFLAGS::        0       ;SAVED FLAGS
1869 003202  000000                     SCSR::  0               ;SAVED CSR
1870 003204  000000                     SMAIN:: 0               ;SAVED MAIN SEG. MMGT.
1871 003206  000000                     SHBASE::        0       ;SAVE HISTOGRAM BASE
1872 003210  000000                     SAUX::  0               ;SAVED AUX. SEG MMGT.
1873 003212  000000                     SCBASE::        0       ;SAVED CHARACTER BASE
1874                                    ;
1875                                    ; AND FINALLY SOME MISCELLANEOUS REGISTERS.
1876                                    ;
1877 003214  000000                     SCFLG:: 0               ; SYS CONFIG FLAG (0=1ST TIME, NZ=SUBSEQUENT).
1878 003216  000000                     LUTAV:: 0               ; LUT AVAILABLE FLAG (NZ = USE LUT).
1879 003220  000000                     SHADLY:: 0              ; SHADING DELAY COUNTER (OPTIONAL).
1880 003222  110224                     HUE::   NTSC8+2         ; CURRENT COLOR POINTER (OPTIONAL).
1881
1882                                    ;STIK:: .BYTE   -1, -1  ; 8 JOY-STICK AVAILABLE FLAGS...
1883                                    ;       .BYTE   -1, -1  ;... -1 = I DUNNO...
1884                                    ;       .BYTE   -1, -1  ;...  0 = NO...
1885                                    ;       .BYTE   -1, -1  ;... +1 = YES.
1886
1887 003224  000000                     DUFLG:: 0               ;"DROPPED UNIT" FLAG.  INHIBITS DPU...
1888                                                            ;...CODE IN "CLEAN-UP".
1889 003226  000000                     NODEV:: 0               ;FLAG TO SAY NO DEVICE.
1890
1891 003230  000001                     INTLAC:: 1              ; INTERLACED MODE
1892 003232  000000                     TEMP1:: 0               ;SOME TEMP LOCATIONS.
```

```
1893 003234  000000                TEMP2:: 0
1894 003236  000000                XXCOMM::        0                       ; XXDP+ COMM BLOCK POINTER.
1895 003240  000000                FREE::  0                               ; 1ST FREE MEMORY ADDRESS...
1896 003242  000000                FRESIZ::        0                       ;...AND SIZE (IN WORDS).
1897 003244  000000                KTFLG:: 0                               ; KT11, MEM AVAIL FLAG -
1898                                                                       ; - 0 = <24K OR NO KT -
1899                                                                       ; - NZ = >24K AND KT.
1900 003246  000074                HZ::    60.                             ; LINE FREQUENCY.
1901 003250  000000                YMAX::  0                               ; MAX Y ACCORDING TO LINE FREQENCY.
1902 003252  000000                MEMFLG:: 0                              ; MEM FLAGS (1'S IN BITS 0-3 REFLECT MEMS AVAIL).
1903 003254  000000                SYCFLG:: 0                              ; SYNC CHAN FLAGS (1'S IN BITS 0-3 REFLECT SYNC CHANS AVAIL)
1904 003256                         CTAB::                                  ; CONFIGURATION TABLES.
1905 003256                         MEMTAB::
1906 003256  000000                CTABM:: 0                               ; MEMORY CONFIG WORK.
1907 003260  000000                        0
1908 003262  000000                        0
1909 003264  000000                        0
1910 003266  177777                        -1                              ; END OF MEM TABLE.
1911 003270                         SYCTAB::
1912 003270  000000                CTABS:: 0                               ; SYNC CHAN CONFIG WORK.
1913 003272  000000                        0
1914 003274  000000                        0
1915 003276  000000                        0
1916 003300  177777                        -1                              ; END OF SYNC CHAN TABLE.
1917 003302                         CTABE::                                 ; END OF CONFIG WORK AREA.
1918
1919                                ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
1920                                ;
1921                                ;       0       =       UNIT NOT TESTED
1922                                ;       100000  =       UNIT ONLINE, NO ERRORS
1923                                ;       10XXXX  =       UNIT ONLINE, ENCOUNTERED XXXX ERRORS
1924                                ;       160000  =       UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
1925                                ;       160001  =       UNIT DROPPED, NOT IDLE AT START
1926                                ;       14XXXX  =       UNIT DROPPED, ENCOUNTERED XXXX ERRORS
1927                                ;
1928 003302                        ERTABL: .BLKW   64.
1929 003502  000000                ERTABE: 0
```

```
                                .SBTTL  GLOBAL TEXT SECTION

                                ;++
                                ; THE GLOBAL TEXT SECTION CONTAINS FORMAT STATEMENTS,
                                ; MESSAGES, AND ASCII INFORMATION THAT ARE USED IN
                                ; MORE THAN ONE TEST.
                                ;--

003504  040  040  116  NXR:    .ASCIZ  /  NON-EXISTANT DEVICE REGISTER/
003543  045  101  040  NXRX:   .ASCIZ  /%A  ADDRESS: %06/
003564  045  101  040  PCSX:   .ASCII  /%A  DPC,DSR EXP'D: %06%A,%06%N/
003622  045  101  040         .ASCIZ  /%A  DPC,DSR REC'D: %06%A,%06/
003657  045  101  040  XYX:    .ASCII  /%A  DXR,DYR EXP'D: %06%A,%06%N/
003715  045  101  040         .ASCIZ  /%A  DXR,DYR REC'D: %06%A,%06/
003752  045  116  045  FUSI:   .ASCII  /%N%A/
003756  040  040  125  USI:    .ASCIZ  /  UNEXPECTED "STOP" INTERRUPT/
004014  040  040  042  NSI:    .ASCIZ  /  "STOP" INTERRUPT EXPECTED, NOT RECEIVED/
004066  045  116  045  FUTO:   .ASCII  /%N%A/
004072  040  040  125  UTO:    .ASCIZ  /  UNEXPECTED DPU "ERROR" INTERRUPT/
004135  040  040  104  NTO:    .ASCIZ  /  DPU "ERROR" INTERRUPT EXP'D, NOT REC'D/
004206  045  116  045  FUMI:   .ASCII  /%N%A/
004212  040  040  125  UMI:    .ASCIZ  /  UNEXPECTED "MATCH" INTERRUPT/
004251  045  116  045  FUSWI:  .ASCII  /%N%A/
004255  040  040  125  USWI:   .ASCIZ  /  UNEXPECTED "SWITCH" INTERRUPT/
004315  045  116  045  FNOINTR:        .ASCII  /%N%A/
004321  040  040  116  NOINTR: .ASCIZ  /  NO INTERRUPT WAS GENERATED/
004356  040  040  111  IFAULT: .ASCIZ  /  INTERRUPT FAULT/
004400  045  101  040  INTX:   .ASCIZ  /%A  CPU PC: %06%A  DPC: %06/
004434  040  040  042  NOINIT: .ASCIZ  /  "BUS-INIT" DIDN'T INITIALIZE DISPLAY/
004503  040  040  123  SSF:    .ASCIZ  /  START-NOP-STOP FAILURE/
004534  040  040  122  RJF:    .ASCIZ  /  RESUME-NOP-JUMP FAILURE/
004566  040  040  112  JSF:    .ASCIZ  /  JUMP SELF FAILURE/
004612  040  040  105  ESF:    .ASCIZ  /  EXTERNAL STOP FAILURE/
004642  040  040  120  PDXI:   .ASCIZ  /  PIXEL DATA XFER INCORRECT/
004676  040  040  116  OPCF:   .ASCIZ  /  NULL OPCODE FAILURE/
004724  045  116  045  OPCFX:  .ASCIZ  /%N%A  OPCODE: %06/
004746  040  040  122  RELF:   .ASCIZ  /  RELOCATION FAILURE/
004773  045  116  045  RELFX:  .ASCIZ  /%N%A  INITIAL DPC: %06%A, RELOC(DSR): %06%N/
005047  040  040  101  APF:    .ASCIZ  /  ABS POINT FAILURE/
005073  040  040  114  LVF:    .ASCIZ  /  LONG VECTOR FAILURE/
005121  045  116  045  LONGX:  .ASCIZ  /%N%A  ORIGIN: %04%A,%04%A  DX,DY: %06%A,%06/
005175  040  040  122  RPF:    .ASCIZ  /  REL POINT FAILURE/
005221  040  040  123  SVF:    .ASCIZ  /  SHORT VECTOR FAILURE/
005250  045  116  045  SHRTX:  .ASCIZ  /%N%A  ORIGIN: %04%A,%04%A  DXY: %06/
005314  040  040  107  GHXF:   .ASCIZ  /  GRAPH-HISTO X FAILURE/
005344  040  040  107  GHYF:   .ASCIZ  /  GRAPH-HISTO Y FAILURE/
005374  045  116  045  GHXYX:  .ASCIZ  /%N%A  AMPL: %06%A  INCR: %02/
005431  040  040  102  BMF:    .ASCIZ  /  BIT MAP FAILURE/
005453  045  116  045  BMX:    .ASCIZ  /%N%A  OPCODE: %06/
005475  040  040  103  CHRF:   .ASCIZ  /  CHAR MODE FAILURE/
005521  045  116  045  CHRFX:  .ASCIZ  /%N%A  BASE: %06%A  CHAR: %03/
005556  040  040  124  PRBHNG: .ASCII  /  TEST ABORTED/
005574  040  040  120  PRBH:   .ASCIZ  /  PIXEL-READ-BACK FAILS OR HANGS DPU/
005641  040  040  111  IMDI:   .ASCIZ  /  IMAGE MEMORY DATA INCORRECT/
005677  045  101  040  IMDIX:  .ASCII  /%A  ADDRESS (X,Y): %04%A,%04/
005733  045  116  045         .ASCIZ  /%N%A  FIELD: %04%A  EXP'D: %04%A  REC'D: %04/
006010  045  116  045  IMDIX2: .ASCIZ  /%N%A  SUSPECT RAM GROUP %01%A, IMAGE MEMORY CHANNEL: %01/
```

```
1988 006101  045  116  045  IMDIXC: .ASCII  /%N%A  ADDRESS (X,Y): %04%A,%04/
1989 006137  045  116  045          .ASCIZ  /%N%A  FIELD: %04%A  EXP'D: %04%A  REC'D: %04/
1990 006214  045  116  045          .ASCIZ  /%N%A  SUSPECT RAM GROUP %01%A,  IMAGE MEMORY CHANNEL: %01/
1991 006306  040  040  114  NLRI:   .ASCIZ  /  LUT READ INTERRUPT EXPECTED, NOT RECEIVED/
1992 006362  040  040  114  NLWI:   .ASCIZ  /  LUT WRITE INTERRUPT EXPECTED, NOT RECEIVED/
1993 006437  040  040  125  ULRI:   .ASCIZ  /  UNEXPECTED LUT READ INTERRUPT/
1994 006477  040  040  125  ULWI:   .ASCIZ  /  UNEXPECTED LUT WRITE INTERRUPT/
1995 006540  045  101  040  LINTX:  .ASCIZ  /%A  CSR: %06/
1996 006555  040  040  114  LAIER:  .ASCIZ  /  LUT AUTO-INCR INCORRECT ON READ/
1997 006617  040  040  114  LAIEW:  .ASCIZ  /  LUT AUTO-INCR INCORRECT ON WRITE/
1998 006662  040  040  114  LDINC:  .ASCIZ  /  LUT DATA INCORRECT/
1999 006707  045  101  040  LRWX:   .ASCIZ  /%A  CSR: %06%A  EXP'D: %06%A  REC'D: %06/
2000 006760  040  040  042  NSINIT: .ASCIZ  /  "SOFT-INIT" DIDN'T INITIALIZE THE DPU/
2001 007030  040  040  042  BRINIT: .ASCIZ  /  "BUS-RESET" DIDN'T INITIALIZE THE DPU/
2002 007100  040  040  104  IDPCE:  .ASCIZ  #  DPC READ/WRITE ERROR#
2003 007127  040  040  104  IDSRE:  .ASCIZ  #  DSR READ/WRITE ERROR#
2004 007156  040  040  103  ICSRE:  .ASCIZ  #  CSR READ/WRITE ERROR#
2005 007205  040  040  115  IMRRE:  .ASCIZ  #  MRR READ/WRITE ERROR#
2006 007234  040  040  115  IMPWPE: .ASCIZ  #  MPM WRITE/MWP (HBASE) READ ERROR#
2007 007277  040  040  115  IMPME:  .ASCIZ  #  MPM READ/WRITE ERROR#
2008 007326  040  040  130  IXRRE:  .ASCIZ  #  XRR READ/WRITE ERROR#
2009 007355  040  040  130  IXPME:  .ASCIZ  #  XPM READ/WRITE ERROR#
2010 007404  040  040  130  IXPWPE: .ASCIZ  #  XPM WRITE/XWP (HBASE) READ ERROR#
2011 007447  040  040  110  IHBAE:  .ASCIZ  #  HBA READ/WRITE ERROR#
2012 007476  040  040  103  ICBAE:  .ASCIZ  #  CBA READ/WRITE ERROR#
2013 007525  040  040  104  JPDPCE: .ASCIZ  #  DJMS/DPOP FAILURE (DPC)#
2014 007557  040  040  104  JPPCSE: .ASCIZ  #  DJMS/DPOP FAILURE (PCS)#
2015 007611  040  040  104  DPDSE:  .ASCIZ  /  DPC(W),DSR(R) DUAL ADRS FAULT/
2016 007651  040  040  104  DSDPE:  .ASCIZ  /  DSR(W),DPC(R) DUAL ADRS FAULT/
2017 007711  040  040  104  DSDXE:  .ASCIZ  /  DSR(W),DXR(R) DUAL ADRS FAULT/
2018 007751  040  040  104  DSDYE:  .ASCIZ  /  DSR(W),DYR(R) DUAL ADRS FAULT/
2019 010011  040  040  104  DSPCE:  .ASCIZ  /  DSR(W),PCS(R) DUAL ADRS FAULT/
2020 010051  040  040  104  DSFLE:  .ASCIZ  /  DSR(W),FLG(R) DUAL ADRS FAULT/
2021 010111  040  040  104  DSCSE:  .ASCIZ  /  DSR(W),CSR(R) DUAL ADRS FAULT/
2022 010151  040  040  104  DSMRE:  .ASCIZ  /  DSR(W),MRR(R) DUAL ADRS FAULT/
2023 010211  040  040  104  DSMPE:  .ASCIZ  /  DSR(W),MPM(R) DUAL ADRS FAULT/
2024 010251  040  040  104  DSXRE:  .ASCIZ  /  DSR(W),XRR(R) DUAL ADRS FAULT/
2025 010311  040  040  104  DSXPE:  .ASCIZ  /  DSR(W),XPM(R) DUAL ADRS FAULT/
2026 010351  040  040  103  CSDSE:  .ASCIZ  /  CSR(W),DSR(R) DUAL ADRS FAULT/
2027 010411  040  040  103  CSPCE:  .ASCIZ  /  CSR(W),PCS(R) DUAL ADRS FAULT/
2028 010451  040  040  103  CSFLE:  .ASCIZ  /  CSR(W),FLG(R) DUAL ADRS FAULT/
2029 010511  040  040  103  CSMRE:  .ASCIZ  /  CSR(W),MRR(R) DUAL ADRS FAULT/
2030 010551  040  040  103  CSMPE:  .ASCIZ  /  CSR(W),MPM(R) DUAL ADRS FAULT/
2031 010611  040  040  103  CSXRE:  .ASCIZ  /  CSR(W),XRR(R) DUAL ADRS FAULT/
2032 010651  040  040  103  CSXPE:  .ASCIZ  /  CSR(W),XPM(R) DUAL ADRS FAULT/
2033 010711  040  040  115  MRDSE:  .ASCIZ  /  MRR(W),DSR(R) DUAL ADRS FAULT/
2034 010751  040  040  115  MRPCE:  .ASCIZ  /  MRR(W),PCS(R) DUAL ADRS FAULT/
2035 011011  040  040  115  MRFLE:  .ASCIZ  /  MRR(W),FLG(R) DUAL ADRS FAULT/
2036 011051  040  040  115  MRCSE:  .ASCIZ  /  MRR(W),CSR(R) DUAL ADRS FAULT/
2037 011111  040  040  115  MRMPE:  .ASCIZ  /  MRR(W),MPM(R) DUAL ADRS FAULT/
2038 011151  040  040  115  MRXRE:  .ASCIZ  /  MRR(W),XRR(R) DUAL ADRS FAULT/
2039 011211  040  040  115  MRXPE:  .ASCIZ  /  MRR(W),XPM(R) DUAL ADRS FAULT/
2040 011251  040  040  115  MPDSE:  .ASCIZ  /  MPM(W),DSR(R) DUAL ADRS FAULT/
2041 011311  040  040  115  MPPCE:  .ASCIZ  /  MPM(W),PCS(R) DUAL ADRS FAULT/
2042 011351  040  040  115  MPFLE:  .ASCIZ  /  MPM(W),FLG(R) DUAL ADRS FAULT/
2043 011411  040  040  115  MPCSE:  .ASCIZ  /  MPM(W),CSR(R) DUAL ADRS FAULT/
2044 011451  040  040  115  MPMRE:  .ASCIZ  /  MPM(W),MRR(R) DUAL ADRS FAULT/
```

```
2045 011511  040  040  115  MPXRE:  .ASCIZ  /  MPM(W),XRR(R) DUAL ADRS FAULT/
2046 011551  040  040  115  MPXPE:  .ASCIZ  /  MPM(W),XPM(R) DUAL ADRS FAULT/
2047 011611  040  040  130  XRDSE:  .ASCIZ  /  XRR(W),DSR(R) DUAL ADRS FAULT/
2048 011651  040  040  130  XRPCE:  .ASCIZ  /  XRR(W),PCS(R) DUAL ADRS FAULT/
2049 011711  040  040  130  XRFLE:  .ASCIZ  /  XRR(W),FLG(R) DUAL ADRS FAULT/
2050 011751  040  040  130  XRCSE:  .ASCIZ  /  XRR(W),CSR(R) DUAL ADRS FAULT/
2051 012011  040  040  130  XRMRE:  .ASCIZ  /  XRR(W),MRR(R) DUAL ADRS FAULT/
2052 012051  040  040  130  XRMPE:  .ASCIZ  /  XRR(W),MPM(R) DUAL ADRS FAULT/
2053 012111  040  040  130  XRXPE:  .ASCIZ  /  XRR(W),XPM(R) DUAL ADRS FAULT/
2054 012151  040  040  130  XPDSE:  .ASCIZ  /  XPM(W),DSR(R) DUAL ADRS FAULT/
2055 012211  040  040  130  XPPCE:  .ASCIZ  /  XPM(W),PCS(R) DUAL ADRS FAULT/
2056 012251  040  040  130  XPFLE:  .ASCIZ  /  XPM(W),FLG(R) DUAL ADRS FAULT/
2057 012311  040  040  130  XPCSE:  .ASCIZ  /  XPM(W),CSR(R) DUAL ADRS FAULT/
2058 012351  040  040  130  XPMRE:  .ASCIZ  /  XPM(W),MRR(R) DUAL ADRS FAULT/
2059 012411  040  040  130  XPMPE:  .ASCIZ  /  XPM(W),MPM(R) DUAL ADRS FAULT/
2060 012451  040  040  130  XPXRE:  .ASCIZ  /  XPM(W),XRR(R) DUAL ADRS FAULT/
2061 012511  040  040  115  MMMACE: .ASCIZ  /  MAIN MEM MGT ACCESS FAULT/
2062 012545  040  040  101  XMMACE: .ASCIZ  /  AUX MEM MGT ACCESS FAULT/
2063 012600  040  040  115  MMMNXE: .ASCIZ  /  MAIN MEM MGT NON-EXIST MEM FAULT/
2064 012643  045  101  040  DPCERA: .ASCIZ  /%A (DPC)  EXP'D: %06%A, REC'D: %06/
2065 012706  045  101  040  DSRERA: .ASCIZ  /%A (DSR)  EXP'D: %06%A, REC'D: %06/
2066 012751  045  101  040  DXRERA: .ASCIZ  /%A (DXR)  EXP'D: %06%A, REC'D: %06/
2067 013014  045  101  040  DYRERA: .ASCIZ  /%A (DYR)  EXP'D: %06%A, REC'D: %06/
2068 013057  045  101  040  PCSERA: .ASCIZ  /%A (PCS)  EXP'D: %06%A, REC'D: %06/
2069 013122  045  101  040  FLGERA: .ASCIZ  /%A (FLG)  EXP'D: %06%A, REC'D: %06/
2070 013165  045  101  040  CSRERA: .ASCIZ  /%A (CSR)  EXP'D: %06%A, REC'D: %06/
2071 013230  045  101  040  MRRERA: .ASCIZ  /%A (MRR)  EXP'D: %06%A, REC'D: %06/
2072 013273  045  101  040  MPMERA: .ASCIZ  /%A (MPM)  EXP'D: %06%A, REC'D: %06/
2073 013336  045  101  040  XRRERA: .ASCIZ  /%A (XRR)  EXP'D: %06%A, REC'D: %06/
2074 013401  045  101  040  XPMERA: .ASCIZ  /%A (XPM)  EXP'D: %06%A, REC'D: %06/
2075 013444  045  101  040  HBAERA: .ASCIZ  /%A (HBA)  EXP'D: %06%A, REC'D: %06/
2076 013507  045  101  040  CBAERA: .ASCIZ  /%A (CBA)  EXP'D: %06%A, REC'D: %06/
2077 013552  040  040  104  ECE:    .ASCIZ  /  DIDN'T GET EXPECTED ERROR CODE/
2078                         ;MORE ERROR MESSAGES --
2079
2080 013613  040  040  103  CRWRZ: .ASCIZ  ''  CURSOR REGISTER WRITE/READ ZEROS ERROR''
2081 013664  040  040  103  CRWRO: .ASCIZ  ''  CURSOR REGISTER WRITE/READ ONES ERROR''
2082 013734  040  040  103  CRDAX: .ASCIZ  ''  CURSOR REGISTER DUAL ADDRESS - WRITE Y CHANGED X''
2083 014017  040  040  103  CRDAY: .ASCIZ  ''  CURSOR REGISTER DUAL ADDRESS - WRITE X CHANGED Y''
2084 014102  040  040  103  CRWRE: .ASCIZ  ''  CURSOR REGISTER WRITE/READ ERROR''
2085 014145  040  040  112  JSEWRZ: .ASCIZ  ''  JOYSTICK ENABLES WRT/RD 0'S ERROR''
2086 014211  040  040  112  JSEWRO: .ASCIZ  ''  JOYSTICK ENABLES WRT/RD 1'S ERROR''
2087 014255  040  040  112  CEZWE: .ASCIZ  ''  JOYSTICK ENABLES ZERO-THEN-WRITE ERROR''
2088 014326  040  040  112  CESWE: .ASCIZ  ''  JOYSTICK ENABLES SET-THEN-WRITE ERROR''
2089 014376  040  040  112  JENOCL: .ASCIZ  ''  J.S. CHANNEL DECODE ERROR - ENABLES DIDN'T CLEAR''
2090 014461  040  040  122  CSRGSNE: .ASCIZ ''  REGISTER ERROR AFTER J.S. SWITCH SET, NO ENABLE''
2091 014543  040  040  122  CSRGENS: .ASCIZ ''  REGISTER ERROR AFTER J.S. SW ENABLE SET, NO SWITCH''
2092 014630  040  040  111  CSIFC:  .ASCIZ  ''  INTERNAL REG. ERROR AFTER CLEARING J.S. SW INTR''
2093 014712  040  040  105  NOSWPI: .ASCIZ  ''  ERROR IN FLAGS REG. -- SHOULD SHOW PENDING SW INTR''
2094 014777  040  040  106  NOSWI:  .ASCIZ  ''  FAULTY J.S. SWITCH INTERRUPT''
2095 015036  040  040  111  SWIBV:  .ASCIZ  ''  INCORRECT VECTOR ON J.S. SWITCH INTERRUPT''
2096 015112  040  040  123  SWNOHLT: .ASCIZ ''  SWITCH INTR REQ DIDN'T HALT DISPLAY PROCESSING - BAD DPC''
2097 015205  040  040  106  SWNOCU: .ASCIZ  ''  FAULTY CURSOR RETRIEVAL ON J.S. SW INTR REQ''
2098
2099 015263  040  040  115  NOCMI:  .ASCIZ  ''  MATCH INTERRUPT FAULTY OR NOT RECEIVED''
2100 015334  040  040  116  NOMST:  .ASCIZ  ''  NO STOP ON MATCH INTERRUPT''
2101 015371  040  040  115  CMNOMI: .ASCIZ  ''  MATCH BUT NO MATCH INTERRUPT''
```

```
2102 015430 040 040 111 CMIVB:  .ASCIZ  "  INCORRECT VECTOR ON CURSOR MATCH INTERRUPT"
2103 015505 040 040 127 MATPCS: .ASCIZ  "  WRONG DPC OR DSR FOR MATCH STOP"
2104 015547 040 040 127 MATXYE: .ASCIZ  "  WRONG DXR OR DYR FOR MATCH STOP"
2105 015611 040 040 127 MATPOS: .ASCIZ  "  WRONG DXR OR DYR AFTER POSITION-RESTORE"
2106 015663 040 040 111 MATIRE: .ASCIZ  "  INTERNAL REGISTER INCORRECT FOR MATCH"
2107 015733 040 040 111 MEREG:  .ASCIZ  "  INTERNAL REGISTER INCORRECT AFTER SETUP FOR MATCH"
2108 016017 040 040 106 MATFLE: .ASCIZ  "  FLAGS REGISTER INCORRECT FOR MATCH"
2109 016064 040 040 125 MATUI:  .ASCIZ  "  UNEXPECTED INTERRUPT ON RESUME AFTER MATCH"
2110 016141 040 040 104 MENDPCS: .ASCIZ "  DPC OR DSR WRONG ON AFTER RESUME FOLLOWING MATCH"
2111 016224 040 040 103 CMPE:   .ASCIZ  "  CURSOR MATCHED AT WRONG POSITION"
2112 016267 040 040 116 NOMAT:  .ASCIZ  "  NO CURSOR MATCH DETECTED"
2113 016322 045 116 045 EXPGT2: .ASCIZ  /%N%A EXP'D: %06%A, %06%N%A REC'D: %06%A, %06/
2114 016377 045 116 045 FCOORD: .ASCII  /%N%A CURSOR  (X,Y)  = %06%A, %06/
2115 016437 045 116 045         .ASCIZ  /%N%A REC'D DXR, DYR = %06%A, %06/
2116 016500 045 116 045 FSYCHAN:        .ASCIZ  "%N%A  SYNC GEN/JOYSTICK CHANNEL = %01"
2117
2118
2119 016546 000         NUL:    .ASCIZ  //
2120 016547 045 116 000 NULCR:  .ASCIZ  /%N/
2121 016552 040 040 122 RLXIE:  .ASCIZ  /  RUN-LENGTH X-INCREMENT FAILURE/
2122 016613 040 040 122 RLX2IE: .ASCIZ  /  RUN-LENGTH X-DOUBLE INCREMENT FAILURE/
2123 016663 040 040 122 RLYUE:  .ASCIZ  /  RUN-LENGTH Y-UP FAILURE/
2124 016715 040 040 122 RLYDE:  .ASCIZ  /  RUN-LENGTH Y-DOWN FAILURE/
2125 016751 040 040 122 RLY2UE: .ASCIZ  /  RUN-LENGTH Y-SKIP UP FAILURE/
2126 017010 040 040 122 RLY2DE: .ASCIZ  /  RUN-LENGTH Y-SKIP DOWN FAILURE/
2127 017051 045 101 040 EXPGOT: .ASCIZ  /%A EXP'D: %06%A, REC'D: %06/
2128 017105 045 101 040 DUAD12: .ASCIZ  /%A REG(W) WRITTEN TO: %06%A REG(R) READ; EXP'D: %06%A, REC'D: %06/
2129 017207 040 040 115 INLE:   .ASCIZ  /  MEM INTERLACE ERROR/
2130 017235 045 101 040 INLEXF: .ASCIZ  /%A  PIXEL READBACK EXP'D: %06%A, REC'D: %06%A, CHAN: %01/
2131 017326 040 040 120 SCPRE:  .ASCIZ  /  PIXEL READBACK ERROR/
2132 017355 040 040 123 SCIDE:  .ASCIZ  /  SYNC CHAN INTERLACE DIFFERENCE/
2133 017416 040 040 103 SCME:   .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
2134 017461 045 101 040 SCMFG:  .ASCIZ  /%A  MFG MEM 0: %06%A, MFG SYNC CHAN 0: %06/
2135 017534 045 116 045 SCM:    .ASCII  /%N%A  MEMORYS:/
2136 017552 045 101 040         .ASCII  /%A  0 = %06/
2137 017565 045 101 054         .ASCII  /%A,  1 = %06/
2138 017601 045 101 054         .ASCII  /%A,  2 = %06/
2139 017615 045 101 054         .ASCIZ  /%A,  3 = %06/
2140 017632 045 116 045 SCS:    .ASCII  /%N%A  SYNC CHANS:/
2141 017653 045 101 040         .ASCII  /%A  0 = %06/
2142 017666 045 101 054         .ASCII  /%A,  1 = %06/
2143 017702 045 101 054         .ASCII  /%A,  2 = %06/
2144 017716 045 101 054         .ASCIZ  /%A,  3 = %06%N/
2145 017735 045 116 045 NOMAN:  .ASCIZ  /%N%A**** MANUAL INTERVENTION NOT ALLOWED -- ABORTING ****%N/
2146                            .EVEN
```

```
                                .SBTTL  GLOBAL ERROR REPORT SECTION

                                ;++
                                ; THE GLOBAL ERROR REPORT SECTION CONTAINS THE PRINTB AND PRINTX
                                ; CALLS THAT ARE USED IN MORE THAN ONE TEST.
                                ; ASCII TEXT STRINGS ARE FOUND IN THE GLOBAL TEXT SECTION.
                                ;--

020032                          BGNMSG  NXRERR          ;NON-EXISTANT DEVICE REGISTER.
020032                                  PRINTX  #NXRX,NODEV     ;NODEV = NEXM ADDRESS.
020056  004737  022276                  JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
020062                                  ENDMSG

020064                          BGNMSG  PCSERR          ;BAD DPC AND/OR DSR SIGNATURE.
020064                                  PRINTX  #PCSX,R2,R3,@DPC,@DSR ;R2,R3 = EXP PC,SR
020120  004737  022276                  JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
020124                                  ENDMSG

020126                          BGNMSG  XYERR           ;BAD DXR AND/OR DYR SIGNATURE.
020126                                  PRINTX  #XYX,R4,R5,@DXR,@DYR ;R4,R5 = EXP X,Y
020162  004737  022276                  JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
020166                                  ENDMSG

020170                          BGNMSG  RLXYE           ;BAD DXR AND/OR DYR SIGNATURE.
020170                                  PRINTX  #XYX,R1,R3,R2,R4        ; R1,R3 = EXP, R2,R4 = REC.
020220  004737  022276                  JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
020224                                  ENDMSG

020226                          BGNMSG  INTERR          ;DISPLAY INTERRUPT ERROR SIGNATURE.
020226                                  PRINTX  #INTX,R1,@DPC   ;R1 = CPU PC
020254                                  PRINTX  #NULCR          ; PRINT A NEWLINE
020274  122737  000377  026107          CMPB    #^O377,INTFLAG  ;WAS ANY INTERRUPT GENERATED?
020302  001011                          BNE     1$              ;BR IF YES
020304                                  PRINTX  #FNOINTR                ; NO -- REPORT THE FACT
020324  000465                          BR      20$
020326  132737  000001  026107  1$:     BITB    #IOKSTP,INTFLAG ; WAS IT ÁN UNEXPECTED "STOP" INTERRUPT?
020334  001411                          BEQ     2$              ; BR IF NO.
020336                                  PRINTX  #FUSI           ; YES -- REPORT IT
020356  000450                          BR      20$
020360  132737  000002  026107  2$:     BITB    #IOKJSM,INTFLAG ; WAS IT AN UNEXPECTED 'MATCH" INTERRUPT?
020366  001411                          BEQ     3$              ; BR IF NO.
020370                                  PRINTX  #FUMI           ; REPORT IF YES.
020410  000433                          BR      20$
020412  132737  000010  026107  3$:     BITB    #IOKJSS,INTFLAG ; WAS IT AN UNEXPECTED "SWITCH" INTERRUPT?
020420  001411                          BEQ     4$              ; BR IF NO
020422                                  PRINTX  #FUSWI          ; REPORT IF YES
020442  000416                          BR      20$
020444  132737  000004  026107  4$:     BITB    #IOKERR,INTFLAG ; WAS IT AN UNEXPECTED 'ERROR" INTERRUPT?
020452  001412                          BEQ     20$
020454                                  PRINTX  #FUTO           ; REPORT IT.
020474  004737  023516                  JSR     PC,ERICHK       ;CHECK THE ERROR CODE
020500  004737  022276          20$:    JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
020504                                  ENDMSG

020506                          BGNMSG  IMDERR          ;IMAGE MEM DATA ERROR.
020506  010137  072506                  MOV     R1,IMCRAM               ;TRANSLATE X ADDR...
020512  006237  072506                  ASR     IMCRAM          ;...TO RAM NUMBER.
```

```
2205 020516                                    PRINTX  #IMDIX,R1,R2,PDF,R3,R4
2206 020552                                    PRINTX  #IMDIX2,IMCRAM,IMC.CH
2207 020602  004737  022276                    JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
2208 020606                                    ENDMSG
2209
2210 020610                            BGNMSG  LINTER          ; LUT INTERRUPT ERROR.
2211 020610                                    PRINTX  #LINTX,@LSR     ; SHOW CSR.
2212 020634  004737  022276                    JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
2213 020640                                    ENDMSG
2214
2215 020642                            BGNMSG  LRWERR          ; LUT READ/WRITE ERROR.
2216 020642                                    PRINTX  #LRWX,@LSR,R1,R2 ; CSR, EXP'D, REC'D.
2217 020672  004737  022276                    JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
2218 020676                                    ENDMSG
2219
2220 020700                            BGNMSG  EXPREC
2221 020700                                    PRINTX  #EXPGOT,R1,R2           ; R1=EXP'D, R2=REC'D.
2222 020724  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQUIRED.
2223 020730                            ENDMSG
2224
2225 020732                            BGNMSG  DPCER
2226 020732                                    PRINTX  #DPCERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2227 020756  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2228 020762                            ENDMSG
2229
2230 020764                            BGNMSG  DSRER
2231 020764                                    PRINTX  #DSRERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2232 021010  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2233 021014                            ENDMSG
2234
2235 021016                            BGNMSG  DXRER
2236 021016                                    PRINTX  #DXRERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2237 021042  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2238 021046                            ENDMSG
2239
2240 021050                            BGNMSG  DYRER
2241 021050                                    PRINTX  #DYRERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2242 021074  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2243 021100                            ENDMSG
2244
2245 021102                            BGNMSG  PCSER
2246 021102                                    PRINTX  #PCSERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2247 021126  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2248 021132                            ENDMSG
2249
2250 021134                            BGNMSG  FLGER
2251 021134                                    PRINTX  #FLGERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2252 021160  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2253 021164                            ENDMSG
2254
2255 021166                            BGNMSG  CSRER
2256 021166                                    PRINTX  #CSRERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2257 021212  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2258 021216                            ENDMSG
2259
2260 021220                            BGNMSG  MRRER
2261 021220                                    PRINTX  #MRRERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
```

```
2262 021244  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2263 021250                            ENDMSG
2264
2265 021252                            BGNMSG  MPMER
2266 021252                                    PRINTX  #MPMERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2267 021276  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2268 021302                            ENDMSG
2269
2270 021304                            BGNMSG  XRRER
2271 021304                                    PRINTX  #XRRERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2272 021330  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2273 021334                            ENDMSG
2274
2275 021336                            BGNMSG  XPMER
2276 021336                                    PRINTX  #XPMERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2277 021362  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2278 021366                            ENDMSG
2279
2280 021370                            BGNMSG  HBAER
2281 021370                                    PRINTX  #HBAERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2282 021414  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2283 021420                            ENDMSG
2284
2285 021422                            BGNMSG  CBAER
2286 021422                                    PRINTX  #CBAERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2287 021446  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2288 021452                            ENDMSG
2289
2290 021454                            BGNMSG  DUADRE
2291 021454                                    PRINTX  #DUAD12,R3,R1,R2        ; R3=WRITTEN, R1=EXP'D, R2=REC'D.
2292 021502  004737  022276                    JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2293 021506                            ENDMSG
2294
2295 021510                            BGNMSG  SYSCON
2296 021510  005737  002512                    TST     MFGFLG                  ; MFG MODE?
2297 021514  001414                            BEQ     1$                      ; NO.
2298 021516                                    PRINTX  #SCMFG,MFGM0,MFGS0
2299 021546                            1$:     PRINTX  #SCM,MEMTAB,MEMTAB+2,MEMTAB+4,MEMTAB+6
2300 021606                                    PRINTX  #SCS,SYCTAB,SYCTAB+2,SYCTAB+4,SYCTAB+6
2301 021646                            ENDMSG
2302
2303 021650                            BGNMSG  MMMAE
2304 021650                                    PRINTX  #MRRERA,R3,R4
2305 021674                                    PRINTX  #NULCR
2306 021714                                    PRINTX  #DPCERA,R1,R2
2307 021740                                    PRINTX  #NULCR                  ;PRINT BLANK LINE
2308 021760                            ENDMSG
2309
2310
2311 021762                            BGNMSG  XMMAE
2312 021762                                    PRINTX  #XRRERA,R3,R4
2313 022006                                    PRINTX  #NULCR
2314 022026                                    PRINTX  #DPCERA,R1,R2
2315 022052                                    PRINTX  #NULCR                  ;PRINT BLANK LINE
2316 022072                            ENDMSG
2317
2318
```

```
2319 022074                                BGNMSG  EXPRC2
2320 022074                                        PRINTX  #EXPGT2,R1,R3,R2,R4     ;R1,R3 = EXP'D, R2,R4 = REC'D
2321 022124  004737  022276                        JSR     PC,EXTEND               ; PRINT EXTENSION, IF REQ'D.
2322 022130                                        ENDMSG
2323
2324 022132                                BGNMSG  PNTCOOR
2325 022132                                        PRINTX  #FCOORD,CSMF4X,CSMF4Y,SDXR,SDYR
2326 022172  004737  022276                        JSR     PC,EXTEND
2327 022176                                        ENDMSG
2328
2329 022200                                BGNMSG  PNTSYCH
2330 022200                                        PRINTX  #FSYCHAN,CSCHAN
2331 022224                                        ENDMSG
2332
2333 022226                                BGNMSG  INLEX
2334 022226                                        PRINTX  #INLEXF,R1,R2,R5
2335 022254                                        PRINTX  #NULCR          ;PRINT BLANK LINE
2336 022274                                ENDMSG
2337                                       ;
2338                                       ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
2339                                       ; TO ANY OF THE ABOVE ERROR SIGNATURES.
2340                                       ;
2341 022276  005727                       EXTEND: TST     (PC)+
2342 022300  000000                       EXTA:   0                       ; 0 = NO EXTENSION.
2343 022302  001402                               BEQ     1$
2344 022304  004777  177770                        JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
2345 022310                                1$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
2346 022330  000207                               RTS     PC
```

```
                                        .SBTTL  INTERNAL REGISTER CHECKER, ERROR CODE HANDLER, DPINIT

                                ;ROUTINES TO CHECK CONTENTS OF INTERNAL REGISTERS.
                                ;  UPON ENTRY, THE EXPECTED DATA SHOULD BE IN LOCATION ''GDDAT''
                                ;
                                ;MACRO TO ASSEMBLE REGISTER CHECK FRONT-END:
                                        .MACRO  CHECK   RGS,RNAM
                                        MOV     #SEL'RGS,-(SP)  ;GET REGISTER-SELECT CODE ONTO THE STACK.
                                        MOV     #99$,-(SP)      ;AND THE ADDRESS OF THE REGISTER-NAME MESSAGE.
                                        JMP     IREGCK          ;THEN GO TO COMMON HANDLER.
                                99$:    .ASCIZ  /%A ''RNAM''/
                                        .EVEN
                                        .ENDM   CHECK

022332                          DSRCHK: CHECK   DSR,<DSR>                ;CHECK THE DSR
022360                          PCSCHK: CHECK   PCS,<PCSAVE>             ;...PCSAVE
022410                          FLGCHK: CHECK   FLG,<FLAGS>              ;...FLAGS
022440                          CSRCHK: CHECK   CSR,<CSR>                ;...CSR
022466                          MMCHK:  CHECK   MRR,<MAIN MEM MGMT>              ;...MAIN-SEG MEM. MGMT.
022526                          HBCHK:  CHECK   HBA,<HISTOGRAM BASE>             ;...HISTOGRAM BASE
022566                          XMCHK:  CHECK   XRR,<AUX MEM MGMT>               ;...AUX-SEG MEM. MGMT.
022624                          CBCHK:  CHECK   CBA,<CHARACTER BASE>             ;...CHARACTER BASE

022664  040     040     111     RGE:    .ASCIZ  /  INCORRECT DATA IN INTERNAL REGISTER .../
022736  045     116     045     GDBAD:  .ASCIZ  /%N%A  EXP'D: %O6%A  REC'D: %O6/
022775  045     116     045     TINERR: .ASCIZ  /%N%A  TEST: /
023012  045     116     045     SIMSG:  .ASCIZ  /%N%A  ... AFTER DOING SOFT INIT./
                                        .EVEN

023054  000000                  LOOPFL::        0               ;FLAG OR DATA FOR TEST-LOOP-DEPENDENT STUFF.
023056  000000                  GDDAT:: 0       ;STORAGE FOR EXPECTED DATA.
023060  000000                  BADDAT: 0       ;STORAGE FOR ACTUAL DATA.
023062  000000                  REGNAM: 0       ;ADDRESS OF REGISTER NAME STRING.
023064  000000                  SIFLAG: 0       ;FLAG TO SAY DOING SOFT INIT.

                                ;COME HERE WITH (SP)=ADDRESS OF NAME STRING, 2(SP)=SELECT CODE.
023066  012637  023062          IREGCK: MOV     (SP)+,REGNAM            ;GET STRING ADDRESS.
023072  012677  157736                  MOV     (SP)+,@DSR              ;SELECT THE REGISTER.
023076  017737  157732  023060        MOV     @DSR,BADDAT             ;GET THE DATA.
023104  012777  000000  157722        MOV     #SELDSR,@DSR            ;SELECT THE REAL DSR
023112  023737  023056  023060        CMP     GDDAT,BADDAT            ;COMPARE EXPECTED W/ ACTUAL.
023120  001410                        BEQ     1$                      ;OK IF EQUAL; JUST EXIT.
023122                                HRDERR  RGE,REGERR              ;IF NOT, SAY SO & PRINT DATA.
023142  000207                  1$:     RTS     PC

023144                          BGNMSG  REGERR
023144                                  PRINTX  REGNAM          ;PRINT THE REGISTER NAME.
023164                                  PRINTX  #GDBAD,GDDAT,BADDAT     ;AND THE DATA.
023214                                  PRINTX  #TINERR
023234                                  PRINTX  TNAM            ;PRINT TEST NAME
023254  005737  023064                  TST     SIFLAG          ;SOFT INIT?
023260  001410                          BEQ     2$              ;BR IF NO
023262                                  PRINTX  #SIMSG          ;PRINT IF YES
023302                          2$:     PRINTX  #NULCR          ;PRINT BLANK LINE
023322                                  ENDMSG
```

```
                                PTINER:
                                        PRINTX  #TINERR         ;PRINT "TEST:"
                                        PRINTX  TNAM            ;PRINT NAME OF TEST
                                        PRINTX  #NULCR          ;PRINT BLANK LINE
023404  000207                          RTS     PC

                                ;ROUTINES TO CHECK FOR THE INITIALIZED STATE OF INTERNAL REGISTERS:
                                ;    IIRCHK  CHECKS THEM ALL.
                                ;    IIRCNF  CHECKS ALL BUT THE FLAGS REGISTER.
                                ;    IIRCNC  CHECKS ALL BUT THE CSR
                                ;    IIRNFC  CHECKS ALL BUT FLAGS AND CSR.
                                ;
023406  012737  100000  023056  IIRCNC: MOV     #CHAR,GDDAT     ;SEE IF FLAGS HAS "CHAR" MODE ONLY.
023414  004737  022410                  JSR     PC,FLGCHK
023420  000414                          BR      IIRNFC          ;BYPASS CSR CHECK.
023422  012737  100000  023056  IIRCHK: MOV     #CHAR,GDDAT     ;SEE IF FLAGS HAS "CHAR" MODE ONLY.
023430  004737  022410                  JSR     PC,FLGCHK
023434  004737  023524          IIRCNF: JSR     PC,ERRCHK       ;SPECIFICALLY CHECK FOR AN ERROR CODE.
023440  012737  000003  023056          MOV     #SELCSR,GDDAT   ;WE SHOULD ONLY SEE SELECT CODE IN CSR.
023446  004737  022440                  JSR     PC,CSRCHK       ;
023452  012737  000001  023056  IIRNFC: MOV     #1,GDDAT        ;EXPECT TO SEE "EMPTY" STATUS IN PCSAVE.
023460  004737  022360                  JSR     PC,PCSCHK       ;...GO CHECK IT.
023464  005037  023056                  CLR     GDDAT           ;REMAINING REGISTERS S/B CLEAR.
023470  004737  022466                  JSR     PC,MMCHK        ;...MAIN MEM MGMT
023474  004737  022526                  JSR     PC,HBCHK        ;...HISTOGRAM BASE
023500  004737  022566                  JSR     PC,XMCHK        ;...AUX MEM MGMT
023504  004737  022624                  JSR     PC,CBCHK        ;...CHARACTER BASE.
023510  005037  023064                  CLR     SIFLAG          ;CLEAR SOFT-INIT FLAG.
023514  000207                          RTS     PC

                                ;ERRCHK: CHECKS THE CSR FOR AN ERROR CODE & REPORTS IT.
023516  012746  000001          ERICHK: MOV     #1,-(SP)        ; SET FLAG TO INHIBIT USE OF "DFERR"
023522  000401                          BR      ERRCK1
023524  005046                  ERRCHK: CLR     -(SP)           ; CLEAR FLAG
023526  012777  000003  157300  ERRCK1: MOV     #SELCSR,@DSR    ;SELECT THE CSR.
023534  017746  157274                  MOV     @DSR,-(SP)      ;READ IT.
023540  012777  000000  157266          MOV     #SELDSR,@DSR    ;SELECT REAL DSR SO PEOPLE WON'T GET MIXED UP..
023546  011637  023060                  MOV     (SP),BADDAT     ;STORE CSR CONTENTS.
023552  042716  177760                  BIC     #^C17,(SP)      ;EXTRACT THE SELECT!WE FIELD.
023556  022627  000003                  CMP     (SP)+,#SELCSR   ;WE SHOULD SEE THE CSR REG SELECT CODE.
023562  001424                          BEQ     1$
023564  005716                          TST     (SP)            ; SHOULD WE DO DFERR?
023566  001011                          BNE     20$             ; BR IF NO
023570                                  DFERR   NOCSR           ;IF NOT, SAY WE CAN'T READ IT.
023610  000466                          BR      2$              ; GO TO EXIT
023612                          20$:    PRINTX  #FNOCSR         ; PRINT MESSAGE "CAN'T READ CSR"
023632  000455                          BR      2$              ; GO TO EXIT

023634  005737  023060          1$:     TST     BADDAT          ;TEST THE ERROR BIT.
023640  100052                          BPL     2$              ;OK IF CLEAR; JUST EXIT.
023642  000337  023060                  SWAB    BADDAT          ;IF SET, GET ERROR CODE
023646  006237  023060                  ASR     BADDAT
023652  006237  023060                  ASR     BADDAT          ;...INTO BITS<4:1>.
023656  042737  177741  023060          BIC     #^C36,BADDAT    ;ZAP OUT JUNK.
023664  013746  023060                  MOV     BADDAT,-(SP)    ;SAVE ERROR CODE * 2 ON STACK
023670  062716  024266                  ADD     #ERCTBL,(SP)    ;GET TO THE MESSAGE ADDRESS TABLE.
023674  013637  024264                  MOV     @(SP)+,ERCNAM   ;GET THE ADDRESS OF THE REG NAME MESSAGE.
```

```
                                                TST     (SP)            ; USE 'DFERR'?
                                                BNE     11$             ; BR IF NO
                                                HRDERR  ERCM,ERCOD      ;REPORT THE BAD NEWS.
                                                BR      2$
                                        11$:    PRINTX  #FERCM          ; PRINT ''CSR ERROR CODE DETECTED =''
                                                PRINTX  ERCNAM          ; ... AND THE ERROR NAME.
                                        2$:     TST     (SP)+
                                                RTS     PC

                                        FNOCSR: .ASCII  /%N%A  ***/
                                        NOCSR:  .ASCIZ  /  CAN'T READ THE CSR/
                                        FERCM:  .ASCII  /%N%A  ***/
                                        ERCM:   .ASCIZ  /  CSR ERROR CODE REC'D = /
                                                .EVEN
                                        BGNMSG  ERCOD
                                                PRINTX  ERCNAM
                                                CMP     BADDAT,#13*2    ; IS IT 'DBUS DATA R/W ERROR'?
                                                BNE     1$              ;BR IF NO
                                                PRINTX  #GDBAD,@DXR,@DYR  ; YES -- DXR = EXP'D, DYR = REC'D
                                        1$:     PRINTX  #TINERR
                                                PRINTX  TNAM
                                                TST     SIFLAG          ;SOFT INIT?
                                                BEQ     2$              ;BR IF NO
                                                PRINTX  #SIMSG          ;PRINT IF YES
                                        2$:     PRINTX  #NULCR          ;PRINT BLANK LINE
                                                ENDMSG
                                        ERCNAM: 0
                                        ERCTBL: 77$,1$,2$,3$,4$,5$,6$,7$
                                                10$,11$,12$,13$,14$,15$,16$,17$

                                        ;ERROR CODE EXPLANATIONS;
                                        77$:    .ASCIZ  /%A... 0:  (ILLEGAL ERROR CODE)/
                                        1$:     .ASCIZ  /%A... 1:  NXM TIMEOUT ON DMA X'FER/
                                        2$:     .ASCIZ  /%A... 2:  (ILLEGAL ERROR CODE)/
                                        3$:     .ASCIZ  /%A... 3:  MEMORY PROTECTION ERROR/
                                        4$:     .ASCIZ  /%A... 4:  SEQUENCE ERROR/
                                        5$:     .ASCIZ  /%A... 5:  RESERVED OPERATION/
                                        6$:     .ASCIZ  /%A... 6:  VECTOR ENDPOINT MISMATCH/
                                        7$:     .ASCIZ  /%A... 7:  U-PROC EXECUTED UNUSED MICROWORD/
                                        10$:    .ASCIZ  /%A... 10:  IMAGE MEMORY ''SYNC'' TIMEOUT/
                                        11$:    .ASCIZ  /%A... 11:  PIXEL READBACK ''DATA AVAILABLE'' TIMEOUT/
                                        12$:    .ASCIZ  /%A... 12:  J.S. STATUS ''DATA READY'' TIMEOUT/
                                        13$:    .ASCIZ  /%A... 13:  DBUS DATA READ-WRITE ERROR/
                                        14$:    .ASCIZ  /%A... 14:  DBUS SIGNAL HUNG/
                                        15$:    .ASCIZ  /%A... 15:  VBUS SIGNAL HUNG/
                                        16$:    .ASCIZ  /%A... 16:  (ILLEGAL ERROR CODE)/
                                        17$:    .ASCIZ  /%A... 17:  (ILLEGAL ERROR CODE)/
                                                .EVEN

                                        ;ROUTINE TO DO A ''SOFT'' INIT ON THE DPU:
                                                .ENABL  LSB
                                        DPINIT: JSR     PC,SAVERS       ; SAVE R0-R5.
                                                MOV     #1,SIFLAG       ; SAY DOING SOFT INIT
                                                MOV     #0,@DSR         ;SELECT THE REAL DSR
                                                TST     @DSR            ;ARE WE STOPPED?
                                                BMI     11$             ;BR IF YES
                                                JSR     PC,RELEAS       ;SEE IF WE'RE HUNG.
```

```
2519 025434  017700  155372                    MOV     @DPC,R0
2520 025440  057700  155370                    BIS     @DSR,R0
2521 025444  057700  155366                    BIS     @DXR,R0
2522 025450  057700  155364                    BIS     @DYR,R0
2523 025454  001010                           BNE     11$
2524 025456                                    DFERR   VSHUNG,VSHNG    ; REPORT HUNG, ALL REGISTERS 0
2525 025476                             11$:
2526 025476  012777  040000  155332           MOV     #GTJSSW,@DXR    ;CLEAR PENDING INTERRUPT
2527 025504  004737  027454                    JSR     PC,WAITF
2528 025510  012703  000004                    MOV     #4,R3           ;SET COUNTER FOR # OF SWITCHES TO CLEAR
2529 025514  012704  160000                    MOV     #WECC,R4        ;SET UP OP TO CLEAR SWITCHES
2530 025520  010477  155314             12$:   MOV     R4,@DYR         ;CLEAR A SWITCH
2531 025524  004737  027454                    JSR     PC,WAITF
2532 025530  062704  000400                    ADD     #CH1,R4
2533 025534  005303                           DEC     R3
2534 025536  001370                           BNE     12$
2535 025540  012777  040000  155270  DPSINI:  MOV     #GTJSSW,@DXR    ;CLEAR PENDING INTERRUPT
2536 025546  004737  027454                    JSR     PC,WAITF
2537 025552  012777  100000  155260           MOV     #100000,@DYR    ;DO THE INIT.
2538 025560  004737  027454                    JSR     PC,WAITF        ;WAIT FOR A COMPLETE
2539 025564  103402                           BCS     1$              ;OK IF STOPPED.
2540 025566  004737  026756                    JSR     PC,RELEAS       ;TRY TO UNHANG.
2541 025572  005002                     1$:   CLR     R2              ;EXPECT DPC TO BE 0.
2542 025574  012703  100000                    MOV     #BIT15,R3       ;EXPECT DSR TO BE 100000.
2543 025600  005777  155226                    TST     @DPC            ;TEST THE DPC, S/B =0.
2544 025604  001003                           BNE     2$              ;ERROR IF NOT 0.
2545 025606  027703  155222                    CMP     @DSR,R3         ;CHECK DSR.
2546 025612  001410                           BEQ     3$
2547 025614                             2$:   DFERR   NSINIT,PCSERR   ;REPORT ERROR IN DPC OR DSR.
2548 025634  012737  000001  023064  3$:      MOV     #1,SIFLAG       ;SAY DOING SOFT INIT.
2549 025642  004737  023422                    JSR     PC,IIRCHK       ;CHECK THE INTERNAL REGISTERS
2550 025646  005004                           CLR     R4              ; DXR AND DYR SHOULD BE 0.
2551 025650  005005                           CLR     R5              ;
2552 025652  005777  155160                    TST     @DXR
2553 025656  001003                           BNE     4$
2554 025660  005777  155154                    TST     @DYR
2555 025664  001410                           BEQ     5$
2556 025666                             4$:   DFERR   NSINIT,XYERR
2557 025706  004737  031436             5$:   JSR     PC,RESTRS       ; RESTORE R0-R5.
2558 025712  005037  023064                    CLR     SIFLAG
2559 025716  000207                           RTS     PC              ;RETURN TO CALLER
2560                                           .DSABL  LSB
2561
2562 025720                             BGNMSG VSHNG
2563 025720                                    PRINTX  #VSHGE
2564 025740                                    ENDMSG
2565
2566 025742       040       040       104  VSHUNG: .ASCIZ  /  DPU HUNG -- ALL REGISTERS READ ZERO/
2567 026010       045       101       040  VSHGE:  .ASCIZ  /%A   BAD M7064 OR NO B-CLOCK OR VBLANK FROM M7061%N/
2568                                           .EVEN
2569
2570
2571
2572 026074                             DPRESET: BGNSEG         ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
2573 026076  004737  025402                    JSR     PC,DPINIT       ; DO SOFT INIT.
2574 026102                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
2575 026104  000207                           RTS     PC
```

```
                                .SBTTL  GLOBAL SUBROUTINES SECTION

                                ;++
                                ; THE GLOBAL SUBROUTINES SECTION CONTAINS THE SUBROUTINES
                                ; THAT ARE USED IN MORE THAN ONE TEST.
                                ;--

                                ;
                                ; DEFAULT DISPLAY INTERRUPT HANDLERS.
                                ; IF DISPLAY TIME-OUT, REPORT DEV FATAL, AND ABORT PASS.
                                ; OTHERWISE, SAVE DPU REGISTERS AND DISMISS.
                                ;

                                ; BIT DEFINITIONS FOR "INTMASK" AND "INTFLAG" BYTES:
                                ;
  000200                              IOKCKIN=BIT7    ; DON'T CHECK FOR BAD INTERRUPTS -- TEST WILL.
  000001                              IOKSTP=BIT0     ; EXPECT "STOP" INTERRUPT
  000002                              IOKJSM=BIT1     ; EXPECT JOYSTICK "MATCH" INTERRUPT.
  000004                              IOKERR=BIT2     ; EXPECT "ERROR" INTERRUPT.
  000010                              IOKJSS=BIT3     ; EXPECT "JOYSTICK SWITCH" INTERRUPT.
                                ;
                                ;INTERRUPT MASK -- SAYS EXPECTING INTERRUPTS
026106  000                     INTMASK:        .BYTE   0
                                ;INTERRUPT FLAG -- SAYS WE GOT ONE (IF POSITIVE)
026107  000                     INTFLAG:        .BYTE   0

                                ;SAVED INTERRUPT VECTOR:
026110  000000                  INTVEC: .WORD   0
                                ;SAVE CPU PC
026112  000000                  INTCPC: .WORD   0

                                ;SUBROUTINE TO ENABLE INTERRUPTS:
026114  010046                  ENAINT: MOV     R0,-(SP)        ;SAVE R0
026116  013700  003052                  MOV     STPV,R0         ;GET POINTER TO VECTORS
026122  012720  026242                  MOV     #DSTP,(R0)+     ;SET UP STOP VECTOR
026126  012720  000340                  MOV     #PRI07,(R0)+
026132  012720  026274                  MOV     #DJM,(R0)+      ;SET UP MATCH VECTOR
026136  012720  000340                  MOV     #PRI07,(R0)+
026142  012720  026210                  MOV     #DTO,(R0)+      ;SET UP ERROR VECTOR
026146  012720  000340                  MOV     #PRI07,(R0)+
026152  012720  026326                  MOV     #DJS,(R0)+      ;SET UP JOYSTICK SWITCH VECTOR
026156  012710  000340                  MOV     #PRI07,(R0)
026162  012600                          MOV     (SP)+,R0        ;RESTORE R0
026164  011646                          MOV     (SP),-(SP)
026166  012766  000000  000002          MOV     #0,2(SP)        ;SET CPU TO LEVEL 0
026174  000002                          RTI

                                ;SUBROUTINE TO DISABLE INTERRUPTS (RAISE PRIORITY TO LEVEL 7)
026176  011646                  DSBINT: MOV     (SP),-(SP)
026200  012766  000340  000002          MOV     #PRI07,2(SP)
026206  000002                          RTI

                                ;ERROR (TIME-OUT) INTERRUPT SERVICE:
026210  013737  003056  026110  DTO:     MOV     TOTV,INTVEC     ;SAVE THE VECTOR
026216  105037  026107                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
026222  132737  000004  026106          BITB    #IOKERR,INTMASK ;EXPECTING ERROR INTERRUPT?
026230  001053                          BNE     DPUSAV          ;BR IF YES; NO ERROR.
```

```
2634 026232  152737  000004  026107          BISB    #IOKERR,INTFLAG ;NO. SET THE ERROR FLAG.
2635 026240  000447                          BR      DPUSAV          ;GO FINISH UP.
2636
2637 026242  013737  003052  026110  DSTP:   MOV     STPV,INTVEC     ;SAVE THE VECTOR
2638 026250  105037  026107                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2639 026254  132737  000001  026106          BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2640 026262  001036                          BNE     DPUSAV          ;BR IF YES
2641 026264  152737  000001  026107          BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2642 026272  000432                          BR      DPUSAV          ;GO FINISH UP.
2643
2644 026274  013737  003054  026110  DJM:    MOV     JSMV,INTVEC     ;SAVE THE VECTOR
2645 026302  105037  026107                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2646 026306  132737  000002  026106          BITB    #IOKJSM,INTMASK ;EXPECTING MATCH INTERRUPT?
2647 026314  001021                          BNE     DPUSAV          ;BR IF YES
2648 026316  152737  000002  026107          BISB    #IOKJSM,INTFLAG ;NO. SET THE ERROR FLAG.
2649 026324  000415                          BR      DPUSAV          ;GO FINISH UP.
2650
2651 026326  013737  003060  026110  DJS:    MOV     JSSV,INTVEC     ;SAVE THE VECTOR
2652 026334  105037  026107                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2653 026340  132737  000010  026106          BITB    #IOKJSS,INTMASK ;EXPECTING SWITCH INTERRUPT?
2654 026346  001004                          BNE     DPUSAV          ;BR IF YES
2655 026350  152737  000010  026107          BISB    #IOKJSS,INTFLAG ;NO. SET THE ERROR FLAG.
2656 026356  000400                          BR      DPUSAV          ;GO FINISH UP.
2657
2658
2659 026360  011637  026112          DPUSAV: MOV     (SP),INTCPC     ;SAVE CPU PC.
2660 026364  010146                          MOV     R1,-(SP)        ;SAVE R1
2661 026366  013701  003034                  MOV     DSR,R1          ;GET DSR ADDRESS INTO R1
2662 026372  017737  154434  003126          MOV     @DPC,IDPC       ;SAVE DISPLAY PC...
2663 026400  011137  003130                  MOV     (R1),IDSR       ;...STATUS...
2664 026404  017737  154426  003132          MOV     @DXR,IDXR       ;...X POSITION...
2665 026412  017737  154422  003134          MOV     @DYR,IDYR       ;...Y POSITION...
2666 026420  012711  000001                  MOV     #SELPCS,(R1)    ;SAVE PCSAVE...
2667 026424  011137  003146                  MOV     (R1),IPCSAV     ;
2668 026430  012711  000002                  MOV     #SELFLG,(R1)    ;SAVE FLAGS...
2669 026434  011137  003150                  MOV     (R1),IFLAGS     ;
2670 026440  012711  000003                  MOV     #SELCSR,(R1)    ;SAVE CSR...
2671 026444  011137  003152                  MOV     (R1),ICSR       ;
2672 026450  012711  000004                  MOV     #SELMRR,(R1)    ;SAVE MAIN MEM-MGMT..
2673 026454  011137  003154                  MOV     (R1),IMAIN      ;
2674 026460  012711  000005                  MOV     #SELHBA,(R1)    ;SAVE HISTO. BASE...
2675 026464  011137  003156                  MOV     (R1),IHBASE     ;
2676 026470  012711  000006                  MOV     #SELXRR,(R1)    ;SAVE AUX. MEM-MGMT..
2677 026474  011137  003160                  MOV     (R1),IAUX       ;
2678 026500  012711  000007                  MOV     #SELCBA,(R1)    ;SAVE CHAR. BASE...
2679 026504  011137  003162                  MOV     (R1),ICBASE     ;
2680 026510  012711  000000                  MOV     #SELDSR,(R1)    ;END UP AT DSR...
2681 026514  011137  003144                  MOV     (R1),ISDSR      ;...AND SAVE IT.
2682                         ;          BITB    #IOKERR,INTFLAG ; DID WE GET AN ERROR INTERRUPT?
2683 026520  105737  026107                  TSTB    INTFLAG         ; ANY ERROR FLAGS SET?
2684 026524  001424                          BEQ     1$              ; BR IF NO
2685 026526  105737  026106                  TSTB    INTMASK         ; YES. IS CHECKING TURNED OFF?
2686 026532  100421                          BMI     1$              ; BR IF YES
2687 026534  013746  022300                  MOV     EXTA,-(SP)      ; SAVE ERROR EXTENSION ADDRESS.
2688 026540  012737  023324  022300          MOV     #PTINER,EXTA    ; SET EXTENSION TO PRINT TEST IN ERROR.
2689 026546  013701  026112                  MOV     INTCPC,R1       ;GET CPU PC.
2690 026552  004737  030216                  JSR     PC,INCERK       ;INCREMENT THE ERROR COUNTS
```

```
 2691 026556                                        ERRDF   0,IFAULT,INTERR ;REPORT ERROR...
 2692 026566  012637  022300                        MOV     (SP)+,EXTA      ;RESTORE EXTA
 2693 026572  004737  030254                        JSR     PC,CKEMAX
 2694
 2695 026576  012601                        1$:     MOV     (SP)+,R1        ;RESTORE R1
 2696 026600  000002                                RTI                     ;...AND DISMISS.
 2697                                       ;
 2698                                       ; AND A SIMILAR ONE FOR THE LUT INTERRUPTS.
 2699                                       ;
 2700 026602  000137                        LDUN:   JMP     @(PC)+
 2701 026604  026606                                LUTSAV
 2702 026606  017737  154230  003136        LUTSAV: MOV     @LSR,ILSR       ; SAVE LUT REGISTERS.
 2703 026614  017737  154224  003140                MOV     @LDR,ILDR
 2704 026622  017737  154220  003142                MOV     @LMR,ILMR
 2705 026630  000002                                RTI                     ; AND DISMISS.
```

```
                                        ;
                                        ; SUBROUTINE TO INITIALIZE THE CURRENT UNIT DISPLAY.
                                        ; CALL "CLRW" TO INIT WITH IM'S IN WRITE MODE.
                                        ; CALL "CLRRW" TO INIT WITH IM'S IN WRITE-IN-RETRACE MODE.
                                        ; CALL "CLRRW1" TO INIT WITH IM'S IN WRITE-IN-RETRACE, 1'S-ONLY MODE.
                                        ;
                                                .ENABL  LSB


                                        ;THE OPERATIONS ARE PERFORMED VIA PROGRAMMED I/O SO THAT EXECUTION OF A DISPLAY
                                        ; FILE DOES NOT HAVE TO TAKE PLACE.
026632  005037  026754        CLRW:   CLR     CLMFCN          ;SAY CLEAR TO "WRITE-ONLY" MODE.
026636  000407                  BR      1$
026640  012737  176076  026754 CLRRW1: MOV     #RDWRT1,CLMFCN  ;SAY CLEAR TO READ/WRITE-1'S-ONLY.
026646  000403                  BR      1$
026650  012737  176074  026754 CLRRW:  MOV     #RDWRT,CLMFCN   ;SAY CLEAR TO READ/WRITE
                        ;1$:    JSR     PC,DPINIT       ;DO INIT (CLEARS MEM, WRITE MODE, DISABLES J.S.) ;@@@I
026656  012777  100000  154154 1$:     MOV     #BIT15,@DYR     ; DO SOFT INIT TO CLEAR MEMORIES
026664  004737  027454          JSR     PC,WAITF
026670  005737  026754          TST     CLMFCN          ;ANY THING ELSE TO DO?
026674  001426                  BEQ     3$                      ;EXIT IF NOT.
026676  042737  176000  026754        BIC     #176000,CLMFCN  ;ZAP OUT OP-CODE.
026704  052737  040000  026754        BIS     #WRTMSR,CLMFCN  ;INSERT CODE FOR WRITING INTO DYR.

026712                          2$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.

026714  013777  026754  154116        MOV     CLMFCN,@DYR     ;WRITE MEMORY STATUS-BA REGISTER
026722  004737  027454          JSR     PC,WAITF        ;WAIT FOR NOT BUSY.
026726  004737  023524          JSR     PC,ERRCHK       ;CHECK FOR ERROR
026732                          ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.

026734  062737  000400  026754        ADD     #CH1,CLMFCN     ;BUMP THE CHANNEL #.
026742  032737  001400  026754        BIT     #CH3,CLMFCN     ;DONE ALL CHANNELS?
026750  001360                  BNE     2$                      ;LOOP IF NOT.
026752  000207                  3$:     RTS     PC                      ;RETURN WHEN DONE ALL 3 CHANNELS
026754  000000                  CLMFCN: 0               ;FUNCTION FOR CLEARING MEMORY.
                                        .DSABL  LSB
                                        ;
                                        ; ANOTHER TO RELEASE THE DPU.
                                        ; EXECUTE "EXT STOP" IF HE'S RUNNING, "BUS-INIT" IF HE'S HUNG.
                                        ;
026756  012777  000000  154050  RELEAS: MOV     #SELDSR,@DSR    ;MAKE SURE WE SELECT REAL DSR
026764  005777  154044          TST     @DSR            ;EXIT IF DPU IS STOPPED.
026770  100410                  BMI     1$              ;EXIT IF HE'S FREE.
026772  005077  154036          CLR     @DSR            ;XCT "EXTERNAL STOP".
026776  004737  027454          JSR     PC,WAITF
027002  103403                  BCS     1$              ;EXIT IF STOP WORKED.
                                                        ;IT DIDN'T, XCT BUS-INIT...
027004                          BRESET                          ; BUS RESET.
027006  004737  027454          JSR     PC,WAITF        ;...AND WAIT AGAIN.
027012  000207                  1$:     RTS     PC
                                        ;
                                        ; SUBROUTINE TO FILL ALL OF FREE CORE WITH PIXEL DATA
                                        ; FOR BIT MAP FUNCTIONS.  (R0) = DATA PATTERN.
                                        ; REGISTERS ARE NOT SAVED.
                                        ;
```

```
2764 027014  010001            FILL1:  MOV     R0,R1           ; FILL WITH (R0).
2765 027016  000402                    SKP2
2766 027020  010001            FILL2:  MOV     R0,R1
2767 027022  005101                    COM     R1              ; FILL WITH ALTERNATE WORDS.
2768 027024  013702  003240            MOV     FREE,R2         ; POINTER.
2769 027030  013703  003242            MOV     FRESIZ,R3       ; WORD COUNT.
2770 027034  042703  100003            BIC     #100003,R3              ; INSURE WC IS EVEN.
2771 027040  010022            1$:     MOV     R0,(R2)+        ; 1ST WORD...
2772 027042  010122                    MOV     R1,(R2)+        ;...NEXT WORD.
2773 027044  162703  000002            SUB     #2,R3
2774 027050  003373                    BGT     1$
2775 027052  000207                    RTS     PC
```

```
                                        ;
                                        ; SUBROUTINE TO BLAST THE LUT RAM WITH COLOR OR GREY-SCALE DATA.
                                        ;
027054  012701  107222          LUMIN:  MOV     #LUMTBL,R1      ; GREY-SCALE TABLE ADDRESS.
027060  012746  000001                  MOV     #1,-(SP)        ; LUT WORDS / SHADE.
027064  000404                          SKP2+2
027066  012701  110222          NTSC:   MOV     #NTSC8,R1       ; BASIC 8 COLOR TABLE ADDRESS.
027072  012746  000040                  MOV     #32.,-(SP)      ; LUT WORDS / COLOR.
027076  000404                          SKP2+2
027100  012701  110242          EXPR:   MOV     #EXPRM,R1       ; EXPERIMENTAL 16 COLOR TABLE.
027104  012746  000020                  MOV     #16.,-(SP)      ; LUT WORDS / COLOR.

027110  005737  003216                  TST     LUTAV
027114  001424                          BEQ     3$              ; RETURN IF NO LUT INSTALLED.
027116  005077  153720                  CLR     @LSR            ; ADDR 0, NO INTERRUPTS.
027122  052777  000100  153716          BIS     #GCOFF,@LMR     ; GAMMA CORRECTION OFF.
027130  011600                  1$:     MOV     (SP),R0         ; SET ITERATION K.
027132  012177  153706                  MOV     (R1)+,@LDR      ; BLAST THE RAM.
027136  005777  153700          2$:     TST     @LSR            ; WAIT FOR READY.
027142  100375                          BPL     .-4
027144  105777  153672                  TSTB    @LSR            ; LAST ADDRESS DONE ??
027150  001406                          BEQ     3$              ; EXIT IF SO.
027152  005300                          DEC     R0
027154  001765                          BEQ     1$              ; GET NEXT COLOR.
027156  016177  177776  153660          MOV     -2(R1),@LDR     ; REPEAT THIS COLOR.
027164  000764                          BR      2$
027166  005726                  3$:     TST     (SP)+           ; FIX THE STACK...
027170  000207                          RTS     PC              ;...AND RETURN.
                                        ;
                                        ; BLAST 16 SHADES OF THE BASIC COLOR IN (R1).
                                        ; LOCATION "SHADLY" SET BY CALLER TO PRODUCE A PLEASING
                                        ; VISUAL EFFECT.
                                        ;
027172  005737  003216          SHAD16: TST     LUTAV
027176  001433                          BEQ     3$              ; RETURN IF NO LUT INSTALLED.
027200                                  BREAK
027202  005077  153634                  CLR     @LSR            ; ADDR 0, NO INTERRUPTS.
027206  052777  000100  153632          BIS     #GCOFF,@LMR     ; GAMMA CORRECT OFF.
027214  042701  177356                  BIC     #^C421,R1       ; SET DIMMEST SHADE THIS COLOR.
027220  005002                          CLR     R2              ; 1ST SHADE IS ALWAYS BLACK.
027222  012703  000020          1$:     MOV     #16.,R3         ; 16 LUT WORDS / SHADE.
027226  010277  153612          2$:     MOV     R2,@LDR         ; WRITE DATA.
027232  005777  153604                  TST     @LSR
027236  100375                          BPL     .-4
027240  013700  003220                  MOV     SHADLY,R0       ; DELAY.
027244  005300                          DEC     R0
027246  001376                          BNE     .-2
027250  105777  153566                  TSTB    @LSR            ; LUT RAM FULL ??
027254  001404                          BEQ     3$              ; EXIT IF SO.
027256  005303                          DEC     R3              ; THIS SHADE DONE ??
027260  001362                          BNE     2$              ; NOT YET, LOOP.
027262  060102                          ADD     R1,R2           ; YES, INCREMENT THE SHADE...
027264  000756                          BR      1$              ;...AND CONTINUE.
027266  000207                  3$:     RTS     PC
```

```
                                        ;
                                        ; SUBROUTINES TO TRANSLATE AN ABSOLUTE X OR Y TO SHORT FORM
                                        ; DELTA X OR Y.  USED IN SVEC AND RPNT TESTS.
                                        ; INPUT R0 = ABS CO-ORD, IN THE RANGE (1000 +/-76).
                                        ; OUTPUT R0 = DX OR DY CORRECTLY POSITIONED AND SIGNED.
                                        ;
027270  020027  001000          SDY:    CMP     R0,#1000
027274  103003                          BHIS    .+10            ;SKP3 IF POS DELTA.
027276  005400                          NEG     R0              ;MAKE NEG DELTA.
027300  062700  000100                  ADD     #100,R0         ;SET SIGN BIT.
027304  042700  177600                  BIC     #^C177,R0       ;R0 = +/-DY IN <6:0>.
027310  000207                          RTS     PC

027312  004737  027270          SDX:    JSR     PC,SDY          ;MAKE A DY...
027316  000300                          SWAB    R0              ;...MOVE TO HI BYTE.
027320  006200                          ASR     R0              ;R0 = +/-DX IN <13:7>.
027322  000207                          RTS     PC
                                        ;
                                        ; SUBROUTINE TO CONVERT 12 BIT OCTAL TO ASCII FOR DISPLAY.
                                        ; R1 = OCTAL X OR Y CO-ORD AT ENTRY.
                                        ; R1,,R2 = 4 DIGIT ASCII EQUIVALENT AT EXIT.
                                        ; R0, R1, AND R2 NOT SAVED AT ENTRY.
                                        ;
027324  012737  030060  003232  CNVRT:  MOV     #'00,TEMP1      ;ASCII 00
027332  013737  003232  003234          MOV     TEMP1,TEMP2     ;HERE TOO.
027340  042701  170000                  BIC     #^C7777,R1      ;STRIP ANY CRAP BITS.
027344  010102                          MOV     R1,R2
027346  000302                          SWAB    R2
027350  006202                          ASR     R2
027352  042702  177770                  BIC     #^C7,R2
027356  150237  003232                  BISB    R2,TEMP1        ;1ST (HI ORDER) DIGIT.
027362  010102                          MOV     R1,R2
027364  012700  000006                  MOV     #6,R0
027370  006202                  1$:     ASR     R2              ;SHIFT OFF 6 BITS.
027372  005300                            DEC     R0
027374  001375                              BNE     1$
027376  042702  177770                  BIC     #^C7,R2
027402  150237  003233                  BISB    R2,TEMP1+1      ;2ND DIGIT.
027406  010102                          MOV     R1,R2
027410  012700  000003                  MOV     #3,R0
027414  006202                  2$:     ASR     R2              ;SHIFT OFF 3 BITS.
027416  005300                            DEC     R0
027420  001375                              BNE     2$
027422  042702  177770                  BIC     #^C7,R2
027426  150237  003234                  BISB    R2,TEMP2        ;3RD DIGIT.
027432  042701  177770                  BIC     #^C7,R1
027436  150137  003235                  BISB    R1,TEMP2+1      ;4TH (AND LAST) DIGIT.
027442  013701  003232                  MOV     TEMP1,R1        ;HI HALF => R1.
027446  013702  003234                  MOV     TEMP2,R2        ;LO HALF => R2.
027452  000207                          RTS     PC              ;AND EXIT.
```

```
2938                                    ;
2939                                    ; SUBROUTINE TO WAIT FOR DISPLAY STOP FLAG.
2940                                    ; RETURN WITH "C" = 1 AS SOON AS STOP OCCURS.
2941                                    ; OTHERWISE, "C" = 0 AND RETURN AFTER APPROX 300 MSEC.
2942                                    ;
2943 027454                            WAITF:  BREAK                   ; DO A SUPVSR BREAK FIRST.
2944 027456  012727  005670                    MOV     #3000.,(PC)+    ; 300 MSEC TIMER.
2945 027462  005670                    1$:     3000.
2946 027464  005777  153344            2$:     TST     @DSR
2947 027470  100407                            BMI     3$              ; EXIT ON STOP FLAG.
2948 027472  004737  027530                    JSR     PC,US100        ; OTHERWISE, WAIT 100 USEC...
2949 027476  005337  027462                    DEC     1$
2950 027502  001370                            BNE     2$              ;...AND TRY AGAIN.
2951 027504  000241                            CLC                     ; C = 0, DPU STILL RUNNING...
2952 027506  000401                            SKP1                    ;...OR HUNG-UP AFTER 300 MSEC.
2953 027510  000261                    3$:     SEC                     ; C = 1, DPU IS STOPPED.
2954 027512  000207                            RTS     PC
2955                                    ;
2956                                    ; SUBROUTINES TO PAUSE (WAIT) IN INCREMENTS OF 100 USEC.
2957                                    ; BASED ON AVERAGE TIME (CPU DEPENDANT) TO EXECUTE
2958                                    ;               DEC R0
2959                                    ;               BNE .-2
2960                                    ; DELAY CONSTANT (K100US) SHOULD BE ADJUSTED AS FOLLOWS:
2961                                    ;       FOR SLOW CPU'S (11/03 THRU 10), K100US = 8.
2962                                    ;       FOR FAST CPU'S (11/34 THRU 70), K100US = 32.
2963                                    ;
2964 027514  012746  023420            PAUSE1: MOV     #10000.,-(SP)   ; 1 SEC TIMER.
2965 027520  000402                            SKP2
2966 027522  012746  011610            PAUS.5: MOV     #5000.,-(SP)    ; 1/2 SEC TIMER.
2967 027526  000402                            SKP2
2968 027530  012746  000001            US100:  MOV     #1,-(SP)        ; 100 USEC TIMER.
2969 027534  011646                            MOV     (SP),-(SP)      ; SAVE THE COUNT
2970 027536  010066  000002                    MOV     R0,2(SP)        ; SAVE R0
2971 027542  013700  027564            1$:     MOV     K100US,R0
2972 027546  005300                            DEC     R0
2973 027550  001376                            BNE     .-2             ; 100 USEC (MORE OR LESS) LOOP.
2974 027552  005316                            DEC     (SP)
2975 027554  001372                            BNE     1$              ; VARIABLE LOOP.
2976 027556  005726                            TST     (SP)+           ; FIX STACK...
2977 027560  012600                            MOV     (SP)+,R0        ; RESTORE R0
2978 027562  000207                            RTS     PC              ;...AND RETURN.
2979 027564  000010                    K100US: 8.                      ; 100 USEC TIMER.
```

```
                                        ;
                                        ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
                                        ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
                                        ;               "C" = 0, ALL ADDRESSES OK.
                                        ;
                                        ;CALL:  MOV ADR1,R1
                                        ;       MOV ADR2,R2
                                        ;       JSR PC,NXM
                                        ;       RETURN          ;TEST "C" AND PROCEED.
                                        ;
027566  012737  027620  000004  NXM:    MOV     #2$,@#4         ; SET BUSERR VECTOR.
027574  012737  000200  000006          MOV     #PRI04,@#6
027602  005003                          CLR     R3              ;FLAG.
027604  005711                  1$:     TST     (R1)            ;TEST THE ADDRESS(ES).
                                                                ;IF ANY TRAP, CONTINUE AT 2$.
027606  020102                          CMP     R1,R2           ;OTHERWISE, CONTINUE HERE.
027610  001407                          BEQ     3$              ;BR IF FINISHED (NO NEXM'S).
027612  062701  000002                  ADD     #2,R1           ;SET NEXT ADDRESS...
027616  000772                          BR      1$              ;...AND CONTINUE.

027620  005103                  2$:     COM     R3              ;GOT ONE, SET FLAG...
027622  012716  027630                  MOV     #3$,(SP)
027626  000002                          RTI                     ;...AND DISMISS INTERRUPT...
027630                          3$:     CLRVEC  #4              ;...AND GIVE BACK THE VECTOR.
027636  005703                          TST     R3              ;DID WE CATCH ONE ??
027640  001401                          BEQ     .+4             ;NO, "C" = 0, SKIP NEXT.
027642  000261                          SEC                     ;YES, "C" = 1, (R1) = NEXM ADDR.
027644  000207                          RTS     PC

                                        ;
                                        ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
                                        ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
                                        ; LOOP COUNTER IS SET BY 'BEGIN.TEST" MACRO.
                                        ;
                                        ; CALL: LOOPTO  ARG
                                        ;
027646                          LOOP:
027646  005737  002510                  TST     NOITS           ; ITERATIONS INHIBITED?
027652  001006                          BNE     1$              ; YES.
027654  005737  003030                  TST     QVP             ; NO.
027660  100403                          BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
027662  005337  027700                  DEC     LOOPK           ; BUMP LOOP COUNTER.
027666  001002                          BNE     2$
027670  000241                  1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
027672  000401                          SKP1
027674  000261                  2$:     SEC                     ;LOOP ENABLED.
027676  000207                          RTS     PC
027700  000000                  LOOPK:  0                       ; LOOP (ITERATION) COUNTER.
```

```
                                        ;
                                        ; PRINT THE NUMBER AND NAME OF EACH TEST AS WE GO ALONG.
                                        ; INCREMENT "TESTK" TO INDICATE THE NUMBER OF TESTS
                                        ; IN THE CURRENT RUN SEQUENCE.
                                        ; CLEAR THE ERROR COUNTER AND SIGNATURE EXTENSION FLAGS.
                                        ;
                                        ; *** NOTE: REQUIRES PRIOR USE OF THE "BEGIN.TEST" MACRO
                                        ;           TO SET NAME AND NUMBER POINTERS.
                                        ;
027702  005037  023064          TSTGO:  CLR     SIFLAG          ; CLEAR "SOFT INIT" FLAG
027706  005037  030130                  CLR     ERRK            ; CLEAR LOCAL ERROR COUNTER.
027712  005037  022300                  CLR     EXTA            ; CLEAR ERROR EXTENSION FLAG.
027716  105037  026106                  CLRB    INTMASK         ; CLEAR INTERRUPT MASK (CHECK ERROR)
027722  013700  003026                  MOV     UNITN,R0        ; GET THE UNIT NUMBER,
027726  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
027730  005737  003226                  TST     NODEV           ; DID STARTUP FIND THE DEVICE?
027734  001430                          BEQ     4$              ; BR IF YES
027736  100010                          BPL     3$              ; BR IF NOT IDLE
027740  052760  160000  003302          BIS     #160000,ERTABL(R0) ; FLAG ERROR IN THE ERROR TABLE
027746                                  ERRDF   1,NXR,NXRERR    ; NO DEVICE HERE -- PRINT IT
027756  000407                          BR      2$
027760  052760  160001  003302  3$:     BIS     #160001,ERTABL(R0)      ; FLAG ERROR IN THE ERROR TABLE
027766                                  ERRDF   2,NOINIT        ; DEVICE NOT IDLE
027776  012737  177777  003224  2$:     MOV     #-1,DUFLG       ; DROP THE UNIT
030004                                  DODU    UNITN
030012                                  DOCLN                   ; ABORT THE PASS
030014  000422                          BR      5$
                                        ;
030016                          4$:     RFLAGS  R0              ; GET THE OPERATOR FLAGS.
030020  032700  001000                  BIT     #PNT,R0         ; PRINT THE TEST NUMBERS?
                                ;       TST     TIDFLG          ; WANNA TYPE THE TEST ID?
                                ;       BNE     1$              ; NAH!
030024  001412                          BEQ     1$              ; BR IF NO
030026                                  PRINTF  TNAM,TNUM       ; PRINT NUMBER AND TITLE.
030052  005237  030070          1$:     INC     TESTK           ; BUMP TEST COUNTER.
030056  004737  026114                  JSR     PC,ENAINT       ; ENABLE INTERRUPTS
030062  000207                  5$:     RTS     PC
030064  000000                  TNUM:   0
030066  000000                  TNAM:   0
030070  000000                  TESTK:  0                       ; NUMBER OF TESTS RUN THIS PASS.
                                        ;
                                        ; AT END OF EACH TEST, PRINT THE NUMBER OF ERRORS RECEIVED
                                        ; IF NORMAL ERROR REPORTING IS DISABLED (FLA:IER).
                                        ;
030072                          TSTEND: RFLAGS  R0
030074  030027  020000                  BIT     R0,#IER
030100  001412                          BEQ     1$              ; BR IF "IER" NOT SET.
030102                                  PRINTF  #ESUM,ERRK      ; PRINT ERROR COUNT.
030126  000207                  1$:     RTS     PC
                                        ;
030130  000000                  ERRK:   0                       ; LOCAL ERROR COUNT.
030132    045     101     040   ESUM:   .ASCIZ  /%A %D%A ERRORS/
030151    105     122     122   EMAXDU: .ASCIZ  /ERROR LIMIT REACHED -- DROPPING UNIT/
                                        .EVEN
                                        ;
                                        ; ROUTINES TO INCREMENT LOCAL ERROR COUNT AND CHECK FOR LIMIT:
                                        ;
```

```
3086 030216  005237  030130          INCERK: INC     ERRK            ; INCREMENT LOCAL ERROR COUNT
3087 030222  010046                          MOV     R0,-(SP)        ; SAVE R0
3088 030224  013700  003026                  MOV     UNITN,R0        ; GET UNIT NUMBER,
3089 030230  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
3090 030232  062700  003302                  ADD     #ERTABL,R0      ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
3091 030236  005210                          INC     (R0)            ; INCREMENT THE DEVICE ERROR COUNT
3092 030240  032710  007777                  BIT     #7777,(R0)      ; DID WE OVERFLOW THE FIELD?
3093 030244  001001                          BNE     1$              ; BR IF NO.
3094 030246  005310                          DEC     (R0)            ; YES -- BACK IT UP TO 7777.
3095 030250  012600                  1$:     MOV     (SP)+,R0        ; RESTORE R0
3096 030252  000207                          RTS     PC              ; RETURN TO CALLER.
3097
3098 030254  010046                  CKEMAX: MOV     R0,-(SP)        ; SAVE R0
3099 030256  013700  003026                  MOV     UNITN,R0        ; GET UNIT NUMBER
3100 030262  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET
3101 030264  016000  003302                  MOV     ERTABL(R0),R0   ; GET ERROR TABLE ENTRY
3102 030270  042700  170000                  BIC     #170000,R0      ; EXTRACT ERROR COUNT FIELD
3103 030274  020037  002516                  CMP     R0,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
3104 030300  103004                          BHIS    1$              ; BR IF YES
3105 030302  023737  030130  002514          CMP     ERRK,LERRMAX    ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
3106 030310  103417                          BLO     2$              ; BR IF NO
3107 030312                          1$:     RFLAGS  R0              ; GET OPERATOR FLAGS
3108 030314  032700  000040                  BIT     #IDU,R0         ; IS DROPPING INHIBITED?
3109 030320  001013                          BNE     2$              ; BR IF YES.
3110 030322  012737  177777  003224          MOV     #-1,DUFLG       ; NO -- DROP THE UNIT
3111 030330                                  ERRDF   4,EMAXDU
3112 030340                                  DODU    UNITN
3113 030346                                  DOCLN
3114 030350  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
3115 030352  000207                          RTS     PC              ; RETURN TO CALLER
3116
3117 030354  010046                  CKDROP: MOV     R0,-(SP)
3118 030356                                  RFLAGS  R0
3119 030360  032700  000040                  BIT     #IDU,R0
3120 030364  001011                          BNE     1$
3121 030366  012600                          MOV     (SP)+,R0
3122 030370  012737  177777  003224          MOV     #-1,DUFLG
3123 030376                                  DODU    UNITN
3124 030404                                  DOCLN                   ;ABORT THE PASS
3125 030406  000401                          BR      2$
3126 030410  012600                  1$:     MOV     (SP)+,R0
3127 030412  000207                  2$:     RTS     PC
```

```
                                        ;
                                        ; SUBROUTINE - CONFIGURE VSV11 SYSTEM.
                                        ;
030414                                  CONFIG:
030414  004737  025402                  CONMEM: JSR     PC,DPINIT
030420  012700  040000                          MOV     #40000,R0       ; PROTECT ALL MEMORIES
030424  010077  152410                  10$:    MOV     R0,@DYR
030430  004737  027454                          JSR     PC,WAITF
030434  062700  000400                          ADD     #CH1,R0
030440  032700  001400                          BIT     #CH3,R0
030444  001367                                  BNE     10$
                                        ;
030446  012777  000000  152362                  MOV     #0,@DXR         ; TRY A PIXEL READ.
030454  004737  027454                          JSR     PC,WAITF
030460  103410                                  BCS     1$              ; BR IF UNSUCCESSFUL.
030462                                          DFERR   PRBH            ; PIXEL READBACK HANGS UP.
030502  012737  144003  023056          1$:     MOV     #144003,GDDAT
030510  004737  022440                          JSR     PC,CSRCHK
030514  005005                                  CLR     R5              ; INIT MEM TABLE PTR.
030516  012777  000013  152310          2$:     MOV     #13,@DSR        ; CLEAR CSR.
030524  005065  003256                          CLR     CTABM(R5)       ; CLEAR TABLE ENTRY.
030530  010501                                  MOV     R5,R1           ; ENABLE MEM R/W.
030532  006201                                  ASR     R1
030534  000301                                  SWAB    R1
030536  052701  040060                          BIS     #40060,R1
030542  010177  152272                          MOV     R1,@DYR
030546  004737  027454                          JSR     PC,WAITF
030552  012777  000000  152256                  MOV     #0,@DXR         ; PIXEL READ.
030560  004737  027454                          JSR     PC,WAITF
030564  012777  000003  152242                  MOV     #3,@DSR
030572  017702  152236                          MOV     @DSR,R2         ; IS MEM THERE?
030576  100440                                  BMI     5$              ; NO.
030600  012777  000000  152226                  MOV     #0,@DSR         ; YES. GET PIXEL DATA.
030606  017702  152222                          MOV     @DSR,R2         ;
030612  052702  176003                          BIS     #176003,R2      ; INVERT PIXEL DATA.
030616  005102                                  COM     R2
030620  010265  003256                          MOV     R2,CTABM(R5)    ; PUT PIXEL DATA IN MEM TABLE.
030624  012703  000010                          MOV     #8.,R3          ; GOT AN EVEN NUMBER OF PIXEL BITS?
030630  005004                                  CLR     R4
030632  006202                                  ASR     R2
030634  006202                          3$:     ASR     R2
030636  032702  000001                          BIT     #1,R2
030642  001401                                  BEQ     4$
030644  005204                                  INC     R4
030646  005303                          4$:     DEC     R3
030650  001371                                  BNE     3$
030652  032704  000001                          BIT     #1,R4
030656  001410                                  BEQ     5$
030660                                          HRDERR  SCPRE,SYSCON    ; NO.
030700  042701  000060                  5$:     BIC     #60,R1          ; YES. RE-PROTECT THE MEM.
030704  010177  152130                          MOV     R1,@DYR
030710  004737  027454                          JSR     PC,WAITF
030714  062705  000002                          ADD     #2,R5           ; NEXT MEM.
030720  020527  000010                          CMP     R5,#10          ; ALL MEMS CHECKED?
030724  001274                                  BNE     2$              ; NO.
030726  012737  000010  003252  6$:     MOV     #BIT3,MEMFLG    ; YES. SET ONE FLAG TO INDICATE MEMS AVAIL.
030734  012701  003256                  MOV     #MEMTAB,R1      ;
```

```
3186 030740  005721                    7$:     TST     (R1)+                   ;
3187 030742  001403                            BEQ     8$                      ;
3188 030744  052737  000020  003252            BIS     #BIT4,MEMFLG            ;
3189 030752  006237  003252            8$:     ASR     MEMFLG                  ;
3190 030756  103370                            BCC     7$                      ;
3191 030760  004737  025402            CONSYC: JSR     PC,DPINIT
3192 030764  005003                            CLR     R3
3193 030766  005005                            CLR     R5                      ; INIT SYNC CHAN TABLE PTR.
3194 030770  012777  000013  152036    1$:     MOV     #13,@DSR                ; CLEAR CSR.
3195 030776  005065  003270                    CLR     CTABS(R5)               ; CLEAR TABLE ENTRY.
3196 031002  010501                            MOV     R5,R1                   ; READ CURSOR STATUS.
3197 031004  006201                            ASR     R1
3198 031006  000301                            SWAB    R1
3199 031010  052701  042100                    BIS     #42100,R1
3200 031014  010177  152016                    MOV     R1,@DXR
3201 031020  004737  027454                    JSR     PC,WAITF
3202 031024  012777  000003  152002            MOV     #3,@DSR                 ; GOT A DATA READY ERROR?
3203 031032  017702  151776                    MOV     @DSR,R2
3204 031036  042702  003777                    BIC     #3777,R2
3205 031042  020227  150000                    CMP     R2,#150000
3206 031046  001412                            BEQ     2$                      ; YES. (NO CHAN HERE).
3207 031050  052765  100000  003270            BIS     #BIT15,CTABS(R5)        ; NO. SET CHAN EXISTS IN TABLE.
3208 031056  017702  151756                    MOV     @DYR,R2                 ; GET INTERLACE BIT.
3209 031062  042702  157777                    BIC     #157777,R2
3210 031066  050265  003270                    BIS     R2,CTABS(R5)            ; SET IT IN TABLE.
3211 031072  050203                            BIS     R2,R3                   ; REMEMBER IT IN R3.
3212 031074  062705  000002            2$:     ADD     #2,R5                   ; NEXT SYNC CHAN.
3213 031100  020527  000010                    CMP     R5,#10                  ; ALL SYNC CHANS CHECKED?
3214 031104  001331                            BNE     1$                      ; NO.
3215 031106  012705  003270                    MOV     #CTABS,R5               ; YES. ALL INTERLACE BITS THE SAME?
3216 031112  010337  003230                    MOV     R3,INTLAC               ; REMEMBER INTERLACE/NON-INTERLACE STATUS
3217 031116  052703  100000                    BIS     #BIT15,R3               ; (R3 HAS LAST INTERLACE BIT FOUND).
3218 031122  005715                    3$:     TST     (R5)                    ; GOT AN ENTRY IN TABLE?
3219 031124  001402                            BEQ     4$                      ; NO.
3220 031126  021503                            CMP     (R5),R3                 ; YES. INTERLACE BIT SAME?
3221 031130  001006                            BNE     5$                      ; NO.
3222 031132  062705  000002            4$:     ADD     #2,R5                   ; YES. NEXT TABLE ENTRY.
3223 031136  020527  003300                    CMP     R5,#CTABS+10            ; DONE?
3224 031142  001367                            BNE     3$                      ; NO.
3225 031144  000410                            BR      6$                      ; YES.
3226
3227 031146                            5$:     HRDERR  SCIDE,SYSCON
3228
3229 031166  005737  002512            6$:     TST     MFGFLG                  ; MANUFACTURING FLAG SET?
3230 031172  001420                            BEQ     9$                      ; NO.
3231 031174  023737  003256  003022            CMP     MEMTAB,MFGM0            ; YES. NEW MEM 0 = MFG MEM 0 MASTER?
3232 031202  001004                            BNE     7$                      ; NO.
3233 031204  023737  003270  003024            CMP     SYCTAB,MFGS0            ; NEW SYNC CHAN 0 = MFG MASTER SYNC CHAN 0?
3234 031212  001410                            BEQ     9$                      ; YES.
3235 031214                            7$:     HRDERR  SCME,SYSCON             ; NO.
3236
3237 031234  012737  000010  003254    9$:     MOV     #BIT3,SYCFLG            ; SET ONE FLAG TO INDICATE SYNC CHANS AVAIL.
3238 031242  012701  003270                    MOV     #SYCTAB,R1              ;
3239 031246  005721                    10$:    TST     (R1)+                   ;
3240 031250  001403                            BEQ     11$                     ;
3241 031252  052737  000020  003254            BIS     #BIT4,SYCFLG            ;
3242 031260  006237  003254            11$:    ASR     SYCFLG                  ;
```

```
3243 031264  103370                    BCC     10$             ;
3244 031266  000207                    RTS     PC
3245
3246
3247
```

```
3249                                    ;
3250                                    ; SUBROUTINE - ENABLE MEM MGT.
3251                                    ;
3252 031270  005737  003244            KTON:   TST     KTFLG           ; GOT KT?
3253 031274  001403                            BEQ     1$              ; NO.
3254 031276  012737  000001  177572            MOV     #1,SR0          ; YES. ENABLE KT11.
3255 031304  000207                    1$:     RTS     PC
3256
3257
3258
3259                                    ;
3260                                    ; SUBROUTINE - DISABLE MEM MGT.
3261                                    ;
3262 031306                            KTOFF:  ;TST    KTFLG           ; GOT KT11?
3263                                            ;BEQ    1$              ; NO.
3264 031306  000240                            NOP
3265 031310  000240                            NOP
3266 031312  012737  000000  177572            MOV     #0,SR0          ; DISABLE KT.
3267 031320  000207                    1$:     RTS     PC
```

```
3269                                    ;
3270                                    ; SUBROUTINE - FILL ALL MEMORY OVER 28K WITH ''STOPN''.
3271                                    ; (NOTE - ENABLE KT BEFORE CALLING THIS).
3272                                    ;
3273 031322  005737  003244            KTBKGD: TST     KTFLG           ; GOT KT?
3274 031326  001430                            BEQ     9$              ; NO. GET OUT.
3275 031330  004737  031270                    JSR     PC,KTON         ; YES. ENABLE KT.
3276 031334  012700  173000                    MOV     #STOPN,R0       ; FILL ALL MEM (>28K) WITH BACKGROUND.
3277 031340  012737  001600  172354            MOV     #1600,@#KIPAR6
3278 031346  012701  140000            1$:     MOV     #140000,R1
3279 031352  012702  150000                    MOV     #150000,R2
3280 031356  010021                    2$:     MOV     R0,(R1)+
3281 031360  020102                            CMP     R1,R2
3282 031362  001375                            BNE     2$
3283 031364  023737  172354  003244            CMP     @#KIPAR6,KTFLG
3284 031372  001404                            BEQ     3$
3285 031374  062737  000200  172354            ADD     #200,@#KIPAR6
3286 031402  000761                            BR      1$
3287
3288 031404  004737  031306            3$:     JSR     PC,KTOFF        ; DISABLE KT.
3289 031410  000207                    9$:     RTS     PC
```

```
3291                                    ;
3292                                    ; SUBROUTINE TO SAVE R0-R5.
3293                                    ;
3294 031412  010037  031462    SAVERS: MOV     R0,SAVED
3295 031416  012700  031464            MOV     #SAVED+2,R0
3296 031422  010120                    MOV     R1,(R0)+
3297 031424  010220                    MOV     R2,(R0)+
3298 031426  010320                    MOV     R3,(R0)+
3299 031430  010420                    MOV     R4,(R0)+
3300 031432  010520                    MOV     R5,(R0)+
3301 031434  000207                    RTS     PC
3302
3303                                    ;
3304                                    ; SUBROUTINE TO RESTORE R0-R5.
3305                                    ;
3306 031436  012700  031464    RESTRS: MOV     #SAVED+2,R0
3307 031442  012001                    MOV     (R0)+,R1
3308 031444  012002                    MOV     (R0)+,R2
3309 031446  012003                    MOV     (R0)+,R3
3310 031450  012004                    MOV     (R0)+,R4
3311 031452  012005                    MOV     (R0)+,R5
3312 031454  013700  031462            MOV     SAVED,R0
3313 031460  000207                    RTS     PC
3314
3315
3316
3317 031462  000000    SAVED:  0
3318 031464  000000            0
3319 031466  000000            0
3320 031470  000000            0
3321 031472  000000            0
3322 031474  000000            0
```

```
3324                                    ;
3325                                    ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
3326                                    ;
3327 031476                            ENVIRN: MEMORY  R0
3328 031500  010037  003240                    MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
3329 031504  062737  000002  003240            ADD     #2,FREE
3330 031512  011037  003242                    MOV     (R0),FRESIZ     ;...AND WORD COUNT.
3331 031516  162737  000004  003242            SUB     #4,FRESIZ
3332 031524  013702  002012                    MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
3333 031530  162737  000007  003242  10$:      SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
3334 031536  005302                           DEC     R2
3335 031540  001373                           BNE     10$
3336
3337 031542  012737  000010  027564            MOV     #8.,K100US      ; ASSUME LSI FOR 100US TIMER.
3338 031550                                    READBUS
3339 031552                                    BCOMPLETE 1$
3340 031554  012737  000040  027564            MOV     #32.,K100US     ; NOT LSI -- CHANGE TIMER.
3341 031562                            1$:     ;MOV    #60.,HZ         ; ASSUME 60 HZ SYSTEM.
3342                                    ;       MOV     #MAXY60,YMAX    ;
3343                                    ;       CLOCK   L,R0
3344                                    ;       BNCOMPLETE 2$
3345                                    ;       CLR     @0(R0)          ; INSURE CLOCK IS OFF.
3346                                    ;       MOV     6(R0),HZ        ; SET ACTUAL FREQUENCY.
3347                                    ;       CMP     HZ,#60.         ;
3348                                    ;       BEQ     2$              ;
3349                                    ;       MOV     #MAXY50,YMAX    ;
3350 031562  004737  031656            2$:     JSR     PC,KTINIT       ; INIT KT11 MEM MGT.
3351 031566  000240  000240                    240,240
3352                                    ;*****************************************************************
3353                                    ; NOW THE FOLLOWING CODE PROVIDES A KEY RESTART FOR
3354                                    ; THE DIAGNOSTIC SUPERVISOR (DRS), AND FOR XXDP+ MONITOR.
3355                                    ; IF EITHER ONE CHANGES, THIS PROBABLY WILL NOT WORK !!
3356                                    ; BUT FOR NOW -- IT'S CONVENIENT FOR DEBUGGING PURPOSES.
3357                                    ;
3358                                    ; THE FOLLOWING OFFSETS APPLY TO DRS REV D.
3359          025570                    XX.ENTRY= 25570         ; DRS (REV D) ENTRY FROM XXDP+.
3360          026050                    SUPVSR=   26050         ; DRS (REV D) SUPERVISOR RESTART.
3361          026072                    XX.EXIT=  26072         ; DRS (REV D) EXIT TO XXDP+.
3362          003726                    XXDP=      3726         ; XXDP+ RESTART ADDRESS (XX3726).
3363                                    ;
3364 031572  104042                    KLUDGE: EMT     42              ; XXDP+ COMM BLOCK POINTER => R0
3365 031574  010037  003236                    MOV     R0,XXCOMM       ; SAVE IT.
3366 031600  042700  007777                    BIC     #7777,R0        ; MASK TO 2K PAGE...
3367 031604  162700  027000                    SUB     #27000,R0       ;...MINUS DRS SIZE (5.75K)...
3368 031610  062700  026050                    ADD     #SUPVSR,R0      ;...PLUS DRS RESTART OFFSET...
3369 031614  010037  000206                    MOV     R0,@#206        ;... = DRS RESTART ADDRESS.
3370 031620  012737  000137  000204            MOV     #137,@#204      ; LOC 204 = JMP DRS-RESTART.
3371 031626  013700  003236                    MOV     XXCOMM,R0       ; GET POINTER AGAIN.
3372 031632  042700  007777                    BIC     #7777,R0        ; MASK TO 2K PAGE...
3373 031636  062700  003726                    ADD     #XXDP,R0        ;...PLUS RESTART OFFSET...
3374 031642  010037  000212                    MOV     R0,@#212        ;... = XXDP+ RESTART ADDRESS.
3375 031646  012737  000137  000210            MOV     #137,@#210      ; LOC 210 = JMP XXDP+
3376 031654  000207                           RTS     PC
3377                                    ;*****************************************************************
3378
```

```
3380                                    ;
3381                                    ; SUBROUTINE - IF OVER 28K & KT11 AVAIL, INIT KT11.
3382                                    ;
3383 031656  005037  003244            KTINIT: CLR     KTFLG
3384 031662  023727  002120  001577            CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
3385 031670  101444                            BLOS    9$              ; NO.
3386 031672  013700  000004                    MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
3387 031676  012737  031770  000004            MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
3388 031704  005737  177572                    TST     @#SR0           ; GOT KT11?
3389 031710  000240                            NOP                     ;  (TRAP IF NO).
3390 031712  013737  002120  003244            MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
3391 031720  042737  000177  003244            BIC     #177,KTFLG      ;
3392 031726  010037  000004                    MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
3393 031732  005000                            CLR     R0              ; R0 = AR DATA.
3394 031734  012701  172340                    MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
3395 031740  012761  077406  177740    1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
3396 031746  010021                            MOV     R0,(R1)+        ; SET KIPAR REG.
3397 031750  062700  000200                    ADD     #200,R0         ; BUMP AR DATA BY "4K".
3398 031754  020027  002000                    CMP     R0,#2000        ; AT "I/O"?
3399 031760  001367                            BNE     1$              ; NO.
3400 031762  012741  177600                    MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
3401 031766  000405                            BR      9$
3402
3403 031770                            2$:     FORRTI
     031770  012716  031776                    MOV     #.+6,(SP)       ; FORCE RTI RETURN TO -
     031774  000002                            RTI                     ; - NEXT LOC.
3404 031776  010037  000004                    MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
3405
3406 032002  000207                    9$:     RTS     PC
```

```
                                .SBTTL  INITIALIZE SECTION

                                ;++
                                ; THE INITIALIZE SECTION CONTAINS THE CODING THAT IS PERFORMED
                                ; AT THE BEGINNING OF EACH PASS.
                                ;--
032004                                  BGNPROT
032004  177777  177777  177777         -1, -1, -1, -1  ; NO DEVICE PROTECTION REQUIRED.
032014                                  ENDPROT

032014                                  BGNINIT
                                ;
                                ; IF "START" OR "RESTART", SET QUICK-PASS FLAG AND BUS-INIT.
                                ; IF "CONTINUE", NOTHING IS REQUIRED.
                                ;
032014  005037  023064                  CLR     SIFLAG          ;CLEAR "SOFT INIT" FLAG
032020                                  READEF  #EF.CONTINUE
032026                                  BNCOMPLETE 1$
032030  023737  003026  002012          CMP     UNITN,L$UNIT    ;UNIT IN RANGE?
032036  103056                          BHIS    4$              ;BR IF NO.
032040  005737  003224                  TST     DUFLG           ;DROPPED UNIT?
032044  100460                          BMI     NXTU            ;BR IF YES
032046  013701  003026                  MOV     UNITN,R1
032052  006301                          ASL     R1
032054  005761  003302                  TST     ERTABL(R1)
032060  001504                          BEQ     SETU
032062  032761  040000  003302          BIT     #BIT14,ERTABL(R1)       ; DROPPED?
032070  001046                          BNE     NXTU
032072                                  EXIT    INIT            ;DO NOTHING IF "CONTINUE".
032076                          1$:     READEF  #EF.NEW
032104                                  BNCOMPLETE NXTU         ;TAKE NEXT UNIT IF NOT NEW PASS.
032106                                  READEF  #EF.START
032114                                  BCOMPLETE 2$
032116                                  READEF  #EF.RESTART
032124                                  BNCOMPLETE 31$
032126                          2$:                             ; 1ST PASS, BUS-INIT...
032126                                  BRESET                          ; BUS RESET.
032130  012737  177777  003030          MOV     #-1,QVP         ;...QUICK VERIFY...
032136  005037  030070                  CLR     TESTK           ;...CLEAR TOTAL TEST COUNT.
032142  004737  031476                  JSR     PC,ENVIRN       ; SET ENVIRONMENT.
032146  012700  003302                  MOV     #ERTABL,R0
032152  005020                  30$:    CLR     (R0)+           ;CLEAR THE ERROR TABLE
032154  020027  003502                  CMP     R0,#ERTABE
032160  103774                          BLO     30$
032162  000404                          BR      4$
032164  005037  003030          31$:    CLR     QVP
032170  000137  032240                  JMP     PASRPT          ;GO REPORT THE STATUS

032174                          4$:
032174  012737  177777  003026  NEWPAS: MOV     #-1,UNITN       ;INIT UNIT NUMBER...
032202  005037  032726                  CLR     DEVCNT          ;CLEAR COUNT OF DEVICES RUNNING
032206                          NXTU:   BREAK
032210  005237  003026                  INC     UNITN           ;...AND SET NEXT UNIT NUMBER.
032214  023737  003026  002012          CMP     UNITN,L$UNIT
032222  103423                          BLO     SETU
032224  012737  177777  003224          MOV     #-1,DUFLG
032232  000401                          BR      11$
```

```
3465 032234                                    DOCLN                   ;ABORT, NO MORE UNITS.
3466 032236  000240                    11$:    NOP
3467 032240                            PASRPT:
3468 032240  023727  002012  000001            CMP     L$UNIT,#1       ;HOW MANY UNITS SELECTED?
3469 032246  101752                            BLOS    NEWPAS          ;BR IF ONLY 1
3470 032250  005737  032726                    TST     DEVCNT          ;ARE ANY STILL RUNNING?
3471 032254  001747                            BEQ     NEWPAS          ;BR IF NO
3472 032256                                    RFLAGS  R0
3473 032260  032700  000100                    BIT     #ISR,R0         ;SHOULD WE PRINT STATISTICS
3474 032264  001343                            BNE     NEWPAS          ;BR IF NO
3475
3476 032266                                    DORPT
3477 032270  000741                            BR      NEWPAS
3478 032272                            10$:
3479
3480 032272                            SETU:   GPHARD  UNITN,R0        ;GET UNIT N P-TABLE POINTER.
3481 032300                                    BNCOMPLETE NXTU         ;BR IF UNIT NOT AVAILABLE.
3482 032302  005037  003224                    CLR     DUFLG           ;CLEAR 'DROPPED'' FLAG.
3483 032306  005237  032726                    INC     DEVCNT
3484 032312  012001                            MOV     (R0)+,R1        ; GET 1ST REGISTER ADDRESS.
3485 032314  012702  003032                    MOV     #DPC,R2
3486 032320  010122                    20$:    MOV     R1,(R2)+
3487 032322  062701  000002                    ADD     #2,R1
3488 032326  020227  003046                    CMP     R2,#LMR
3489 032332  101772                            BLOS    20$
3490
3491
3492 032334  012001                            MOV     (R0)+,R1        ; GET 1ST VECTOR ADDRESS.
3493 032336  011002                            MOV     (R0),R2         ; GET INTERRUPT PRIORITY
3494 032340  012037  003050                    MOV     (R0)+,DPRI      ; SET INTERRUPT PRIORITY.
3495 032344  010137  003052                    MOV     R1,STPV         ; SET STOP VECTOR POINTER...
3496 032350  012721  026242                    MOV     #DSTP,(R1)+     ;...VECTOR...
3497 032354  010221                            MOV     R2,(R1)+        ;...AND PRIORITY.
3498 032356  010137  003054                    MOV     R1,JSMV         ; DITTO JOY-STICK MATCH.
3499 032362  012721  026274                    MOV     #DJM,(R1)+
3500 032366  010221                            MOV     R2,(R1)+
3501 032370  010137  003056                    MOV     R1,TOTV         ; DITTO TIME-OUT.
3502 032374  012721  026210                    MOV     #DTO,(R1)+
3503 032400  010221                            MOV     R2,(R1)+
3504 032402  010137  003060                    MOV     R1,JSSV         ; DITTO JOY-STICK SWITCH.
3505 032406  012721  026326                    MOV     #DJS,(R1)+
3506 032412  010221                            MOV     R2,(R1)+
3507 032414  010137  003062                    MOV     R1,LUTV         ; DITTO LUT DONE.
3508 032420  012721  026602                    MOV     #LDUN,(R1)+
3509 032424  010221                            MOV     R2,(R1)+
3510 032426  012037  003216                    MOV     (R0)+,LUTAV     ; GET "LUT AVAILABLE" ENTRY
3511 032432  012737  001676  003250            MOV     #MAXY60,YMAX    ; ASSUME 60HZ FREQUENCY.
3512 032440  012737  000074  003246            MOV     #60.,HZ         ;
3513 032446  005720                            TST     (R0)+           ; WHAT IS SELECTED FREQUENCY?
3514 032450  001406                            BEQ     21$             ; BR IF 60HZ
3515 032452  012737  000062  003246            MOV     #50.,HZ         ; SAY 50 HZ
3516 032460  012737  001776  003250            MOV     #MAXY50,YMAX    ; AND SET UP FOR 512 VISIBLE LINES
3517 032466                            21$:
3518 032466  005037  003064                    CLR     OPTI
3519 032472  005737  002502                    TST     DPUMOD          ; DPU ONLY?
3520 032476  001003                            BNE     1$              ; SKIP NEXT IF DPU-ONLY.
3521 032500  012737  040000  003064            MOV     #I,OPTI         ;SET OPTIONAL I(NTENSIFY) BIT.
```

```
3522 032506                     1$:
3523                            ;       TST     QVP             ;1ST PASS ??
3524                            ;       BEQ     5$              ;NO, SKIP THE PASS 1 STUFF.
```

```
                                        ;
                                        ; 1ST PASS, CHECK THAT DEVICE ADDRESSES ARE VALID, AND
                                        ; THAT THE DISPLAY STATUS IS PROPERLY INITIALIZED.
                                        ;
032506  013701  003026                  MOV     UNITN,R1
032512  006301                          ASL     R1
032514  052761  100000  003302          BIS     #BIT15,ERTABL(R1)       ; SAY DEVICE RUNNING
032522  005037  022300                  CLR     EXTA                    ; CLEAR ERROR EXTENSION FLAG.
032526  023727  002012  000001          CMP     L$UNIT,#1               ; ARE WE TESTING MULTIPLE UNITS?
032534  101416                          BLOS    10$                     ; BR IF NO.
032536                                  RFLAGS  R0                      ; YES -- GET OPERATOR FLAGS.
032540  032700  001000                  BIT     #PNT,R0                 ; SHOULD WE PRINT UNIT #?
032544  001412                          BEQ     10$                     ; BR IF NOT.
032546                                  PRINTF  #PUNIT,UNITN            ; PRINT THE UNIT #
032572                          10$:
032572  005037  003226                  CLR     NODEV
032576  013701  003032                  MOV     DPC,R1
032602  013702  003040                  MOV     DYR,R2
032606  004737  027566                  JSR     PC,NXM                  ;TEST ALL 4 DPU REGISTERS...
032612  103003                          BCC     2$                      ;...AND BR IF ALL OK.
032614  010137  003226                  MOV     R1,NODEV                ;FLAG DEVICE AS NON-EXISTENT
032620  000416                          BR      3$                      ;DROP THIS UNIT.
032622  012777  000000  150204  2$:     MOV     #0,@DSR                 ;SELECT REAL DSR
032630  005777  150200                  TST     @DSR                    ;IS UNIT STOPPED?
032634  100413                          BMI     4$                      ;BR IF DISPLAY INITIALIZED OK.
032636  013701  003026                  MOV     UNITN,R1
032642  006301                          ASL     R1
032644  005261  003302                  INC     ERTABL(R1)
032650  012737  000001  003226          MOV     #1,NODEV                ;FLAG AS NOT IDLE
032656  012737  177777  003224  3$:     MOV     #-1,DUFLG
                                        ;
032664  005737  003216          4$:     TST     LUTAV                   ; NOW, SEE IF LUT INSTALLED.
032670  001412                          BEQ     5$                      ; BR IF NO
032672  013701  003042                  MOV     LSR,R1
032676  013702  003046                  MOV     LMR,R2
032702  004737  027566                  JSR     PC,NXM                  ; TEST ALL LUT REGISTERS...
032706  103003                          BCC     5$                      ;...BR IF IT'S THERE.
032710  012737  000000  003216          MOV     #0,LUTAV                ; CLEAR "LUT AVAILABLE" FLAG.
                                        ;
                                        ; FINALLY, SET CPU PRIORITY AND WE'RE DONE.
                                        ;
032716                          5$:     SETPRI  #PRI00                  ;ENABLE INTERRUPTS.
032724                                  ENDINIT
                                        ;
032726  000000                  DEVCNT: 0                               ;COUNT OF RUNNING DEVICES
032730     045     116     045  PUNIT:  .ASCIZ  /%N%N%A***** TESTING UNIT %D2%A *****/
                                        .EVEN
                                        ;
                                        ; SUBROUTINE - IF USER PATCHABLE FLAG "FORCER" IS SET (NON-ZERO),
                                        ; THEN FORCE RETURN TO THE NEXT ERROR CALL (HRDERR) IN SEQUENCE.
                                        ;
032776  005737  003020          FORCET: TST     FORCER                  ; FORCE ERROR TYPOUT?
033002  001402                          BEQ     1$                      ; NO.
033004  062716  000004                  ADD     #4,(SP)                 ; YES. FORCE RTN TO THE HRDERR CALL.
```

```
   3583 033010  000207                1$:     RTS     PC              ;
```

```
                                .SBTTL  ADD AND DROP UNITS SECTIONS

                                ;++
                                ; THE ADD-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO BE (A) ADDED TO THE TEST LIST FOR THE FIRST TIME,
                                ;    OR (B) RE-INSERTED IF IT HAD BEEN PREVIOUSLY DROPPED.
                                ;--
033012                                  BGNAU
033012  010001                          MOV     R0,R1           ; GET UNIT TO BE ADDED (R0)
033014  006301                          ASL     R1              ; MAKE IT A WORD INDEX
033016  052761  100000  003302          BIS     #100000,ERTABL(R1)      ; SET THE "ACTIVE" BIT
033024  042761  040000  003302          BIC     #40000,ERTABL(R1)       ; CLEAR THE "DROPPED" BIT
033032                                  PRINTF  #1$,R0
033054                                  EXIT    AU
033060      045     116     045 1$:     .ASCIZ  /%N%A  UNIT %D%A ADDED/
                                        .EVEN

033106                                  ENDAU           ; UNUSED.
                                ;++
                                ; THE DROP-UNIT SECTION CONTAINS THE CODING THAT CAUSES A DEVICE
                                ; TO BE REMOVED FROM THE TEST LIST.
                                ;
                                ; SUPVSR DOES THE "DROPPING".  THIS IS JUST TO TELL THE MAN.
                                ; "DROPPED" UNITS ARE RE-SELECTED ON OPERATOR "STA" OR "ADD"
                                ; COMMAND, OTHERWISE REMAIN INACTIVE.  THE "DISPLAY" COMMAND
                                ; WILL PRINT ALL DROPPED UNITS, AND THE P-TABLES OF THOSE
                                ; WHICH ARE STILL ACTIVE.
                                ; UPON ENTRY, R0 CONTAINS THE UNIT TO BE DROPPED.

033110                                  BGNDU
033110  012737  177777  003224          MOV     #-1,DUFLG
033116  010001                          MOV     R0,R1
                                ;       MOVB    #-1,STIK(R1)    ;REMOVE "JOY-STICK" FLAG.
033120  006301                          ASL     R1
033122  052761  140000  003302          BIS     #140000,ERTABL(R1)      ; SAY DROPPED
033130  000240  000240  000240          240,240,240             ; ?????????
033136                                  PRINTF  #1$,R0
033160                                  EXIT    DU
033164      045     116     045 1$:     .ASCIZ  /%N%A  UNIT %D%A DROPPED/
                                        .EVEN
033214                                  ENDDU
                                ;++
                                ; AUTO-DROP CODE SECTION.
                                ;--
033216                                  BGNAUTO
033216                                  ENDAUTO         ; UNUSED.
```

```
                                .SBTTL  CLEAN-UP AND REPORT CODING SECTIONS

                                ;++
                                ; THE CLEANUP CODING SECTION CONTAINS THE CODING THAT IS
                                ; EXECUTED AT THE END OF EACH PASS (OR SUB-PASS).
                                ; USE TO RETURN DEVICE UNDER TEST TO A NEUTRAL STATE.
                                ;--
033220                                  BGNCLN
033220  005737  003224                  TST     DUFLG           ;"DROPPED" FLAG IS SET ON...
033224  100414                          BMI     1$              ;...AND GROSS DPU FAULT...
                                                                ;...DON'T TRY TO XCT DPU CODE.

033226  004737  026756                  JSR     PC,RELEAS       ;CHECK FOR "HUNG" DPU.
033232  012777  040000  147576          MOV     #GTJSSW,@DXR    ;CLEAR PENDING SWITCH INTERRUPT
033240  004737  027454                  JSR     PC,WAITF
033244  012777  100000  147566          MOV     #100000,@DYR    ;DO SOFT INIT.
033252  004737  027454                  JSR     PC,WAITF
033256  005737  003216          1$:     TST     LUTAV
033262  001402                          BEQ     2$
033264  005077  147552                  CLR     @LSR            ; IF LUT, CLEAR STATUS.
033270                          2$:     ENDCLN
                                ;++
                                ; THE REPORT CODING SECTION CONTAINS THE
                                ; "PRINTS" CALLS THAT GENERATE STATISTICAL REPORTS.
                                ;--
033272                                  BGNRPT
033272                                  PRINTS  #DEVSUM
033312  010246                          MOV     R2,-(SP)
033314  010346                          MOV     R3,-(SP)
033316  010446                          MOV     R4,-(SP)
033320  012704  003302                  MOV     #ERTABL,R4      ; GET START OF ERROR TABLE.
033324  005003                          CLR     R3              ; CLEAR UNIT NUMBER
033326  011402                  1$:     MOV     (R4),R2         ; GET ERROR TABLE ENTRY & TEST IT.
033330  001467                          BEQ     4$              ; ZERO IF UNIT NOT RUN
033332  100066                          BPL     4$
033334  032702  040000                  BIT     #BIT14,R2       ; WAS UNIT DROPPED?
033340  001015                          BNE     2$              ; BR IF YES
033342  042702  170000                  BIC     #^C7777,R2      ; GET ERROR COUNT FIELD
033346                                  PRINTS  #DEVONL,R3,R2   ; PRINT
033372  000446                          BR      4$
033374  020227  160000          2$:     CMP     R2,#160000      ; WAS UNIT NON-EXISTENT?
033400  001012                          BNE     3$              ; BR IF NO
033402                                  PRINTS  #DEVNXR,R3
033424  000431                          BR      4$
033426  020227  160001          3$:     CMP     R2,#160001      ; WAS UNIT NOT READY AT STARTUP?
033432  001012                          BNE     30$             ; BR IF NO.
033434                                  PRINTS  #DEVNRD,R3
033456  000414                          BR      4$
033460  042702  170000          30$:    BIC     #^C7777,R2
033464                                  PRINTS  #DEVDRO,R3,R2
033510  062704  000002          4$:     ADD     #2,R4
033514  005203                          INC     R3
033516  020427  003502                  CMP     R4,#ERTABE
033522  103701                          BLO     1$
033524  012604                          MOV     (SP)+,R4
033526  012603                          MOV     (SP)+,R3
033530  012602                          MOV     (SP)+,R2
```

```
3689 033532                                ENDRPT          ; UNUSED.
3690
3691
3692 033534     045     116     045  DEVSUM: .ASCIZ  /%N%ADEVICE STATUS SUMMARY:%N/
3693 033571     045     101     040  DEVONL: .ASCIZ  /%A  UNIT %D3%A   ONLINE,  ERRORS = %D%N/
3694 033641     045     101     040  DEVNXR: .ASCIZ  /%A  UNIT %D3%A   DROPPED, NON-EXISTENT REGISTER%N/
3695 033723     045     101     040  DEVNRD: .ASCIZ  /%A  UNIT %D3%A   DROPPED, NOT READY AT STARTUP%N/
3696 034004     045     101     040  DEVDRO: .ASCIZ  /%A  UNIT %D3%A   DROPPED, ERRORS = %D%N/
3697                                        .EVEN
```

```
3699                                    .SBTTL
3700                                    .SBTTL  HARDWARE TEST SECTION.
3701                                    .SBTTL
3702                                    .SBTTL  TEST #          DESCRIPTION
3703                                    .SBTTL  ------          -----------
3704                                    .SBTTL
3705            000000                  TN=0
3706                                    .NLIST  MC,ME
```

```
3708
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 1    -   STATIC RESET
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 1.)

3709                            ;
3710                            ; EXECUTE A BUS RESET INSTRUCTION AND CHECK THE STATE OF THE DPU.
3711                            ; THIS TEST IS ALMOST IDENTICAL TO THE "SOFT-INIT" ROUTINE (DPINIT:)
3712                            ; USED TO INIT THE DPU AT THE START OF MANY OF THE TESTS IN THIS
3713                            ; PROGRAM.  THE DIFFERENCES ARE THAT INSTEAD OF A SOFTWARE DPU INIT,
3714                            ; A BUS RESET IS USED, AND THAT THE ERROR MESSAGE READS "BUS RESET",
3715                            ; INSTEAD OF "SOFT-INIT".
3716                            ;
3717 034122  004737  027702    STRSET: JSR     PC,TSTGO        ; TITLE.
3718
3719 034126                     1$:
3720 034126                             BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3721 034130                             BRESET                          ; BUS RESET.
3722 034132  004737  027454            JSR     PC,WAITF        ;WAIT FOR A COMPLETE
3723 034136  005002                    CLR     R2              ;EXPECT DPC TO BE 0.
3724 034140  012703  100000            MOV     #BIT15,R3       ;EXPECT DSR TO BE 100000.
3725 034144  005777  146662            TST     @DPC            ;TEST THE DPC, S/B =0.
3726 034150  001003                    BNE     2$              ;ERROR IF NOT 0.
3727 034152  027703  146656            CMP     @DSR,R3         ;CHECK DSR.
3728 034156  001410                    BEQ     3$
3729 034160                     2$:     DFERR   BRINIT,PCSERR   ;REPORT ERROR IN DPC OR DSR.
3730 034200  004737  023422     3$:     JSR     PC,IIRCHK       ;CHECK THE INTERNAL REGISTERS
3731 034204  005004                    CLR     R4
3732 034206  005005                    CLR     R5              ;DXR AND DYR SHOULD BE 0.
3733 034210  005777  146622            TST     @DXR
3734 034214  001003                    BNE     4$
3735 034216  005777  146616            TST     @DYR
3736 034222  001410                    BEQ     5$
3737 034224                     4$:     DFERR   BRINIT,XYERR
3738
3739 034244                     5$:     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3740 034246                             COMEND  1$
     034246  004737  027646            JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
     034252  103401                    BCS     64$
     034254  000402                    BR      65$

     034256  000137  034126    64$:    JMP     1$

     034262  004737  030072    65$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
3741 034266                             EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
3742
3743 034272                             END.TEST

                                ;*************************************************************************
                                ;*
                                ;*      END TEST 1
                                ;*
                                ;*************************************************************************
```

```
3745
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 2    -   RESET ONES
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 1.)

3746                            ;
3747                            ; SET ALL APPLICABLE REGISTERS TO ONES, CHECK THAT ONES GOT SET, THEN
3748                            ; EXECUTE A BUS RESET INSTRUCTION AND CHECK THE STATE OF THE DPU.
3749                            ; THIS IS THE SAME AS THE "STATIC RESET" TEST ABOVE, EXCEPT THAT
3750                            ; THE REGISTERS ARE SET TO ONES FIRST.
3751                            ; R1=EXPECTED, R2=RECEIVED, R0,R3=WORK REGS.
3752                            ;
3753 034340  004737  027702    RESET1: JSR     PC,TSTGO        ; TITLE.
3754 034344                    1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3755
3756 034346  012701  177763            MOV     #177763,R1
3757 034352                            WTDSRA  R1,CSR
      034352  012700  000013            MOV     #SETCSR,R0      ; WRITE R1 INTO CSR .
      034356  050100                    BIS     R1,R0
      034360  010077  146450            MOV     R0,@DSR
3758 034364                            RDDSRA  CSR,R2
      034364  012777  000003  146442    MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      034372  017702  146436            MOV     @DSR,R2
3759 034376                            IFERROR ICSRE,EXPREC,,CKLOOP
      034376  020102                    CMP     R1,R2           ; OK?
      034400  001411                    BEQ     64$             ; YES.
      034402                            HRDERR  ICSRE,EXPREC,CKLOOP     ; NO.
      034424                    64$:
3760
3761 034424                            WTDSRA  #177760,MRR
      034424  012777  177774  146402    MOV     #177760&^C17!SETMRR,@DSR
3762 034432                            WTDSRA  #170000,MPM
      034432  012777  170015  146374    MOV     #170000&^C17!SETMPM,@DSR
3763 034440                            RDDSRA  MRR,R2
      034440  012777  000004  146366    MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
      034446  017702  146362            MOV     @DSR,R2
3764 034452  010203                    MOV     R2,R3
3765 034454  042702  170000            BIC     #170000,R2      ; MRR OK?
3766 034460  012701  007777            MOV     #7777,R1
3767 034464                            IFERROR IMRRE,EXPREC,,CKLOOP
      034464  020102                    CMP     R1,R2           ; OK?
      034466  001411                    BEQ     65$             ; YES.
      034470                            HRDERR  IMRRE,EXPREC,CKLOOP     ; NO.
      034512                    65$:
3768 034512  010302                    MOV     R3,R2           ; MPM OK?
3769 034514  042702  007777            BIC     #7777,R2
3770 034520  012701  170000            MOV     #170000,R1
3771 034524                            IFERROR IMPME,EXPREC,,CKLOOP
      034524  020102                    CMP     R1,R2           ; OK?
      034526  001411                    BEQ     66$             ; YES.
      034530                            HRDERR  IMPME,EXPREC,CKLOOP     ; NO.
      034552                    66$:
3772
```

```
3773 034552                                    WTDSRA  #177760,XRR
     034552  012777  177776  146254          MOV     #177760&^C17!SETXRR,@DSR
3774 034560                                    WTDSRA  #170000,XPM
     034560  012777  170017  146246          MOV     #170000&^C17!SETXPM,@DSR
3775 034566                                    RDDSRA  XRR,R2
     034566  012777  000006  146240          MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
     034574  017702  146234                  MOV     @DSR,R2
3776 034600  010203                           MOV     R2,R3
3777 034602  042702  170000                  BIC     #170000,R2      ; XRR OK?
3778 034606  012701  007777                  MOV     #7777,R1
3779 034612                                    IFERROR IXRRE,EXPREC,,CKLOOP
     034612  020102                           CMP     R1,R2           ; OK?
     034614  001411                           BEQ     67$             ; YES.
     034616                                    HRDERR  IXRRE,EXPREC,CKLOOP     ; NO.
     034640                                67$:
3780 034640  010302                           MOV     R3,R2           ; XPM OK?
3781 034642  042702  007777                  BIC     #7777,R2
3782 034646  012701  170000                  MOV     #170000,R1
3783 034652                                    IFERROR IXPME,EXPREC,,CKLOOP
     034652  020102                           CMP     R1,R2           ; OK?
     034654  001411                           BEQ     68$             ; YES.
     034656                                    HRDERR  IXPME,EXPREC,CKLOOP     ; NO.
     034700                                68$:
3784
3785 034700                                10$:    BRESET                          ; BUS RESET.
3786 034702  004737  027454                       JSR     PC,WAITF        ;WAIT FOR A COMPLETE
3787 034706  005002                               CLR     R2              ;EXPECT DPC TO BE 0.
3788 034710  012703  100000                       MOV     #BIT15,R3       ;EXPECT DSR TO BE 100000.
3789 034714  005777  146112                       TST     @DPC            ;TEST THE DPC, S/B =0.
3790 034720  001003                               BNE     11$             ;ERROR IF NOT 0.
3791 034722  027703  146106                       CMP     @DSR,R3         ;CHECK DSR.
3792 034726  001410                               BEQ     12$
3793 034730                                11$:    DFERR   BRINIT,PCSERR   ;REPORT ERROR IN DPC OR DSR.
3794 034750  004737  023422                12$:    JSR     PC,IIRCHK       ;CHECK THE INTERNAL REGISTERS
3795 034754  005004                               CLR     R4
3796 034756  005005                               CLR     R5              ;DXR AND DYR SHOULD BE 0.
3797 034760  005777  146052                       TST     @DXR
3798 034764  001003                               BNE     13$
3799 034766  005777  146046                       TST     @DYR
3800 034772  001410                               BEQ     14$
3801 034774                                13$:    DFERR   BRINIT,XYERR
3802
3803 035014                                14$:    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3804
3805 035016                                        COMEND  1$
     035016  004737  027646                       JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
     035022  103401                               BCS     69$
     035024  000402                               BR      70$

     035026  000137  034344                69$:    JMP     1$

     035032  004737  030072                70$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
3806 035036                                        EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
3807
3808 035042                                        END.TEST

                                           ;*******************************************************************************
```

```
;*
;*      END TEST 2
;*
;*****************************************************************************
```

```
3810
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 3    -   SOFT INIT ONES
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 10.)

3811                                    ;
3812                                    ; SET ALL APPLICABLE REGISTERS TO ONES, CHECK THAT ONES GOT SET, THEN
3813                                    ; EXECUTE SOFT INIT AND CHECK THE STATE OF THE DPU.
3814                                    ; THIS IS THE SAME AS THE "RESET ONES" TEST ABOVE, EXCEPT THAT
3815                                    ; A SOFT INIT IS EXECUTED INSTEAD OF A RESET INSTRUCTION.
3816                                    ; R1=EXPECTED, R2=RECEIVED, R0,R3=WORK REGS.
3817                                    ;
3818 035114  004737  027702            SINIT1: JSR     PC,TSTGO        ; TITLE.
3819 035120                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3820
3821 035122  012701  177763                    MOV     #177763,R1
3822 035126                                    WTDSRA  R1,CSR
     035126  012700  000013                    MOV     #SETCSR,R0      ; WRITE R1 INTO CSR .
     035132  050100                            BIS     R1,R0
     035134  010077  145674                    MOV     R0,@DSR
3823 035140                                    RDDSRA  CSR,R2
     035140  012777  000003  145666            MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     035146  017702  145662                    MOV     @DSR,R2
3824 035152                                    IFERROR ICSRE,EXPREC,,CKLOOP
     035152  020102                            CMP     R1,R2           ; OK?
     035154  001411                            BEQ     64$             ; YES.
     035156                                    HRDERR  ICSRE,EXPREC,CKLOOP     ; NO.
     035200                             64$:
3825
3826 035200                                    WTDSRA  #177760,MRR
     035200  012777  177774  145626            MOV     #177760&^C17!SETMRR,@DSR
3827 035206                                    WTDSRA  #170000,MPM
     035206  012777  170015  145620            MOV     #170000&^C17!SETMPM,@DSR
3828 035214                                    RDDSRA  MRR,R2
     035214  012777  000004  145612            MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
     035222  017702  145606                    MOV     @DSR,R2
3829 035226  010203                            MOV     R2,R3
3830 035230  042702  170000                    BIC     #170000,R2      ; MRR OK?
3831 035234  012701  007777                    MOV     #7777,R1
3832 035240                                    IFERROR IMRRE,EXPREC,,CKLOOP
     035240  020102                            CMP     R1,R2           ; OK?
     035242  001411                            BEQ     65$             ; YES.
     035244                                    HRDERR  IMRRE,EXPREC,CKLOOP     ; NO.
     035266                             65$:
3833 035266  010302                            MOV     R3,R2           ; MPM OK?
3834 035270  042702  007777                    BIC     #7777,R2
3835 035274  012701  170000                    MOV     #170000,R1
3836 035300                                    IFERROR IMPME,EXPREC,,CKLOOP
     035300  020102                            CMP     R1,R2           ; OK?
     035302  001411                            BEQ     66$             ; YES.
     035304                                    HRDERR  IMPME,EXPREC,CKLOOP     ; NO.
     035326                             66$:
3837
```

```
                                                WTDSRA  #177760,XRR
                                                MOV     #177760&^C17!SETXRR,@DSR
                                                WTDSRA  #170000,XPM
                                                MOV     #170000&^C17!SETXPM,@DSR
                                                RDDSRA  XRR,R2
                                                MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
                                                MOV     @DSR,R2
                                                MOV     R2,R3
                                                BIC     #170000,R2      ; XRR OK?
                                                MOV     #7777,R1
                                                IFERROR IXRRE,EXPREC,,CKLOOP
                                                CMP     R1,R2           ; OK?
                                                BEQ     67$             ; YES.
                                                HRDERR  IXRRE,EXPREC,CKLOOP     ; NO.
                                        67$:
                                                MOV     R3,R2           ; MPM OK?
                                                BIC     #7777,R2
                                                MOV     #170000,R1
                                                IFERROR IXPME,EXPREC
                                                CMP     R1,R2           ; OK?
                                                BEQ     68$             ; YES.
                                                HRDERR  IXPME,EXPREC,   ; NO.
                                        68$:

                                                MOV     #100000,@DYR    ;DO SOFT INIT.
                                                JSR     PC,WAITF        ;WAIT FOR STOP
                                                CLR     R2              ;EXPECT DPC TO BE 0.
                                                MOV     #BIT15,R3       ;EXPECT DSR TO BE 100000.
                                                TST     @DPC            ;TEST THE DPC, S/B =0.
                                                BNE     11$             ;ERROR IF NOT 0.
                                                CMP     @DSR,R3         ;CHECK DSR.
                                                BEQ     12$
                                        11$:    DFERR   NSINIT,PCSERR   ;REPORT ERROR IN DPC OR DSR.
                                        12$:    JSR     PC,IIRCHK       ;CHECK THE INTERNAL REGISTERS
                                                CLR     R4
                                                CLR     R5              ;DXR AND DYR SHOULD BE 0.
                                                TST     @DXR
                                                BNE     13$
                                                TST     @DYR
                                                BEQ     14$
                                        13$:    DFERR   NSINIT,XYERR

                                        14$:    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.

                                                COMEND  1$
                                                JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
                                                BCS     69$
                                                BR      70$

                                        69$:    JMP     1$

                                        70$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
                                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>

                                                END.TEST
```

```
;*****************************************************************************
;*
;*      END TEST 3
;*
;*****************************************************************************
```

```
3876
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 4    -   REGISTERS UNIQUE ADDRESS
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 10.)

3877                            ;
3878                            ; ALL REGISTERS ARE SET TO THEIR NORMAL STATIC STATE VIA A SOFT INIT.
3879                            ; EACH APPLICABLE REGISTER, IN TURN, IS SET TO ITS CAPABILITY FOR ONES,
3880                            ; AND THEN ALL OF THE OTHER REGISTERS ARE CHECKED THAT THEY REMAINED
3881                            ; AT THEIR STATIC STATE.
3882                            ;
3883 035704  004737  027702    DUADR:  JSR     PC,TSTGO        ; PRINT TITLE.
3884 035710                     1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3885 035712                             INITDP
      035712  004737  025402            JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
3886 035716  012703  177760            MOV     #177760,R3
3887 035722  010377  145106            MOV     R3,@DSR         ; WRITE DSR.
3888 035726  012701  000000            MOV     #0,R1
3889 035732  017702  145074            MOV     @DPC,R2         ; READ DPC INTO R2.
3890 035736                            IFERROR DSDPE,DUADRE
      035736  020102                    CMP     R1,R2           ; OK?
      035740  001410                    BEQ     64$             ; YES.
      035742                            HRDERR  DSDPE,DUADRE,   ; NO.
      035762                    64$:
3891 035762  017702  145050            MOV     @DXR,R2         ; READ DXR INTO R2.
3892 035766                            IFERROR DSDXE,DUADRE
      035766  020102                    CMP     R1,R2           ; OK?
      035770  001410                    BEQ     65$             ; YES.
      035772                            HRDERR  DSDXE,DUADRE,   ; NO.
      036012                    65$:
3893 036012  017702  145022            MOV     @DYR,R2         ; READ DYR INTO R2.
3894 036016                            IFERROR DSDYE,DUADRE
      036016  020102                    CMP     R1,R2           ; OK?
      036020  001410                    BEQ     66$             ; YES.
      036022                            HRDERR  DSDYE,DUADRE,   ; NO.
      036042                    66$:
3895 036042                            ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3896 036044                            BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3897 036046                            INITDP
      036046  004737  025402            JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
3898 036052  012703  177760            MOV     #177760,R3
3899 036056                            WTDSRA  R3,DSR
      036056  012700  000010            MOV     #SETDSR,R0      ; WRITE R3 INTO DSR .
      036062  050300                    BIS     R3,R0
      036064  010077  144744            MOV     R0,@DSR
3900 036070  012701  000001            MOV     #1,R1
3901 036074                            RDDSRA  PCS,R2
      036074  012777  000001  144732    MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
      036102  017702  144726            MOV     @DSR,R2
3902 036106                            IFERROR DSPCE,DUADRE
      036106  020102                    CMP     R1,R2           ; OK?
      036110  001410                    BEQ     67$             ; YES.
      036112                            HRDERR  DSPCE,DUADRE,   ; NO.
```

```
                036132                          67$:
3903            036132  012701  100000                  MOV     #100000,R1
3904            036136                          RDDSRA  FLG,R2
                036136  012777  000002  144670          MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
                036144  017702  144664                  MOV     @DSR,R2
3905            036150                          IFERROR DSFLE,DUADRE
                036150  020102                  CMP     R1,R2           ; OK?
                036152  001410                  BEQ     68$             ; YES.
                036154                          HRDERR  DSFLE,DUADRE,   ; NO.
                036174                          68$:
3906            036174  012701  000003                  MOV     #3,R1
3907            036200                          RDDSRA  CSR,R2
                036200  012777  000003  144626          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
                036206  017702  144622                  MOV     @DSR,R2
3908            036212                          IFERROR DSCSE,DUADRE
                036212  020102                  CMP     R1,R2           ; OK?
                036214  001410                  BEQ     69$             ; YES.
                036216                          HRDERR  DSCSE,DUADRE,   ; NO.
                036236                          69$:
3909            036236  012701  000000                  MOV     #0,R1
3910            036242                          RDDSRA  MRR,R2
                036242  012777  000004  144564          MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
                036250  017702  144560                  MOV     @DSR,R2
3911            036254  042702  170000                  BIC     #170000,R2      ; MASK OUT MPM.
3912            036260                          IFERROR DSMRE,DUADRE
                036260  020102                  CMP     R1,R2           ; OK?
                036262  001410                  BEQ     70$             ; YES.
                036264                          HRDERR  DSMRE,DUADRE,   ; NO.
                036304                          70$:
3913            036304                          RDDSRA  MPM,R2
                036304  012777  000004  144522          MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
                036312  017702  144516                  MOV     @DSR,R2
3914            036316  042702  007777                  BIC     #7777,R2        ; MASK OUT MRR.
3915            036322                          IFERROR DSMPE,DUADRE
                036322  020102                  CMP     R1,R2           ; OK?
                036324  001410                  BEQ     71$             ; YES.
                036326                          HRDERR  DSMPE,DUADRE,   ; NO.
                036346                          71$:
3916            036346                          RDDSRA  XRR,R2
                036346  012777  000006  144460          MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
                036354  017702  144454                  MOV     @DSR,R2
3917            036360  042702  170000                  BIC     #170000,R2      ; MASK OUT XPM.
3918            036364                          IFERROR DSXRE,DUADRE
                036364  020102                  CMP     R1,R2           ; OK?
                036366  001410                  BEQ     72$             ; YES.
                036370                          HRDERR  DSXRE,DUADRE,   ; NO.
                036410                          72$:
3919            036410                          RDDSRA  XPM,R2
                036410  012777  000006  144416          MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
                036416  017702  144412                  MOV     @DSR,R2
3920            036422  042702  007777                  BIC     #7777,R2        ; MASK OUT XRR.
3921            036426                          IFERROR DSXPE,DUADRE
                036426  020102                  CMP     R1,R2           ; OK?
                036430  001410                  BEQ     73$             ; YES.
                036432                          HRDERR  DSXPE,DUADRE,   ; NO.
                036452                          73$:
3922            036452                          ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
```

```
                                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
                                                INITDP
                036456  004737  025402         JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
3925            036462  012703  177760         MOV     #177760,R3
3926                                           WTDSRA  R3,CSR
                036466  012700  000013         MOV     #SETCSR,R0      ; WRITE R3 INTO CSR .
                036472  050300                 BIS     R3,R0
                036474  010077  144334         MOV     R0,@DSR
3927            036500  012701  100000         MOV     #100000,R1
3928                                           RDDSRA  DSR,R2
                036504  012777  000000  144322 MOV     #SELDSR,@DSR    ; READ DSR INTO R2 .
                036512  017702  144316         MOV     @DSR,R2
3929            036516                         IFERROR CSDSE,DUADRE
                036516  020102                 CMP     R1,R2           ; OK?
                036520  001410                 BEQ     74$             ; YES.
                036522                         HRDERR  CSDSE,DUADRE,   ; NO.
                036542                  74$:
3930            036542  012701  000001         MOV     #1,R1
3931            036546                         RDDSRA  PCS,R2
                036546  012777  000001  144260 MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
                036554  017702  144254         MOV     @DSR,R2
3932            036560                         IFERROR CSPCE,DUADRE
                036560  020102                 CMP     R1,R2           ; OK?
                036562  001410                 BEQ     75$             ; YES.
                036564                         HRDERR  CSPCE,DUADRE,   ; NO.
                036604                  75$:
3933            036604  012701  100000         MOV     #100000,R1
3934            036610                         RDDSRA  FLG,R2
                036610  012777  000002  144216 MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
                036616  017702  144212         MOV     @DSR,R2
3935            036622                         IFERROR CSFLE,DUADRE
                036622  020102                 CMP     R1,R2           ; OK?
                036624  001410                 BEQ     76$             ; YES.
                036626                         HRDERR  CSFLE,DUADRE,   ; NO.
                036646                  76$:
3936            036646  012701  000000         MOV     #0,R1
3937            036652                         RDDSRA  MRR,R2
                036652  012777  000004  144154 MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
                036660  017702  144150         MOV     @DSR,R2
3938            036664  042702  170000         BIC     #170000,R2      ; MASK OUT MPM.
3939            036670                         IFERROR CSMRE,DUADRE
                036670  020102                 CMP     R1,R2           ; OK?
                036672  001410                 BEQ     77$             ; YES.
                036674                         HRDERR  CSMRE,DUADRE,   ; NO.
                036714                  77$:
3940            036714                         RDDSRA  MPM,R2
                036714  012777  000004  144112 MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
                036722  017702  144106         MOV     @DSR,R2
3941            036726  042702  007777         BIC     #7777,R2        ; MASK OUT MRR.
3942            036732                         IFERROR CSMPE,DUADRE
                036732  020102                 CMP     R1,R2           ; OK?
                036734  001410                 BEQ     78$             ; YES.
                036736                         HRDERR  CSMPE,DUADRE,   ; NO.
                036756                  78$:
3943            036756                         RDDSRA  XRR,R2
                036756  012777  000006  144050 MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
                036764  017702  144044         MOV     @DSR,R2
```

```
                                                BIC     #170000,R2      ; MASK OUT XPM.
                                                IFERROR CSXRE,DUADRE
                                                CMP     R1,R2           ; OK?
                                                BEQ     79$             ; YES.
                                                HRDERR  CSXRE,DUADRE,   ; NO.
                                        79$:
                                                RDDSRA  XPM,R2
                                                MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
                                                MOV     @DSR,R2
                                                BIC     #7777,R2        ; MASK OUT XRR.
                                                IFERROR CSXPE,DUADRE
                                                CMP     R1,R2           ; OK?
                                                BEQ     80$             ; YES.
                                                HRDERR  CSXPE,DUADRE,   ; NO.
                                        80$:
                                                ENDSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
                                                BGNSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
                                                INITDP
                                                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
                                                MOV     #7777,R3
                                                WTDSRA  R3,MRR
                                                MOV     #SETMRR,R0      ; WRITE R3 INTO MRR .
                                                BIS     R3,R0
                                                MOV     R0,@DSR
                                                MOV     #100000,R1
                                                RDDSRA  DSR,R2
                                                MOV     #SELDSR,@DSR    ; READ DSR INTO R2 .
                                                MOV     @DSR,R2
                                                IFERROR MRDSE,DUADRE
                                                CMP     R1,R2           ; OK?
                                                BEQ     81$             ; YES.
                                                HRDERR  MRDSE,DUADRE,   ; NO.
                                        81$:
                                                MOV     #1,R1
                                                RDDSRA  PCS,R2
                                                MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
                                                MOV     @DSR,R2
                                                IFERROR MRPCE,DUADRE
                                                CMP     R1,R2           ; OK?
                                                BEQ     82$             ; YES.
                                                HRDERR  MRPCE,DUADRE,   ; NO.
                                        82$:
                                                MOV     #100000,R1
                                                RDDSRA  FLG,R2
                                                MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
                                                MOV     @DSR,R2
                                                IFERROR MRFLE,DUADRE
                                                CMP     R1,R2           ; OK?
                                                BEQ     83$             ; YES.
                                                HRDERR  MRFLE,DUADRE,   ; NO.
                                        83$:
                                                MOV     #3,R1
                                                RDDSRA  CSR,R2
                                                MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
                                                MOV     @DSR,R2
                                                IFERROR MRCSE,DUADRE
                                                CMP     R1,R2           ; OK?
```

```
        037276  001410                          BEQ     84$             ; YES.
        037300                                  HRDERR  MRCSE,DUADRE,   ; NO.
        037320                          84$:
3966    037320  012701  000000                  MOV     #0,R1
3967    037324                                  RDDSRA  MPM,R2
        037324  012777  000004  143502          MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
        037332  017702  143476                  MOV     @DSR,R2
3968    037336  042702  007777                  BIC     #7777,R2        ; MASK OUT MRR.
3969    037342                                  IFERROR MRMPE,DUADRE
        037342  020102                          CMP     R1,R2           ; OK?
        037344  001410                          BEQ     85$             ; YES.
        037346                                  HRDERR  MRMPE,DUADRE,   ; NO.
        037366                          85$:
3970    037366                                  RDDSRA  XRR,R2
        037366  012777  000006  143440          MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
        037374  017702  143434                  MOV     @DSR,R2
3971    037400  042702  170000                  BIC     #170000,R2      ; MASK OUT XPM.
3972    037404                                  IFERROR MRXRE,DUADRE
        037404  020102                          CMP     R1,R2           ; OK?
        037406  001410                          BEQ     86$             ; YES.
        037410                                  HRDERR  MRXRE,DUADRE,   ; NO.
        037430                          86$:
3973    037430                                  RDDSRA  XPM,R2
        037430  012777  000006  143376          MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
        037436  017702  143372                  MOV     @DSR,R2
3974    037442  042702  007777                  BIC     #7777,R2        ; MASK OUT XRR.
3975    037446                                  IFERROR MRXPE,DUADRE
        037446  020102                          CMP     R1,R2           ; OK?
        037450  001410                          BEQ     87$             ; YES.
        037452                                  HRDERR  MRXPE,DUADRE,   ; NO.
        037472                          87$:
3976    037472                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3977    037474                                  BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3978    037476                                  INITDP
        037476  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
3979    037502  012703  170000                  MOV     #170000,R3
3980    037506                                  WTDSRA  R3,MPM
        037506  012700  000015                  MOV     #SETMPM,R0      ; WRITE R3 INTO MPM .
        037512  050300                          BIS     R3,R0
        037514  010077  143314                  MOV     R0,@DSR
3981    037520  012701  100000                  MOV     #100000,R1
3982    037524                                  RDDSRA  DSR,R2
        037524  012777  000000  143302          MOV     #SELDSR,@DSR    ; READ DSR INTO R2 .
        037532  017702  143276                  MOV     @DSR,R2
3983    037536                                  IFERROR MPDSE,DUADRE
        037536  020102                          CMP     R1,R2           ; OK?
        037540  001410                          BEQ     88$             ; YES.
        037542                                  HRDERR  MPDSE,DUADRE,   ; NO.
        037562                          88$:
3984    037562  012701  000001                  MOV     #1,R1
3985    037566                                  RDDSRA  PCS,R2
        037566  012777  000001  143240          MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
        037574  017702  143234                  MOV     @DSR,R2
3986    037600                                  IFERROR MPPCE,DUADRE
        037600  020102                          CMP     R1,R2           ; OK?
        037602  001410                          BEQ     89$             ; YES.
        037604                                  HRDERR  MPPCE,DUADRE,   ; NO.
```

```
                037624
                                                        89$:
3987 037624  012701  100000                                 MOV     #100000,R1
3988 037630                                                 RDDSRA  FLG,R2
     037630  012777  000002  143176                        MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
     037636  017702  143172                                MOV     @DSR,R2
3989 037642                                                 IFERROR MPFLE,DUADRE
     037642  020102                                        CMP     R1,R2           ; OK?
     037644  001410                                        BEQ     90$             ; YES.
     037646                                                 HRDERR  MPFLE,DUADRE,   ; NO.
     037666                                             90$:
3990 037666  012701  000003                                 MOV     #3,R1
3991 037672                                                 RDDSRA  CSR,R2
     037672  012777  000003  143134                        MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     037700  017702  143130                                MOV     @DSR,R2
3992 037704                                                 IFERROR MPCSE,DUADRE
     037704  020102                                        CMP     R1,R2           ; OK?
     037706  001410                                        BEQ     91$             ; YES.
     037710                                                 HRDERR  MPCSE,DUADRE,   ; NO.
     037730                                             91$:
3993 037730  012701  000000                                 MOV     #0,R1
3994 037734                                                 RDDSRA  MRR,R2
     037734  012777  000004  143072                        MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
     037742  017702  143066                                MOV     @DSR,R2
3995 037746  042702  170000                                BIC     #170000,R2      ; MASK OUT MPM.
3996 037752                                                 IFERROR MPMRE,DUADRE
     037752  020102                                        CMP     R1,R2           ; OK?
     037754  001410                                        BEQ     92$             ; YES.
     037756                                                 HRDERR  MPMRE,DUADRE,   ; NO.
     037776                                             92$:
3997 037776                                                 RDDSRA  XRR,R2
     037776  012777  000006  143030                        MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
     040004  017702  143024                                MOV     @DSR,R2
3998 040010  042702  170000                                BIC     #170000,R2      ; MASK OUT XPM.
3999 040014                                                 IFERROR MPXRE,DUADRE
     040014  020102                                        CMP     R1,R2           ; OK?
     040016  001410                                        BEQ     93$             ; YES.
     040020                                                 HRDERR  MPXRE,DUADRE,   ; NO.
     040040                                             93$:
4000 040040                                                 RDDSRA  XPM,R2
     040040  012777  000006  142766                        MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
     040046  017702  142762                                MOV     @DSR,R2
4001 040052  042702  007777                                BIC     #7777,R2        ; MASK OUT XRR.
4002 040056                                                 IFERROR MPXPE,DUADRE
     040056  020102                                        CMP     R1,R2           ; OK?
     040060  001410                                        BEQ     94$             ; YES.
     040062                                                 HRDERR  MPXPE,DUADRE,   ; NO.
     040102                                             94$:
4003 040102                                                 ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4004 040104                                                 BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4005 040106                                                 INITDP
     040106  004737  025402                                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4006 040112  012703  007777                                MOV     #7777,R3
4007 040116                                                 WTDSRA  R3,XRR
     040116  012700  000016                                MOV     #SETXRR,R0      ; WRITE R3 INTO XRR .
     040122  050300                                        BIS     R3,R0
     040124  010077  142704                                MOV     R0,@DSR
4008 040130  012701  100000                                MOV     #100000,R1
```

```
4009 040134                                        RDDSRA  DSR,R2
     040134  012777  000000  142672                MOV     #SELDSR,@DSR    ; READ DSR INTO R2 .
     040142  017702  142666                        MOV     @DSR,R2
4010 040146                                        IFERROR XRDSE,DUADRE
     040146  020102                                CMP     R1,R2           ; OK?
     040150  001410                                BEQ     95$             ; YES.
     040152                                        HRDERR  XRDSE,DUADRE,   ; NO.
     040172                             95$:
4011 040172  012701  000001                        MOV     #1,R1
4012 040176                                        RDDSRA  PCS,R2
     040176  012777  000001  142630                MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
     040204  017702  142624                        MOV     @DSR,R2
4013 040210                                        IFERROR XRPCE,DUADRE
     040210  020102                                CMP     R1,R2           ; OK?
     040212  001410                                BEQ     96$             ; YES.
     040214                                        HRDERR  XRPCE,DUADRE,   ; NO.
     040234                             96$:
4014 040234  012701  100000                        MOV     #100000,R1
4015 040240                                        RDDSRA  FLG,R2
     040240  012777  000002  142566                MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
     040246  017702  142562                        MOV     @DSR,R2
4016 040252                                        IFERROR XRFLE,DUADRE
     040252  020102                                CMP     R1,R2           ; OK?
     040254  001410                                BEQ     97$             ; YES.
     040256                                        HRDERR  XRFLE,DUADRE,   ; NO.
     040276                             97$:
4017 040276  012701  000003                        MOV     #3,R1
4018 040302                                        RDDSRA  CSR,R2
     040302  012777  000003  142524                MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     040310  017702  142520                        MOV     @DSR,R2
4019 040314                                        IFERROR XRCSE,DUADRE
     040314  020102                                CMP     R1,R2           ; OK?
     040316  001410                                BEQ     98$             ; YES.
     040320                                        HRDERR  XRCSE,DUADRE,   ; NO.
     040340                             98$:
4020 040340  012701  000000                        MOV     #0,R1
4021 040344                                        RDDSRA  MRR,R2
     040344  012777  000004  142462                MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
     040352  017702  142456                        MOV     @DSR,R2
4022 040356  042702  170000                        BIC     #170000,R2      ; MASK OUT MPM.
4023 040362                                        IFERROR XRMRE,DUADRE
     040362  020102                                CMP     R1,R2           ; OK?
     040364  001410                                BEQ     99$             ; YES.
     040366                                        HRDERR  XRMRE,DUADRE,   ; NO.
     040406                             99$:
4024 040406                                        RDDSRA  MPM,R2
     040406  012777  000004  142420                MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
     040414  017702  142414                        MOV     @DSR,R2
4025 040420  042702  007777                        BIC     #7777,R2        ; MASK OUT MPM.
4026 040424                                        IFERROR XRMPE,DUADRE
     040424  020102                                CMP     R1,R2           ; OK?
     040426  001410                                BEQ     100$            ; YES.
     040430                                        HRDERR  XRMPE,DUADRE,   ; NO.
     040450                             100$:
4027 040450                                        RDDSRA  XPM,R2
     040450  012777  000006  142356                MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
     040456  017702  142352                        MOV     @DSR,R2
```

```
                                                BIC     #7777,R2        ; MASK OUT XRR.
                                                IFERROR XRXPE,DUADRE
                                                CMP     R1,R2           ; OK?
                                                BEQ     101$            ; YES.
                                                HRDERR  XRXPE,DUADRE,   ; NO.
                                        101$:
                                                ENDSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
                                                BGNSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
                                                INITDP
                                                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
                                                MOV     #170000,R3
                                                WTDSRA  R3,XPM
                                                MOV     #SETXPM,R0      ; WRITE R3 INTO XPM .
                                                BIS     R3,R0
                                                MOV     R0,@DSR
                                                MOV     #100000,R1
                                                RDDSRA  DSR,R2
                                                MOV     #SELDSR,@DSR    ; READ DSR INTO R2 .
                                                MOV     @DSR,R2
                                                IFERROR XPDSE,DUADRE
                                                CMP     R1,R2           ; OK?
                                                BEQ     102$            ; YES.
                                                HRDERR  XPDSE,DUADRE,   ; NO.
                                        102$:
                                                MOV     #1,R1
                                                RDDSRA  PCS,R2
                                                MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
                                                MOV     @DSR,R2
                                                IFERROR XPPCE,DUADRE
                                                CMP     R1,R2           ; OK?
                                                BEQ     103$            ; YES.
                                                HRDERR  XPPCE,DUADRE,   ; NO.
                                        103$:
                                                MOV     #100000,R1
                                                RDDSRA  FLG,R2
                                                MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
                                                MOV     @DSR,R2
                                                IFERROR XPFLE,DUADRE
                                                CMP     R1,R2           ; OK?
                                                BEQ     104$            ; YES.
                                                HRDERR  XPFLE,DUADRE,   ; NO.
                                        104$:
                                                MOV     #3,R1
                                                RDDSRA  CSR,R2
                                                MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
                                                MOV     @DSR,R2
                                                IFERROR XPCSE,DUADRE
                                                CMP     R1,R2           ; OK?
                                                BEQ     105$            ; YES.
                                                HRDERR  XPCSE,DUADRE,   ; NO.
                                        105$:
                                                MOV     #0,R1
                                                RDDSRA  MRR,R2
                                                MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
                                                MOV     @DSR,R2
                                                BIC     #170000,R2      ; MASK OUT MPM.
                                                IFERROR XPMRE,DUADRE
```

```
        040772  020102                          CMP     R1,R2           ; OK?
        040774  001410                          BEQ     106$            ; YES.
        040776                                  HRDERR  XPMRE,DUADRE,   ; NO.
        041016                          106$:
  4051  041016                                  RDDSRA  MPM,R2
        041016  012777  000004  142010          MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
        041024  017702  142004                  MOV     @DSR,R2
  4052  041030  042702  007777                  BIC     #7777,R2        ; MASK OUT MRR.
  4053  041034                                  IFERROR XPMPE,DUADRE
        041034  020102                          CMP     R1,R2           ; OK?
        041036  001410                          BEQ     107$            ; YES.
        041040                                  HRDERR  XPMPE,DUADRE,   ; NO.
        041060                          107$:
  4054  041060                                  RDDSRA  XRR,R2
        041060  012777  000006  141746          MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
        041066  017702  141742                  MOV     @DSR,R2
  4055  041072  042702  170000                  BIC     #170000,R2      ; MASK OUT XPM.
  4056  041076                                  IFERROR XPXRE,DUADRE
        041076  020102                          CMP     R1,R2           ; OK?
        041100  001410                          BEQ     108$            ; YES.
        041102                                  HRDERR  XPXRE,DUADRE,   ; NO.
        041122                          108$:
  4057  041122                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
  4058  041124                                  COMEND  1$
        041124  004737  027646                  JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
        041130  103401                          BCS     109$
        041132  000402                          BR      110$

        041134  000137  035710          109$:   JMP     1$

        041140  004737  030072          110$:   JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
  4059  041144                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
  4060  041150                                  END.TEST

                                  ;*************************************************************************
                                  ;*
                                  ;*      END TEST 4
                                  ;*
                                  ;*************************************************************************
```

```
4062
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 5    -   INCREMENTING REGISTERS
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 10.)

4063                            ;
4064                            ; THE READ/WRITE CAPABILITIES OF THE REGISTERS WILL BE CHECKED
4065                            ; BY "INCREMENTING" EACH REGISTER FROM ITS LOWEST POSSIBLE
4066                            ; VALUE, MINUS ONE, TO ITS HIGHEST POSSIBLE VALUE, PLUS ONE,
4067                            ; IE. 1 TO 177777 TO 0.
4068                            ; R1=EXPECTED, R2=RECEIVED, R0,R3=WORK REGS.
4069                            ;
4070 041232                    INCREG: COMBEG
      041232  004737  027702            JSR     PC,TSTGO        ; TITLE.
      041236  004737  026074            JSR     PC,DPRESET      ; DO SOFT INIT.
4071  041242  012701  000023    1$:     MOV     #SELCSR+20,R1   ; INCREMENT CSR.
4072
4073 041246                            BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4074 041250                    2$:     WTDSRA  R1,CSR
      041250  012700  000013            MOV     #SETCSR,R0      ; WRITE R1 INTO CSR .
      041254  050100                    BIS     R1,R0
      041256  010077  141552            MOV     R0,@DSR
4075 041262                            RDDSRA  CSR,R2
      041262  012777  000003  141544    MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      041270  017702  141540            MOV     @DSR,R2
4076 041274                            IFERROR ICSRE,EXPREC
      041274  020102                    CMP     R1,R2           ; OK?
      041276  001410                    BEQ     64$             ; YES.
      041300                            HRDERR  ICSRE,EXPREC,   ; NO.
      041320                    64$:
4077 041320                            CKLOOP          ; LOOP IF ERROR.
4078  041322  062701  000020          ADD     #20,R1          ; INCREMENT.
4079  041326  020127  000003          CMP     R1,#SELCSR      ; DONE?
4080  041332  001346                  BNE     2$              ; NO.
4081  041334                          ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4082
4083  041336  012701  000001          MOV     #1,R1           ; INCREMENT MRR.
4084  041342  012703  000020          MOV     #20,R3
4085
4086 041346                            BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4087 041350                    3$:     WTDSRA  R3,MRR
      041350  012700  000014            MOV     #SETMRR,R0      ; WRITE R3 INTO MRR .
      041354  050300                    BIS     R3,R0
      041356  010077  141452            MOV     R0,@DSR
4088 041362                            RDDSRA  MRR,R2
      041362  012777  000004  141444    MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
      041370  017702  141440            MOV     @DSR,R2
4089 041374                            IFERROR IMRRE,EXPREC
      041374  020102                    CMP     R1,R2           ; OK?
      041376  001410                    BEQ     65$             ; YES.
      041400                            HRDERR  IMRRE,EXPREC,   ; NO.
      041420                    65$:
4090 041420                            CKLOOP          ; LOOP IF ERROR.
```

```
4091 041422  062703  000020                ADD     #20,R3          ; INCREMENT.
4092 041426  062701  000001                ADD     #1,R1
4093 041432  020127  000001                CMP     R1,#1           ; DONE?
4094 041436  001403                        BEQ     4$              ; YES.
4095 041440  042701  170000                BIC     #170000,R1      ; NO.
4096 041444  000741                        BR      3$              ;
4097
4098 041446                          4$:   ENDSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4099
4100 041450  012703  000020                MOV     #20,R3          ; INCREMENT MPM.
4101
4102 041454                                BGNSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4103 041456                          5$:   WTDSRA  R3,MPM
     041456  012700  000015                MOV     #SETMPM,R0      ; WRITE R3 INTO MPM .
     041462  050300                        BIS     R3,R0
     041464  010077  141344                MOV     R0,@DSR
4104 041470  010301                        MOV     R3,R1           ;GET WRITTEN DATA
4105 041472  042701  007777                BIC     #7777,R1        ;LEAVE MASK BITS IN R1
4106 041476                                RDDSRA  MPM,R2
     041476  012777  000004  141330        MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
     041504  017702  141324                MOV     @DSR,R2
4107 041510                                IFERROR IMPME,EXPREC
     041510  020102                        CMP     R1,R2           ; OK?
     041512  001410                        BEQ     66$             ; YES.
     041514                                HRDERR  IMPME,EXPREC,   ; NO.
     041534                          66$:
4108 041534  005001                        CLR     R1              ;ASSUME WRITE-PROTECT BIT IS 0
4109 041536  032703  001000                BIT     #BIT9,R3        ;DID WE WRITE IT TO 1?
4110 041542  001402                        BEQ     50$             ;BR IF NOT.
4111 041544  052701  100000                BIS     #BIT15,R1       ;SET BIT 15 FOR HBASE COMPARE IF YES.
4112 041550                          50$:  RDDSRA  HBA,R2
     041550  012777  000005  141256        MOV     #SELHBA,@DSR    ; READ HBA INTO R2 .
     041556  017702  141252                MOV     @DSR,R2
4113 041562                                IFERROR IMPWPE,EXPREC
     041562  020102                        CMP     R1,R2           ; OK?
     041564  001410                        BEQ     67$             ; YES.
     041566                                HRDERR  IMPWPE,EXPREC,  ; NO.
     041606                          67$:
4114 041606                                CKLOOP                  ; LOOP IF ERROR.
4115 041610  062703  000020                ADD     #20,R3          ; INCREMENT.
4116 041614  020327  000020                CMP     R3,#20          ; DONE?
4117 041620  001316                        BNE     5$              ; NO.
4118 041622                                ENDSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4119
4120 041624  012701  000001                MOV     #1,R1           ; INCREMENT XRR.
4121 041630  012703  000020                MOV     #20,R3
4122
4123 041634                                BGNSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4124 041636                          6$:   WTDSRA  R3,XRR
     041636  012700  000016                MOV     #SETXRR,R0      ; WRITE R3 INTO XRR .
     041642  050300                        BIS     R3,R0
     041644  010077  141164                MOV     R0,@DSR
4125 041650                                RDDSRA  XRR,R2
     041650  012777  000006  141156        MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
     041656  017702  141152                MOV     @DSR,R2
4126 041662                                IFERROR IXRRE,EXPREC
     041662  020102                        CMP     R1,R2           ; OK?
```

```
        041664  001410                          BEQ     68$             ; YES.
        041666                                  HRDERR  IXRRE,EXPREC,   ; NO.
        041706                          68$:
  4127  041706                                  CKLOOP                  ; LOOP IF ERROR.
  4128  041710  062703  000020                  ADD     #20,R3          ; INCREMENT.
  4129  041714  062701  000001                  ADD     #1,R1
  4130  041720  020127  000001                  CMP     R1,#1           ; DONE?
  4131  041724  001403                          BEQ     7$              ; YES.
  4132  041726  042701  170000                  BIC     #170000,R1      ; NO.
  4133  041732  000741                          BR      6$              ;
  4134
  4135  041734                          7$:     ENDSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
  4136
  4137  041736  012703  000020                  MOV     #20,R3          ; INCREMENT XPM.
  4138
  4139  041742                                  BGNSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
  4140  041744                          8$:     WTDSRA  R3,XPM
        041744  012700  000017                  MOV     #SETXPM,R0      ; WRITE R3 INTO XPM .
        041750  050300                          BIS     R3,R0
        041752  010077  141056                  MOV     R0,@DSR
  4141  041756  010301                          MOV     R3,R1           ;GET WRITTEN DATA
  4142  041760  042701  007777                  BIC     #7777,R1        ;LEAVE MASK BITS IN R1
  4143  041764                                  RDDSRA  XPM,R2
        041764  012777  000006  141042          MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
        041772  017702  141036                  MOV     @DSR,R2
  4144  041776                                  IFERROR IXPME,EXPREC
        041776  020102                          CMP     R1,R2           ; OK?
        042000  001410                          BEQ     69$             ; YES.
        042002                                  HRDERR  IXPME,EXPREC,   ; NO.
        042022                          69$:
  4145  042022  005001                          CLR     R1              ;ASSUME WRITE-PROTECT BIT IS 0
  4146  042024  032703  001000                  BIT     #BIT9,R3        ;DID WE WRITE IT TO 1?
  4147  042030  001402                          BEQ     80$             ;BR IF NOT.
  4148  042032  052701  040000                  BIS     #BIT14,R1       ;SET BIT 14 FOR HBASE COMPARE IF YES.
  4149  042036                          80$:    RDDSRA  HBA,R2
        042036  012777  000005  140770          MOV     #SELHBA,@DSR    ; READ HBA INTO R2 .
        042044  017702  140764                  MOV     @DSR,R2
  4150  042050                                  IFERROR IXPWPE,EXPREC
        042050  020102                          CMP     R1,R2           ; OK?
        042052  001410                          BEQ     70$             ; YES.
        042054                                  HRDERR  IXPWPE,EXPREC,  ; NO.
        042074                          70$:
  4151  042074                                  CKLOOP                  ; LOOP IF ERROR.
  4152  042076  062703  000020                  ADD     #20,R3          ; INCREMENT.
  4153  042102  020327  000020                  CMP     R3,#20          ; DONE?
  4154  042106  001316                          BNE     8$              ; NO.
  4155  042110                                  ENDSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
  4156
  4157  042112                                  COMEND  1$
        042112  004737  027646                  JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
        042116  103401                          BCS     71$
        042120  000402                          BR      72$

        042122  000137  041242          71$:    JMP     1$

        042126  004737  030072          72$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
  4158  042132                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
4159
4160 042136                                    END.TEST

                                        ;*****************************************************************************
                                        ;*
                                        ;*      END TEST 5
                                        ;*
                                        ;*****************************************************************************
```

```
4162
                                ;*****************************************************************************
                                ;*
                                ;*      BEGIN TEST 6    -   DPU START-STOP
                                ;*
                                ;*****************************************************************************

                                ; (ITERATION COUNT = 20.)

4163                            ;
4164                            ; FIRST INSURE THAT START, RESUME, DNOP, DJMP, AND STOP
4165                            ; ALL FUNCTION CORRECTLY.
4166                            ; IF NOT, ABORT, NO POINT IN CONTINUING.
4167                            ;
4168 042210                    SSTST:  COMBEG
     042210  004737  027702             JSR     PC,TSTGO        ; TITLE.
     042214  004737  026074             JSR     PC,DPRESET      ; DO SOFT INIT.
4169
4170 042220                    1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4171
4172 042222  012702  042526             MOV     #SST2,R2        ;SET 1ST EXP'D DPC...
4173 042226  017703  140602             MOV     @DSR,R3
4174 042232  042703  176000             BIC     #^C1777,R3
4175 042236  052703  172000             BIS     #STOP,R3        ;...AND DSR.
4176 042242  012777  042516  140562     MOV     #SST1,@DPC      ;XCT START, DNOPS, AND STOP.
4177 042250  004737  027454             JSR     PC,WAITF        ;WAIT 1/2 SEC (MORE OR LESS).
4178 042254  027702  140552             CMP     @DPC,R2         ;DPC SHOULD POINT TO 'RT2''.
4179 042260  001411                     BEQ     2$              ;BR IF IT'S RIGHT.
4180 042262                             DFERR   SSF,PCSERR      ;START-NOP-STOP FAILS.
4181 042302  000421                     BR      3$              ;...BYPASS RESUME TEST.
4182 042304  012702  042550     2$:     MOV     #SST3,R2        ;OK, SET NEXT EXP'D DPC.
4183 042310  005277  140516             INC     @DPC            ;XCT RESUME, DNOP'S, AND DJMP.
4184 042314  004737  027454             JSR     PC,WAITF
4185 042320  027702  140506             CMP     @DPC,R2         ;NOW DPC SHOULD BE AT 'RT3''.
4186 042324  001412                     BEQ     4$              ;BR IF SO
4187 042326                             DFERR   RJF,PCSERR      ;RESUME-NOP-JUMP FAILS.
4188 042346  004737  026756     3$:     JSR     PC,RELEAS       ;UNHANG IF HUNG
4189 042352  004737  023422     4$:     JSR     PC,IIRCHK       ;VERIFY THAT INTERNAL REGS REMAIN STATIC
4190 042356                             ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4191
4192
4193                            ; SO FAR, SO GOOD.  NOW LET'S PUT HIM IN A JUMP-SELF
4194                            ; LOOP, AND SEE IF WE CAN STOP HIM FROM THE CPU.
4195                            ;
4196
4197 042360                             BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4198 042362                             DPSTART #SST2
     042362  012777  042526  140442     MOV     #SST2,@DPC      ; START THE DPU.
     042370  004737  027454             JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4199 042374  005277  140432             INC     @DPC            ;RESUME, XCT DJMP SELF.
4200 042400  004737  027454             JSR     PC,WAITF
4201 042404  103011                     BCC     5$              ;BR IF HE'S DOING IT.
4202 042406                             HRDERR  JSF             ;JUMP SELF FAILS.
4203 042426  000417                     BR      6$
4204 042430  000240  000240     5$:     240,240
4205 042434  005077  140374             CLR     @DSR            ;XCT 'EXTERNAL STOP''.
4206 042440  004737  027454             JSR     PC,WAITF
```

```
4207 042444  103410                          BCS     6$              ;BR IF HE STOPPED.
4208 042446                                  HRDERR  ESF             ;EXT STOP FAILS.
4209 042466  004737  023422          6$:     JSR     PC,IIRCHK       ;MAKE SURE INTERNALS STILL OK.
4210 042472  004737  026756                  JSR     PC,RELEAS       ;IT DIDN'T, GO RELEASE DPU.
4211 042476                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4212
4213 042500                                  LOOPTO  1$              ; REPEAT 'TIL LOOPER EXPIRES.
     042500  004737  027646                  JSR     PC,LOOP
     042504  103645                          BCS     1$
4214 042506  004737  030072                  JSR     PC,TSTEND       ; PRINT ERROR SUMMARY IF REQ'D.
4215 042512                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
4216                                 ;
4217                                 ; DISPLAY FILE FOR THIS TEST.
4218                                 ;
4219 042516  164000  164000  164000  SST1:   DNOP,DNOP,DNOP
4220 042524  173000                          STOPN
4221 042526  164000  164000  164000  SST2:   DNOP,DNOP,DNOP
4222 042534  160000  042546                  DJMP,.+10
4223 042540  173000  173000  173000          STOPN,STOPN,STOPN       ;SHOULD SKIP OVER THESE...
4224 042546  173000                          STOPN                   ;...AND LAND HERE.
4225 042550  164000  164000          SST3:   DNOP,DNOP
4226 042554  160000  042554                  DJMP,.-2                ;JUMP SELF FOR 'EXT STOP'.
4227 042560  173000  173000                  STOPN,STOPN             ;INSURANCE.
4228
4229 042564                                  END.TEST

                                 ;*******************************************************************************
                                 ;*
                                 ;*      END TEST 6
                                 ;*
                                 ;*******************************************************************************
```

```
4231
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 7    -   DPU OPCODES
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 5.)

4232                                    ;
4233                                    ; FIRST, TEST ALL THOSE OPCODES WHICH XFER PIXEL DATA TO THE
4234                                    ; DPU, OPCODES 100000 THRU 130000, (CHAR THRU REL POINT).
4235                                    ; USE ALL 10 BITS OF THE PIXEL DATA FIELD.
4236                                    ;
4237 042632                            OPCOD:  COMBEG
      042632  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      042636  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
4238  042642  005037  023054                    CLR     LOOPFL          ;CLEAR "LOOP-FLAG" FOR C-BASE DATA.
4239  042646  012737  043412  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
4240  042654  012702  043446            14$:    MOV     #OPC1+6,R2      ;SET EXPECTED DPC.
4241  042660  012737  102000  043440            MOV     #CHAR!LO,OPC1   ;INIT OPCODE = CHAR, LEVEL 0.
4242  042666  012703  172000            1$:     MOV     #STOP,R3        ;SET EXPECTED DSR.
4243
4244  042672                            2$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4245
4246  042674  012777  043440  140130            MOV     #OPC1,@DPC      ;START, XCT OPCODE+DATA.
4247  042702  004737  027454                    JSR     PC,WAITF
4248  042706  027703  140122                    CMP     @DSR,R3         ;DOES DSR = STOP + DATA ??
4249  042712  001410                            BEQ     3$              ;YES
4250  042714                                    HRDERR  PDXI,PCSERR     ;PIXEL DATA XFER WRONG.
4251  042734  013737  043440  023056    3$:     MOV     OPC1,GDDAT      ;GET OPCODE
4252  042742  042737  003777  023056            BIC     #3777,GDDAT     ;CLEAR OUT INTENSITY
4253  042750  004737  022410                    JSR     PC,FLGCHK       ;SEE THAT OPCODE IS IN FLAGS.
4254  042754  012737  000001  023056            MOV     #1,GDDAT        ;EXPECT "EMPTY" ON PCSAVE...
4255  042762  004737  022360                    JSR     PC,PCSCHK       ;...AND GO CHECK FOR IT.
4256  042766  012737  000003  023056            MOV     #SELCSR,GDDAT   ;EXPECT CLEAR CSR (EXC. SEL)...
4257  042774  004737  022440                    JSR     PC,CSRCHK       ;...AND CHECK FOR IT.
4258  043000  013737  023054  023056            MOV     LOOPFL,GDDAT    ;EXPECT CBASE EITHER 0 OR DNOP...
4259  043006  004737  022624                    JSR     PC,CBCHK        ;...AND CHECK FOR IT.
4260  043012  005037  023056                    CLR     GDDAT           ;EXPECT 0 IN REMAINING REGS...
4261  043016  004737  022466                    JSR     PC,MMCHK        ;...MAIN MEM MGMT.
4262  043022  004737  022566                    JSR     PC,XMCHK        ;...AUX MEM MGMT.
4263  043026  004737  022526                    JSR     PC,HBCHK        ;...HISTOGRAM BASE.
4264  043032                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4265
4266  043034  020327  173777                    CMP     R3,#STOP!1777   ;ALL DATA BITS DONE ??
4267  043040  001404                            BEQ     4$              ;BR IF SO.
4268  043042  005203                            INC     R3              ;NO BUMP DATA BITS...
4269  043044  005237  043440                    INC     OPC1            ;...HERE TOO...
4270  043050  000710                            BR      2$              ;...AND CONTINUE.
4271
4272  043052  023727  043440  133777    4$:     CMP     OPC1,#RPNT!3777 ;DONE THRU RPNT ???
4273  043060  001407                            BEQ     5$              ;BR IF SO.
4274  043062  062737  004000  043440            ADD     #BIT11,OPC1     ;NO, SET NEXT OPCODE...
4275  043070  042737  001777  043440            BIC     #1777,OPC1      ;...LEVEL 0...
4276  043076  000673                            BR      1$              ;...AND CONTINUE.
4277                                  ;
```

```
                                        ; NOW OF THE REMAINING OPCODES (134000 - 176000), EXECUTE ALL
                                        ; EXCEPT BM0/BM1(134000/136000), DJMP(160000), AND STOP(172000),
                                        ; WITH BITS <9:0> OF THE OPCODE = ZERO, AND NO FOLLOWING DATA.
                                        ; THEY ALL (INCLUDING THOSE UNDEFINED/SPARE) SHOULD APPEAR AS
                                        ; NOPS, HAVING NO AFFECT ON THE DPU AS FAR AS FINAL REGISTER
                                        ; CONTENT IS CONCERNED.
                                        ; NOTE: THIS TEST WILL LEAVE IM CHAN 0 IN A "PROTECT" STATE.
                                        ;
043100  017704  137732          5$:     MOV     @DXR,R4         ; ALL REGISTERS SHOULD REMAIN...
043104  017705  137730                  MOV     @DYR,R5         ;...UNCHANGED.
043110  012737  140000  043440          MOV     #140000,OPC1    ; FIRST OPCODE = 1ST SPARE.

043116                          6$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.

043120  012777  043440  137704          MOV     #OPC1,@DPC
043126  004737  027454                  JSR     PC,WAITF
043132  027702  137674                  CMP     @DPC,R2
043136  001003                          BNE     7$
043140  027703  137670                  CMP     @DSR,R3
043144  001411                          BEQ     8$
043146                          7$:     HRDERR  OPCF,PCSERR     ;PC OR SR WRONG AFTER NULL.
043166  000416                          BR      11$
043170  027704  137642          8$:     CMP     @DXR,R4
043174  001003                          BNE     9$
043176  027705  137636                  CMP     @DYR,R5
043202  001410                          BEQ     11$
043204                          9$:     HRDERR  OPCF,XYERR      ;XR OR YR WRONG AFTER NULL.
043224  012737  130000  023056  11$:    MOV     #RPNT,GDDAT     ;SAY WE SHOULD SEE 'RPNT" OP IN FLAGS.
043232  023727  043440  144000          CMP     OPC1,#RNLN      ;GO TO RUN-LENGTH YET?
043240  103403                          BLO     10$             ;BR IF NOT
043242  012737  144000  023056          MOV     #RNLN,GDDAT     ;IF YES, EXPECT TO SEE IT.
043250  004737  022410          10$:    JSR     PC,FLGCHK       ;CHECK FLAGS FOR THE GRAPHIC OP.
043254  012737  000003  023056          MOV     #SELCSR,GDDAT   ;...CHECK OTHER REGS, EXCEPT HBASE & CBASE
043262  004737  022440                  JSR     PC,CSRCHK       ;...CSR
043266  012737  000001  023056          MOV     #1,GDDAT
043274  004737  022360                  JSR     PC,PCSCHK       ;...PCSAVE
043300  005037  023056                  CLR     GDDAT           ;    OTHERS S/B 0
043304  004737  022466                  JSR     PC,MMCHK        ;...MAIN MEM MGMT
043310  004737  022566                  JSR     PC,XMCHK        ;...AUX MEM MGMT
043314                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.

043316  062737  002000  043440  15$:    ADD     #BIT10,OPC1     ;SET NEXT OPCODE.
043324  023727  043440  160000          CMP     OPC1,#DJMP
043332  001771                          BEQ     15$             ; BYPASS DJMP...
043334  023727  043440  172000          CMP     OPC1,#STOP
043342  001765                          BEQ     15$             ;...AND STOP.
043344  023727  043440  162000          CMP     OPC1,#IMREAD    ; BYPASS DMA PIXEL READ.
043352  001761                          BEQ     15$             ;
043354  005737  043440                  TST     OPC1            ; DONE THRU LAST OPCODE ??
043360  001256                          BNE     6$              ;CONTINUE IF NOT.

043362  013737  043442  023054  12$:    MOV     OPC1+2,LOOPFL           ;GET DATA FOR CHAR-BASE
043370                                  LOOPTO  13$             ;REPEAT 'TIL LOOPER EXPIRES. (BRANCH OUT OF RANGE) ;aaaI
043370  004737  027646                  JSR     PC,LOOP
043374  103404                          BCS     13$
043376  004737  030072          30$:    JSR     PC,TSTEND
043402                          EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
 4333 043406  000137  042654          13$:    JMP     14$             ;USE JMP TO GET BACK FOR LOOP.
 4334                                 ;
 4335                                 ; ON ERROR, SHOW THE FAILING DPU OPCODE.
 4336                                 ;
 4337 043412                          20$:    PRINTX  #OPCFX,OPC1     ; EXTENSION SHOWS BAD OPCODE.
 4338 043436  000207                          RTS     PC
 4339                                 ;
 4340                                 ; DISPLAY CODE.
 4341                                 ;
 4342 043440  102000  164000  173000  OPC1:   CHAR!LO, DNOP, STOPN    ; DISPLAY CODE.
 4343
 4344 043446                          END.TEST

                                     ;*******************************************************************************
                                     ;*
                                     ;*      END TEST 7
                                     ;*
                                     ;*******************************************************************************
```

```
4346
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 8    -   INCREMENTING HISTOGRAM BASE ADDRESS
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 5.)

4347
4348                            ;
4349                            ; THE READ/WRITE CAPABILITY OF THE BASE HISTOGRAM COORDINATE WILL BE
4350                            ; CHECKED BY "INCREMENTING" IT FROM 2 TO 1776 TO 0.
4351                            ;  (NOTE THAT THE WRITTEN DATA WILL APPEAR SHIFTED 1 PLACE TO THE RIGHT.)
4352                            ; R1= EXPECTED, R2=RECEIVED, R0,R3=WORK REGS.
4353                            ;
4354
4355 043544                    INCHBA: COMBEG
      043544  004737  027702            JSR     PC,TSTGO        ; TITLE.
      043550  004737  026074            JSR     PC,DPRESET      ; DO SOFT INIT.
4356 043554  012703  000002    1$:     MOV     #2,R3
4357 043560                            BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4358 043562  010337  043712    2$:     MOV     R3,11$          ; SET VALUE FOR HBA CODE.
4359 043566  010301                    MOV     R3,R1
4360 043570  006201                    ASR     R1
4361 043572  012777  043710  137232    MOV     #10$,@DPC       ; WRITE HBA.
4362 043600  004737  027454            JSR     PC,WAITF
4363 043604                            RDDSRA  HBA,R2
      043604  012777  000005  137222    MOV     #SELHBA,@DSR    ; READ HBA INTO R2 .
      043612  017702  137216            MOV     @DSR,R2
4364 043616                            IFERROR IHBAE,EXPREC
      043616  020102                    CMP     R1,R2           ; OK?
      043620  001410                    BEQ     64$             ; YES.
      043622                            HRDERR  IHBAE,EXPREC,   ; NO.
      043642                    64$:
4365 043642                            CKLOOP          ; LOOP ON ERROR.
4366 043644  062703  000002            ADD     #2,R3           ; INCREMENT.
4367 043650  042703  176001            BIC     #176001,R3      ; MASK UNWANTED.
4368 043654  020327  000000            CMP     R3,#0           ; DONE?
4369 043660  001340                    BNE     2$              ; NO.
4370 043662                    30$:    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4371 043664                            COMEND  1$
      043664  004737  027646            JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
      043670  103401                    BCS     65$
      043672  000402                    BR      66$

      043674  000137  043554    65$:    JMP     1$

      043700  004737  030072    66$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
4372 043704                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
4373
4374                            ;
4375                            ; DISPLAY CODE.
4376                            ;
4377 043710  150000    10$:    SETHB
4378 043712  000000    11$:    0
4379 043714  173000            STOPN
```

```
4380
4381 043716                    END.TEST

                        ;*****************************************************************************
                        ;*
                        ;*      END TEST 8
                        ;*
                        ;*****************************************************************************
```

```
4383
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 9    -   INCREMENTING CHARACTER BASE ADDRESS
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 2.)

4384
4385
4386                            ;
4387                            ; THE READ/WRITE CAPABILITY OF THE BASE CHARACTER ADDRESS WILL BE
4388                            ; CHECKED BY "INCREMENTING" IT FROM 1 TO 177777 TO 0.
4389                            ; R1= EXPECTED, R2=RECEIVED, R0,R3=WORK REGS.
4390                            ;
4391 044014                    INCCBA: COMBEG
      044014  004737  027702            JSR     PC,TSTGO        ; TITLE.
      044020  004737  026074            JSR     PC,DPRESET      ; DO SOFT INIT.
4392 044024  012701  000001    1$:      MOV     #1,R1
4393 044030                            BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4394 044032  010137  044152    2$:      MOV     R1,11$          ; SET VALUE FOR CBA CODE.
4395 044036  012777  044150  136766    MOV     #10$,@DPC       ; WRITE CBA.
4396 044044  004737  027454            JSR     PC,WAITF
4397 044050                            RDDSRA  CBA,R2
      044050  012777  000007  136756    MOV     #SELCBA,@DSR    ; READ CBA INTO R2 .
      044056  017702  136752            MOV     @DSR,R2
4398 044062                            IFERROR ICBAE,EXPREC
      044062  020102                    CMP     R1,R2           ; OK?
      044064  001410                    BEQ     64$             ; YES.
      044066                            HRDERR  ICBAE,EXPREC,   ; NO.
      044106                    64$:
4399 044106                             CKLOOP          ; LOOP ON ERROR.
4400 044110  062701  000001            ADD     #1,R1           ; INCREMENT.
4401 044114  020127  000000            CMP     R1,#0           ; DONE?
4402 044120  001344                    BNE     2$              ; NO.
4403 044122                    30$:    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4404 044124                            COMEND  1$
      044124  004737  027646            JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
      044130  103401                    BCS     65$
      044132  000402                    BR      66$

      044134  000137  044024    65$:    JMP     1$

      044140  004737  030072    66$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
4405 044144                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
4406
4407                            ;
4408                            ; DISPLAY CODE.
4409                            ;
4410 044150  152000    10$:     SETCB
4411 044152  000000    11$:     0
4412 044154  173000             STOPN
4413
4414 044156                      END.TEST

                                ;*************************************************************************
```

```
;*
;*      END TEST 9
;*
;*****************************************************************************
```

```
4416
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 10   -   DJMS/DPOP
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 10.)

4417
4418                                    ;
4419                                    ; CHECK THE OPERATION OF THE DJMS AND DPOP OP CODES.
4420                                    ;
4421 044222                            JMSPOP: COMBEG
     044222  004737  027702                     JSR     PC,TSTGO        ; TITLE.
     044226  004737  026074                     JSR     PC,DPRESET      ; DO SOFT INIT.
4422
4423 044232                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4424
4425 044234                                    DPSTART #10$            ; DO A DJMS.
     044234  012777  044630  136570             MOV     #10$,@DPC       ; START THE DPU.
     044242  004737  027454                     JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4426 044246  012701  044646                     MOV     #21$,R1
4427 044252  017702  136554                     MOV     @DPC,R2         ; READ DPC INTO R2.
4428 044256                                    IFERROR JPDPCE,EXPREC,,CKLOOP
     044256  020102                             CMP     R1,R2           ; OK?
     044260  001411                             BEQ     64$             ; YES.
     044262                                     HRDERR  JPDPCE,EXPREC,CKLOOP    ; NO.
     044304                             64$:
4429 044304  012701  044634                     MOV     #11$,R1
4430 044310                                    RDDSRA  PCS,R2
     044310  012777  000001  136516             MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
     044316  017702  136512                     MOV     @DSR,R2
4431 044322                                    IFERROR JPPCSE,EXPREC,,CKLOOP
     044322  020102                             CMP     R1,R2           ; OK?
     044324  001411                             BEQ     65$             ; YES.
     044326                                     HRDERR  JPPCSE,EXPREC,CKLOOP    ; NO.
     044350                             65$:
4432 044350                                    DPCONT                  ; NOW DO A DPOP.
     044350  052777  000001  136454             BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
     044356  004737  027454                     JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4433 044362  012701  044636                     MOV     #12$,R1
4434 044366  017702  136440                     MOV     @DPC,R2         ; READ DPC.
4435 044372                                    IFERROR JPDPCE,EXPREC,,CKLOOP
     044372  020102                             CMP     R1,R2           ; OK?
     044374  001411                             BEQ     66$             ; YES.
     044376                                     HRDERR  JPDPCE,EXPREC,CKLOOP    ; NO.
     044420                             66$:
4436 044420                                    DPCONT                  ; DO A DJMS (WITH AUX SEG BIT SET).
     044420  052777  000001  136404             BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
     044426  004737  027454                     JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4437 044432  012701  044647                     MOV     #21$!AUXSEG,R1
4438 044436  017702  136370                     MOV     @DPC,R2         ; READ DPC INTO R2.
4439 044442                                    IFERROR JPDPCE,EXPREC,,CKLOOP
     044442  020102                             CMP     R1,R2           ; OK?
     044444  001411                             BEQ     67$             ; YES.
     044446                                     HRDERR  JPDPCE,EXPREC,CKLOOP    ; NO.
```

```
     044470                                 67$:
4440 044470  012701  044642                      MOV     #13$,R1
4441 044474                                      RDDSRA  PCS,R2
     044474  012777  000001  136332              MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
     044502  017702  136326                      MOV     @DSR,R2
4442 044506                                      IFERROR JPPCSE,EXPREC,,CKLOOP
     044506  020102                              CMP     R1,R2           ; OK?
     044510  001411                              BEQ     68$             ; YES.
     044512                                      HRDERR  JPPCSE,EXPREC,CKLOOP    ; NO.
     044534                                 68$:
4443 044534                                      DPCONT                  ; NOW DPOP AGAIN.
     044534  052777  000001  136270              BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
     044542  004737  027454                      JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4444 044546  012701  044644                      MOV     #20$,R1
4445 044552  017702  136254                      MOV     @DPC,R2         ; READ DPC.
4446 044556                                      IFERROR JPDPCE,EXPREC
     044556  020102                              CMP     R1,R2           ; OK?
     044560  001410                              BEQ     69$             ; YES.
     044562                                      HRDERR  JPDPCE,EXPREC,  ; NO.
     044602                                 69$:
4447 044602                                      ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4448
4449 044604                                      COMEND  1$
     044604  004737  027646                      JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
     044610  103401                              BCS     70$
     044612  000402                              BR      71$

     044614  000137  044232              70$:    JMP     1$

     044620  004737  030072              71$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
4450 044624                                      EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
4451                                     ;
4452                                     ; DISPLAY CODE
4453                                     ;
4454 044630  160001  044644             10$:    DJMS,20$
4455 044634  173000                     11$:    STOPN
4456 044636  160001  044645             12$:    DJMS,20$!AUXSEG
4457 044642  173000                     13$:    STOPN
4458
4459 044644  173000                     20$:    STOPN
4460 044646  165000                     21$:    DPOP
4461 044650  173000                     22$:    STOPN
4462
4463 044652                                      END.TEST

                                         ;*****************************************************************************
                                         ;*
                                         ;*      END TEST 10
                                         ;*
                                         ;*****************************************************************************
```

```
4465                                    ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 11   -   MAIN MEM MGT ACCESS (OVER 28K)
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 1.)

4466
4467                                    ;
4468                                    ; CHECK THAT A DISPLAY FILE CAN BE ACCESSED VIA MAIN MEM MGT.
4469                                    ; IN MEMORY OVER 28K.
4470                                    ;
4471 044744                            MMMAC:  COMBEG
     044744  004737  027702                     JSR     PC,TSTGO        ; TITLE.
     044750  004737  026074                     JSR     PC,DPRESET      ; DO SOFT INIT.
4472 044754  005737  003244                     TST     KTFLG           ; GOT KT & >28K?
4473 044760  001016                             BNE     MMMACA          ; YES.
4474 044762                                     RFLAGS  R0              ; ARE WE PRINTING TEST NOS.?
4475 044764  032700  001000                     BIT     #PNT,R0         ; ?
4476 044770  001410                             BEQ     1$              ; BR IF NO -- DON'T PRINT
4477 044772                                     PRINTF  #MNOKT          ; NO.
4478 045012                            1$:      EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
4479
4480
4481 045016  004737  031322            MMMACA: JSR     PC,KTBKGD       ; SET UP BACKGROUND.
4482 045022  012703  001600            1$:     MOV     #1600,R3        ; INIT BANK PTR. (TO START OF 28K PAGE)
4483
4484 045026                            2$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4485
4486 045030  010337  172354                     MOV     R3,@#KIPAR6     ; SET KT TO TEST PAGE.
4487 045034  010301                             MOV     R3,R1           ; SET UP MRR.
4488 045036  006301                             ASL     R1
4489 045040  006301                             ASL     R1
4490 045042  006301                             ASL     R1
4491 045044  006301                             ASL     R1
4492 045046                                     WTDSRA  R1,MRR
     045046  012700  000014                     MOV     #SETMRR,R0      ; WRITE R1 INTO MRR .
     045052  050100                             BIS     R1,R0
     045054  010077  135754                     MOV     R0,@DSR
4493 045060  012705  140000                     MOV     #140000,R5      ; INIT LOC PTR.
4494 045064  004737  031270                     JSR     PC,KTON         ; ENABLE KT.
4495 045070  012715  164000                     MOV     #DNOP,(R5)      ; REPLACE ''STOPN'' WITH 'DNOP'' IN TEST LOC.
4496 045074  004737  031306                     JSR     PC,KTOFF        ; DISABLE KT.
4497 045100                                     DPSTART #0
     045100  012777  000000  135724             MOV     #0,@DPC ; START THE DPU.
     045106  004737  027454                     JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4498 045112  000413                             BR      4$
4499
4500 045114  004737  031270            3$:     JSR     PC,KTON         ; ENABLE KT.
4501 045120  012715  164000                     MOV     #DNOP,(R5)      ; REPLACE ''STOPN'' WITH 'DNOP'' IN TEST LOC.
4502 045124  004737  031306                     JSR     PC,KTOFF        ; DISABLE KT.
4503 045130                                     DPCONT
     045130  052777  000001  135674             BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
     045136  004737  027454                     JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4504 045142  010501                    4$:     MOV     R5,R1           ; R1 = EXPECTED DPC, AND -
```

```
4505 045144  042701  170000                BIC     #170000,R1      ; -
4506 045150  062701  000004                ADD     #4,R1           ; - (R3 ALREADY = EXPECTED MRR).
4507 045154  017702  135652                MOV     @DPC,R2         ; READ DPC INTO R2.
4508 045160                                RDDSRA  MRR,R4
     045160  012777  000004  135646        MOV     #SELMRR,@DSR    ; READ MRR INTO R4 .
     045166  017704  135642                MOV     @DSR,R4
4509 045172                                IFERRX2 MMMACE,MMMAE
     045172  020102                        CMP     R1,R2           ; 1ST ITEM OK?
     045174  001002                        BNE     65$             ; NO.
     045176  020304                        CMP     R3,R4           ; 2ND ITEM OK?
     045200  001410                        BEQ     64$             ; YES.
     045202                         65$:   HRDERR  MMMACE,MMMAE,   ; NO.
     045222                         64$:
4510 045222  004737  031270                JSR     PC,KTON         ; ENABLE KT.
4511 045226  012715  173000                MOV     #STOPN,(R5)     ; REPLACE ''DNOP'' WITH ''STOPN''.
4512 045232  004737  031306                JSR     PC,KTOFF        ; DISABLE KT.
4513
4514 045236  062705  000002                ADD     #2,R5           ; ADVANCE LOC PTR.
4515 045242  020527  147774                CMP     R5,#147774      ; DONE ENOUGH LOCS?
4516 045246  001322                        BNE     3$              ; NO.
4517 045250                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4518 045252  020337  003244                CMP     R3,KTFLG        ; YES. ALL BANKS DONE?
4519 045256  001403                        BEQ     9$              ; YES.
4520 045260  062703  000200                ADD     #200,R3         ; NO. ADVANCE BANK PTR.
4521 045264  000660                        BR      2$
4522
4523 045266                         9$:    COMEND  1$
     045266  004737  027646                JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
     045272  103401                        BCS     66$
     045274  000402                        BR      67$

     045276  000137  045022         66$:   JMP     1$

     045302  004737  030072         67$:   JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
4524 045306  004737  031306                JSR     PC,KTOFF        ; DISABLE KT11.
4525 045312                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
4526
4527
4528 045316     045     116     045  MNOKT:  .ASCIZ  /%N%A  !!! NO KT, OR <32K - NO TEST !!!/
4529                                        .EVEN
4530 045366                                END.TEST

                                ;*****************************************************************************
                                ;*
                                ;*      END TEST 11
                                ;*
                                ;*****************************************************************************
```

```
4532
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 12   -   AUX MEM MGT ACCESS (OVER 28K)
                                ;*
                                ;*************************************************************************

                                  (ITERATION COUNT = 1.)

4533
4534
4535                            ;
4536                            ; CHECK THAT A DISPLAY FILE CAN BE ACCESSED VIA AUX MEM MGT.
4537                            ; IN MEMORY OVER 28K.
4538 045456                    ;
     045456  004737  027702     XMMAC:  COMBEG
     045462  004737  026074             JSR     PC,TSTGO        ; TITLE.
4539 045466  005737  003244             JSR     PC,DPRESET      ; DO SOFT INIT.
4540 045472  001016                     TST     KTFLG           ; GOT KT & >28K?
4541 045474                             BNE     XMMACA          ; YES.
4542 045476  032700  001000             RFLAGS  R0
4543 045502  001410                     BIT     #PNT,R0         ; ARE WE PRINTING?
4544 045504                             BEQ     1$              ; BR IF NO
4545 045524                     1$:     PRINTF  #MNOKT          ; NO.
4546                                     EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
4547
4548 045530  004737  031322     XMMACA: JSR     PC,KTBKGD       ; SET UP BACKGROUND.
4549 045534  012703  001600     1$:     MOV     #1600,R3        ; INIT BANK PTR.
4550
4551 045540                     2$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4552
4553 045542  010337  172354             MOV     R3,@#KIPAR6     ; SET KT TO TEST PAGE.
4554 045546  010301                     MOV     R3,R1           ; SET UP XRR.
4555 045550  006301                     ASL     R1
4556 045552  006301                     ASL     R1
4557 045554  006301                     ASL     R1
4558 045556  006301                     ASL     R1
4559 045560                             WTDSRA  R1,XRR
     045560  012700  000016             MOV     #SETXRR,R0      ; WRITE R1 INTO XRR .
     045564  050100                     BIS     R1,R0
     045566  010077  135242             MOV     R0,@DSR
4560 045572  012705  140000             MOV     #140000,R5      ; INIT LOC PTR.
4561 045576  004737  031270             JSR     PC,KTON         ; ENABLE KT.
4562 045602  012715  164000             MOV     #DNOP,(R5)      ; REPLACE "STOPN" WITH "DNOP" IN TEST LOC.
4563 045606  004737  031306             JSR     PC,KTOFF        ; DISABLE KT.
4564 045612                             DPSTART #10$
     045612  012777  046030  135212     MOV     #10$,@DPC       ; START THE DPU.
     045620  004737  027454             JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4565 045624  000413                     BR      4$
4566
4567 045626  004737  031270     3$:     JSR     PC,KTON         ; ENABLE KT.
4568 045632  012715  164000             MOV     #DNOP,(R5)      ; REPLACE "STOPN" WITH "DNOP" IN TEST LOC.
4569 045636  004737  031306             JSR     PC,KTOFF        ; DISABLE KT.
4570 045642                             DPCONT
     045642  052777  000001  135162     BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
     045650  004737  027454             JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4571 045654  010501                     MOV     R5,R1           ; R1 = EXPECTED DPC, AND -
```

```
4572 045656  042701  170000                BIC     #170000,R1      ; -
4573 045662  062701  000005                ADD     #5,R1           ; - (R3 ALREADY = EXPECTED XRR).
4574 045666  017702  135140                MOV     @DPC,R2         ; READ DPC INTO R2.
4575 045672                                RDDSRA  XRR,R4
     045672  012777  000006  135134        MOV     #SELXRR,@DSR    ; READ XRR INTO R4 .
     045700  017704  135130                MOV     @DSR,R4
4576 045704                                IFERRX2 XMMACE,XMMAE
     045704  020102                        CMP     R1,R2           ; 1ST ITEM OK?
     045706  001002                        BNE     65$             ; NO.
     045710  020304                        CMP     R3,R4           ; 2ND ITEM OK?
     045712  001410                        BEQ     64$             ; YES.
     045714                          65$:  HRDERR  XMMACE,XMMAE,   ; NO.
     045734                          64$:
4577 045734  004737  031270                JSR     PC,KTON         ; ENABLE KT.
4578 045740  012715  173000                MOV     #STOPN,(R5)     ; REPLACE "DNOP" WITH "STOPN".
4579 045744  004737  031306                JSR     PC,KTOFF        ; DISABLE KT.
4580
4581 045750  062705  000002                ADD     #2,R5           ; ADVANCE LOC PTR.
4582 045754  020527  147774                CMP     R5,#147774      ; DONE ENOUGH LOCS?
4583 045760  001322                        BNE     3$              ; NO.
4584 045762                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4585 045764  020337  003244                CMP     R3,KTFLG        ; YES. ALL BANKS DONE?
4586 045770  001403                        BEQ     9$              ; YES.
4587 045772  062703  000200                ADD     #200,R3         ; NO. ADVANCE BANK PTR.
4588 045776  000660                        BR      2$
4589
4590 046000                          9$:   COMEND  1$
     046000  004737  027646                JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
     046004  103401                        BCS     66$
     046006  000402                        BR      67$

     046010  000137  045534          66$:  JMP     1$

     046014  004737  030072          67$:  JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
4591 046020  004737  031306                JSR     PC,KTOFF        ; DISABLE KT11.
4592 046024                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
4593
4594                                    ;
4595                                    ; DISPLAY CODE.
4596                                    ;
4597 046030  160001  000001          10$:  DJMS,AUXSEG
4598 046034  173000                        STOPN
4599
4600 046036                                END.TEST

                                    ;*****************************************************************************
                                    ;*
                                    ;*      END TEST 12
                                    ;*
                                    ;*****************************************************************************
```

```
4605
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 13   -   STOP INTERRUPT
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 40.)

4606                            ;
4607                            ; CHECK THE ACTION OF THE STOP INTERRUPT.
4608                            ; ALSO CHECKED IS THE ACTION OF THE CSR ''FORCE STOP INTR'' BIT.
4609                            ; REPEAT 40 TIMES AFTER QUICK PASS.
4610                            ;
4611 046110  004737  027702    STPTST: JSR     PC,TSTGO        ;TITLE
4612
4613 046114                     19$:    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4614
4615 046116  004737  025402            JSR     PC,DPINIT       ;INIT THE DPU SO WE CAN CHECK INTERNAL REGS.  ;@@@I
4616 046122                            INTSET  STP
     046122  012737  177601  026106            MOV     #^O177400!IOKCKIN!IOKSTP,INTMASK        ; PRIME FLAG, EXPECT STP
4617 046130  012777  046742  134674            MOV     #STP1,@DPC      ;XCT ''STOP AND INT''.
4618 046136  004737  027454    1$:     JSR     PC,WAITF
4619 046142  105737  026107            TSTB    INTFLAG         ; DID WE GET AN INTERRUPT?
4620 046146  001412                    BEQ     2$              ;BR IF INTERRUPT REC'D.
4621 046150  012701  046136            MOV     #1$,R1          ;PC AT WAIT TIME.
4622 046154                            HRDERR  NSI,INTERR      ;INT EXP'D, NOT REC'D.
4623 046174  004737  023422    2$:     JSR     PC,IIRCHK       ;CHECK INTERNALS FOR INIT'ED STATE.
4624 046200                            INTSET  STP
     046200  012737  177601  026106            MOV     #^O177400!IOKCKIN!IOKSTP,INTMASK        ; PRIME FLAG, EXPECT STP
4625 046206  012777  046746  134616            MOV     #STP2,@DPC      ;XCT ''STOP'', ENABLE STILL SET.
4626 046214  004737  027454    3$:     JSR     PC,WAITF
4627 046220  105737  026107            TSTB    INTFLAG         ;DID WE GET AN INTERRUPT?
4628 046224  001412                    BEQ     4$              ;BR IF INTERRUPT REC'D.
4629 046226  012701  046214            MOV     #3$,R1          ;SET PC AT WAIT TIME.
4630 046232                            HRDERR  NSI,INTERR      ;INT EXP'D, NOT REC'D.
4631 046252  004737  023422    4$:     JSR     PC,IIRCHK       ;CHECK INTERNALS FOR INIT'ED STATE
4632 046256                            INTSET
     046256  012737  177600  026106            MOV     #^O177400!IOKCKIN,INTMASK       ; PRIME FLAG, NO INTR. EXPECTED.
4633 046264  012777  046752  134540            MOV     #STP3,@DPC      ;XCT ''I OFF'' AND ''STOP''.
4634 046272  004737  027454    5$:     JSR     PC,WAITF
4635 046276  123727  026107  000377            CMPB    INTFLAG,#^O377  ;DID WE GET AN INTERRUPT?
4636 046304  001412                    BEQ     6$              ;BR IF NO INTERRUPT.
4637 046306  012701  046272            MOV     #5$,R1
4638 046312                            HRDERR  USI,INTERR      ;INT REC'D, NOT EXP'D.
4639 046332  004737  023422    6$:     JSR     PC,IIRCHK       ;CHECK INTERNALS FOR INIT'ED STATE
4640 046336                            INTSET
     046336  012737  177600  026106            MOV     #^O177400!IOKCKIN,INTMASK       ; PRIME FLAG, NO INTR. EXPECTED.
4641 046344  012777  046760  134460            MOV     #STP4,@DPC      ;XCT ''I ON'' AND ''STOP N''.
4642 046352  004737  027454    7$:     JSR     PC,WAITF
4643 046356  123727  026107  000377            CMPB    INTFLAG,#^O377  ;DID WE GET AN INTERRUPT?
4644 046364  001412                    BEQ     8$              ;BR IF NO INTERRUPT.
4645 046366  012701  046352            MOV     #7$,R1
4646 046372                            HRDERR  USI,INTERR      ;INT REC'D, NOT EXP'D.
4647 046412  012777  000053  134414    8$:     MOV     #FORCSI!WRTCSR,@DSR     ;SET BIT TO FORCE STOP INTR.
4648 046420                            INTSET  STP
     046420  012737  177601  026106            MOV     #^O177400!IOKCKIN!IOKSTP,INTMASK        ; PRIME FLAG, EXPECT STP
```

```
4649 046426  012777  046752  134376          MOV     #STP3,@DPC      ;XCT "I-OFF" AND STOP
4650 046434  004737  027454          9$:     JSR     PC,WAITF        ;WAIT FOR DONE
4651 046440  105737  026107                  TSTB    INTFLAG         ;DID WE GET AN INTERRUPT?
4652 046444  001412                          BEQ     10$             ;BR IF YES -- OK
4653 046446  012701  046434                  MOV     #9$,R1          ;GET CPU PC FOR ERR REPORT
4654 046452                                  HRDERR  NSI,INTERR      ;INTR EXP'D, NOT REC'D
4655 046472                          10$:    INTSET  STP
     046472  012737  177601  026106          MOV     #^O177400!IOKCKIN!IOKSTP,INTMASK        ; PRIME FLAG, EXPECT STP
4656 046500  012777  046760  134324          MOV     #STP4,@DPC      ;XCT "I-ON" AND "STOP N"
4657 046506  004737  027454          11$:    JSR     PC,WAITF        ;WAIT FOR DONE.
4658 046512  105737  026107                  TSTB    INTFLAG         ;DID WE GET AN INTERRUPT?
4659 046516  001412                          BEQ     12$             ;...BR IF YES -- OK
4660 046520  012701  046506                  MOV     #11$,R1         ;GET PC
4661 046524                                  HRDERR  NSI,INTERR      ;REPORT IF NOT
4662 046544  012777  000013  134262  12$:    MOV     #0!WRTCSR,@DSR  ;CLEAR THE CSR
4663 046552                                  SETPRI  #PRI07          ;LOCK OUT INTERRUPTS
4664 046560  105037  026106                  CLRB    INTMASK         ;DON'T EXPECT ANY INTERRUPTS.
4665 046564  012777  046742  134240          MOV     #STP1,@DPC      ;XCT "STOP AND INTR"
4666 046572  004737  027454                  JSR     PC,WAITF        ;WAIT FOR DONE
4667 046576  012737  100001  023056          MOV     #BIT15!PINTRA,GDDAT     ;EXPECT PENDING INTR
4668 046604  004737  022410                  JSR     PC,FLGCHK       ;CHECK FLAGS FOR IT.
4669 046610  012777  000012  134216          MOV     #CLFLGS,@DSR    ;CLEAR THE INTERRUPT.
4670 046616  012737  100000  023056          MOV     #100000,GDDAT
4671 046624  004737  022410                  JSR     PC,FLGCHK       ;...AND VERIFY IT.
4672 046630                                  INTSET
     046630  012737  177600  026106          MOV     #^O177400!IOKCKIN,INTMASK       ; PRIME FLAG, NO INTR. EXPECTED.
4673 046636                                  SETPRI  #PRI00          ;ALLOW INTERRUPTS
4674 046644  000240                  13$:    NOP                     ;....AND SEE IF WE GET INTR
4675 046646  000240                          NOP
4676 046650  000240                          NOP
4677 046652  000240                          NOP
4678 046654  004737  027454                  JSR     PC,WAITF
4679 046660  105737  026107                  TSTB    INTFLAG         ;DID WE GET AN INTERRUPT?
4680 046664  100412                          BMI     14$             ;OK IF NO INTR
4681 046666  012701  046644                  MOV     #13$,R1         ;SAVE CPU PC
4682 046672                                  HRDERR  USI,INTERR      ;SAY UNEXPECTED INTR
4683 046712  004737  023434          14$:    JSR     PC,IIRCNF       ;CHECK INTERNALS FOR INIT'ED STATE
4684 046716                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4685
4686 046720                                  LOOPTO  18$             ; REPEAT 'TIL LOOPER EXPIRES.
     046720  004737  027646                  JSR     PC,LOOP
     046724  103404                          BCS     18$
4687 046726  004737  030072                  JSR     PC,TSTEND
4688 046732                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
4689 046736  000137  046114          18$:    JMP     19$     ;JUMP TO START OF LOOP
4690                                 ;
4691                                 ; DISPLAY CODE
4692                                 ;
4693 046742  164002  173400          STP1:   SYNC+1,STOPI            ;STOP AND INTERRUPT.
4694 046746  164002  172000          STP2:   SYNC+1,STOP             ;AGAIN.  ENABLE SHOULD HAVE...
4695                                                                 ;...BEEN SAVED.
4696 046752  164002  171000  172000  STP3:   SYNC+1,SIOFF,STOP       ;ENABLE OFF AND STOP (NO INT).
4697 046760  164002  171400  173000  STP4:   SYNC+1,SION,STOPN       ;ENABLE ON, STOP WITHOUT INTERRUPT.
4698
4699 046766                                  END.TEST

                                    ;*******************************************************************************
```

```
                        ;*
                        ;*      END TEST 13
                        ;*
                        ;*****************************************************************************
```

```
4701
                                ;*****************************************************************************
                                ;*
                                ;*      BEGIN TEST 14   -   DPU TIME-OUT INTERRUPT
                                ;*
                                ;*****************************************************************************

                                ; (ITERATION COUNT = 100.)

4702                            ;
4703                            ; SEND THE DISPLAY TO A NON-EXISTENT MEMORY LOCATION (IF AVAILABLE)
4704                            ; AND VERIFY THAT HE RESPONDS WITH A TIME-OUT ERROR INTERRUPT.
4705                            ; REPEAT 100 TIMES AFTER QUICK PASS.
4706                            ;
4707 047050                    TOTVST: COMBEG
     047050  004737  027702             JSR     PC,TSTGO        ; TITLE.
     047054  004737  026074             JSR     PC,DPRESET      ; DO SOFT INIT.
4708 047060                             READBUS                         ;WHAT KIND OF CPU?      ;***B
4709 047062                             BNCOMPLETE 1$                   ;BR IF UNIBUS           ;***B
4710 047064  023727  002120  007577     CMP     L$HIME,#7577            ;> 124K ON LSI-11?      ;***B
4711 047072  103416                     BLO     1$                      ;BR IF NO.              ;***B
4712 047074                             RFLAGS  R0                      ;YES. GET FLAGS.        ;***B
4713 047076  032700  001000             BIT     #PNT,R0                 ;ARE WE PRINTING?       ;***B
4714 047102  001410                     BEQ     19$                     ;BR IF NO.              ;***B
4715 047104                             PRINTF  #NOTOT                  ;SAY WE CAN'T TEST.     ;***B
4716 047124                    19$:     EXIT    TST                     ;EXIT THE TEST.         ;***B
4717
4718 047130                    1$:      BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4719 047132  012777  140014  133674     MOV     #SETMRR!140000,@DSR     ;SET RELOC TO 600000, SINCE THE...
4720                                                                    ;...I/O PAGE IS REALLY 760000.
4721
4722 047140  012777  000012  133666     MOV     #CLFLGS,@DSR    ;CLEAR THE FLAGS & PENDING INTERUPTS.
4723 047146  012777  000013  133660     MOV     #0!WRTCSR,@DSR  ;CLEAR THE ERROR CODE.
4724 047154                             INTSET  ERR
     047154  012737  177604  026106     MOV     #^0177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4725 047162  012777  160000  133642     MOV     #160000,@DPC    ;XCT DISPLAY START...
4726 047170  005000                10$:  CLR     R0              ; ... SHOULD GET TIME-OUT.
4727 047172  005300                11$:  DEC     R0
4728 047174  001376                     BNE     11$
4729 047176  105737  026107             TSTB    INTFLAG         ; DID WE GET PROPER INTERRUPT?
4730 047202  001412                     BEQ     2$              ; BR IF YES
4731 047204  012701  047170             MOV     #10$,R1         ;IT DIDN'T, GET ERROR PC...
4732 047210                             HRDERR  NTO,INTERR      ;NO TIME-OUT INTERRUPT.
4733
4734 047230                    2$:
4735 047230  012737  104003  023056     MOV     #BIT15!<1*BIT11>!SELCSR,GDDAT   ;SET UP GDDAT W/ ERROR CODE.
4736 047236  004737  022440             JSR     PC,CSRCHK       ;CHECK FOR THE ERROR (NXM)
4737 047242  012737  100000  023056     MOV     #100000,GDDAT   ;SAY FLAGS SHOULD HAVE NO PENDING INTRS
4738 047250  004737  022410             JSR     PC,FLGCHK       ;...CHECK FOR IT.
4739 047254  012777  000013  133552     MOV     #0!WRTCSR,@DSR  ;CLEAR ERROR CODE
4740 047262  005037  023056             CLR     GDDAT           ;EXPECT 0'S FOR...
4741 047266  004737  022526             JSR     PC,HBCHK        ;...H-BASE
4742 047272  004737  022624             JSR     PC,CBCHK        ;...C-BASE
4743 047276  004737  022566             JSR     PC,XMCHK        ;...AND AUX MMGT.
4744 047302                             ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4745
4746 047304                             LOOPTO  1$              ; REPEAT...
```

```
        047304  004737  027646                  JSR     PC,LOOP
        047310  103707                          BCS     1$
   4747 047312  000400                          BR      5$              ;...TIL LOOPER EXPIRES.
   4748
   4749 047314  012777  000014  133512  5$:     MOV     #SETMRR!0,@DSR          ;INSURE MAIN-SEG RELOC IS 0...
   4750 047322  012777  000000  133504          MOV     #SELDSR,@DSR            ;SELECT REAL DSR
   4751 047330  004737  030072                  JSR     PC,TSTEND
   4752
   4753 047334                                  END.TEST

                                        ;*************************************************************************
                                        ;*
                                        ;*      END TEST 14
                                        ;*
                                        ;*************************************************************************

   4754 047336      045     116     045  NOTOT:  .ASCIZ  /%N%A   *** MEMORY > 124K. CANNOT TEST TIMEOUT. ***/    ;***B
   4755                                         .EVEN
```

```
4757
                                ;*****************************************************************************
                                ;*
                                ;*      BEGIN TEST 15   -   ERROR CODES
                                ;*
                                ;*****************************************************************************

                                ; (ITERATION COUNT = 10.)

4758
4759                            ;
4760                            ; GENERATE ALL ERROR CODES NOT TESTED ELSEWHERE IN THE PROGRAM
4761                            ; AND POSSIBLE TO GENERATE WITH SOFTWARE.  CHECK FOR THE RETURN
4762                            ; OF THE APPROPRIATE ERROR CODE.
4763                            ;
4764 047466                    ERCODS: COMBEG
      047466  004737  027702            JSR     PC,TSTGO        ; TITLE.
      047472  004737  026074            JSR     PC,DPRESET      ; DO SOFT INIT.
4765 047476                    1$:
4766                            ; TEST MPE - MEMORY PROTECTION (MAIN SEGMENT).
4767
4768 047476                            BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4769 047500                            INITDP
      047500  004737  025402            JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4770 047504                            WTDSRA  #170000,MPM     ; LIMIT MAIN SEG TO 4K
      047504  012777  170015  133322    MOV     #170000&^C17!SETMPM,@DSR
4771 047512                            INTSET  ERR
      047512  012737  177604  026106    MOV     #^0177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4772 047520                            DPSTART #ECMPE1         ; FORCE MPE ON MAIN SEGMENT
      047520  012777  052420  133304    MOV     #ECMPE1,@DPC    ; START THE DPU.
      047526  004737  027454            JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4773 047532  105737  026107            TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4774 047536  001412                    BEQ     10$             ; BR IF YES.
4775 047540  012701  047526            MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4776 047544                            HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4777 047564                    10$:
4778 047564  012701  114003            MOV     #MPE,R1
4779 047570                            RDDSRA  CSR,R2
      047570  012777  000003  133236    MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      047576  017702  133232            MOV     @DSR,R2
4780 047602                            IFERROR ECE,CSRER
      047602  020102                    CMP     R1,R2           ; OK?
      047604  001410                    BEQ     64$             ; YES.
      047606                            HRDERR  ECE,CSRER,      ; NO.
      047626                    64$:
4781 047626                            ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4782
4783                            ; TEST MPE - MEMORY PROTECTION (AUXILIARY SEGMENT).
4784
4785 047630                            BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4786 047632                            INITDP
      047632  004737  025402            JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4787 047636                            WTDSRA  #170000,XPM
      047636  012777  170017  133170    MOV     #170000&^C17!SETXPM,@DSR
4788 047644                            INTSET  ERR
      047644  012737  177604  026106    MOV     #^0177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4789 047652                            DPSTART #ECMPE2         ; FORCE MPE ON AUXILIARY SEGMENT.
```

```
                047652  012777  052424  133152          MOV     #ECMPE2,@DPC    ; START THE DPU.
                047660  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
          4790  047664  105737  026107                  TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
          4791  047670  001412                          BEQ     11$             ; BR IF YES.
          4792  047672  012701  047660                  MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
          4793  047676                                  HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
          4794  047716                          11$:
          4795  047716  012701  114003                  MOV     #MPE,R1
          4796  047722                                  RDDSRA  CSR,R2
                047722  012777  000003  133104          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
                047730  017702  133100                  MOV     @DSR,R2
          4797  047734                                  IFERROR ECE,CSRER
                047734  020102                          CMP     R1,R2           ; OK?
                047736  001410                          BEQ     65$             ; YES.
                047740                                  HRDERR  ECE,CSRER,      ; NO.
                047760                          65$:
          4798  047760                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
          4799  047762  005737  002502                  TST     DPUMOD          ; ARE WE IN DPU-MODE ONLY?
          4800  047766  001132                          BNE     3$              ; BR IF YES -- CAN'T DO READBACK
          4801                                                                  ; TO CHECK FOR WRITE-PROTECT VIOLATIONS.
          4802
          4803                          ; TEST MPE - MEMORY WRITE PROTECTION (MAIN SEGMENT).
          4804
          4805  047770                                  BGNSEG
          4806  047772                                  INITDP
                047772  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
          4807  047776                                  WTDSRA  #BIT9,MPM       ; SET MAIN-SEG WRITE PROTECT
                047776  012777  001015  133030          MOV     #BIT9@^C17!SETMPM,@DSR
          4808  050004                                  INTSET  ERR
                050004  012737  177604  026106          MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
          4809  050012                                  DPSTART #ECMPE3         ; FORCE MPE BY TRYING DMA READBACK
                050012  012777  052436  133012          MOV     #ECMPE3,@DPC    ; START THE DPU.
                050020  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
          4810  050024  105737  026107                  TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
          4811  050030  001412                          BEQ     12$             ; BR IF YES.
          4812  050032  012701  050020                  MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
          4813  050036                                  HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
          4814  050056                          12$:
          4815  050056  012701  114003                  MOV     #MPE,R1         ; SET EXPECTED ERROR CODE 'MPE'
          4816  050062                                  RDDSRA  CSR,R2
                050062  012777  000003  132744          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
                050070  017702  132740                  MOV     @DSR,R2
          4817  050074                                  IFERROR ECE,CSRER
                050074  020102                          CMP     R1,R2           ; OK?
                050076  001410                          BEQ     66$             ; YES.
                050100                                  HRDERR  ECE,CSRER,      ; NO.
                050120                          66$:
          4818  050120                                  ENDSEG
          4819
          4820
          4821                          ; TEST MPE - MEMORY WRITE PROTECTION (AUXILIARY SEGMENT).
          4822
          4823  050122                                  BGNSEG
          4824  050124                                  INITDP
                050124  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
          4825  050130                                  WTDSRA  #BIT9,XPM       ; SET AUX-SEG WRITE PROTECT
                050130  012777  001017  132676          MOV     #BIT9@^C17!SETXPM,@DSR
```

```
4826 050136                                    INTSET  ERR
     050136  012737  177604  026106            MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4827 050144                                    DPSTART #ECMPE4                 ; FORCE MPE BY TRYING DMA READBACK
     050144  012777  052452  132660            MOV     #ECMPE4,@DPC            ; START THE DPU.
     050152  004737  027454                    JSR     PC,WAITF                ; WAIT FOR DISPLAY STOP.
4828 050156  105737  026107                    TSTB    INTFLAG                 ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4829 050162  001412                            BEQ     13$                     ; BR IF YES.
4830 050164  012701  050152                    MOV     #.-12,R1                ; NO. SO GET PC FOR ERROR REPORT,
4831 050170                                    HRDERR  NTO,INTERR              ; ... AND REPORT THE PROBLEM.
4832 050210                                13$:
4833 050210  012701  114003                    MOV     #MPE,R1                 ; SET EXPECTED ERROR CODE 'MPE''
4834 050214                                    RDDSRA  CSR,R2
     050214  012777  000003  132612            MOV     #SELCSR,@DSR            ; READ CSR INTO R2 .
     050222  017702  132606                    MOV     @DSR,R2
4835 050226                                    IFERROR ECE,CSRER
     050226  020102                            CMP     R1,R2                   ; OK?
     050230  001410                            BEQ     67$                     ; YES.
     050232                                    HRDERR  ECE,CSRER,              ; NO.
     050252                                67$:
4836 050252                                    ENDSEG
4837
4838
4839 050254                                3$:
4840                                       ; TEST RSVDOP - RESERVED OPCODE (ILLEGAL INSTRUCTION)
4841
4842 050254                                    BGNSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4843 050256                                    INITDP
     050256  004737  025402                    JSR     PC,DPINIT               ; GO DO SOFT INIT, CHECK THE STATE.
4844 050262                                    INTSET  ERR
     050262  012737  177604  026106            MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4845 050270                                    DPSTART #ECRSV1                 ;START DPU ON ILLEGAL INSTRUCTION.
     050270  012777  052466  132534            MOV     #ECRSV1,@DPC            ; START THE DPU.
     050276  004737  027454                    JSR     PC,WAITF                ; WAIT FOR DISPLAY STOP.
4846 050302  105737  026107                    TSTB    INTFLAG                 ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4847 050306  001412                            BEQ     14$                     ; BR IF YES.
4848 050310  012701  050276                    MOV     #.-12,R1                ; NO. SO GET PC FOR ERROR REPORT,
4849 050314                                    HRDERR  NTO,INTERR              ; ... AND REPORT THE PROBLEM.
4850 050334                                14$:
4851 050334  012701  124003                    MOV     #RSVDOP,R1
4852 050340                                    RDDSRA  CSR,R2
     050340  012777  000003  132466            MOV     #SELCSR,@DSR            ; READ CSR INTO R2 .
     050346  017702  132462                    MOV     @DSR,R2
4853 050352                                    IFERROR ECE,CSRER
     050352  020102                            CMP     R1,R2                   ; OK?
     050354  001410                            BEQ     68$                     ; YES.
     050356                                    HRDERR  ECE,CSRER,              ; NO.
     050376                                68$:
4854 050376                                    ENDSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4855
4856                                       ; TEST RSVDOP - ODD Y COORDINATE (BIT 0 SET)
4857
4858 050400                                    BGNSEG                  ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4859 050402                                    INITDP
     050402  004737  025402                    JSR     PC,DPINIT               ; GO DO SOFT INIT, CHECK THE STATE.
4860 050406  012777  000113  132420            MOV     #ENECHK!WRTCSR,@DSR     ;ENABLE RSVD-OP ERROR CHECKING.
4861 050414                                    INTSET  ERR
     050414  012737  177604  026106            MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
```

```
                                                DPSTART #ECRSV2          ;START DPU ON ODD Y VALUE.
 4862 050422                                    MOV     #ECRSV2,@DPC     ; START THE DPU.
      050422  012777  052472  132402            JSR     PC,WAITF         ; WAIT FOR DISPLAY STOP.
      050430  004737  027454                    TSTB    INTFLAG          ; DID WE GET THE EXPECTED ERROR INTERRUPT?
 4863 050434  105737  026107                    BEQ     15$              ; BR IF YES.
 4864 050440  001412                            MOV     #.-12,R1         ; NO. SO GET PC FOR ERROR REPORT,
 4865 050442  012701  050430                    HRDERR  NTO,INTERR       ; ... AND REPORT THE PROBLEM.
 4866 050446
 4867 050466                                15$:
 4868 050466  012701  124103                    MOV     #RSVDOP!ENECHK,R1
 4869 050472                                    RDDSRA  CSR,R2
      050472  012777  000003  132334            MOV     #SELCSR,@DSR     ; READ CSR INTO R2 .
      050500  017702  132330                    MOV     @DSR,R2
 4870 050504                                    IFERROR ECE,CSRER
      050504  020102                            CMP     R1,R2            ; OK?
      050506  001410                            BEQ     69$              ; YES.
      050510                                    HRDERR  ECE,CSRER,       ; NO.
      050530                                69$:
 4871 050530                                    ENDSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 4872
 4873                                       ; TEST RSVDOP - ODD X VALUE (BIT 0 SET)
 4874
 4875 050532                                    BGNSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 4876 050534                                    INITDP
      050534  004737  025402                    JSR     PC,DPINIT        ; GO DO SOFT INIT, CHECK THE STATE.
 4877 050540  012777  000113  132266            MOV     #ENECHK!WRTCSR,@DSR      ;ENABLE RSVD-OP ERROR CHECKING.
 4878 050546                                    INTSET  ERR
      050546  012737  177604  026106            MOV     #^O177400!IOKCKIN!IOKERR,INTMASK         ; PRIME FLAG, EXPECT ERR
 4879 050554                                    DPSTART #ECRSV3          ;START DPU ON ODD X VALUE.
      050554  012777  052502  132250            MOV     #ECRSV3,@DPC     ; START THE DPU.
      050562  004737  027454                    JSR     PC,WAITF         ; WAIT FOR DISPLAY STOP.
 4880 050566  105737  026107                    TSTB    INTFLAG          ; DID WE GET THE EXPECTED ERROR INTERRUPT?
 4881 050572  001412                            BEQ     16$              ; BR IF YES.
 4882 050574  012701  050562                    MOV     #.-12,R1         ; NO. SO GET PC FOR ERROR REPORT,
 4883 050600                                    HRDERR  NTO,INTERR       ; ... AND REPORT THE PROBLEM.
 4884 050620                                16$:
 4885 050620  012701  124103                    MOV     #RSVDOP!ENECHK,R1
 4886 050624                                    RDDSRA  CSR,R2
      050624  012777  000003  132202            MOV     #SELCSR,@DSR     ; READ CSR INTO R2 .
      050632  017702  132176                    MOV     @DSR,R2
 4887 050636                                    IFERROR ECE,CSRER
      050636  020102                            CMP     R1,R2            ; OK?
      050640  001410                            BEQ     70$              ; YES.
      050642                                    HRDERR  ECE,CSRER,       ; NO.
      050662                                70$:
 4888 050662                                    ENDSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 4889
 4890                                       ; TEST SEQERR - SEQUENCE ERROR (DJMS FROM SUBROUTINE)
 4891
 4892 050664                                    BGNSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 4893 050666                                    INITDP
      050666  004737  025402                    JSR     PC,DPINIT        ; GO DO SOFT INIT, CHECK THE STATE.
 4894 050672                                    INTSET  ERR
      050672  012737  177604  026106            MOV     #^O177400!IOKCKIN!IOKERR,INTMASK         ; PRIME FLAG, EXPECT ERR
 4895 050700                                    DPSTART #ECSEQ1
      050700  012777  052512  132124            MOV     #ECSEQ1,@DPC     ; START THE DPU.
      050706  004737  027454                    JSR     PC,WAITF         ; WAIT FOR DISPLAY STOP.
 4896 050712  105737  026107                    TSTB    INTFLAG          ; DID WE GET THE EXPECTED ERROR INTERRUPT?
```

```
4897 050716  001412                        BEQ     17$             ; BR IF YES.
4898 050720  012701  050706                MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4899 050724                                HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4900 050744                          17$:
4901 050744  012701  120003                MOV     #SEQERR,R1
4902 050750                                RDDSRA  CSR,R2
     050750  012777  000003  132056        MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     050756  017702  132052                MOV     @DSR,R2
4903 050762                                IFERROR ECE,CSRER
     050762  020102                        CMP     R1,R2           ; OK?
     050764  001410                        BEQ     71$             ; YES.
     050766                                HRDERR  ECE,CSRER,      ; NO.
     051006                          71$:
4904 051006                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4905
4906                                ; TEST SEQERR - SEQUENCE ERROR (CHARACTER DATA IN CHAR. SUBROUTINE)
4907
4908 051010                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4909 051012                                INITDP
     051012  004737  025402                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4910 051016                                INTSET  ERR
     051016  012737  177604  026106        MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4911 051024                                DPSTART #ECSEQ2
     051024  012777  052530  132000        MOV     #ECSEQ2,@DPC    ; START THE DPU.
     051032  004737  027454                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4912 051036  105737  026107                TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4913 051042  001412                        BEQ     18$             ; BR IF YES.
4914 051044  012701  051032                MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4915 051050                                HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4916 051070                          18$:
4917 051070  012701  120003                MOV     #SEQERR,R1
4918 051074                                RDDSRA  CSR,R2
     051074  012777  000003  131732        MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     051102  017702  131726                MOV     @DSR,R2
4919 051106                                IFERROR ECE,CSRER
     051106  020102                        CMP     R1,R2           ; OK?
     051110  001410                        BEQ     72$             ; YES.
     051112                                HRDERR  ECE,CSRER,      ; NO.
     051132                          72$:
4920 051132                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4921
4922
4923                                ; TEST SEQERR - SEQUENCE ERROR (DJMS IN CHARACTER SUBROUTINE)
4924
4925 051134                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4926 051136                                INITDP
     051136  004737  025402                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4927 051142                                INTSET  ERR
     051142  012737  177604  026106        MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4928 051150                                DPSTART #ECSEQ3
     051150  012777  052576  131654        MOV     #ECSEQ3,@DPC    ; START THE DPU.
     051156  004737  027454                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4929 051162  105737  026107                TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4930 051166  001412                        BEQ     19$             ; BR IF YES.
4931 051170  012701  051156                MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4932 051174                                HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4933 051214                          19$:
```

```
4934 051214  012701  120003                    MOV     #SEQERR,R1
4935 051220                                    RDDSRA  CSR,R2
     051220  012777  000003  131606            MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     051226  017702  131602                    MOV     @DSR,R2
4936 051232                                    IFERROR ECE,CSRER
     051232  020102                            CMP     R1,R2           ; OK?
     051234  001410                            BEQ     73$             ; YES.
     051236                                    HRDERR  ECE,CSRER,      ; NO.
     051256                          73$:
4937 051256                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4938
4939
4940                                ; TEST SEQERR - SEQUENCE ERROR (BIT-MAP-0 IN SUBROUTINE)
4941
4942 051260                                    BGNSEG          ; >>>>>>>>>>>>>>>>>>>>>>>>>>>>> BEGIN SEGMENT.
4943 051262                                    INITDP
     051262  004737  025402                    JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4944 051266                                    INTSET  ERR
     051266  012737  177604  026106            MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4945 051274                                    DPSTART #ECSEQ4
     051274  012777  052630  131530            MOV     #ECSEQ4,@DPC    ; START THE DPU.
     051302  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4946 051306  105737  026107                    TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4947 051312  001412                            BEQ     20$             ; BR IF YES.
4948 051314  012701  051302                    MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4949 051320                                    HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4950 051340                          20$:
4951 051340  012701  120003                    MOV     #SEQERR,R1
4952 051344                                    RDDSRA  CSR,R2
     051344  012777  000003  131462            MOV     #SELCSR,@DSR    ; RFAD CSR INTO R2 .
     051352  017702  131456                    MOV     @DSR,R2
4953 051356                                    IFERROR ECE,CSRER
     051356  020102                            CMP     R1,R2           ; OK?
     051360  001410                            BEQ     74$             ; YES.
     051362                                    HRDERR  ECE,CSRER,      ; NO.
     051402                          74$:
4954 051402                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4955
4956
4957                                ; TEST SEQERR - SEQUENCE ERROR (BIT-MAP-1 IN SUBROUTINE)
4958
4959 051404                                    BGNSEG          ; >>>>>>>>>>>>>>>>>>>>>>>>>>>>> BEGIN SEGMENT.
4960 051406                                    INITDP
     051406  004737  025402                    JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4961 051412                                    INTSET  ERR
     051412  012737  177604  026106            MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4962 051420                                    DPSTART #ECSEQ5
     051420  012777  052654  131404            MOV     #ECSEQ5,@DPC    ; START THE DPU.
     051426  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4963 051432  105737  026107                    TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4964 051436  001412                            BEQ     21$             ; BR IF YES.
4965 051440  012701  051426                    MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4966 051444                                    HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4967 051464                          21$:
4968 051464  012701  120003                    MOV     #SEQERR,R1
4969 051470                                    RDDSRA  CSR,R2
     051470  012777  000003  131336            MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
```

```
        051476  017702  131332                  MOV     @DSR,R2
   4970 051502                                  IFERROR ECE,CSRER
        051502  020102                          CMP     R1,R2           ; OK?
        051504  001410                          BEQ     75$             ; YES.
        051506                                  HRDERR  ECE,CSRER,      ; NO.
        051526                          75$:
   4971 051526                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
   4972
   4973
   4974                                 ; TEST SEQERR - SEQUENCE ERROR (DMA READBACK FROM SUBROUTINE)
   4975
   4976 051530                                  BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
   4977 051532                                  INITDP
        051532  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
   4978 051536                                  INTSET  ERR
        051536  012737  177604  026106          MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
   4979 051544                                  DPSTART #ECSEQ6
        051544  012777  052676  131260          MOV     #ECSEQ6,@DPC    ; START THE DPU.
        051552  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
   4980 051556  105737  026107                  TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
   4981 051562  001412                          BEQ     22$             ; BR IF YES.
   4982 051564  012701  051552                  MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
   4983 051570                                  HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
   4984 051610                          22$:
   4985 051610  012701  120003                  MOV     #SEQERR,R1
   4986 051614                                  RDDSRA  CSR,R2
        051614  012777  000003  131212          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
        051622  017702  131206                  MOV     @DSR,R2
   4987 051626                                  IFERROR ECE,CSRER
        051626  020102                          CMP     R1,R2           ; OK?
        051630  001410                          BEQ     76$             ; YES.
        051632                                  HRDERR  ECE,CSRER,      ; NO.
        051652                          76$:
   4988 051652                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
   4989
   4990
   4991                                 ; TEST SEQERR - SEQUENCE ERROR (ILLEGAL DMA READBACK RESTART)
   4992
   4993 051654                                  BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
   4994 051656                                  INITDP
        051656  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
   4995 051662                                  INTSET  ERR
        051662  012737  177604  026106          MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
   4996 051670                                  DPSTART #ECSEQ7
        051670  012777  052726  131134          MOV     #ECSEQ7,@DPC    ; START THE DPU.
        051676  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
   4997 051702  105737  026107                  TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
   4998 051706  001412                          BEQ     23$             ; BR IF YES.
   4999 051710  012701  051676                  MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
   5000 051714                                  HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
   5001 051734                          23$:
   5002 051734  012701  120003                  MOV     #SEQERR,R1
   5003 051740                                  RDDSRA  CSR,R2
        051740  012777  000003  131066          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
        051746  017702  131062                  MOV     @DSR,R2
   5004 051752                                  IFERROR ECE,CSRER
        051752  020102                          CMP     R1,R2           ; OK?
```

```
        051754  001410                          BEQ     77$             ; YES.
        051756                                  HRDERR  ECE,CSRER,      ; NO.
        051776                          77$:
5005    051776                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5006
5007                                    ; TEST SYNCTO - SYNC TIMEOUT FROM IMAGE MEMORY.
5008
5009    052000                                  BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5010    052002                                  INITDP
        052002  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
5011    052006                                  INTSET  ERR
        052006  012737  177604  026106          MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
5012    052014                                  DPSTART #ECSTO
        052014  012777  052754  131010          MOV     #ECSTO,@DPC     ; START THE DPU.
        052022  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
5013    052026  105737  026107                  TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
5014    052032  001412                          BEQ     24$             ; BR IF YES.
5015    052034  012701  052022                  MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
5016    052040                                  HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
5017    052060                          24$:
5018    052060  012701  140003                  MOV     #SYNCTO,R1
5019    052064                                  RDDSRA  CSR,R2
        052064  012777  000003  130742          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
        052072  017702  130736                  MOV     @DSR,R2
5020    052076                                  IFERROR ECE,CSRER
        052076  020102                          CMP     R1,R2           ; OK?
        052100  001410                          BEQ     78$             ; YES.
        052102                                  HRDERR  ECE,CSRER,      ; NO.
        052122                          78$:
5021    052122                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5022
5023                                    ; TEST DRDYTO - DATA READY TIMEOUT ON CURSOR READBACK
5024
5025    052124                                  BGNSEG
5026    052126                                  INITDP
        052126  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
5027    052132  105037  026106                  CLRB    INTMASK
5028    052136  012777  040000  130672          MOV     #GTJSSW,@DXR    ; SAY READ PENDING COORDS (BUT THERE ARE NONE)
5029    052144  004737  027454                  JSR     PC,WAITF        ; WAIT FOR READY
5030    052150  012701  150003                  MOV     #DRDYTO,R1      ; SET EXPECTED ERROR CODE 'DRDYTO' INTO R1
5031    052154                                  RDDSRA  CSR,R2          ; GET THE CSR
        052154  012777  000003  130652          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
        052162  017702  130646                  MOV     @DSR,R2
5032    052166                                  IFERROR ECE,CSRER
        052166  020102                          CMP     R1,R2           ; OK?
        052170  001410                          BEQ     79$             ; YES.
        052172                                  HRDERR  ECE,CSRER,      ; NO.
        052212                          79$:
5033    052212                                  ENDSEG
5034
5035                                    ; TEST DAVTO - DATA AVAILABLE TIMEOUT PIXEL READBACK.
5036
5037    052214                                  BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5038    052216                                  INITDP
        052216  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
5039    052222  012777  040000  130610          MOV     #40000,@DYR     ; PROTECT ALL MEMORYS.
5040    052230  004737  027454                  JSR     PC,WAITF        ;
```

```
 5041 052234  012777  040400  130576          MOV     #40400,@DYR     ;
 5042 052242  004737  027454                  JSR     PC,WAITF        ;
 5043 052246  012777  041000  130564          MOV     #41000,@DYR     ;
 5044 052254  004737  027454                  JSR     PC,WAITF        ;
 5045 052260  012777  041400  130552          MOV     #41400,@DYR     ;
 5046 052266  004737  027454                  JSR     PC,WAITF        ;
 5047 052272  012777  000000  130536          MOV     #0,@DXR         ; TRY PIXEL READ.
 5048 052300  004737  027454                  JSR     PC,WAITF        ;
 5049 052304  103410                          BCS     2$              ; BRANCH IF UNSUCCESSFUL.
 5050 052306                                  DFERR   PRBH            ; PIXEL READBACK HUNG UP.
 5051 052326  012701  144003          2$:     MOV     #DAVTO,R1
 5052 052332                                  RDDSRA  CSR,R2
      052332  012777  000003  130474          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      052340  017702  130470                  MOV     @DSR,R2
 5053 052344                                  IFERROR ECE,CSRER
      052344  020102                          CMP     R1,R2           ; OK?
      052346  001410                          BEQ     80$             ; YES.
      052350                                  HRDERR  ECE,CSRER,      ; NO.
      052370                          80$:
 5054 052370                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 5055
 5056 052372                                  COMEND  1$
      052372  004737  027646                  JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
      052376  103401                          BCS     81$
      052400  000402                          BR      82$

      052402  000137  047476          81$:    JMP     1$

      052406  004737  030072          82$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
 5057 052412                                  EXIT    TST     ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
 5058
 5059                                 ;
 5060                                 ; INTERUPT SERVICE FOR THIS TEST.
 5061                                 ;
 5062 052416  000002                  ECISVC: RTI             ; RETURN FROM INTERUPT.
 5063
 5064                                 ;
 5065                                 ; DISPLAY FILES.
 5066                                 ;
 5067 052420  164000                  ECMPE1: DNOP                    ; MPE - MEMORY PROTECTION ERROR.
 5068 052422  173000                          STOPN                   ; ... ON MAIN SEGMENT
 5069
 5070 052424  160001  052433          ECMPE2: DJMS,1$!AUXSEG          ; CALL AUXILIARY SEGMENT
 5071 052430  173000                          STOPN
 5072 052432  164000                  1$:     DNOP                    ;SHOULD GET MPE ERROR
 5073 052434  173000                          STOPN
 5074
 5075 052436  162000                  ECMPE3: IMREAD                  ;TRY DMA READBACK
 5076 052440  000001  000001  000001          1,1,1,160000            ;...TO WRITE PROTECTED MAIN SEG.
 5077 052450  173000                          STOPN
 5078
 5079 052452  162000                  ECMPE4: IMREAD                  ;TRY DMA READBACK ...
 5080 052454  000001  000001  000001          1,1,1,160000!AUXSEG     ;...TO WRITE-PROTECTED AUX SEG.
 5081 052464  173000                          STOPN
 5082
 5083 052466  161777                  ECRSV1: 161777                  ; RSVDOP - RESERVED OPCODE, OPERATION.
 5084 052470  173000                          STOPN
```

```
5085
5086 052472  114000  000000  000001  ECRSV2: APNT,0,1                ; RSVDOP - ODD Y VALUE.
5087 052500  173000                          STOPN
5088
5089 052502  114000  000001  000000  ECRSV3: APNT,1,0                ; RSVDOP - ODD X VALUE.
5090 052510  173000                          STOPN
5091
5092 052512  160001  052520          ECSEQ1: DJMS,1$                 ; SEQERR - SEQUENCE ERROR.
5093 052516  173000                          STOPN
5094 052520  160001  052524          1$:     DJMS,2$                 ;DJMS FROM SUBROUTINE.
5095 052524  173000                  2$:     STOPN
5096 052526  173000                          STOPN
5097
5098 052530  114000  000000  000000  ECSEQ2: APNT,0,0                ; SEQERR.
5099 052536  152000  052550                  SETCB,2$
5100 052542  103774                          CHAR!ALL
5101 052544  000000                  1$:     0
5102 052546  173000                          STOPN
5103 052550  000000                  2$:     0                       ;DATA AS 1ST WORD -- STILL IN CHAR MODE.
5104 052552  000000  000000  000000          0,0,0,0,0,0,0,0
5105 052572  165000                          DPOP
5106 052574  173000                          STOPN
5107
5108 052576  114000  000000  000000  ECSEQ3: APNT,0,0                ; SEQERR.
5109 052604  152000  052616                  SETCB,2$
5110 052610  103774                          CHAR!ALL
5111 052612  000000                  1$:     0
5112 052614  173000                          STOPN
5113 052616  160001  052624          2$:     DJMS,3$                 ;DJMS IN CHAR SUBROUTINE.
5114 052622  173000                          STOPN
5115 052624  173000                  3$:     STOPN
5116 052626  173000                          STOPN
5117
5118 052630  170140          ECSEQ4: CLRMEM                          ; SEQERR.
5119 052632  114000  000000  000000          APNT,0,0
5120 052640  160001  052646                  DJMS,1$
5121 052644  173000                          STOPN
5122 052646  134000                  1$:     BM04
5123 052650  052650                  2$:     2$
5124 052652  173000                          STOPN
5125
5126 052654  114000  000000  000000  ECSEQ5: APNT,0,0                ; SEQERR.
5127 052662  160001  052670                  DJMS,1$
5128 052666  173000                          STOPN
5129 052670  136000                  1$:     BM14
5130 052672  052672                  2$:     2$
5131 052674  173000                          STOPN
5132
5133 052676  114000  000000  000000  ECSEQ6: APNT,0,0                ; SEQERR.
5134 052704  160001  052712                  DJMS,1$
5135 052710  173000                          STOPN
5136 052712  162000                  1$:     IMREAD                  ;DMA READBACK FROM SUBROUTINE.
5137 052714  000001                          1
5138 052716  000001                          1
5139 052720  000001                          1
5140 052722  052750                          ECSQ7B
5141 052724  173000                          STOPN
```

```
5142
5143 052726  114000  000000  000000  ECSEQ7: APNT,0,0
5144 052734  162001                          IMREAD!1                        ;ILLEGAL DMA READBACK RESTART.
5145 052736  000001                          1
5146 052740  052750                          ECSQ7B
5147 052742  173000                          STOPN
5148 052744  162101                          IMREAD!101
5149 052746  173000                          STOPN
5150 052750                          ECSQ7B: .BLKW   2
5151
5152 052754  176000                ECSTO:    PROTEC!CH0
5153 052756  176400                          PROTEC!CH1
5154 052760  177000                          PROTEC!CH2
5155 052762  177400                          PROTEC!CH3
5156 052764  114000  040000  000000          APNT,I!0,0
5157 052772  173000                          STOPN
5158
5159 052774                                  END.TEST

                                    ;*****************************************************************************
                                    ;*
                                    ;*      END TEST 15
                                    ;*
                                    ;*****************************************************************************
```

```
5161                                    ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 16   -   ADDRESS RELOCATE
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 20.)

5162                                    ;
5163                                    ; TEST ALL COMBINATIONS OF DPC AND RELOCATE FACTOR
5164                                    ; WHERE DPC + REL = RELT1, A LOCATION WITHIN THE PROGRAM
5165                                    ; TEST RELIES OF THE FACT THAT (DPC + REL) > 18 BITS WILL
5166                                    ; WRAP AROUND IN THE FINAL DPC.
5167                                    ; REPEAT 20 TIMES AFTER QUICK PASS.
5168                                    ;
5169 053050                            RELTST: COMBEG
      053050  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      053054  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
5170 053060  012737  053274  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5171 053066  012777  000000  127740            MOV     #SELDSR,@DSR
5172 053074  017703  127734                    MOV     @DSR,R3
5173 053100  042703  176000                    BIC     #^C1777,R3
5174 053104  052703  172000                    BIS     #STOP,R3        ;...AND STATUS.
5175
5176 053110  012704  053322            5$:     MOV     #RELT1,R4       ;SET INITIAL DPC
5177 053114  012702  053324                    MOV     #RELT1+2,R2     ;SET INITIAL EXPECTED DPC
5178 053120  012705  000014                    MOV     #SETMRR!0,R5    ;SET INITIAL RELOCATION (0)
5179
5180 053124                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5181
5182 053126  010577  127702                    MOV     R5,@DRR         ;SET 1ST (NEXT) REL VALUE...
5183 053132  010477  127674                    MOV     R4,@DPC         ;...XCT -- RELOCATE TO 'RELT1''
5184 053136  004737  027454                    JSR     PC,WAITF
5185 053142  027702  127664                    CMP     @DPC,R2         ;CHECK DPC = R4 (INITIAL) + 2
5186 053146  001003                            BNE     2$
5187 053150  027703  127660                    CMP     @DSR,R3         ;...AND DSR = STOP+PIXEL DATA.
5188 053154  001410                            BEQ     3$
5189 053156                            2$:     HRDERR  RELF,PCSERR     ;RELOCATE FAILURE.
5190 053176  004737  026756            3$:     JSR     PC,RELEAS       ;IN CASE HE 'HUNG' SOMEWHERE.
5191 053202                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5192
5193 053204  020527  177774                    CMP     R5,#SETMRR!177760       ;MAX RELOCATION DONE?
5194 053210  001414                            BEQ     4$              ;ALL DONE IF SO.
5195
5196 053212  062705  000020                    ADD     #BIT4,R5        ;BUMP RELOCATION BY 1 (32 WORDS)
5197 053216  162702  000100                    SUB     #100,R2         ;REDUCE EXPECTED...
5198 053222  162704  000100                    SUB     #100,R4         ;... AND INITIAL DPC BY 32 WORDS)
5199 053226  100336                            BPL     1$              ;CONTINUE 'TIL WE UNDERFLOW
5200 053230  005705                            TST     R5              ;SEE IF ALREADY FORCED HIGH BITS
5201 053232  100734                            BMI     1$              ;JUST CONTINUE IF YES
5202 053234  052705  140000                    BIS     #140000,R5      ;FORCE HIGH RELOCATION BITS
5203 053240  000731                            BR      1$              ;...THEN CONTINUE
5204
5205 053242                            4$:     LOOPTO  5$              ; REPEAT 'TIL LOOPER EXPIRES.
      053242  004737  027646                    JSR     PC,LOOP
      053246  103720                            BCS     5$
```

```
 5206 053250  012777  000014  127556          MOV     #SETMRR!0,@DSR  ;INSURE RR IS 0
 5207 053256  012777  000000  127550          MOV     #SELDSR,@DSR    ;...AND DSR IS SELECTED
 5208 053264  004737  030072                  JSR     PC,TSTEND
 5209 053270                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
 5210                                 ;
 5211                                 ; ON ERROR, SHOW RELOCATE PARAMETERS.
 5212                                 ;
 5213 053274                         20$:    PRINTX  #RELFX,R4,R5    ; EXTENSION SHOWS RELOC FACTORS.
 5214 053320  000207                         RTS     PC
 5215
 5216                                 ;DISPLAY FILE FOR RELOCATE TEST
 5217 053322  173000                 RELT1:  STOPN
 5218 053324  173000                         STOPN
 5219 053326                                 END.TEST

                                     ;*****************************************************************************
                                     ;*
                                     ;*      END TEST 16
                                     ;*
                                     ;*****************************************************************************
```

```
5221
                                ;*****************************************************************************
                                ;*
                                ;*      BEGIN TEST 17   -   CHARACTERS
                                ;*
                                ;*****************************************************************************

                                ; (ITERATION COUNT = 20.)

5222                            ;
5223                            ; USE SETCB, CHAR, AND DPOP AS A SUBROUTINE CALL AND RETURN.
5224                            ; USE ALL ASCII CODES (HI AND LO BYTES) ALTERING THE BASE
5225                            ; ADDRESS SUCH THAT EACH CHAR/BASE COMBINATION YIELDS THE
5226                            ; SAME SUBROUTINE ADDRESS.
5227                            ; VERIFY THE DPC AT EACH SUBROUTINE ENTRY AND AGAIN AFTER
5228                            ; FINAL DPOP.
5229                            ; DO THE TEST FOR BOTH MAIN AND AUXILIARY SEGMENT SPECIFIED IN C-BASE.
5230                            ; REPEAT 20 TIMES AFTER A QUICK PASS.
5231                            ;
5232                            ; NOTE THAT AN ERROR WHERE CHAR CODE = -1 IMPLIES THAT
5233                            ; CHAR MODE DIDN'T TERMINATE WITH THE FINAL DPOP.
5234                            ; IN EVERY OTHER CASE, EXP DPC = SETCB + 2(CC) + 4.
5235                            ;
5236                            ; ALSO, ANY ERROR IMPLIES THAT WE'VE LOST CONTROL AND
5237                            ; CAN'T CONTINUE.
5238                            ;
5239                            ; CHAR ARRAY SIZE = 67. WORDS (OPCODE + 128 CHAR + DJMP,RETURN).
5240                            ;
5241 053374                    CHRTST: COMBEG
      053374  004737  027702            JSR     PC,TSTGO        ; TITLE.
      053400  004737  026074            JSR     PC,DPRESET      ; DO SOFT INIT.
5242
5243 053404  012737  054110  054066      MOV     #CT.SUB,CHSEG   ;GET INITIAL BASE ADDRESS, MAIN SEGMENT.
5244 053412  012737  054036  022300      MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5245 053420  013701  003240            MOV     FREE,R1         ; BUFFER ADDRESS => R1.
5246 053424  010137  054104            MOV     R1,CT.ASC       ; SET "DJMP FREE" CODE.
5247 053430  005002                    CLR     R2              ;1ST CHAR CODE => R2
5248 053432  012721  100000            MOV     #CHAR,(R1)+     ;SET CHAR OPCODE...
5249 053436  110221            6$:     MOVB    R2,(R1)+        ;...AND 128 CHARACTERS...
5250 053440  005202                     INC     R2
5251 053442  020227  000200               CMP     R2,#128.
5252 053446  001373                       BNE     6$
5253 053450  012721  160000            MOV     #DJMP,(R1)+     ;...AND THE RETURN LINK.
5254 053454  012721  054106            MOV     #CT.RET,(R1)+
5255
5256 053460  017703  127350            MOV     @DSR,R3
5257 053464  042703  176000            BIC     #^C1777,R3
5258 053470  052703  172000            BIS     #STOP,R3        ;SET EXP DSR FOR ERROR.
5259
5260 053474                    5$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5261 053476  013737  054066  054100      MOV     CHSEG,CT.ASC-4  ;INIT INITIAL BASE ADDRESS.
5262 053504  013737  054066  054116      MOV     CHSEG,CT.BAS    ;INIT BASE ADDRESS IN CHAR SUBTRN.
5263 053512  005001                    CLR     R1              ;R1 = CURRENT CHAR.
5264
5265 053514  013702  054066            MOV     CHSEG,R2        ;GET SEGMENT SELECTOR
5266 053520  042702  177776            BIC     #^CAUXSEG,R2    ;...FOR DPC READBACK
5267 053524  052702  054114            BIS     #CT.SUB+4,R2    ;...AND SET EXPECTED DPC.
```

```
5268 053530  012777  054070  127274          MOV     #CT,@DPC        ;START, XCT 1ST CHAR.
5269 053536  000403                          SKP3                    ;SKIP OVER RESUME
5270 053540  012777  000001  127264  1$:     MOV     #1,@DPC         ;RESUME, XCT NEXT CHAR.
5271 053546  004737  027454                  JSR     PC,WAITF
5272 053552  027702  127254                  CMP     @DPC,R2         ;CHECK DPC = R2
5273 053556  001410                          BEQ     2$              ;ERROR IF IT'S WRONG.
5274 053560                                  HRDERR  CHRF,PCSERR     ;CHAR MODE FAILS.
5275 053600  013746  003240          2$:     MOV     FREE,-(SP)      ;GET START OF MAIN DISPLAY FILE,
5276 053604  005701                          TST     R1                      ;FINAL DPOP?
5277 053606  100003                          BPL     15$             ;SKIP AHEAD IF NOT.     @@@I
5278 053610  062716  000203                  ADD     #203,(SP)       ;SAY PCSAVE IS EMPTY, WITH FINAL ADDRESS.
5279 053614  000405                          BR      16$             ;...THEN GO CHECK IT.   @@@I
5280 053616  062716  000004          15$:    ADD     #4,(SP)         ;  ADD OFFSET FOR FIRST TRIP,
5281 053622  060116                          ADD     R1,(SP)         ;  AND THE CHARACTER,
5282 053624  042716  000001                  BIC     #1,(SP)         ;MAKE EVEN, 2 CHARS/WORD
5283 053630  012637  023056          16$:    MOV     (SP)+,GDDAT     ;EXPECTED DATA FOR PCSAVE
5284 053634  004737  022360                  JSR     PC,PCSCHK       ;...CHECK PCSAVE.
5285 053640  012737  100000  023056          MOV     #CHAR,GDDAT     ;NOW CHECK FOR "CHAR" OPCODE IN FLAGS
5286 053646  004737  022410                  JSR     PC,FLGCHK       ;
5287 053652  012737  000003  023056          MOV     #SELCSR,GDDAT   ;... AND CSR
5288 053660  004737  022440                  JSR     PC,CSRCHK       ;
5289 053664  013737  054116  023056          MOV     CT.BAS,GDDAT    ;CHECK THE CHARACTER BASE
5290 053672  004737  022624                  JSR     PC,CBCHK        ;
5291 053676                                  CKLOOP                  ;LOOP ON ERROR.
5292
5293 053700  005701                          TST     R1              ;HAVE WE TERMINATED ??
5294 053702  100414                          BMI     3$              ;EXIT IF SO.
5295 053704  162737  000002  054116          SUB     #2,CT.BAS       ;NO, DECR BASE ADDRESS.
5296 053712  005201                          INC     R1              ;INCR CHAR COUNT.
5297 053714  020127  000200                  CMP     R1,#128.        ;128 CHARS DONE ??
5298 053720  001307                          BNE     1$              ;LOOP IF NOT.
5299 053722  012701  177777                  MOV     #-1,R1          ;YES, SET TERMINATE FLAG...
5300 053726  012702  054110                  MOV     #CT.RET+2,R2    ;...ADJUST EXPECTED DPC...
5301 053732  000702                          BR      1$              ;...AND LOOP TO XCT FINAL DPOP.
5302
5303 053734                          3$:     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5304 053736  032737  000001  054066          BIT     #AUXSEG,CHSEG           ;DID WE DO AUX SEGMENT?
5305 053744  001004                          BNE     31$             ;BR IF YES.     @@@I
5306 053746  052737  000001  054066          BIS     #AUXSEG,CHSEG   ;NO. SO DO IT NOW
5307 053754  000647                          BR      5$              ;LOOP BACK AFTER SETTING SEG BIT.       @@@I
5308 053756  042737  000001  054066  31$:    BIC     #AUXSEG,CHSEG   ;DID AUX, SO CLEAR THE BIT
5309 053764                                  LOOPTO  5$              ; REPEAT TIL LOOPER EXPIRES.
     053764  004737  027646                  JSR     PC,LOOP
     053770  103641                          BCS     5$
5310 053772  005737  002502                  TST     DPUMOD          ; ARE WE IN DPU-MODE ONLY?
5311 053776  001013                          BNE     40$             ; BR=YES.
5312 054000  004737  026074                  JSR     PC,DPRESET      ; DO SOFT INIT. & CHECK STATE
5313 054004                                  DPSTART #CHDISP
     054004  012777  054122  127020          MOV     #CHDISP,@DPC    ; START THE DPU.
     054012  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
5314 054016  004737  027514                  JSR     PC,PAUSE1
5315 054022  004737  027514                  JSR     PC,PAUSE1
5316 054026                          40$:
5317 054026  004737  030072                  JSR     PC,TSTEND
5318 054032                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
5319                                 ;
5320                                 ; ON ERROR, SHOW BASE ADDRESS AND CURRENT CHAR CODE.
```

```
                                        ;
054036                          20$:    PRINTX  #CHRFX,CT.BAS,R1 ; ERROR EXTENSION.
054064  000207                          RTS     PC
                                        ;
054066  054110                  CHSEG:  CT.SUB  ;CHARACTER BASE CONTROL, WITH SEGMENT SELECT.
                                        ;
                                        ; DISPLAY CODE FOR CHAR MODE TEST.
                                        ;
054070  164000  164000  164000  CT:     DNOP,DNOP,DNOP
054076  152000  054110                  SETCB,CT.SUB                ;INITIAL BASE ADDR = CT.SUB.
054102  160000                          DJMP                        ;XFER PC TO THE CHAR...
054104  000000                  CT.ASC: 0                           ;...DATA ARRAY.
054106  173000                  CT.RET: STOPN                       ;FINAL "DPOP" SHOULD RETURN HERE
                                        
054110  054112                  CT.SUB: .+2                         ;SUBROUTINE ENTRY PONNTER.
054112  173000                          STOPN                       ;STOP HERE IF CHAR ADDRESSES...
                                                                    ;...COME OUT RIGHT.
054114  152000                          SETCB                       ;ON RESUME, SET NEW BASE ADDR...
054116  054110                  CT.BAS: CT.SUB
054120  165000                          DPOP                        ;...AND DPOP TO NEXT CHAR.
                                        
                                        ; DISPLAY FILE FOR CHARACTER DISPLAY:
                                        
054122  176074  176474  177074  CHDISP: RDWRT!CH0,RDWRT!CH1,RDWRT!CH2,RDWRT!CH3
                                ;***B   CLRMEM
054132  116000                          APNT!LO                     ;***B
054134  170100                          SETMEM                      ;***B
054136  152000  110302                  SETCB, ACAT
054142  117774  001000  001500          APNT!ALL,1000,1500
054150  100000                          CHAR
054152     040     041     042          .ASCII  ? !''#$%&'()**+,-./?
                                        .EVEN
054172  114000  001000  001400          APNT,1000,1400
054200  100000                          CHAR
054202     060     061     062          .ASCII  /0123456789/
054214  114000  001000  001300          APNT,1000,1300
054222  100000                          CHAR
054224     072     073     074          .ASCII  /:;<=>?@ /
                                        .EVEN
054234  114000  001000  001200          APNT,1000,1200
054242  100000                          CHAR
054244     101     102     103          .ASCII  /ABCDEFGHIJKLM /
                                        .EVEN
054262  114000  001000  001100          APNT,1000,1100
054270  100000                          CHAR
054272     116     117     120          .ASCII  /NOPQRSTUVWXYZ /
                                        .EVEN
054310  114000  001000  001000          APNT,1000,1000
054316  100000                          CHAR
054320     133     134     135          .ASCII  /[\]^_ /
                                        .EVEN
                                        
054326  117774  001000  000602          APNT!ALL,1000,602
054334  100000                          CHAR
054336     040     041     042          .ASCII  ? !''#$%&'()**+,-./?
                                        .EVEN
054356  114000  001000  000502          APNT,1000,502
```

```
 5378 054364  100000                       CHAR
 5379 054366     060      061      062      .ASCII  /0123456789/
 5380 054400  114000  001000  000402       APNT,1000,402
 5381 054406  100000                       CHAR
 5382 054410     072      073      074      .ASCII  /:;<=>?@ /
 5383                                       .EVEN
 5384 054420  114000  001000  000302       APNT,1000,302
 5385 054426  100000                       CHAR
 5386 054430     101      102      103      .ASCII  /ABCDEFGHIJKLM /
 5387                                       .EVEN
 5388 054446  114000  001000  000202       APNT,1000,202
 5389 054454  100000                       CHAR
 5390 054456     116      117      120      .ASCII  /NOPQRSTUVWXYZ /
 5391                                       .EVEN
 5392 054474  114000  001000  000102       APNT,1000,102
 5393 054502  100000                       CHAR
 5394 054504     133      134      135      .ASCII  /[\]^_ /
 5395                                       .EVEN
 5396 054512  173000                       STOPN
 5397
 5398 054514                                END.TEST

;*****************************************************************************
;*
;*      END TEST 17
;*
;*****************************************************************************
```

```
5400
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 18   -   ABSOLUTE POINTS
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 10.)

5401                                    ;
5402                                    ; VERIFY THAT WE CAN EXECUTE ABSOLUTE POINTS FOR THE FULL
5403                                    ; RANGE OF X/Y.
5404                                    ;
5405                                    ; PLOT ALL X POINTS, HOLDING Y AT 0.
5406                                    ; PLOT ALL Y POINTS, HOLDING X AT 0.
5407                                    ; PLOT ALL POINTS ON THE 45 (X = Y).
5408                                    ;
5409                                    ; REPEAT EACH POINT 10 TIMES AFTER QUICK PASS.
5410                                    ;
5411 054566  004737  027702            APTST:  JSR     PC,TSTGO        ;TITLE
5412 054572  012737  055076  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5413 054600  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT (IF THERE IS ONE).
5414
5415 054604  005737  002502                    TST     DPUMOD
5416 054610  001002                            BNE     .+6             ; SKIP NEXT IF DPU-ONLY.
5417 054612  004737  026650                    JSR     PC,CLRRW        ; CLEAR AND SET R/W MODE.
5418
5419 054616  012737  177777  003232            MOV     #-1,TEMP1       ;USE AS XFER SWITCH.
5420 054624  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF THE LOOPER.
5421 054632  005004                    8$:     CLR     R4              ;EXPECTED X ADDRESS.
5422 054634  005005                            CLR     R5              ;EXPECTED Y ADDRESS.
5423 054636  005037  055146                    CLR     ABPX            ;INIT X...
5424 054642  053737  003064  055146            BIS     OPTI,ABPX       ;...INTENSIFY IS OPTIONAL...
5425 054650  005037  055150                    CLR     ABPY            ;...AND Y AT 0.
5426
5427 054654                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5428
5429 054656  012777  055140  126146    9$:     MOV     #ABP,@DPC       ;START DISPLAY...
5430 054664  004737  027454                    JSR     PC,WAITF        ;WAIT FOR DONE FLAG.
5431 054670  027704  126142                    CMP     @DXR,R4         ;CHECK DXR = R4...
5432 054674  001003                            BNE     2$
5433 054676  027705  126136                    CMP     @DYR,R5         ;...AND DYR = R5.
5434 054702  001410                            BEQ     3$
5435 054704                            2$:     HRDERR  APF,XYERR       ;ABS POINT FAILS.
5436 054724                            3$:     LOOPTO  9$              ; REPEAT...TIL LOOPER EXPIRES.
     054724  004737  027646                    JSR     PC,LOOP
     054730  103752                            BCS     9$
5437 054732                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5438
5439 054734  013737  003234  027700            MOV     TEMP2,LOOPK     ;RESET LOOPER.
5440 054742  005737  003232                    TST     TEMP1           ;TEST XFER SWITCH.
5441 054746  003027                            BGT     5$              ;INCR BOTH X AND Y.
5442 054750  001413                            BEQ     4$              ;INCR Y ONLY.
5443
5444 054752  062704  000002                    ADD     #2,R4           ;INCR X ONLY.
5445 054756  062737  000002  055146            ADD     #2,ABPX
5446 054764  020427  001776                    CMP     R4,#MAXX
```

```
5447 054770  101731                    BLOS    1$              ;...AND LOOP 'TIL DONE.
5448 054772  005237  003232            INC     TEMP1           ;THEN, BUMP SWITCH...
5449 054776  000715                    BR      8$              ;...AND REINIT.
5450
5451 055000  062705  000002      4$:   ADD     #2,R5           ;INCR Y ONLY.
5452 055004  062737  000002  055150  ADD     #2,ABPY
5453 055012  020527  001776          CMP     R5,#MAXY
5454 055016  101716                    BLOS    1$              ;...AND LOOP 'TIL DONE.
5455 055020  005237  003232            INC     TEMP1           ;THEN, BUMP SWITCH...
5456 055024  000702                    BR      8$              ;...AND REINIT.
5457
5458 055026  012700  000002      5$:   MOV     #2,R0           ;INCR X AND Y.
5459 055032  060004                    ADD     R0,R4           ;INCR X'S...
5460 055034  060037  055146            ADD     R0,ABPX
5461 055040  060005                    ADD     R0,R5           ;...AND Y'S...
5462 055042  060037  055150            ADD     R0,ABPY
5463 055046  020527  001776            CMP     R5,#MAXY
5464 055052  101700                    BLOS    1$              ;...AND LOOP 'TIL DONE.
5465
5466 055054  005737  002502      10$:  TST     DPUMOD
5467 055060  001002                    BNE     11$
5468 055062  004737  027514            JSR     PC,PAUSE1       ; PAUSE IF VIDEO ON.
5469 055066  004737  030072      11$:  JSR     PC,TSTEND
5470 055072                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
5471                                ;
5472                                ; ON ERROR, SHOW ORIGIN AND DELTAS.
5473                                ;
5474 055076                    20$:    PRINTX  #LONGX,ABP+2,ABP+4,ABPX,ABPY ; EXTENSION.
5475 055136  000207                    RTS     PC
5476                                ;
5477                                ; DISPLAY FILE FOR APNT TEST.
5478                                ;
5479 055140  117774            ABP:    APNT!ALL
5480 055142  000000  000000            0,0
5481 055146  000000            ABPX:   0
5482 055150  000000            ABPY:   0
5483 055152  173000                    STOPN
5484
5485 055154                            END.TEST

;*****************************************************************************
;*
;*      END TEST 18
;*
;*****************************************************************************
```

```
5487
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 19   -   LONG VECTORS
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 10.)

5488                                    ;
5489                                    ; GENERATE LONG VECTORS FROM 0,0 VARYING THE LENGTH FROM
5490                                    ; MAX(X) TO HALF(XY) TO MAX(Y).
5491                                    ; THEN REPEAT FROM MAX,MAX USING NEGATIVE DELTAS.
5492                                    ;
5493                                    ;       [THIS MAY BE WORST CASE !!!]
5494                                    ;
5495                                    ; REPEAT EACH VECTOR 10 TIMES AFTER QUICK PASS.
5496                                    ;
5497 055224  004737  027702            LVTST:  JSR     PC,TSTGO        ;TITLE
5498 055230  012737  055544  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5499 055236  004737  027054                    JSR     PC,LUMIN        ; BLAST THE LUT.
5500
5501 055242  005737  002502                    TST     DPUMOD
5502 055246  001002                            BNE     .+6             ; SKIP NEXT IF DPU-ONLY.
5503 055250  004737  026650                    JSR     PC,CLRRW        ; CLEAR AND SET R/W MODE.
5504
5505 055254  005037  055610                    CLR     LV+2            ;INIT ORIGIN AT 0,0.
5506 055260  005037  055612                    CLR     LV+4
5507 055264  012704  001776                    MOV     #MAXX,R4        ;INIT EXP X AT MAX...
5508 055270  005005                            CLR     R5              ;...AND Y AT 0.
5509 055272  010437  055616                    MOV     R4,LVX          ;INIT X LENGTH...
5510 055276  053737  003064  055616            BIS     OPTI,LVX        ;...OPTIONAL INT...
5511 055304  010537  055620                    MOV     R5,LVY          ;...AND Y LENGTH.
5512 055310  012702  177776                    MOV     #-2,R2          ;INIT EXP X...
5513 055314  012703  000002                    MOV     #2,R3           ;...AND Y INCR'S.
5514 055320  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
5515
5516 055326                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5517
5518 055330  012777  055606  125474    9$:     MOV     #LV,@DPC        ;START 'EM UP.
5519 055336  004737  027454                    JSR     PC,WAITF
5520 055342  027704  125470                    CMP     @DXR,R4         ;CHECK DXR = R4...
5521 055346  001003                            BNE     2$
5522 055350  027705  125464                    CMP     @DYR,R5         ;...AND DYR = R5.
5523 055354  001410                            BEQ     3$
5524 055356                            2$:     HRDERR  LVF,XYERR       ;LONG VECTOR FAILS.
5525 055376                            3$:     LOOPTO  9$              ; REPEAT...
     055376  004737  027646                    JSR     PC,LOOP
     055402  103752                            BCS     9$
5526 055404                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5527
5528 055406  013737  003234  027700            MOV     TEMP2,LOOPK     ;RESET LOOPER.
5529 055414  033727  055616  001777            BIT     LVX,#1777
5530 055422  001411                            BEQ     4$              ;BR IF DELTA X = 0.
5531 055424  162737  000002  055616            SUB     #2,LVX          ;OTHERWISE, DECR DX...
5532 055432  062737  000002  055620            ADD     #2,LVY          ;..AND INCR DY.
5533 055440  060204                            ADD     R2,R4           ;SET NEXT EXP X...
```

```
5534 055442  060305                          ADD     R3,R5           ;...AND EXP Y.
5535 055444  000730                          BR      1$              ;...AND CONTINUE.
5536
5537 055446  033727  055616  020000  4$:     BIT     LVX,#MXY        ;ARE WE IN THE NEG CYCLE ??
5538 055454  001022                          BNE     10$             ;WE'RE DONE IF SO.
5539 055456  012737  001776  055610          MOV     #MAXX,LV+2      ;OTHERWISE, REINIT ORIGIN...
5540 055464  012737  001776  055612          MOV     #MAXY,LV+4      ;...AT MAX.MAX.
5541 055472  052737  021776  055616          BIS     #MXY!MAXX,LVX   ;INIT X LENGTH AT -MAX.
5542 055500  012737  020000  055620          MOV     #MXY!0,LVY      ;INIT Y LENGTH AT -0.
5543 055506  005004                          CLR     R4              ;INIT EXP X = 0...
5544 055510  012705  001776                  MOV     #MAXY,R5        ;...AND Y = MAX...
5545 055514  005402                          NEG     R2              ;...REVERSE EXP INCR'S.
5546 055516  005403                          NEG     R3
5547 055520  000702                          BR      1$              ;AND GO 'ROUND.
5548
5549 055522  005737  002502          10$:    TST     DPUMOD
5550 055526  001002                          BNE     11$
5551 055530  004737  027514                  JSR     PC,PAUSE1       ; PAUSE IF VIDEO ON.
5552 055534  004737  030072          11$:    JSR     PC,TSTEND
5553 055540                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
5554                                 ;
5555                                 ; ON ERROR, SHOW ORIGIN AND DELTAS.
5556                                 ;
5557 055544                          20$:    PRINTX  #LONGX,LV+2,LV+4,LVX,LVY ; EXTENSION.
5558 055604  000207                          RTS     PC
5559                                 ;
5560                                 ; DISPLAY FILE FOR LONG VECTOR TEST.
5561                                 ;
5562 055606  114000                  LV:     APNT
5563 055610  000000  000000                  0,0
5564 055614  113774                          LVEC!ALL
5565 055616  000000                  LVX:    0
5566 055620  000000                  LVY:    0
5567 055622  173000                          STOPN
5568
5569 055624                                  END.TEST

;*****************************************************************************
;*
;*      END TEST 19
;*
;*****************************************************************************
```

```
5571
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 20   -   RELATIVE POINTS
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 10.)

5572                            ;
5573                            ; VERIFY THAT WE CAN EXECUTE RELATIVE POINTS FROM 1000,1000
5574                            ; FOR THE FULL RANGE OF X/Y (-76 TO +76).
5575                            ;
5576                            ; PLOT ALL X POINTS, HOLDING Y AT -76.
5577                            ; PLOT ALL Y POINTS, HOLDING X AT -76.
5578                            ; PLOT ALL POINTS ON THE 45 (X = Y).
5579                            ;
5580                            ; REPEAT EACH POINT 10 TIMES AFTER QUICK PASS.
5581                            ;
5582 055676  004737  027702    RPTST:  JSR     PC,TSTGO        ;TITLE
5583 055702  012737  056242  022300        MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5584 055710  004737  027054            JSR     PC,LUMIN        ; BLAST THE LUT.
5585
5586 055714  005737  002502            TST     DPUMOD
5587 055720  001002                    BNE     .+6             ; SKIP NEXT IF DPU-ONLY.
5588 055722  004737  026650            JSR     PC,CLRRW        ; CLEAR AND SET R/W MODE.
5589
5590 055726  012737  177777  003232        MOV     #-1,TEMP1       ;USE AS XFER SWITCH.
5591 055734  013737  027700  003234        MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
5592 055742  012704  000702        8$:     MOV     #1000-76,R4     ;INIT EXPECTED X...
5593 055746  010405                    MOV     R4,R5           ;...AND Y.
5594 055750  012737  037576  056310        MOV     #37576,RPD      ;INIT DX,DY AT -MAX,-MAX.
5595 055756  053737  003064  056310        BIS     OPTI,RPD        ;OPTIONAL INTENSIFY.
5596
5597 055764                        1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5598
5599 055766  012777  056300  125036  9$:   MOV     #RPT,@DPC       ;START DISPLAY
5600 055774  004737  027454            JSR     PC,WAITF
5601 056000  027704  125032            CMP     @DXR,R4         ;CHECK DXR = R4...
5602 056004  001003                    BNE     2$
5603 056006  027705  125026            CMP     @DYR,R5         ;...AND DYR = R5.
5604 056012  001410                    BEQ     3$
5605 056014                        2$:     HRDERR  RPF,XYERR       ;REL POINT FAILS.
5606 056034                        3$:     LOOPTO  9$              ; REPEAT...
     056034  004737  027646            JSR     PC,LOOP
     056040  103752                    BCS     9$
5607 056042                            ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5608
5609 056044  013737  003234  027700        MOV     TEMP2,LOOPK     ;RESET LOOPER.
5610 056052  005737  003232            TST     TEMP1           ;TEST XFER SWITCH.
5611 056056  003036                    BGT     5$              ;INCR X AND Y.
5612 056060  001420                    BEQ     4$              ;INCR Y ONLY.
5613
5614 056062  062704  000002            ADD     #2,R4           ;INCR X ONLY.
5615 056066  010400                    MOV     R4,R0
5616 056070  004737  027312            JSR     PC,SDX          ;CALCULATE +/- DX IN R0.
5617 056074  042737  037600  056310        BIC     #37600,RPD      ;CLEAR OLD...
```

```
5618 056102  050037  056310            BIS     R0,RPD          ;...SET NEX DX.
5619 056106  020427  001076            CMP     R4,#1076
5620 056112  101724                    BLOS    1$              ;AND LOOP 'TIL DONE.
5621 056114  005237  003232            INC     TEMP1           ;THEN, BUMP SWITCH...
5622 056120  000710                    BR      8$              ;...AND REINIT.
5623
5624 056122  062705  000002      4$:   ADD     #2,R5           ;INCR Y ONLY.
5625 056126  010500                    MOV     R5,R0
5626 056130  004737  027270            JSR     PC,SDY          ;CALCULATE +/- DY IN R0.
5627 056134  110037  056310            MOVB    R0,RPD          ;SET NEW DY.
5628 056140  020527  001076            CMP     R5,#1076
5629 056144  101707                    BLOS    1$              ;AND LOOP 'TIL DONE.
5630 056146  005237  003232            INC     TEMP1           ;THEN, BUMP SWITCH...
5631 056152  000673                    BR      8$              ;...AND REINIT.
5632
5633 056154  062704  000002      5$:   ADD     #2,R4           ;INCR X AND Y.
5634 056160  010405                    MOV     R4,R5           ; (Y = X)
5635 056162  010500                    MOV     R5,R0
5636 056164  004737  027270            JSR     PC,SDY          ;CALCULATE +/- DY IN R0.
5637 056170  010001                    MOV     R0,R1           ;SAVE FOR A MOMENT.
5638 056172  000300                    SWAB    R0
5639 056174  006200                    ASR     R0              ;REPOSITION FOR DX.
5640 056176  050100                    BIS     R1,R0           ;COMBINE DX AND DY.
5641 056200  042737  037777  056310    BIC     #37777,RPD      ;CLEAR OLD...
5642 056206  050037  056310            BIS     R0,RPD          ;...SET NEW DX,DY.
5643 056212  020527  001076            CMP     R5,#1076
5644 056216  101662                    BLOS    1$              ;AND LOOP 'TIL DONE.
5645
5646 056220  005737  002502      10$:  TST     DPUMOD
5647 056224  001002                    BNE     11$
5648 056226  004737  027514            JSR     PC,PAUSE1       ; PAUSE IF VIDEO ON.
5649 056232  004737  030072      11$:  JSR     PC,TSTEND
5650 056236                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
5651                                   ;
5652                                   ; ON ERROR, SHOW ORIGIN AND DELTAS.
5653                                   ;
5654 056242                    20$:   PRINTX  #SHRTX,RPT+2,RPT+4,RPD ; EXTENSION.
5655 056276  000207                    RTS     PC
5656                                   ;
5657                                   ;DISPLAY FILE FOR REL POINT TEST.
5658                                   ;
5659 056300  114000            RPT:   APNT
5660 056302  001000  001000            1000,1000               ;ORIGIN AT CENTER SCREEN.
5661 056306  133774                    RPNT!ALL
5662 056310  000000            RPD:   0                       ;DELTA X AND DELTA Y.
5663 056312  173000                    STOPN
5664
5665 056314                            END.TEST

                                   ;*************************************************************************
                                   ;*
                                   ;*      END TEST 20
                                   ;*
                                   ;*************************************************************************
```

```
                                ;*****************************************************************************
                                ;*
                                ;*      BEGIN TEST 21   -   SHORT VECTORS
                                ;*
                                ;*****************************************************************************

                                ; (ITERATION COUNT = 10.)

                                ;
                                ; VERIFY THAT WE CAN EXECUTE SHORT VECTORS FROM 1000,1000
                                ; FOR ALL VALUES OF DELTA X AND Y.
                                ; SWEEP VECTORS FROM MAX,MAX, CCW THRU 360 DEGREES.
                                ;
                                ; REPEAT EACH VECTOR 10 TIMES AFTER QUICK PASS.
                                ;
056364  004737  027702          SVTST:  JSR     PC,TSTGO        ;TITLE
056370  012737  056742  022300          MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
056376  004737  027054                  JSR     PC,LUMIN        ; BLAST LUT.
056402  005737  002502                  TST     DPUMOD
056406  001002                          BNE     .+6             ; SKIP NEXT IF DPU-ONLY.
056410  004737  026650                  JSR     PC,CLRRW        ; CLEAR AND SET R/W MODE.
056414  012704  001074                  MOV     #1074,R4        ;INIT EXPECTED X...
056420  012705  001076                  MOV     #1076,R5        ;...AND Y TOO.
056424  012737  017076  057010          MOV     #17076,SVD      ;INIT DX +74, DX +76.
056432  053737  003064  057010          BIS     OPTI,SVD        ;...INT OPTIONAL.
056440  005037  003232                  CLR     TEMP1           ;USE AS XFER SWITCH.
056444  013737  027700  003234          MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPER.

056452                          1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.

056454  012777  057000  124350  9$:     MOV     #SVT,@DPC       ;START DISPLAY
056462  004737  027454                  JSR     PC,WAITF
056466  027704  124344                  CMP     @DXR,R4         ;CHECK DXR = R4...
056472  001003                          BNE     2$
056474  027705  124340                  CMP     @DYR,R5         ;...AND DYR = R5.
056500  001411                          BEQ     3$
056502                          2$:     HRDERR  SVF,XYERR       ;SHORT VECTOR FAILS.
056522  000403                          BR      30$
056524                          3$:     LOOPTO  9$              ; REPEAT...
056524  004737  027646                  JSR     PC,LOOP
056530  103751                          BCS     9$
056532                          30$:    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.

056534  013737  003234  027700          MOV     TEMP2,LOOPK     ;RESET LOOPER.
056542  005737  003232                  TST     TEMP1           ;TEST XFER SWITCH...
056546  001033                          BNE     7$              ;...BR IF SET.
056550  020427  000702                  CMP     R4,#1000-76
056554  001413                          BEQ     5$              ;BR IF X MIN.
056556  162704  000002                  SUB     #2,R4           ;OTHERWISE, DECR X.
056562  010400                          MOV     R4,R0
056564  004737  027312                  JSR     PC,SDX          ;TRANSLATE TO DX.
056570  042737  037600  057010          BIC     #37600,SVD      ;CLEAR OLD...
056576  050037  057010                  BIS     R0,SVD          ;...SET NEW DX...
056602  000723                          BR      1$              ;...AND CONTINUE.

056604  020527  000702          5$:     CMP     R5,#1000-76
```

```
5714 056610  001410                        BEQ     6$              ;BR IF Y MIN.
5715 056612  162705  000002                SUB     #2,R5           ;OTHERWISE, DECR Y.
5716 056616  010500                        MOV     R5,R0
5717 056620  004737  027270                JSR     PC,SDY          ;TRANSLATE TO DY.
5718 056624  110037  057010                MOVB    R0,SVD          ;SET NEW DY...
5719 056630  000710                        BR      1$              ;...AND CONTINUE.
5720
5721 056632  005137  003232          6$:   COM     TEMP1           ;SET XFER SWITCH.
5722 056636  020427  001076          7$:   CMP     R4,#1000+76
5723 056642  001413                        BEQ     8$              ;BR IF X IS MAX.
5724 056644  062704  000002                ADD     #2,R4           ;OTHERWISE, INCR X.
5725 056650  010400                        MOV     R4,R0
5726 056652  004737  027312                JSR     PC,SDX          ;TRANSLATE TO DX.
5727 056656  042737  037600  057010        BIC     #37600,SVD      ;CLEAR OLD...
5728 056664  050037  057010                BIS     R0,SVD          ;...SET NEW DX...
5729 056670  000670                        BR      1$              ;...AND CONTINUE.
5730
5731 056672  020527  001076          8$:   CMP     R5,#1000+76
5732 056676  001410                        BEQ     10$             ;BR (DONE) IF Y IS MAX.
5733 056700  062705  000002                ADD     #2,R5           ;OHTERWISE, INCR Y.
5734 056704  010500                        MOV     R5,R0
5735 056706  004737  027270                JSR     PC,SDY          ;TRANSLATE TO DY.
5736 056712  110037  057010                MOVB    R0,SVD          ;SET NEW DY...
5737 056716  000655                        BR      1$              ;...AND CONTINUE.
5738
5739 056720  005737  002502          10$:  TST     DPUMOD
5740 056724  001002                        BNE     11$
5741 056726  004737  027514                JSR     PC,PAUSE1       ; PAUSE IF VIDEO ON.
5742 056732  004737  030072          11$:  JSR     PC,TSTEND
5743 056736                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
5744                                ;
5745                                ; ON ERROR, SHOW ORIGIN AND DELTAS.
5746                                ;
5747 056742                        20$:    PRINTX  #SHRTX,SVT+2,SVT+4,SVD ; EXTENSION.
5748 056776  000207                        RTS     PC
5749                                ;
5750                                ; DISPLAY CODE FOR SHORT VECTORS.
5751                                ;
5752 057000  114000                SVT:    APNT
5753 057002  001000  001000               1000,1000               ;ORIGIN AT CENTER SCREEN.
5754 057006  107774                        SVEC!ALL
5755 057010  000000                SVD:    0                       ;DELTA X AND DELTA Y.
5756 057012  173000                        STOPN
5757
5758 057014                                END.TEST

;*************************************************************************
;*
;*      END TEST 21
;*
;*************************************************************************
```

```
5760
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 22   -   RUN-LENGTH
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 10.)

5761
5762                                    ;
5763                                    ; ALL VARIATIONS OF RUN LENTH MODE ARE DONE AND THE END POINTS ARE CHECKED.
5764                                    ; IF TESTING IN DPU-ONLY MODE, JUST THE 'NEWLINE'' COMMANDS ARE TESTED.
5765                                    ;
5766 057062                            RLMOD:  COMBEG
      057062  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      057066  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
5767  057072  004737  026650                    JSR     PC,CLRRW        ; CLEAR FOR READ/WRITE.
5768  057076  005737  002502            RLMOD1: TST     DPUMOD          ; ARE WE IN DPU-ONLY MODE?
5769  057102  001121                            BNE     RLNEWL          ; IF YES, JUST DO 'NEWLINE'' TESTS.
5770                                                                    ; IF NO, DO ALL TESTS ( WRITE TO SCREEN ).
5771                                                                    ; TEST X INCREMENT.
5772  057104  012705  000001            1$:     MOV     #1,R5           ; INIT COUNT FOR X TEST.
5773  057110                                    DPSTART #XINC
      057110  012777  057722  123714            MOV     #XINC,@DPC      ; START THE DPU.
      057116  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
5774  057122                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5775  057124  110537  057743            2$:     MOVB    R5,XINCB+1      ; MODIFY COUNT IN DISPLAY FILE.
5776  057130                                    DPCONT
      057130  052777  000001  123674            BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
      057136  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
5777  057142  010501                            MOV     R5,R1           ; R1 = EXPECTED DXR.
5778  057144  006301                            ASL     R1              ;
5779  057146  012703  000000                    MOV     #0,R3           ; R3 = EXPECTED DYR.
5780  057152  017702  123660                    MOV     @DXR,R2         ; READ DXR INTO R2.
5781  057156  017704  123656                    MOV     @DYR,R4         ; READ DYR INTO R4.
5782  057162                                    IFERRX2 RLXIE,RLXYE
      057162  020102                            CMP     R1,R2           ; 1ST ITEM OK?
      057164  001002                            BNE     65$             ; NO.
      057166  020304                            CMP     R3,R4           ; 2ND ITEM OK?
      057170  001410                            BEQ     64$             ; YES.
      057172                            65$:    HRDERR  RLXIE,RLXYE,    ; NO.
      057212                            64$:
5783  057212  005205                            INC     R5                      ; INCREMENT X VALUE.
5784  057214  020527  000200                    CMP     R5,#200         ; DONE?
5785  057220  001341                            BNE     2$              ; NO.
5786  057222                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5787                                                                    ; TEST X2 INCREMENT.
5788  057224  012705  000001                    MOV     #1,R5           ; INIT COUNT FOR X2 TEST.
5789  057230                                    DPSTART #X2INC
      057230  012777  057752  123574            MOV     #X2INC,@DPC     ; START THE DPU.
      057236  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
5790  057242                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5791  057244  110537  057773            3$:     MOVB    R5,X2INCB+1     ; MODIFY COUNT IN DISPLAY FILE.
5792  057250                                    DPCONT
      057250  052777  000001  123554            BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
      057256  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
```

```
 5793 057262  010501                        MOV     R5,R1           ; R1 = EXPECTED DXR.
 5794 057264  006301                        ASL     R1              ;
 5795 057266  006301                        ASL     R1              ;
 5796 057270  012703  000000                MOV     #0,R3           ; R3 = EXPECTED DYR.
 5797 057274  017702  123536                MOV     @DXR,R2         ; READ DXR INTO R2.
 5798 057300  017704  123534                MOV     @DYR,R4         ; READ DYR INTO R4.
 5799 057304                                IFERRX2 RLX2IE,RLXYE
      057304  020102                        CMP     R1,R2           ; 1ST ITEM OK?
      057306  001002                        BNE     67$             ; NO.
      057310  020304                        CMP     R3,R4           ; 2ND ITEM OK?
      057312  001410                        BEQ     66$             ; YES.
      057314                         67$:   HRDERR  RLX2IE,RLXYE,   ; NO.
      057334                         66$:
 5800 057334  005205                        INC     R5              ; INCREMENT X VALUE.
 5801 057336  020527  000200                CMP     R5,#200         ; DONE?
 5802 057342  001340                        BNE     3$              ; NO.
 5803 057344                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 5804
 5805                                ; TEST Y UP.
 5806 057346                         RLNEWL: BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 5807 057350                                DPSTART #YUP
      057350  012777  060002  123454        MOV     #YUP,@DPC       ; START THE DPU.
      057356  004737  027454                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 5808 057362  012701  000000                MOV     #0,R1           ; R1 = EXPECTED DXR.
 5809 057366  012703  000002                MOV     #2,R3           ; R3 = EXPECTED DYR.
 5810 057372  017702  123440                MOV     @DXR,R2         ; READ DXR INTO R2.
 5811 057376  017704  123436                MOV     @DYR,R4         ; READ DYR INTO R4.
 5812 057402                                IFERRX2 RLYUE,RLXYE
      057402  020102                        CMP     R1,R2           ; 1ST ITEM OK?
      057404  001002                        BNE     65$             ; NO.
      057406  020304                        CMP     R3,R4           ; 2ND ITEM OK?
      057410  001410                        BEQ     64$             ; YES.
      057412                         65$:   HRDERR  RLYUE,RLXYE,    ; NO.
      057432                         64$:
 5813 057432                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 5814
 5815                                ; TEST Y DOWN.
 5816 057434                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 5817 057436                                DPSTART #YDN
      057436  012777  060016  123366        MOV     #YDN,@DPC       ; START THE DPU.
      057444  004737  027454                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 5818 057450  012701  000000                MOV     #0,R1           ; R1 = EXPECTED DXR.
 5819 057454  012703  000000                MOV     #0,R3           ; R3 = EXPECTED DYR.
 5820 057460  017702  123352                MOV     @DXR,R2         ; READ DXR INTO R2.
 5821 057464  017704  123350                MOV     @DYR,R4         ; READ DYR INTO R4.
 5822 057470                                IFERRX2 RLYDE,RLXYE
      057470  020102                        CMP     R1,R2           ; 1ST ITEM OK?
      057472  001002                        BNE     67$             ; NO.
      057474  020304                        CMP     R3,R4           ; 2ND ITEM OK?
      057476  001410                        BEQ     66$             ; YES.
      057500                         67$:   HRDERR  RLYDE,RLXYE,    ; NO.
      057520                         66$:
 5823 057520                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 5824
 5825                                ; TEST SKIP UP.
 5826 057522                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 5827 057524                                DPSTART #Y2UP
```

```
        057524  012777  060032  123300          MOV     #Y2UP,@DPC      ; START THE DPU.
        057532  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 5828   057536  012701  000000                  MOV     #0,R1           ; R1 = EXPECTED DXR.
 5829   057542  012703  000004                  MOV     #4,R3           ; R3 = EXPECTED DYR.
 5830   057546  017702  123264                  MOV     @DXR,R2         ; READ DXR INTO R2.
 5831   057552  017704  123262                  MOV     @DYR,R4         ; READ DYR INTO R4.
 5832   057556                                  IFERRX2 RLY2UE,RLXYE
        057556  020102                          CMP     R1,R2           ; 1ST ITEM OK?
        057560  001002                          BNE     69$             ; NO.
        057562  020304                          CMP     R3,R4           ; 2ND ITEM OK?
        057564  001410                          BEQ     68$             ; YES.
        057566                          69$:    HRDERR  RLY2UE,RLXYE,   ; NO.
        057606                          68$:
 5833   057606                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 5834
 5835                                   ; TEST SKIP DOWN.
 5836   057610                                  BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 5837   057612                                  DPSTART #Y2DN
        057612  012777  060046  123212          MOV     #Y2DN,@DPC      ; START THE DPU.
        057620  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 5838   057624  012701  000000                  MOV     #0,R1           ; R1 = EXPECTED DXR.
 5839   057630  012703  000000                  MOV     #0,R3           ; R3 = EXPECTED DYR.
 5840   057634  017702  123176                  MOV     @DXR,R2         ; READ DXR INTO R2.
 5841   057640  017704  123174                  MOV     @DYR,R4         ; READ DYR INTO R4.
 5842   057644                                  IFERRX2 RLY2DE,RLXYE
        057644  020102                          CMP     R1,R2           ; 1ST ITEM OK?
        057646  001002                          BNE     71$             ; NO.
        057650  020304                          CMP     R3,R4           ; 2ND ITEM OK?
        057652  001410                          BEQ     70$             ; YES.
        057654                          71$:    HRDERR  RLY2DE,RLXYE,   ; NO.
        057674                          70$:
 5843   057674                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 5844
 5845   057676                          RLMXIT: COMEND  RLMOD1
        057676  004737  027646                  JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
        057702  103401                          BCS     64$
        057704  000402                          BR      65$

        057706  000137  057076          64$:    JMP     RLMOD1

        057712  004737  030072          65$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
 5846   057716                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>
 5847
 5848
 5849
 5850                                   ;
 5851                                   ; DISPLAY CODE.
 5852                                   ;
 5853
 5854   057722  114000  000000  000000  XINC:   APNT,0,0                ; X INCREMENT.
 5855   057730  173000                          STOPN
 5856   057732  114000  040000  000000  XINCA:  APNT,I!0,0
 5857   057740  144000                          RNLN
 5858   057742  000777                  XINCB:  000777
 5859   057744  173000                          STOPN
 5860   057746  160000  057732                  DJMP,XINCA
 5861
```

```
 5862 057752  114000  000000  000000  X2INC:   APNT,0,0                    ; X2 INCREMENT.
 5863 057760  173000                           STOPN
 5864 057762  114000  040000  000000  X2INCA:  APNT,I!0,0
 5865 057770  144100                           RNLN!DBLPIX
 5866 057772  000777                X2INCB:    000777
 5867 057774  173000                           STOPN
 5868 057776  160000  057762                   DJMP,X2INCA
 5869
 5870 060002  114000  000000  000000  YUP:     APNT,0,0                    ; Y UP.
 5871 060010  144000  000377                   RNLN,377
 5872 060014  173000                           STOPN
 5873
 5874 060016  114000  000000  000002  YDN:     APNT,0,2                    ; Y DOWN.
 5875 060024  144040  000377                   RNLN!YDOWN,377
 5876 060030  173000                           STOPN
 5877
 5878 060032  114000  000000  000000  Y2UP:    APNT,0,0                    ; Y SKIP UP.
 5879 060040  144400  000377                   RNLN!RLSKIP,377
 5880 060044  173000                           STOPN
 5881
 5882 060046  114000  000000  000004  Y2DN:    APNT,0,4                    ; Y SKIP DOWN.
 5883 060054  144440  000377                   RNLN!RLSKIP!YDOWN,377
 5884 060060  173000                           STOPN
 5885
 5886 060062                                   END.TEST

                        ;*****************************************************************************
                        ;*
                        ;*      END TEST 22
                        ;*
                        ;*****************************************************************************
```

```
5888
                                ;*****************************************************************************
                                ;*
                                ;*      BEGIN TEST 23   -   CURSOR REGISTERS/SWITCH/MATCH
                                ;*
                                ;*****************************************************************************

                                ; (ITERATION COUNT = 3.)

5889                            ;
5890                            ; THIS TEST, USING THE LOADABLE CURSOR POSITION REGISTERS AND "SOFT" SWITCH,
5891                            ; PERFORMS THE FOLLOWING CHECKS ON THE CURSOR REGISTERS, JOYSTICK SWITCH, AND
5892                            ; CURSOR MATCH AND ASSOCIATED LOGIC FOR EACH AVAILABLE SYNC GENERATOR CHANNEL:
5893                            ;
5894                            ;  (1)  WRITE/READ TEST OF THE CURSOR POSITION REGISTERS USING DATA OF ALL 1'S
5895                            ;       AND ALL 0'S, USING PROGRAMMED I/O (VIA DXR AND DYR) TO CHECK FOR STUCK
5896                            ;       0'S, STUCK 1'S AND DUAL ADDRESSING.
5897                            ;
5898                            ;  (2)  WRITE/READ TEST OF THE CURSOR POSITION REGISTERS VIA THE DISPLAY FILE
5899                            ;       INSTRUCTION LDCP (LOAD CURSOR POSITION) AND CURD (CURSOR READ), USING
5900                            ;       AN INCREMENTING COUNT PATTERN ON THE X POSITION (0 - 3776) AND A
5901                            ;       DECREMENTING COUNT PATTERN ON THE Y POSITION (3776 - 0).
5902                            ;
5903                            ;  (3)  WRITE/READ TEST OF THE SWITCH ENABLE, MATCH ENABLE, AND CROSSHAIR
5904                            ;       INTENSITY ENABLE BITS USING PROGRAMMED I/O TO SET AND CLEAR ALL BITS.
5905                            ;
5906                            ;  (4)  WRITE/READ TEST OF THE ENABLE BITS VIA DISPLAY FILE INSTRUCTION
5907                            ;       LDJSS (LOAD JOYSTICK STATUS), USING ALL COMBINATIONS OF LOAD-ENABLES
5908                            ;       AND DATA.
5909                            ;
5910                            ;  (5)  SET THE SWITCH ENABLE AND MATCH ENABLE BITS, THEN PERFORM A WRITE TO
5911                            ;       THE JOYSTICK STATUS REGISTER OF ANOTHER CHANNEL WITH DATA CONFIGURED
5912                            ;       TO SET THE ENABLE BITS; VERIFY THAT THE ENABLES OF THE CHANNEL UNDER
5913                            ;       TEST WERE CLEARED.  FAILURE HERE INDICATES A FAULTY CHANNEL ADDRESS
5914                            ;       DECODER IN THE SYNC CHANNEL UNDER TEST.
5915                            ;
5916                            ;  (6)  TEST OPERATION OF THE JOYSTICK SWITCH INTERRUPT.
5917                            ;
5918                            ;  (7)  TEST OPERATION OF THE CURSOR MATCH INTERRUPT. FLOATING 1'S AND 0'S
5919                            ;       PATTERNS ARE USED IN THE CURSOR X AND Y POSITIONS TO VERIFY THE A
5920                            ;       MATCH OCCURS ON THE SELECTED POINT WHILE DRAWING A VECTOR THROUGH IT,
5921                            ;       AND THAT A MATCH DOES NOT OCCUR WHILE DRAWING VECTORS AROUND THE
5922                            ;       SELECTED POINT PARALLEL TO THE X AND Y AXES.
5923                            ;
5924
5925 060152  005737  002502            TST     DPUMOD          ; DPU-MODE ONLY?
5926 060156  001402                    BEQ     CSMTST          ; BR IF NOT -- OK TO EXECUTE.
5927 060160  000137  065016            JMP     CSMXIT          ; YES -- JUST GO TO EXIT.
```

```
5929
5930 060164                                CSMTST: COMBEG
     060164  004737  027702                        JSR     PC,TSTGO        ; TITLE.
     060170  004737  026074                        JSR     PC,DPRESET      ; DO SOFT INIT.
5931 060174  004737  030414                1$:     JSR     PC,CONFIG               ; IDENTIFY THE AVAILABLE SYNC CHANNELS.
5932 060200  004737  027054                        JSR     PC,LUMIN
5933 060204  005037  065140                CSMITR: CLR     CSCHAN          ; START TESTING AT CHANNEL 0.
5934 060210  012737  065152  022300        CSMLOOP: MOV    #PRSYCH,EXTA    ; SET UP TO PRINT CHANNEL NUMBER ON ERROR.
5935 060216  004737  026074                        JSR     PC,DPRESET      ; DO SOFT INITIALIZE & CHECK REGISTERS.
5936 060222  013702  065140                        MOV     CSCHAN,R2       ; GET CHANNEL NUMBER INTO R2
5937 060226  006302                        ASL     R2              ; ... AND MAKE IT A WORD OFFSET.
5938 060230  005762  003270                        TST     SYCTAB(R2)      ; IS THERE A MODULE HERE?
5939 060234  001002                        BNE     1$              ; BR IF YES.
5940 060236  000137  064756                        JMP     CSMNEXT         ; IF NOT, GO SELECT NEXT CHANNEL.
5941 060242  006302                1$:     ASL     R2              ; SHIFT IT LEFT 5 MORE TIMES TO GET
5942 060244  006302                        ASL     R2
5943 060246  006302                        ASL     R2
5944 060250  006302                        ASL     R2
5945 060252  006302                        ASL     R2              ; ... CHANNEL NUMBER INTO BITS <7:6>
5946 060254  010237  065142                MOV     R2,CSCH.6       ; STORE IT FOR FUTURE USE.
5947 060260  006302                        ASL     R2              ; SHIFT IT LEFT 2 MORE TIMES TO GET
5948 060262  006302                        ASL     R2              ; ... IT INTO BITS <9:8>
5949 060264  010237  065144                MOV     R2,CSCH.8       ; STORE IT FOR FUTURE USE.
5950 060270  006302                        ASL     R2              ; SHIFT LEFT 3 MORE TIMES ...
5951 060272  006302                        ASL     R2              ; ... TO GET ...
5952 060274  006302                        ASL     R2              ; ... CHANNEL NUMBER INTO BITS <12:11>,
5953 060276  010237  065146                MOV     R2,CSCH.11      ; AND STORE IT FOR FUTURE USE.
5954 060302  112737  000000  026106        MOVB    #0,INTMASK      ; DON'T EXPECT ANY INTERRUPT.
5955 060310  004737  026114                JSR     PC,ENAINT       ; BUT LOOK FOR BAD ONES.
5956
5957                                ;WRITE 0'S INTO BOTH CURSOR REGISTERS AND VERIFY THAT ALL 0'S ARE RETURNED.
5958
5959 060314                                CSMT1:  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
5960 060316  012701  100000                        MOV     #BIT15,R1       ; EXPECT CURSOR X = 0 (WITH FLAG SET)
5961 060322  010103                        MOV     R1,R3           ; DITTO FOR CURSOR Y
5962 060324  013705  065150                        MOV     CSRETRY,R5      ; SET UP R5 AS RETRY COUNTER ;a@aR
5963 060330                                1$:     BISW3   CSCH.11,#WRTCPX!0,@DXR ; WRITE CURSOR X TO 0.
     060330  012746  140000                        MOV     #WRTCPX!0,-(SP)
     060334  053716  065146                        BIS     CSCH.11,(SP)
     060340  012677  122472                        MOV     (SP)+,@DXR
5964 060344  004737  027454                        JSR     PC,WAITF                ; WAIT FOR READY.
5965 060350                                        BISW3   CSCH.11,#WRTCPY!0,@DYR ; WRITE CURSOR Y TO 0.
     060350  012746  140000                        MOV     #WRTCPY!0,-(SP)
     060354  053716  065146                        BIS     CSCH.11,(SP)
     060360  012677  122454                        MOV     (SP)+,@DYR
5966 060364  004737  065474                        JSR     PC,GETCURS              ; READ CURSOR, GET DXR,DYR INTO R2,R4
5967 060370  020102                        CMP     R1,R2           ; CHECK DXR (X-CURSOR)
5968 060372  001002                        BNE     2$              ; BR IF ERROR
5969 060374  020304                        CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
5970 060376  001412                        BEQ     3$              ; BR IF OK.
5971 060400  005305                2$:     DEC     R5              ; DECREMENT THE RETRY COUNTER ;a@aR
5972 060402  100352                        BPL     1$              ; GO TRY AGAIN IF NOT NEGATIVE ;a@aR
5973 060404                                HRDERR  CRWRZ,RLXYE     ; REPORT THE ERROR.
5974 060424                        3$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
```

```
5976
5977                                    ;WRITE 1'S INTO BOTH CURSOR REGISTERS AND VERIFY THAT ALL 1'S ARE RETURNED.
5978
5979 060426                             CSMT2:  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
5980 060430  012701  103776                     MOV     #BIT15!^O3776,R1 ; EXPECT CURSOR X = 3776 (WITH FLAG SET)
5981 060434  010103                             MOV     R1,R3           ; DITTO FOR CURSOR Y
5982 060436  013705  065150                     MOV     CSRETRY,R5      ; @@@R
5983 060442                             1$:     BISW3   CSCH.11,#WRTCPX!^O3776,@DXR ; WRITE CURSOR X TO 3776.
     060442  012746  143776                     MOV     #WRTCPX!^O3776,-(SP)
     060446  053716  065146                     BIS     CSCH.11,(SP)
     060452  012677  122360                     MOV     (SP)+,@DXR
5984 060456  004737  027454                     JSR     PC,WAITF                ; WAIT FOR READY.
5985 060462                                     BISW3   CSCH.11,#WRTCPY!^O3776,@DYR ; WRITE CURSOR Y TO 3776.
     060462  012746  143776                     MOV     #WRTCPY!^O3776,-(SP)
     060466  053716  065146                     BIS     CSCH.11,(SP)
     060472  012677  122342                     MOV     (SP)+,@DYR
5986 060476  004737  065474                     JSR     PC,GETCURS              ; READ CURSOR, GET DXR,DYR INTO R2,R4
5987 060502  020102                             CMP     R1,R2           ; CHECK DXR (X-CURSOR)
5988 060504  001002                             BNE     2$              ; BR IF ERROR
5989 060506  020304                             CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
5990 060510  001412                             BEQ     3$              ; BR IF OK.
5991 060512  005305                     2$:     DEC     R5              ; DECREMENT THE RETRY COUNTER ;@@@R
5992 060514  100352                             BPL     1$              ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
5993 060516                                     HRDERR  CRWRO,RLXYE     ; REPORT THE ERROR.
5994 060536                             3$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
```

```
 5996
 5997                                   ;CHECK DUAL ADDRESSING -- VERIFY THAT Y NOT CHANGED BY WRITING X.
 5998
 5999 060540  012701  100000      CSMT3:  MOV     #BIT15!0,R1     ; EXPECT CURSOR X TO BE 0
 6000 060544  012703  103776              MOV     #BIT15!^O03776,R3 ; EXPECT CURSOR Y TO BE ALL 1'S
 6001 060550                              BISW3   CSCH.11,#WRTCPX,R5 ; SET UP WRITE CURSOR W/ CHANNEL
      060550  012705  140000              MOV     #WRTCPX,R5
      060554  053705  065146              BIS     CSCH.11,R5
 6002
 6003 060560                      1$:     BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
 6004 060562  013700  065150              MOV     CSRETRY,R0      ; @@@R
 6005 060566                      10$:    BISW3   R5,R3,@DYR      ; LOAD CURSOR Y WITH A PATTERN.
      060566  010346                      MOV     R3,-(SP)
      060570  050516                      BIS     R5,(SP)
      060572  012677  122242              MOV     (SP)+,@DYR
 6006 060576  004737  027454              JSR     PC,WAITF        ; WAIT FOR STOP.
 6007 060602                              BISW3   R5,R1,@DXR      ; LOAD OPPOSITE PATTERN INTO CURSOR X.
      060602  010146                      MOV     R1,-(SP)
      060604  050516                      BIS     R5,(SP)
      060606  012677  122224              MOV     (SP)+,@DXR
 6008 060612  004737  065474              JSR     PC,GETCURS      ; GET THE CURSOR INTO R2 & R4.
 6009 060616  020102                      CMP     R1,R2           ; CHECK DXR (X-CURSOR)
 6010 060620  001002                      BNE     2$              ; BR IF ERROR
 6011 060622  020304                      CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
 6012 060624  001412                      BEQ     3$              ; BR IF OK.
 6013 060626  005300              2$:     DEC     R0              ; DECREMENT THE RETRY COUNTER ;@@@R
 6014 060630  100356                      BPL     10$             ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
 6015 060632                              HRDERR  CRDAX,RLXYE     ; REPORT THE ERROR.
 6016 060652                      3$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
 6017
 6018 060654  032701  003776              BIT     #^O03776,R1     ; DID WE WRITE 1'S INTO X YET?
 6019 060660  001004                      BNE     4$              ; BR IF YES, SO WE'RE DONE.
 6020 060662  010301                      MOV     R3,R1           ; NO, SO SWAP PATTERNS.
 6021 060664  012703  100000              MOV     #BIT15,R3       ; ... SET UP TO WRITE Y TO 0.
 6022 060670  000733                      BR      1$              ; LOOP BACK TO DO IT.
 6023 060672                      4$:
```

```
                                ;CHECK DUAL ADDRESSING -- VERIFY THAT X NOT CHANGED BY WRITING Y.

060672  012701  103776         CSMT4:  MOV     #BIT15!^O3776,R1 ; EXPECT CURSOR X TO BE ALL 1'S
060676  012703  100000                 MOV     #BIT15!0,R3     ; EXPECT CURSOR Y TO BE 0
060702                                 BISW3   CSCH.11,#WRTCPX,R5 ; SET UP WRITE CURSOR W/ CHANNEL
060702  012705  140000                 MOV     #WRTCPX,R5
060706  053705  065146                 BIS     CSCH.11,R5

060712                          1$:    BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
060714  013700  065150                 MOV     CSRETRY,R0      ; @@@R
060720                          10$:    BISW3   R5,R1,@DXR      ; LOAD CURSOR X WITH A PATTERN.
060720  010146                         MOV     R1,-(SP)
060722  050516                         BIS     R5,(SP)
060724  012677  122106                 MOV     (SP)+,@DXR
060730  004737  027454                 JSR     PC,WAITF
060734                                 BISW3   R5,R3,@DYR      ; LOAD OPPOSITE PATTERN INTO CURSOR Y.
060734  010346                         MOV     R3,-(SP)
060736  050516                         BIS     R5,(SP)
060740  012677  122074                 MOV     (SP)+,@DYR
060744  004737  065474                 JSR     PC,GETCURS              ; GET THE CURSOR INTO R2 & R4.
060750  020102                         CMP     R1,R2           ; CHECK DXR (X-CURSOR)
060752  001002                         BNE     2$              ; BR IF ERROR
060754  020304                         CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
060756  001412                         BEQ     3$              ; BR IF OK.
060760  005300                  2$:    DEC     R0              ; DECREMENT THE RETRY COUNTER ;@@@R
060762  100356                         BPL     10$             ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
060764                                 HRDERR  CRDAY,RLXYE     ; REPORT THE ERROR.
061004                          3$:    ENDSEG          ;>>>>>>>>>>>>>END SEGMENT

061006  032703  003776                 BIT     #^O3776,R3      ; DID WE WRITE 1'S INTO Y YET?
061012  001004                         BNE     4$              ; BR IF YES, SO WE'RE DONE.
061014  010103                         MOV     R1,R3           ; NO, SO SWAP PATTERNS.
061016  012701  100000                 MOV     #BIT15,R1       ; ... SET UP TO WRITE X TO 0.
061022  000733                         BR      1$              ; LOOP BACK TO DO IT.
061024                          4$:
```

```
                                        ; DO WRITE/READ TEST USING DISPLAY FILE.

061024  005001                  CSMT5:  CLR     R1                  ; INIT THE X PATTERN TO 0
061026  012703  003776                  MOV     #^O3776,R3          ; INIT THE Y PATTERN TO ALL 1'S
061032                                  BISW3   CSCH.6,#LDCP,CSMF1A ; SET UP THE DISPLAY FILE
061032  012746  175000                  MOV     #LDCP,-(SP)
061036  053716  065142                  BIS     CSCH.6,(SP)
061042  012637  065200                  MOV     (SP)+,CSMF1A
061046                                  BISW3   CSCH.8,#CURD,CSMF1B ; ... BY INSERTING THE CHANNEL.
061046  012746  146100                  MOV     #CURD,-(SP)
061052  053716  065144                  BIS     CSCH.8,(SP)
061056  012637  065206                  MOV     (SP)+,CSMF1B

061062  010137  065202          1$:     MOV     R1,CSMF1A+2         ; INSERT X COORD. TO BE WRITTEN
061066  010337  065204                  MOV     R3,CSMF1A+4         ; ... AND THE Y COORDINATE.
061072  013705  065150                  MOV     CSRETRY,R5
061076                                  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
061100                          2$:     DPSTART #CSMF1A             ; EXECUTE LOAD THEN READ OF CURSOR.
061100  012777  065200  121724          MOV     #CSMF1A,@DPC        ; START THE DPU.
061106  004737  027454                  JSR     PC,WAITF            ; WAIT FOR DISPLAY STOP.
061112  052701  100000                  BIS     #BIT15,R1           ; EXPECT THE CURSOR FLAG BIT IN DXR
061116  052703  100000                  BIS     #BIT15,R3           ; ... AND DYR.
061122  004737  065520                  JSR     PC,RDDXY            ; GET MASKED DXR AND DYR INTO R2 AND R4.
061126  020102                          CMP     R1,R2               ; CHECK DXR FOR X CURSOR.
061130  001002                          BNE     3$                  ; BR IF INCORRECT.
061132  020304                          CMP     R3,R4               ; CHECK DYR FOR Y CURSOR.
061134  001412                          BEQ     4$                  ; BR IF OK.
061136  005305                  3$:     DEC     R5
061140  100357                          BPL     2$
061142                                  HRDERR  CRWRE,RLXYE         ; ERROR -- DXR OR DYR WRITE/READ ERROR.
061162                          4$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT

061164  062701  100002                  ADD     #2!BIT15,R1         ; INCREMENT THE X PATTERN.
061170  162703  100002                  SUB     #2!BIT15,R3         ; DECREMENT THE Y PATTERN.
061174  032701  003776                  BIT     #^O3776,R1          ; DID WE INCREMENT X AROUND TO 0?
061200  001330                          BNE     1$                  ; LOOP BACK IF NO.
061202                          30$:
```

```
                                ; *** SPACE HERE FOR TESTING GENERAL DUAL ADDRESSING (ALL DBUS CODES & CHANNELS)

                                ; VERIFY THAT THE SWITCH, MATCH AND CROSSHAIR ENABLES CAN BE CLEARED TO 0'S.

                                CSMT6:  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
                                        JSR     PC,DSBINT       ; DISABLE INTERRUPTS
                                        MOV     #BIT15,R1       ; EXPECT DXR TO BE 0
                                        MOV     R1,R3           ; DITTO FOR DYR.
                                        BISW3   CSCH.8,#WJSS!^O77,@DXR ; FIRST SET ALL ENABLES.
061216 012746 042077                    MOV     #WJSS!^O77,-(SP)
061222 053716 065144                    BIS     CSCH.8,(SP)
061226 012677 121604                    MOV     (SP)+,@DXR
                                        JSR     PC,WAITF                ; WAIT FOR READY.
                                        BISW3   CSCH.8,#WJSS!BIT5!BIT3!BIT1,@DXR ; THEN CLEAR THEM.
061236 012746 042052                    MOV     #WJSS!BIT5!BIT3!BIT1,-(SP)
061242 053716 065144                    BIS     CSCH.8,(SP)
061246 012677 121564                    MOV     (SP)+,@DXR
                                        MOV     CSRETRY,R5              ; @@@R
                                1$:     JSR     PC,GETJSE               ; GET THE ENABLES INTO R2 AND R4.
                                        CMP     R1,R2           ; CHECK DXR (X-CURSOR)
                                        BNE     2$              ; BR IF ERROR
                                        CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
                                        BEQ     3$              ; BR IF OK.
                                2$:     DEC     R5              ; DECREMENT THE RETRY COUNTER ;@@@R
                                        BPL     1$              ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
                                        HRDERR  JSEWRZ,RLXYE    ; REPORT THE ERROR.
                                3$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT

                                ; VERIFY THAT THE SWITCH, MATCH AND CROSSHAIR ENABLES CAN BE SET TO 1'S.

                                CSMT7:  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
                                        MOV     #BIT15!JSSIES!JSMIES,R1 ; EXPECT DXR TO HAVE SW & MAT SET.
                                        MOV     #BIT15!CHIE,R3  ; EXPECT DYR TO HAVE CROSSHAIR ENABLE SET.
                                        BISW3   CSCH.8,#WJSS!BIT5!BIT3!BIT1,@DXR ; FIRST CLEAR THEM.
061332 012746 042052                    MOV     #WJSS!BIT5!BIT3!BIT1,-(SP)
061336 053716 065144                    BIS     CSCH.8,(SP)
061342 012677 121470                    MOV     (SP)+,@DXR
                                        JSR     PC,WAITF                ; WAIT FOR READY.
                                        BISW3   CSCH.8,#WJSS!^O77,@DXR ; THEN SET THEM.
061352 012746 042077                    MOV     #WJSS!^O77,-(SP)
061356 053716 065144                    BIS     CSCH.8,(SP)
061362 012677 121450                    MOV     (SP)+,@DXR
                                        MOV     CSRETRY,R5              ; @@@R
                                1$:     JSR     PC,GETJSE               ; GET THE ENABLES INTO R2 AND R4.
                                        CMP     R1,R2           ; CHECK DXR (X-CURSOR)
                                        BNE     2$              ; BR IF ERROR
                                        CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
                                        BEQ     3$              ; BR IF OK.
                                2$:     DEC     R5              ; DECREMENT THE RETRY COUNTER ;@@@R
                                        BPL     1$              ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
                                        HRDERR  JSEWRO,RLXYE    ; REPORT THE ERROR.
                                3$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
```

```
                                ; TEST ENABLES WITH GENERAL WRITE/READ FROM DISPLAY FILE.

061434  112737  000000  026106  CSMT8:  MOVB    #0,INTMASK
061442  004737  026114                  JSR     PC,ENAINT       ; ENABLE INTERRUPTS, TO FIND BAD ONES.
061446  013704  065144                  MOV     CSCH.B,R4       ; GET A COPY OF THE CHANNEL #, FOR INSERTING.
061452                                  BISW3   R4,#CUIOFF!CUOFF,CSMF2A ; SET UP TO CLEAR ALL.
061452  012746  146052                  MOV     #CUIOFF!CUOFF,-(SP)
061456  050416                          BIS     R4,(SP)
061460  012637  065212                  MOV     (SP)+,CSMF2A
061464                                  BISW3   R4,#LDJSS,CSMF2B        ; SET UP TO GENERAL WRITE.
061464  012746  146000                  MOV     #LDJSS,-(SP)
061470  050416                          BIS     R4,(SP)
061472  012637  065214                  MOV     (SP)+,CSMF2B
061476                                  BISW3   R4,#CURD,CSMF2C ; SET UP FOR READ.
061476  012746  146100                  MOV     #CURD,-(SP)
061502  050416                          BIS     R4,(SP)
061504  012637  065216                  MOV     (SP)+,CSMF2C
061510                                  BISW3   R4,#LDJSS!^O77,CSMF2D   ; SET UP TO SET ALL.
061510  012746  146077                  MOV     #LDJSS!^O77,-(SP)
061514  050416                          BIS     R4,(SP)
061516  012637  065222                  MOV     (SP)+,CSMF2D

061522                          1$:     BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
061524  012701  100000                  MOV     #BIT15,R1               ; ASSUME JSSIE AND JSMIE WILL BE 0.
061530  010103                          MOV     R1,R3                   ; ASSUME CHIE WILL BE 0.
061532  013705  065214                  MOV     CSMF2B,R5               ; GET A COPY OF THE WRITTEN BITS.
061536  032705  000002                  BIT     #JSWE,R5                ; ENABLED TO WRITE SWITCH ENABLE?
061542  001405                          BEQ     2$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
061544  032705  000001                  BIT     #JSWD,R5                ; WRITE TO 1?
061550  001402                          BEQ     2$                      ; BR IF NO -- LEAVE AT 0.
061552  052701  004000                  BIS     #JSSIES,R1              ; YES -- SET EXPECTED STATUS.
061556                          2$:
061556  032705  000010                  BIT     #JMWE,R5                ; ENABLED TO WRITE MATCH ENABLE?
061562  001405                          BEQ     3$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
061564  032705  000004                  BIT     #JMWD,R5                ; WRITE TO 1?
061570  001402                          BEQ     3$                      ; BR IF NO -- LEAVE AT 0.
061572  052701  010000                  BIS     #JSMIES,R1              ; YES -- SET EXPECTED STATUS.
061576                          3$:
061576  032705  000040                  BIT     #JCWE,R5                ; ENABLED TO WRITE CROSSHAIR ENABLE?
061602  001405                          BEQ     4$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
061604  032705  000020                  BIT     #JCWD,R5                ; WRITE TO 1?
061610  001402                          BEQ     4$                      ; BR IF NO -- LEAVE AT 0.
061612  052703  010000                  BIS     #CHIE,R3                ; YES -- SET EXPECTED STATUS.

061616  013700  065150          4$:     MOV     CSRETRY,R0              ; @@@R

061622                          40$:    DPSTART #CSMF2A         ; EXECUTE CLEAR ALL, THEN GENERAL WRITE.
061622  012777  065212  121202          MOV     #CSMF2A,@DPC    ; START THE DPU.
061630  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
061634  004737  065566                  JSR     PC,RDJSE        ; GET DXR, DYR INTO R2, R4.
061640  020102                          CMP     R1,R2           ; CHECK DXR (X-CURSOR)
061642  001002                          BNE     41$             ; BR IF ERROR
061644  020304                          CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
061646  001412                          BEQ     43$             ; BR IF OK.
061650  005300                  41$:    DEC     R0              ; DECREMENT THE RETRY COUNTER ;@@@R
061652  100363                          BPL     40$             ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
```

```
                                        HRDERR  CEZWE,RLXYE     ; REPORT THE ERROR.
                                43$:
061674  012701  114000                  MOV     #BIT15!JSSIES!JSMIES,R1 ; ASSUME SWITCH & MATCH ENABLES ARE 1.
061700  012703  110000                  MOV     #BIT15!CHIE,R3          ; ASSUME CROSSHAIR ENABLE IS 1.

061704  032705  000002                  BIT     #JSWE,R5                ; ENABLED TO WRITE SWITCH ENABLE?
061710  001405                          BEQ     5$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
061712  032705  000001                  BIT     #JSWD,R5                ; WRITE TO 0?
061716  001002                          BNE     5$                      ; BR IF NO -- LEAVE AT 1.
061720  042701  004000                  BIC     #JSSIES,R1              ; YES -- CLEAR EXPECTED STATUS.

                                5$:
061724  032705  000010                  BIT     #JMWE,R5                ; ENABLED TO WRITE MATCH ENABLE?
061730  001405                          BEQ     6$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
061732  032705  000004                  BIT     #JMWD,R5                ; WRITE TO 0?
061736  001002                          BNE     6$                      ; BR IF NO -- LEAVE AT 1.
061740  042701  010000                  BIC     #JSMIES,R1              ; YES -- CLEAR EXPECTED STATUS.
                                6$:
061744  032705  000040                  BIT     #JCWE,R5                ; ENABLED TO WRITE CROSSHAIR ENABLE?
061750  001407                          BEQ     7$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
061752  032705  000020                  BIT     #JCWD,R5                ; WRITE TO 0?
061756  001004                          BNE     7$                      ; BR IF NO -- LEAVE AT 1.
061760  042703  010000                  BIC     #CHIE,R3                ; YES -- CLEAR EXPECTED STATUS.
061764  013700  065150                  MOV     CSRETRY,R0              ; @@@R
                                7$:
061770                                  DPCONT          ; RESTART, TO SET ALL, THEN DO GENERAL WRITE.
061770  052777  000001  121034          BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
061776  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
062002  004737  027454                  JSR     PC,WAITF                        ; WAIT FOR DONE.
062006  004737  065566                  JSR     PC,RDJSE                        ; GET DXR, DYR INTO R2, R4.
062012  020102                          CMP     R1,R2           ; CHECK DXR (X-CURSOR)
062014  001002                          BNE     72$             ; BR IF ERROR
062016  020304                          CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
062020  001412                          BEQ     73$             ; BR IF OK.
062022  005300                  72$:    DEC     R0              ; DECREMENT THE RETRY COUNTER ;@@@R
062024  100361                          BPL     7$              ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
062026                                  HRDERR  CESWE,RLXYE     ; REPORT THE ERROR.

062046                          73$:    ENDSEG          ;>>>>>>>>>>>>END SEGMENT
062050  005237  065214                  INC     CSMF2B          ; INCREMENT THE LDJSS WRITE DATA.
062054  032737  000077  065214          BIT     #^077,CSMF2B            ; DONE ALL COMBOS?
062062  001402                          BEQ     8$                      ; BR TO EXIT IF YES.
062064  000137  061522                  JMP     1$                      ; NO, SO LOOP BACK FOR ANOTHER ROUND.
062070                          8$:
```

```
                                        ; CHECK THAT WRITING JS AND JM ENABLES OF OTHER CHANNELS CLEARS ENABLES OF
                                        ; THE CHANNEL UNDER TEST.
                                        ;
062070  004737  026074        CSMT9:    JSR     PC,DPRESET              ; DO SOFT INITIALIZE.
062074  105037  026106                  CLRB    INTMASK
062100  013705  065144                  MOV     CSCH.8,R5               ; GET CHANNEL # IN R5<9:8>

062104  062705  000400        1$:       ADD     #BIT8,R5                ; INCREMENT TARGET CHANNEL
062110  042705  176377                  BIC     #^C<BIT9!BIT8>,R5       ; MASK THE CHANNEL FIELD.
062114  020537  065144                  CMP     R5,CSCH.8               ; BACK TO THE CHANNEL UNDER TEST?
062120  001451                          BEQ     2$                      ; BR = YES. DONE

062122                                  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
062124                                  BISW3   CSCH.8,#WRTJSS!JCWE!JMWE!JMWD!JSWE!JSWD,@DXR    ; SET ENABLES
062124  012746  042057                  MOV     #WRTJSS!JCWE!JMWE!JMWD!JSWE!JSWD,-(SP)
062130  053716  065144                  BIS     CSCH.8,(SP)
062134  012677  120676                  MOV     (SP)+,@DXR
062140  004737  027454                  JSR     PC,WAITF
062144                                  BISW3   R5,#WRTJSS!JMWE!JMWD!JSWE!JSWD,@DXR
062144  012746  042017                  MOV     #WRTJSS!JMWE!JMWD!JSWE!JSWD,-(SP)
062150  050516                          BIS     R5,(SP)
062152  012677  120660                  MOV     (SP)+,@DXR
062156  004737  027454                  JSR     PC,WAITF
062162  012701  100000                  MOV     #BIT15,R1       ; EXPECT ENABLES TO BE CLEAR.
062166  012703  100000                  MOV     #BIT15,R3
062172  013700  065150                  MOV     CSRETRY,R0      ; @@@R
062176  004737  065542        10$:      JSR     PC,GETJSE       ; LOAD R2 & R4 W/ DXR, DYR.
062202  020102                          CMP     R1,R2           ; CHECK DXR (X-CURSOR)
062204  001002                          BNE     12$             ; BR IF ERROR
062206  020304                          CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
062210  001412                          BEQ     13$             ; BR IF OK.
062212  005300                12$:      DEC     R0              ; DECREMENT THE RETRY COUNTER ;@@@R
062214  100370                          BPL     10$             ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
062216                                  HRDERR  JENOCL,RLXYE    ; REPORT THE ERROR.

062236                          13$:    ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
062240  000137  062104                  JMP     1$                      ; LOOP BACK TO TRY ANOTHER TARGET.

062244                          2$:
```

```
                                        ;
                                        ;TEST THE SOFT JOYSTICK SWITCH INTERRUPT OPERATION:
                                        ;
                                        ;   (1)  CLEAR SWITCH ENABLE, SET SWITCH, SEE THAT NO INTERRUPT PENDING.
                                        ;   (2)  CLEAR SWITCH, SET SWITCH ENABLE, SEE THAT NO INTERRUPT PENDING.
                                        ;   (3)  SET BOTH SWITCH ENABLE AND SWITCH, SEE THAT FLAGS SHOWS PENDING
                                        ;        INTERRUPT,
                                        ;   (4)  REPEAT (3), BUT ALLOW INTERRUPT AND CHECK RESULTS
                                        ;
                                        ;   (5)  REPEAT (4) BUT FROM DISPLAY FILE, SEE THAT SWITCH INTERRUPT STOPS
                                        ;        PROCESSING AND GETS CURSOR.
                                        ;

                                        ;(1) TEST THAT NO INTERRUPT REQUEST OCCURS WITH ENABLE CLEAR AND SWITCH SET.

062244                                  CSMT10: BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
062246  004737  026176                          JSR     PC,DSBINT               ; DISABLE INTERRUPTS.
062252                                          INITDP                          ; INITIALIZE THE DPU.
062252  004737  025402                          JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
062256                                          BISW3   CSCH.8,#WJSS!JSWE!0,@DXR  ; CLEAR SWITCH ENABLE.
062256  012746  042002                          MOV     #WJSS!JSWE!0,-(SP)
062262  053716  065144                          BIS     CSCH.8,(SP)
062266  012677  120544                          MOV     (SP)+,@DXR
062272  004737  027454                          JSR     PC,WAITF
062276                                          BISW3   CSCH.8,#WECC!SWCHON,@DYR  ; SET THE SWITCH.
062276  012746  160002                          MOV     #WECC!SWCHON,-(SP)
062302  053716  065144                          BIS     CSCH.8,(SP)
062306  012677  120526                          MOV     (SP)+,@DYR
062312  004737  027454                          JSR     PC,WAITF
062316  004737  023422                          JSR     PC,IIRCHK               ; CHECK INTERNAL REGS FOR INIT STATE.
062322                                  1$:     BISW3   CSCH.8,#WECC,@DYR       ;TURN OFF THE SWITCH
062322  012746  160000                          MOV     #WECC,-(SP)
062326  053716  065144                          BIS     CSCH.8,(SP)
062332  012677  120502                          MOV     (SP)+,@DYR
062336  004737  027454                          JSR     PC,WAITF
062342  012777  040000  120466                  MOV     #GTJSSW,@DXR            ;DISMISS INTR REQ, IF ANY
062350  004737  027454                          JSR     PC,WAITF
062354                                          WTDSRA  #0,FLG                  ;CLEAR PENDING INTERRUPT
062354  012777  000012  120452                  MOV     #0&^C17!SETFLG,@DSR
062362                                          ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
```

```
                                        ;(2) TEST THAT NO INTERRUPT REQUEST OCCURS WITH ENABLE SET AND SWITCH CLEAR.

                                CSMT11: BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
 062366  004737  026176                 JSR     PC,DSBINT                       ; DISABLE INTERRUPTS.
 062372                                 INITDP                                  ; INITIALIZE THE DPU.
 062372  004737  025402                 JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
 062376                                 BISW3   CSCH.8,#WECC!0,@DYR  ; CLEAR THE SWITCH.
 062376  012746  160000                 MOV     #WECC!0,-(SP)
 062402  053716  065144                 BIS     CSCH.8,(SP)
 062406  012677  120426                 MOV     (SP)+,@DYR
 062412  004737  027454                 JSR     PC,WAITF
 062416                                 BISW3   CSCH.8,#WJSS!JSWE!JSWD,@DXR  ; SET SWITCH ENABLE.
 062416  012746  042003                 MOV     #WJSS!JSWE!JSWD,-(SP)
 062422  053716  065144                 BIS     CSCH.8,(SP)
 062426  012677  120404                 MOV     (SP)+,@DXR
 062432  004737  027454                 JSR     PC,WAITF
 062436  004737  023422                 JSR     PC,IIRCHK                       ; CHECK INTERNAL REGS FOR INIT STATE.
 062442                         1$:     INITDP
 062442  004737  025402                 JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
 062446                                 ENDSEG          ;>>>>>>>>>>>>END SEGMENT
```

```
                                        ;(3) SET SWITCH AND ENABLE, SEE PENDING INTERRUPT
                                        
                                CSMT12: BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
                                        INITDP
062452  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
062456  004737  026176                  JSR     PC,DSBINT
062462                                  BISW3   CSCH.8,#WJSS!JSWE!JSWD,@DXR     ; ENABLE SWITCH INTERRUPT.
062462  012746  042003                  MOV     #WJSS!JSWE!JSWD,-(SP)
062466  053716  065144                  BIS     CSCH.8,(SP)
062472  012677  120340                  MOV     (SP)+,@DXR
062476  004737  027454                  JSR     PC,WAITF
062502                                  BISW3   CSCH.8,#WECC!SWCHON,@DYR        ; TURN THE SWITCH ON.
062502  012746  160002                  MOV     #WECC!SWCHON,-(SP)
062506  053716  065144                  BIS     CSCH.8,(SP)
062512  012677  120322                  MOV     (SP)+,@DYR
062516  004737  027454                  JSR     PC,WAITF
062522  012701  100016                  MOV     #CHAR!JSLCKO!PINTRB!PVEC2,R1    ; EXPECT FLAGS TO SHOW PENDING.
062526                                  RDDSRA  FLG,R2                  ; READ THE FLAGS REGISTER INTO R2.
062526  012777  000002  120300          MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
062534  017702  120274                  MOV     @DSR,R2
062540                                  IFERROR NOSWPI,EXPREC
062540  020102                          CMP     R1,R2           ; OK?
062542  001410                          BEQ     64$             ; YES.
062544                                  HRDERR  NOSWPI,EXPREC,  ; NO.
062564                          64$:
062564                          1$:
062564                                  BISW3   CSCH.8,#WECC,@DYR       ; TURN OFF THE SWITCH.
062564  012746  160000                  MOV     #WECC,-(SP)
062570  053716  065144                  BIS     CSCH.8,(SP)
062574  012677  120240                  MOV     (SP)+,@DYR
062600  004737  027454                  JSR     PC,WAITF
062604                                  BISW3   CSCH.8,#WJSS!JSWE!0,@DXR
062604  012746  042002                  MOV     #WJSS!JSWE!0,-(SP)
062610  053716  065144                  BIS     CSCH.8,(SP)
062614  012677  120216                  MOV     (SP)+,@DXR
062620  004737  027454                  JSR     PC,WAITF
062624  012777  040000  120204          MOV     #GTJSSW,@DXR            ;DISMISS INTR REQ, IF ANY
062632  004737  027454                  JSR     PC,WAITF
                                
062636                                  WTDSRA  #0,FLG                  ; CLEAR THE FLAGS AND INTERRUPT.
062636  012777  000012  120170          MOV     #0&^C17!SETFLG,@DSR
062644  004737  023422                  JSR     PC,IIRCHK               ; CHECK THE INTERNAL REGISTERS
062650                          2$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
062652  004737  026074                  JSR     PC,DPRESET              ; DO SOFT INITIALIZE.
```

```
                                        ;(4) SET SWITCH AND ENABLE, CHECK FOR PROPER INTERRUPT.

                                CSMT13: BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
                                        INITDP
                                        JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
                                        JSR     PC,ENAINT
                                        INTSET  JSS
                                        MOV     #^O177400!IOKCKIN!IOKJSS,INTMASK        ; PRIME FLAG, EXPECT JSS
                                        BISW3   CSCH.8,#WJSS!JSWE!JSWD,@DXR     ; ENABLE SWITCH INTERRUPT.
                                        MOV     #WJSS!JSWE!JSWD,-(SP)
                                        BIS     CSCH.8,(SP)
                                        MOV     (SP)+,@DXR
                                        JSR     PC,WAITF
                                        BISW3   CSCH.8,#WECC!SWCHON,@DYR        ; TURN THE SWITCH ON.
                                        MOV     #WECC!SWCHON,-(SP)
                                        BIS     CSCH.8,(SP)
                                        MOV     (SP)+,@DYR
                                        JSR     PC,WAITF
                                        MOV     #100,R2
                                10$:    TSTB    INTFLAG         ; DID WE INTERRUPT?
                                        BPL     11$                     ; BR IF YES.
                                        DEC     R2
                                        BGT     10$
                                11$:    TSTB    INTFLAG         ; DID WE GET GOOD INTERRUPT?
                                        BEQ     2$              ; BR IF YES
                                        MOV     #10$,R1         ; GET CPU PC.
                                        HRDERR  NOSWI,INTERR
                                        JMP     3$                      ; MIGHT AS WELL QUIT.

                                2$:

                                3$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
                                        JSR     PC,DPRESET      ; DO SOFT INIT.
```

```
                                ;(5) CHECK SWITCH INTERRUPT OPERATION FROM DISPLAY FILE.

063020                  CSMT14: BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
063022                          INITDP
063022  004737  025402                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
063026  004737  026114                JSR     PC,ENAINT               ; ENABLE INTERRUPTS.
063032                                BISW3   CSCH.6,#LDCP,CSMF3A     ; SET UP DISPLAY FILE TO LOAD CURSOR,
063032  012746  175000                MOV     #LDCP,-(SP)
063036  053716  065142                BIS     CSCH.6,(SP)
063042  012637  065230                MOV     (SP)+,CSMF3A
063046                                BISW3   CSCH.8,#CUIS,CSMF3B     ; ... AND SET SWITCH ENABLE
063046  012746  146013                MOV     #CUIS,-(SP)
063052  053716  065144                BIS     CSCH.8,(SP)
063056  012637  065236                MOV     (SP)+,CSMF3B
063062                                BISW3   CSCH.6,#LDECC!SWCHON,CSMF3C ; ... AND TURN ON SWITCH.
063062  012746  175402                MOV     #LDECC!SWCHON,-(SP)
063066  053716  065142                BIS     CSCH.6,(SP)
063072  012637  065240                MOV     (SP)+,CSMF3C
063076  112737  000377  026107        MOVB    #^O377,INTFLAG          ; PRIME THE INTERRUPT FLAG.
063104                                DPSTART #CSMF3A         ; START THE DPU.
063104  012777  065230  117720        MOV     #CSMF3A,@DPC    ; START THE DPU.
063112  004737  027454                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
063116  004737  027454                JSR     PC,WAITF                ; WAIT FOR STOP.
063122  017702  117704                MOV     @DPC,R2         ; SAVE THE DPC.
063126  105737  026107                TSTB    INTFLAG         ; TEST FOR GOOD INTERRUPT.
063132  001412                        BEQ     1$                      ; BR IF GOOD.
063134  012701  063116                MOV     #.-16,R1
063140                                HRDERR  NOSWI,INTERR

063160  020227  065252        1$:     CMP     R2,#CSMF3D              ; DID THE DPC HIT THE STOPN?
063164  103412                        BLO     2$                      ; BR IF NO -- OK
063166  012701  065244                MOV     #CSMF3C+4,R1            ; EXPECT A DPC PRIOR TO STOPN.
063172                                HRDERR  SWNOHLT,EXPREC

063212                  2$:     BISW3   #BIT15,CSMF3A+2,R1      ; EXPECT DXR TO HAVE X CURSOR.
063212  013701  065232                MOV     CSMF3A+2,R1
063216  052701  100000                BIS     #BIT15,R1
063222                                BISW3   #BIT15,CSMF3A+4,R3      ; EXPECT DYR TO HAVE Y CURSOR.
063222  013703  065234                MOV     CSMF3A+4,R3
063226  052703  100000                BIS     #BIT15,R3
063232  005737  065150                TST     CSRETRY
063236  001416                        BEQ     3$
063240  004737  065520                JSR     PC,RDDXY                ; GET MASKED DXR, DYR INTO R2,R4.
063244                                IFERRX2 SWNOCU,RLXYE
063244  020102                        CMP     R1,R2           ; 1ST ITEM OK?
063246  001002                        BNE     65$             ; NO.
063250  020304                        CMP     R3,R4           ; 2ND ITEM OK?
063252  001410                        BEQ     64$             ; YES.
063254                  65$:    HRDERR  SWNOCU,RLXYE,   ; NO.
063274                  64$:

063274  004737  026176        3$:     JSR     PC,DSBINT
063300                                ENDSEG          ;>>>>>>>>>>>>END SEGMENT
063302  004737  026074                JSR     PC,DPRESET              ; DO SOFT INIT.
```

```
                                        ; VERIFY THE CURSOR MATCH OPERATION.
                                        ;
063306  004737  026074          CSMT15: JSR     PC,DPRESET              ; DO SOFT INIT.
063312                                  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
063314  112737  000000  026106          MOVB    #0,INTMASK              ; DON'T EXPECT INTERRUPTS
063322                                  BISW3   CSCH.8,#CUIM!CUON,CSMF4B        ; INSERT CHANNEL # IN FILE.
063322  012746  146076                  MOV     #CUIM!CUON,-(SP)
063326  053716  065144                  BIS     CSCH.8,(SP)
063332  012637  065320                  MOV     (SP)+,CSMF4B
063336                                  BISW3   CSCH.6,#LDCP,CSMF4LC
063336  012746  175000                  MOV     #LDCP,-(SP)
063342  053716  065142                  BIS     CSCH.6,(SP)
063346  012637  065324                  MOV     (SP)+,CSMF4LC
063352                                  BISW3   CSCH.6,#LDECC!SWCHON,CSMF4C
063352  012746  175402                  MOV     #LDECC!SWCHON,-(SP)
063356  053716  065142                  BIS     CSCH.6,(SP)
063362  012637  065314                  MOV     (SP)+,CSMF4C
063366                                  BISW3   CSCH.8,#CURD!JSWE!JMWE!JCWE,CSMF4J
063366  012746  146152                  MOV     #CURD!JSWE!JMWE!JCWE,-(SP)
063372  053716  065144                  BIS     CSCH.8,(SP)
063376  012637  065346                  MOV     (SP)+,CSMF4J
063402  012737  000002  065422          MOV     #2,CSMF4F+2                     ; ASSUME NON-INTERLACED POSITION
063410  012737  020002  065440          MOV     #M!2,CSMF4G+2
063416  012737  020002  065456          MOV     #M!2,CSMF4H+2
063424  004737  026176                  JSR     PC,DSBINT
063430  112737  000202  026106          MOVB    #IOKCKIN!IOKJSM,INTMASK ; SAY MATCH INTR OK, WE WILL CHECK.
063436  032737  000001  027700          BIT     #BIT0,LOOPK     ; ARE WE IN AN ODD-NUMBERED ITERATION?
063444  001406                          BEQ     11$             ; BR IF NO.
063446                                  DPSTART #CSMF4V         ; YES --START THE SETUP FILE FOR VISUAL.
063446  012777  065254  117356          MOV     #CSMF4V,@DPC    ; START THE DPU.
063454  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
063460  000405                          BR      1$
063462                          11$:    DPSTART #CSMF4U         ; START TO PUT MEMORIES IN WRITE-ONLY MODE.
063462  012777  065270  117342          MOV     #CSMF4U,@DPC    ; START THE DPU.
063470  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
063474                          1$:     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.

063476  004737  026114                  JSR     PC,ENAINT               ; ENABLE INTERRUPTS.
063502  012737  065026  065024          MOV     #CSMPTB,CSMPY           ; CSMPY = TABLE POINTER FOR Y POSITIONS
063510  012737  065026  065022  2$:     MOV     #CSMPTB,CSMPX           ; CSMPX = TABLE POINTER FOR X POSITIONS
063516  017737  001302  065330          MOV     @CSMPY,CSMF4Y           ; SETUP Y CURSOR POSITION.
063524  017737  001272  065326  3$:     MOV     @CSMPX,CSMF4X           ; SETUP X CURSOR POSITION.
063532  005737  003230                  TST     INTLAC                  ; ARE WE IN INTERLACED MODE?
063536  001027                          BNE     4$                      ; BR IF YES -- POSITIONING IS OK.
063540  032737  000002  065330          BIT     #BIT1,CSMF4Y            ; NO -- ARE WE ON AN ODD LINE?
063546  001012                          BNE     30$                     ; BR IF YES -- CAN WRITE ABOVE, NOT BELOW
063550  012737  000004  065422          MOV     #4,CSMF4F+2             ; EVEN LINE -- SETUP FOR Y+4.
063556  012737  020004  065440          MOV     #M!4,CSMF4G+2           ; ... AND BACK DOWN BY 4
063564  012737  020002  065456          MOV     #M!2,CSMF4H+2           ; ... AND BELOW Y BY 2.
063572  000411                          BR      4$
063574  012737  000002  065422  30$:     MOV     #2,CSMF4F+2             ; ODD LINE -- SETUP FOR Y+2,
063602  012737  020002  065440          MOV     #M!2,CSMF4G+2           ; ... AND BACK DOWN TO Y,
063610  012737  020004  065456          MOV     #M!4,CSMF4H+2           ; ... BUT DOWN TO Y-4.

063616                          4$:     BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
063620  112737  000200  026106          MOVB    #IOKCKIN,INTMASK                ; SAY WE'LL CHECK UNEXP. INTR.
```

```
 6420 063626  013705  065150                      MOV     CSRETRY,R5
 6421 063632  013701  065326                      MOV     CSMF4X,R1
 6422 063636  013703  065330                      MOV     CSMF4Y,R3
 6423 063642  052701  100000                      BIS     #BIT15,R1
 6424 063646  052703  100000                      BIS     #BIT15,R3
 6425 063652                                      BISW3   CSCH.8,#WECC,@DYR       ; CLEAR THE SWITCH
      063652  012746  160000                      MOV     #WECC,-(SP)
      063656  053716  065144                      BIS     CSCH.8,(SP)
      063662  012677  117152                      MOV     (SP)+,@DYR
 6426 063666  004737  027454                      JSR     PC,WAITF
 6427 063672  112737  000377  026107  43$:       MOVB    #^O377,INTFLAG          ; PRIME THE INTERRUPT FLAG
 6428 063700                                      DPSTART #CSMF4D
      063700  012777  065334  117124              MOV     #CSMF4D,@DPC    ; START THE DPU.
      063706  004737  027454                      JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 6429 063712  017702  117114                      MOV     @DPC,R2
 6430 063716  105737  026107                      TSTB    INTFLAG         ; ANY INTERRUPTS?
 6431 063722  002414                              BLT     403$
 6432 063724  010146                              MOV     R1,-(SP)
 6433 063726  012701  063710                      MOV     #.-14.,R1
 6434 063732                                      HRDERR  IFAULT,INTERR
 6435 063752  012601                              MOV     (SP)+,R1
 6436 063754                      403$:
 6437 063754  004737  065520                      JSR     PC,RDDXY
 6438 063760  020102                              CMP     R1,R2
 6439 063762  001002                              BNE     44$
 6440 063764  020304                              CMP     R3,R4
 6441 063766  001414                              BEQ     45$
 6442 063770  005305                      44$:    DEC     R5
 6443 063772  100337                              BPL     43$
 6444 063774                                      HRDERR  CRWRE,RLXYE
 6445 064014  000137  064710                      JMP     8$      ; GO TRY THE NEXT COORDINATES.
 6446 064020                      45$:
 6447 064020  112737  000202  026106              MOVB    #IOKCKIN!IOKJSM,INTMASK ; SAY WE'LL CHECK FOR MATCH
 6448 064026  112737  000377  026107              MOVB    #^O377,INTFLAG          ; PRIME THE INTERRUPT FLAG.
 6449 064034                                      DPCONT
      064034  052777  000001  116770              BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
      064042  004737  027454                      JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 6450 064046  012701  000100                      MOV     #100,R1                 ; SETUP A WATCHDOG COUNTER.
 6451 064052  105737  026107              40$:    TSTB    INTFLAG         ; DID WE GET AN INTERRUPT?
 6452 064056  100002                              BPL     41$                     ; BR IF YES.
 6453 064060  005301                              DEC     R1
 6454 064062  003373                              BGT     40$
 6455 064064  017702  116742              41$:    MOV     @DPC,R2         ; GET DPC FOR ERROR.
 6456 064070  017737  116742  003170              MOV     @DXR,SDXR
 6457 064076  017737  116736  003172              MOV     @DYR,SDYR
 6458 064104  105737  026107                      TSTB    INTFLAG
 6459 064110  001417                              BEQ     5$                      ; WAS IT A GOOD INTERRUPT?
 6460 064112  002434                              BLT     50$                     ; BR IF NO INTERRUPT OCCURRED.
 6461 064114  010146                              MOV     R1,-(SP)
 6462 064116  012701  064052                      MOV     #40$,R1
 6463 064122                                      HRDERR  NOCMI,INTERR
 6464 064142  012601                              MOV     (SP)+,R1
 6465 064144  000137  064204                      JMP     50$
 6466
 6467 064150                      5$:
 6468 064150  027727  116656  065372              CMP     @DPC,#CSMF4K+2  ;DID WE GET TO THE STOPN (NO MATCH)?
 6469 064156  001040                              BNE     51$                     ; BR IF NO
```

```
                                                HRDERR  NOMST,PNTCOOR
                                                JMP     7$              ; GO SEE IF OTHER THINGS CHECK OUT.

                                        50$:    ; COME HERE IF NO MATCH INTERRUPT OCCURRED.
                                                CMP     R2,#CSMF4K              ; DID WE DO A MATCH STOP?
                                                BNE     501$                    ; BR IF NO
                                                HRDERR  CMNOMI,PNTCOOR  ;YES
                                                BR      51$                     ; GO CHECK OTHER MATCH DATA.
                                        501$:   HRDERR  NOMAT,PNTCOOR   ;NO MATCH AT ALL.
                                                JMP     7$                      ; GO SEE IF IT MATCHES ELSEWHERE.


                                        51$:    ;COME HERE IF WE THINK WE GOT A MATCH

                                                MOV     #CSMF4K,R1              ; EXPECT DPC TO BE JUST AFTER MATCH.
                                                MOV     #LVEC!ALL,R3            ; EXPECT DSR TO SHOW LVEC & ALL PIX BITS
                                                MOV     @DPC,R2         ; GET ACTUAL DPC.
                                                MOV     @DSR,R4         ; AND THE DSR.
                                                IFERRX2 MATPCS,EXPRC2
                                                CMP     R1,R2           ; 1ST ITEM OK?
                                                BNE     65$             ; NO.
                                                CMP     R3,R4           ; 2ND ITEM OK?
                                                BEQ     64$             ; YES.
                                        65$:    HRDERR  MATPCS,EXPRC2,  ; NO.
                                        64$:

                                                MOV     CSMF4X,R1               ; GET EXPECTED MATCH X
                                                MOV     CSMF4Y,R3               ; ... AND THE Y.
                                                MOV     @DXR,R2         ; GET DXR, SHOULD HOLD MATCH X
                                                MOV     @DYR,R4         ; GET DYR, SHOULD HOLD MATCH Y.
                                                IFERRX2 MATXYE,EXPRC2
                                                CMP     R1,R2           ; 1ST ITEM OK?
                                                BNE     67$             ; NO.
                                                CMP     R3,R4           ; 2ND ITEM OK?
                                                BEQ     66$             ; YES.
                                        67$:    HRDERR  MATXYE,EXPRC2,  ; NO.
                                        66$:

                                                MOV     #RSTPOS,@DXR            ; RESTORE GRAPHIC POSITION TO DXR, DYR
                                                JSR     PC,WAITF                        ; WAIT FOR STOP.
                                                ADD     #4,R1                   ; X SHOULD BE X(MATCH)+4
                                                MOV     @DXR,R2         ; GET ACTUAL DXR
                                                MOV     @DYR,R4         ; ... AND DYR.
                                                IFERRX2 MATPOS,EXPRC2   ;
                                                CMP     R1,R2           ; 1ST ITEM OK?
                                                BNE     69$             ; NO.
                                                CMP     R3,R4           ; 2ND ITEM OK?
                                                BEQ     68$             ; YES.
                                        69$:    HRDERR  MATPOS,EXPRC2,  ; NO.
                                        68$:

                                        6$:     MOV     #LVEC,R1                ; GET OP-CODE EXPECTED FOR FLAGS.
                                                RDDSRA  FLG,R2                  ; GET FLAGS.
                                                MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
                                                MOV     @DSR,R2
                                                IFERROR MATFLE,FLGER,7$
                                                CMP     R1,R2           ; OK?
```

```
        064476  001410                          BEQ     7$                      ; YES.
        064500                                  HRDERR  MATFLE,FLGER,           ; NO.
        064520                          7$:
  6506  064520  112737  000377  026107  7$:     MOVB    #^O377,INTFLAG                  ; PRIME THE INTERRUPT FLAG.
  6507  064526                                  DPSTART #CSMF4E                 ; START AGAIN TO SEE IF WE GET BAD MATCH
        064526  012777  065372  116276          MOV     #CSMF4E,@DPC            ; START THE DPU.
        064534  004737  027454                  JSR     PC,WAITF                ; WAIT FOR DISPLAY STOP.
  6508  064540  004737  027454                  JSR     PC,WAITF
  6509  064544  105737  026107                  TSTB    INTFLAG                 ; DID WE GET AN INTERRUPT?
  6510  064550  002421                          BLT     17$                             ; BR IF NO
  6511  064552  001446                          BEQ     18$                             ; BR IF GOT "GOOD" INTERRUPT.
  6512  064554  017702  116252                  MOV     @DPC,R2
  6513  064560  010146                          MOV     R1,-(SP)
  6514  064562  012701  064542                  MOV     #.-16.,R1
  6515  064566                                  HRDERR  MATUI,INTERR
  6516  064606  012601                          MOV     (SP)+,R1
  6517  064610  000137  064710                  JMP     8$
  6518
  6519  064614                          17$:
  6520  064614  012701  065472                  MOV     #CSMF4I+2,R1                    ; GET EXPECT STOP DPC.
  6521  064620  012703  173200                  MOV     #STOPN!GOLD,R3                  ; DSR SHOULD HAVE STOP + GOLD COLOR.
  6522  064624  017702  116202                  MOV     @DPC,R2                 ; GET THE DPC.
  6523  064630  017704  116200                  MOV     @DSR,R4                 ; GET THE DSR
  6524  064634                                  IFERRX2 MENDPCS,EXPRC2
        064634  020102                          CMP     R1,R2                   ; 1ST ITEM OK?
        064636  001002                          BNE     71$                     ; NO.
        064640  020304                          CMP     R3,R4                   ; 2ND ITEM OK?
        064642  001410                          BEQ     70$                     ; YES.
        064644                          71$:    HRDERR  MENDPCS,EXPRC2,         ; NO.
        064664                          70$:
  6525  064664  000137  064710                  JMP     8$
  6526
  6527  064670                          18$:    HRDERR  MATUI,PNTCOOR
  6528
  6529  064710                          8$:     ENDSEG          ;>>>>>>>>>>>>END SEGMENT
  6530  064712  062737  000002  065022          ADD     #2,CSMPX                        ; BUMP POINTER TO X DATA.
  6531  064720  023727  065022  065136          CMP     CSMPX,#CSMPTE                   ; IS X POINTER AT END OF TABLE?
  6532  064726  103002                          BHIS    9$                              ; BR IF YES -- GO DO NEXT Y.
  6533  064730  000137  063524                  JMP     3$                              ; LOOP BACK.
  6534
  6535  064734  062737  000002  065024  9$:     ADD     #2,CSMPY
  6536  064742  023727  065024  065136          CMP     CSMPY,#CSMPTE                   ; IS Y POINTER AT END OF TABLE?
  6537  064750  103002                          BHIS    10$                             ; BR IF YES -- WE'RE DONE.
  6538  064752  000137  063510                  JMP     2$                              ; NO -- GO DO NEXT Y.
  6539  064756                          10$:
  6540
  6541  064756                          CSMNEXT:
  6542  064756  005237  065140                  INC     CSCHAN                          ; INCREMENT TO NEXT CHANNEL
  6543  064762  023727  065140  000004          CMP     CSCHAN,#4                       ; DID WE DO ALL AVAILABLE CHANNELS?
  6544  064770  002002                          BGE     1$                              ; BR IF YES -- GO TO EXIT.
  6545  064772  000137  060210                  JMP     CSMLOOP                         ; NO. LOOP BACK TO TEST NEXT CHANNEL.
  6546  064776                          1$:
  6547  064776                          CSMEND: COMEND  CSMITR
        064776  004737  027646                  JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
        065002  103401                          BCS     64$
        065004  000402                          BR      65$
```

```
        065006  000137  060204          64$:    JMP     CSMITR

        065012  004737  030072          65$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
  6548  065016                          CSMXIT: EXIT    TST
  6549
  6550
  6551
  6552                                  ; TABLE OF MATCH POINTS -- ALL VALUES ARE USED FOR X AND Y:
  6553
  6554  065022  000000                  CSMPX:  .WORD   0
  6555  065024  000000                  CSMPY:  .WORD   0
  6556
  6557  065026  000000  000002  000004  CSMPTB: .WORD   0,BIT1,BIT2,BIT3,BIT4          ; FIRST, FLOATING 1 ON 0'S, BITS 2
  6558  065040  000040  000100  000200          .WORD   BIT5,BIT6,BIT7,BIT8,BIT9        ; ... THRU 9.
  6559          001776                          A1=^O1776               ;ALL 1'S FOR POSITION
  6560  065052  001776  001774  001772          .WORD   A1,A1&^CBIT1,A1&^CBIT2,A1&^CBIT3,A1&^CBIT4      ; THEN, FLOATING 0 ON 1'S
  6561  065064  001736  001676  001576          .WORD   A1&^CBIT5,A1&^CBIT6,A1&^CBIT7,A1&^CBIT8,A1&^CBIT9
  6562  065076  001252  000524  001462          .WORD   1252,524,1462,636       ; ALTERNATING BITS AND PAIRS.
  6563  065106  000006  000016  000036          .WORD   6,16,36,76,176,376
  6564  065122  001770  001760  001740          .WORD   1770,1760,1740,1700,1600,1400
  6565  065136  000000                  CSMPTE: .WORD   0
  6566
  6567  065140  000000                  CSCHAN: .WORD   0
  6568  065142  000000                  CSCH.6: .WORD   0
  6569  065144  000000                  CSCH.8: .WORD   0
  6570  065146  000000                  CSCH.11:        .WORD   0
  6571
  6572  065150  000002                  CSRETRY:        .WORD   2
  6573
  6574                                  ; EXTENDED PRINTOUT FOR SYNC CHANNEL:
  6575  065152                          PRSYCH: PRINTX  #FSYCHAN,CSCHAN
  6576  065176  000207                          RTS     PC
  6577
  6578                                  ;DISPLAY FILES FOR CURSOR/SWITCH/MATCH TEST --
  6579
  6580  065200  175000  000000  000000  CSMF1A: .WORD   LDCP,0,0        ;LOAD CURSOR POSITION REGISTERS
  6581  065206  146100                  CSMF1B: .WORD   CURD            ;READ THEM BACK
  6582  065210  173000                          .WORD   STOPN
  6583
  6584  065212  146052                  CSMF2A: .WORD   CUIOFF!CUOFF    ; CLEAR ALL ENABLES
  6585  065214  146012                  CSMF2B: .WORD   CUIOFF          ; DO GENERAL WRITE
  6586  065216  146100  173000          CSMF2C: .WORD   CURD,STOPN      ; READ CURSOR (ENABLES COME TOO), THEN STOP
  6587  065222  146077                  CSMF2D: .WORD   CUON!^O77       ; SET ALL ENABLES
  6588  065224  160000  065214                  .WORD   DJMP,CSMF2B     ; THEN GO DO GENERAL WRITE AND READ.
  6589
  6590  065230  175000  001332  002464  CSMF3A: .WORD   LDCP,^O1332,^O2464      ; LOAD CURSOR WITH SOME DATA
  6591  065236  146013                  CSMF3B: .WORD   CUIS            ; ENABLE SWITCH INTERRUPT
  6592  065240  175402                  CSMF3C: .WORD   LDECC!SWCHON    ; TURN ON THE SOFT SWITCH
  6593  065242  164000  164000  164000          .WORD   DNOP,DNOP,DNOP,DNOP  ; DO SOME NO-OPS -- SHOULD STOP
  6594  065252  173000                  CSMF3D: .WORD   STOPN            ; STOP -- SWITCH DIDN'T WORK
  6595
  6596                                  ;DISPLAY FILE FOR CURSOR MATCH TEST:
  6597
  6598  065254  176074  176474  177074  CSMF4V: .WORD   RDWRT!CH0,RDWRT!CH1,RDWRT!CH2,RDWRT!CH3  ; MEMORY SETUP, R/W.
  6599  065264  160000  065300                  .WORD   DJMP,CSMF4A
  6600
  6601  065270  176034  176434  177034  CSMF4U: .WORD   WRT!CH0,WRT!CH1,WRT!CH2,WRT!CH3 ; MEMORY SETUP, WRT-ONLY.
```

```
6602 065300  170140                  CSMF4A: .WORD   CLRMEM           ; CLEAR THE MEMORIES.
6603 065302  160001  065320                  .WORD   DJMS,CSMF4B              ; GO TURN ON CURSOR AND MATCH ENABLE
6604 065306  160001  065314  173000          .WORD   DJMS,CSMF4C,STOPN        ; TURN SWITCH ON, AND STOP
6605 065314  175402  165000          CSMF4C: .WORD   LDECC!SWCHON,DPOP        ; TURN SWITCH ON, AND RETURN.
6606
6607 065320  146076  165000          CSMF4B: .WORD   CUON!CUIM,DPOP           ; ENABLE CURSOR AND MATCH.
6608 065324  175000                  CSMF4LC: .WORD  LDCP             ; LOAD CURSOR POSITION.
6609 065326  000000                  CSMF4X:  .WORD  0                ; X
6610 065330  000000                  CSMF4Y:  .WORD  0                ; Y
6611 065332  165000                          .WORD   DPOP                     ; RETURN.
6612
6613 065334  114000  160001  065326  CSMF4D: .WORD   APNT,DJMS,CSMF4X         ; SET MATCH POSITION
6614 065342  160001  065324                  .WORD   DJMS,CSMF4LC             ; LOAD CURSOR POSITION.
6615 065346  146152  173000          CSMF4J: .WORD   CURD!JSWE!JMWE!JCWE,STOPN                ; STOP FOR CHECK.
6616 065352  160001  065320                  .WORD   DJMS,CSMF4B              ; GO TURN ON MATCH ENABLE
6617 065356  160001  065314                  .WORD   DJMS,CSMF4C              ; GO TURN ON SWITCH
6618 065362  113774                          .WORD   LVEC!ALL                 ; SET TO WHITE.
6619 065364  040004  000000                  .WORD   I!4,0                    ; DRAW, SHOULD MATCH
6620 065370  173000                  CSMF4K: .WORD   STOPN            ; ... WITH DPC AT F4K
6621 065372  114000  160001  065326  CSMF4E: .WORD   APNT,DJMS,CSMF4X         ; RESTART. SET POINT
6622 065400  113600                          .WORD   LVEC!YELLOW              ; CHANGE COLOR
6623 065402  000002  000000                  .WORD   2,0                      ; ... THEN MOVE TO X+2, Y.
6624 065406  041776  000000  061776          .WORD   I!MAXX,0,I!M!MAXX,0      ; DRAW OUT TO RIGHT AND BACK.
6625 065416  113300                          .WORD   LVEC!VIOLET
6626 065420  020002  000002          CSMF4F:  .WORD M!2,2                     ; GO TO ABOVE MATCH POINT
6627 065424  040000  001776  040000          .WORD   I!0,MAXY,I!0,M!MAXY      ; ... AND DRAW UP AND BACK.
6628 065434  113500                          .WORD   LVEC!EGGBL
6629 065436  020002  020002          CSMF4G: .WORD   M!2,M!2
6630 065442  061776  000000  041776          .WORD   I!M!MAXX,0,I!MAXX,0      ; DRAW LEFT AND BACK.
6631 065452  113200                          .WORD   LVEC!GOLD
6632 065454  000002  020002          CSMF4H: .WORD   2,M!2
6633 065460  040000  021776  040000          .WORD   I!0,M!MAXY,I!0,MAXY
6634 065470  173000  173000          CSMF4I: .WORD   STOPN,STOPN
6635
6636                                 ;SUBROUTINES:
6637                                 ;GET CURSOR COORDINATES, AND READ MASKED DXR, DYR INTO R2, R4 --
6638 065474  004737  027454          GETCURS: JSR    PC,WAITF                         ; WAIT FOR STOP BIT.
6639 065500                                   BISW3  CSCH.8,#WRTJSS!BIT6,@DXR         ; RETRIEVE CURSOR FROM SELECTED CHAN.
     065500  012746  042100                  MOV     #WRTJSS!BIT6,-(SP)
     065504  053716  065144                  BIS     CSCH.8,(SP)
     065510  012677  115322                  MOV     (SP)+,@DXR
6640 065514  004737  027454                  JSR     PC,WAITF                 ; WAIT FOR STOP.
6641 065520                          RDDXY:  BICW3   #^C<BIT15!^O3777>,@DXR,R2  ; DXR TO R2, LEAVE FLAG & COORD.
     065520  017702  115312                  MOV     @DXR,R2
     065524  042702  074000                  BIC     #^C<BIT15!^O3777>,R2
6642 065530                                  BICW3   #^C<BIT15!^O3777>,@DYR,R4  ; DYR TO R4, LEAVE FLAG & COORD.
     065530  017704  115304                  MOV     @DYR,R4
     065534  042704  074000                  BIC     #^C<BIT15!^O3777>,R4
6643 065540  000207                          RTS     PC
6644
6645                                 ;GET JOYSTICK ENABLES-
6646 065542  004737  027454          GETJSE: JSR     PC,WAITF                 ; WAIT FOR STOP.
6647 065546                                  BISW3   CSCH.8,#WRTJSS!BIT6,@DXR         ; READ CURSOR
     065546  012746  042100                  MOV     #WRTJSS!BIT6,-(SP)
     065552  053716  065144                  BIS     CSCH.8,(SP)
     065556  012677  115254                  MOV     (SP)+,@DXR
6648 065562  004737  027454                  JSR     PC,WAITF
```

```
 6649 065566                            RDJSE:  BICW3   #^C<BIT15!JSSIES!JSMIES>,@DXR,R2 ;DXR TO R2, LEAVE ENABLES
      065566  017702  115244                    MOV     @DXR,R2
      065572  042702  063777                    BIC     #^C<BIT15!JSSIES!JSMIES>,R2
 6650 065576                                    BICW3   #^C<BIT15!CHIE>,@DYR,R4 ;DYR TO R4.
      065576  017704  115236                    MOV     @DYR,R4
      065602  042704  067777                    BIC     #^C<BIT15!CHIE>,R4
 6651 065606  000207                            RTS     PC
 6652
 6653 065610                                    END.TEST

                                        ;*******************************************************************************
                                        ;*
                                        ;*      END TEST 23
                                        ;*
                                        ;*******************************************************************************
```

```
6655
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 24   -   GRAPH-HISTOGRAM X
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 10.)

6656 065664  005737  002502                    TST     DPUMOD          ;DPU ONLY ???
6657 065670  001402                            BEQ     GHXTST          ;NO, EXECUTE.
6658 065672                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
6659                                    ;
6660                                    ; PLOT GRAPH X = 1600 TO 200 (DELTA 30), FOR Y = 76 TO 2.
6661                                    ; WHEN DONE, REPEAT IN HISTO MODE, WITH BASE = X-20.
6662                                    ; DO THE SAME FOR CONNECTED-GRAPH AND FILLED HISTOGRAM, BUT AT LOWER
6663                                    ; INTENSITY.  RUN THE MEMORIES IN "WRITE-1'S-ONLY" MODE TO SEE STUFF.
6664                                    ; REPEAT EACH INCREMENT 10 TIMES AFTER QUICK PASS.
6665                                    ;
6666 065676  004737  027702            GHXTST: JSR     PC,TSTGO        ;TITLE
6667 065702  012737  066300  022300            MOV     #20$,EXTA       ;SET ERROR EXTENSION ADDRESS.
6668 065710  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
6669 065714  004737  026640                    JSR     PC,CLRRW1       ;INIT IN R/W-1'S-ONLY MODE.
6670 065720  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
6671 065726  012737  123774  066350            MOV     #GHX!ALL,XA-2   ;SET G/H-X MODE WITH MAX INTENSITY
6672 065734  012737  000000  066332            MOV     #0,GHXTYP       ;START WITH GRAPH FUNCTION
6673 065742  013737  066332  066352  8$:       MOV     GHXTYP,XA       ;SETUP THE FUNCTION IN FILE
6674 065750  052737  001600  066352            BIS     #1600,XA        ;... AND SET IN THE X AMPLITUDE
6675 065756  012737  001560  066336            MOV     #1560,GHXT+2    ;INIT BASE = X-20
6676 065764  012737  174176  066340            MOV     #GYI!76,YI      ;INIT Y INCREMENT = 76.
6677 065772  012704  001600                    MOV     #1600,R4        ;INIT EXPECTED X.
6678
6679
6680 065776                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6681
6682 066000  013701  066340                    MOV     YI,R1
6683 066004  042701  177700                    BIC     #^C77,R1        ;SET CURRENT INCR IN R1.
6684 066010  005005                    2$:     CLR     R5              ;INIT EXP Y.
6685 066012  052737  010000  066352            BIS     #GHIINH,XA
6686 066020  012777  066334  115004            MOV     #GHXT,@DPC      ;START NEW INCREMENT.
6687 066026  000402                            SKP2
6688 066030  005277  114776            3$:     INC     @DPC            ;RESUME, PLOT NEXT POINT.
6689 066034  004737  027454                    JSR     PC,WAITF
6690 066040  032737  010000  066352            BIT     #GHIINH,XA
6691 066046  001001                            BNE     13$
6692 066050  060105                            ADD     R1,R5           ;EXP Y = Y + INCR.
6693 066052  027704  114760            13$:    CMP     @DXR,R4         ;CHECK RETURNED X = X AMP...
6694 066056  001003                            BNE     4$
6695 066060  027705  114754                    CMP     @DYR,R5         ;...AND Y = Y + INCR.
6696 066064  001410                            BEQ     5$
6697 066066                            4$:     HRDERR  GHXF,XYERR      ;GHX FAILS.
6698 066106  042737  010000  066352  5$:       BIC     #GHIINH,XA
6699 066114  020527  001776                    CMP     R5,#MAXY        ;DONE THIS INCREMENT ??
6700 066120  103743                            BLO     3$              ;NOT YET, CONTINUE.
6701 066122                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6702 066124                                    LOOPTO  1$              ; REPEAT...
     066124  004737  027646                    JSR     PC,LOOP
```

```
066130  103722                          BCS     1$
6703
6704
6705 066132  013737  003234  027700      MOV     TEMP2,LOOPK     ;RESET LOOPER.
6706 066140  023727  066340  174102      CMP     YI,#GYI!2
6707 066146  001414                      BEQ     6$              ;BR IF LAST INCR DONE.
6708 066150  162737  000002  066340      SUB     #2,YI           ;OTHERWISE, SET NEXT INCR...
6709 066156  162737  000030  066352      SUB     #30,XA          ;...AMPLITUDE...
6710 066164  162704  000030              SUB     #30,R4
6711 066170  162737  000030  066336      SUB     #30,GHXT+2      ;...AND BASE LINE.
6712 066176  000677                      BR      1$              ;DO NEXT SET.
6713
6714 066200  005737  066332      6$:     TST     GHXTYP          ;JUST DID GRAPH?
6715 066204  001004                      BNE     30$             ;...BR AHEAD IF NOT.
6716 066206  012737  040000  066332      MOV     #HST,GHXTYP     ;YES. SET HISTO MODE
6717 066214  000652                      BR      8$              ;...AND GO DO IT.
6718 066216  023727  066332  060000  30$:     CMP     GHXTYP,#FHST    ;JUST DID FILLED-HISTO?
6719 066224  001417                      BEQ     10$             ;DONE IF YES.
6720 066226  162737  000400  066350      SUB     #BIT8,XA-2      ;IF NOT, DECREMENT INTENSITY.
6721 066234  023727  066332  040000      CMP     GHXTYP,#HST     ;JUST DID HISTO.?
6722 066242  001004                      BNE     31$             ;...BR AHEAD IF NOT.
6723 066244  012737  020000  066332      MOV     #CGRPH,GHXTYP   ;IF YES, NOW DO CONN-GRAPH
6724 066252  000633                      BR      8$              ;...
6725 066254  012737  060000  066332  31$:     MOV     #FHST,GHXTYP    ;DID C-GRPH, NOW DO F-HIST
6726 066262  000627                      BR      8$              ;...
6727
6728 066264  004737  027514      10$:    JSR     PC,PAUSE1
6729 066270  004737  030072              JSR     PC,TSTEND
6730 066274                              EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
6731
6732                                     ; ON ERROR, SHOW X AMPLITUDE AND Y INCREMENT.
6733                                     ;
6734 066300                      20$:    PRINTX  #GHXYX,XA,YI    ;ERROR EXTENSION.
6735 066330  000207                      RTS     PC
6736
6737                                     ;DISPLAY CODE FOR GRAPH/HISTO X TEST.
6738                                     ;
6739 066332  000000              GHXTYP: 0               ;FUNCTION TYPE
6740 066334  150000  000000      GHXT:   SETHB,0         ;SET BASE LINE.
6741 066340  174100              YI:     GYI!0           ;SET Y INCR.
6742 066342  114000                      APNT
6743 066344  000000  000000              0,0
6744 066350  123774                      GHX!ALL
6745 066352  000000              XA:     0               ;X AMPLITUDE (BIT 14 = HST MODE).
6746 066354  173000                      STOPN
6747 066356  160000  066350              DJMP,XA-2
6748
6749 066362                              END.TEST

                                    ;*******************************************************************************
                                    ;*
                                    ;*      END TEST 24
                                    ;*
                                    ;*******************************************************************************
```

```
6751
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 25   -   GRAPH-HISTOGRAM Y
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 10.)

6752 066436  005737  002502                    TST     DPUMOD          ;DPU ONLY ??
6753 066442  001402                            BEQ     GHYTST          ;NO, EXECUTE.
6754 066444                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
6755                                    ;
6756                                    ; PLOT GRAPH Y = 1600 TO 200 (DELTA 30), FOR X = 76 TO 2.
6757                                    ; WHEN DONE, REPEAT IN HISTO MODE, WITH BASE = Y-20.
6758                                    ; DO THE SAME FOR CONNECTED-GRAPH AND FILLED HISTOGRAM, BUT AT LOWER
6759                                    ; INTENSITY.  RUN THE MEMORIES IN "WRITE-1'S-ONLY" MODE TO SEE STUFF.
6760                                    ; REPEAT EACH INCREMENT 10 TIMES AFTER QUICK PASS.
6761                                    ;
6762 066450  004737  027702            GHYTST: JSR     PC,TSTGO        ;TITLE
6763 066454  012737  067052  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
6764 066462  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
6765 066466  004737  026640                    JSR     PC,CLRRW1       ;INIT IN R/W-1'S-ONLY MODE.
6766 066472  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
6767 066500  012737  127774  067122            MOV     #GHY!ALL,YA-2   ;SET G/H-Y MODE WITH MAX INTENSITY
6768 066506  012737  000000  067104            MOV     #0,GHYTYP       ;START WITH GRAPH FUNCTION
6769 066514  013737  067104  067124    8$:     MOV     GHYTYP,YA       ;SETUP THE FUNCTION IN FILE
6770 066522  052737  001600  067124            BIS     #1600,YA        ;... AND SET IN THE Y AMPLITUDE
6771 066530  012737  001560  067110            MOV     #1560,GHYT+2    ;INIT BASE LINE = Y-20.
6772 066536  012737  174176  067112            MOV     #GXI!76,XI      ;INIT X INCREMENT = 76.
6773 066544  012705  001600                    MOV     #1600,R5        ;INIT EXP Y.
6774
6775
6776 066550                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6777
6778 066552  013701  067112                    MOV     XI,R1
6779 066556  042701  177700                    BIC     #^C77,R1        ;SET CURRENT INCR IN R1.
6780 066562  005004                    2$:     CLR     R4              ;INIT EXP X...
6781 066564  052737  010000  067124            BIS     #GHIINH,YA
6782 066572  012777  067106  114232            MOV     #GHYT,@DPC      ;START NEW INCREMENT.
6783 066600  000402                            SKP2
6784 066602  005277  114224            3$:     INC     @DPC            ;RESUME, PLOT NEXT POINT.
6785 066606  004737  027454                    JSR     PC,WAITF
6786 066612  032737  010000  067124            BIT     #GHIINH,YA
6787 066620  001001                            BNE     13$
6788 066622  060104                            ADD     R1,R4           ;EXP X = X + INCR.
6789 066624  027704  114206            13$:    CMP     @DXR,R4         ;CHECK RETURNED X = X + INCR...
6790 066630  001003                            BNE     4$
6791 066632  027705  114202                    CMP     @DYR,R5         ;...AND Y = Y AMPL.
6792 066636  001410                            BEQ     5$
6793 066640                            4$:     HRDERR  GHYF,XYERR      ;GHY FAILS.
6794 066660  042737  010000  067124    5$:     BIC     #GHIINH,YA
6795 066666  020427  001776                    CMP     R4,#MAXX        ;DONE THIS INCREMENT ??
6796 066672  103743                            BLO     3$              ;NOT YET, CONTINUE.
6797 066674                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6798 066676                                    LOOPTO  1$              ; REPEAT...
     066676  004737  027646                    JSR     PC,LOOP
```

```
      066702  103722                        BCS     1$
 6799
 6800
 6801 066704  013737  003234  027700        MOV     TEMP2,LOOPK     ;RESET LOOPER.
 6802 066712  023727  067112  174102        CMP     XI,#GXI!2
 6803 066720  001414                        BEQ     6$              ;BR IF LAST INCR DONE.
 6804 066722  162737  000002  067112        SUB     #2,XI           ;OTHERWISE, SET NEXT INCR...
 6805 066730  162737  000030  067124        SUB     #30,YA          ;...AMPLITUDE...
 6806 066736  162705  000030                SUB     #30,R5
 6807 066742  162737  000030  067110        SUB     #30,GHYT+2      ;...AND BASE LINE.
 6808 066750  000677                        BR      1$              ;DO NEXT SET.
 6809
 6810 066752  005737  067104        6$:     TST     GHYTYP          ;JUST DID GRAPH?
 6811 066756  001004                        BNE     30$             ;...BR AHEAD IF NOT.
 6812 066760  012737  040000  067104        MOV     #HST,GHYTYP     ;YES. SET HISTO MODE
 6813 066766  000652                        BR      8$              ;...AND GO DO IT.
 6814 066770  023727  067104  060000  30$:  CMP     GHYTYP,#FHST    ;JUST DID FILLED-HISTO?
 6815 066776  001417                        BEQ     10$             ;DONE IF YES.
 6816 067000  162737  000400  067122        SUB     #BIT8,YA-2      ;IF NOT, DECREMENT INTENSITY.
 6817 067006  023727  067104  040000        CMP     GHYTYP,#HST     ;JUST DID HISTO.?
 6818 067014  001004                        BNE     31$             ;...BR AHEAD IF NOT.
 6819 067016  012737  020000  067104        MOV     #CGRPH,GHYTYP   ;IF YES, NOW DO CONN-GRAPH
 6820 067024  000633                        BR      8$              ;...
 6821 067026  012737  060000  067104  31$:  MOV     #FHST,GHYTYP    ;DID C-GRPH, NOW DO F-HIST
 6822 067034  000627                        BR      8$              ;...
 6823
 6824 067036  004737  027514        10$:    JSR     PC,PAUSE1
 6825 067042  004737  030072                JSR     PC,TSTEND
 6826 067046                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
 6827                                ;
 6828                                ; ON ERROR, SHOW Y AMPLITUDE AND X INCREMENT.
 6829                                ;
 6830 067052                        20$:    PRINTX  #GHXYX,YA,XI    ; ERROR EXTENSION.
 6831 067102  000207                        RTS     PC
 6832                                ;
 6833                                ;DISPLAY CODE FOR GRAPH/HISTO Y TEST.
 6834                                ;
 6835 067104  000000                GHYTYP: 0                       ;FUNCTION TYPE
 6836 067106  150000  000000        GHYT:   SETHB,0                 ;SET BASE LINE.
 6837 067112  174100                XI:     GXI!0                   ;SET X INCR.
 6838 067114  114000                        APNT
 6839 067116  000000  000000                0,0
 6840 067122  127774                        GHY!ALL
 6841 067124  000000                YA:     0                       ;Y AMPLITUDE (BIT 14 = HST MODE).
 6842 067126  173000                        STOPN
 6843 067130  160000  067122                DJMP,YA-2
 6844
 6845 067134                                END.TEST

                                ;*************************************************************************
                                ;*
                                ;*      END TEST 25
                                ;*
                                ;*************************************************************************
```

```
6847
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 26   -   BIT MAP (1)
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 10.)

6848 067202  005737  002502                    TST     DPUMOD          ;DPU ONLY ??
6849 067206  001402                            BEQ     BM1TST          ;NO, EXECUTE.
6850 067210                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
6851                                    ;
6852                                    ; EXECUTE BIT MAP MODE 1 WITH 4 AND 8 BIT PIXELS FOR ALL
6853                                    ; POSSIBLE RUN LENGTHS (0 THRU 777).  TEST FINAL X/Y FOR EACH.
6854                                    ; IF VISUAL, DISPLAYS A WEDGE STARTING AT 0,0 WITH EVERY
6855                                    ; OTHER PIXEL (IN X) TURNED OFF.
6856                                    ; REPEAT AFTER QUICK AS BEFORE.
6857                                    ; PIXEL BUFFER SIZE = 256. WORDS (512. 8 BIT PIXELS MAX).
6858                                    ;
6859                                    ;       IN THE NEW DPU, A COUNT OF 0 DOES 512. PIXELS, SO IT'S LAST.
6860 067214  004737  027702            BM1TST: JSR     PC,TSTGO        ; TITLE.
6861 067220  012737  067506  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
6862 067226  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
6863 067232  013737  027700  003234            MOV     LOOPK,TEMP2     ; SAVE A COPY OF THE LOOPER.
6864 067240  012700  000377                    MOV     #377,R0         ; SET PIXEL DATA PATTERN...
6865 067244  004737  027014                    JSR     PC,FILL1        ;...AND FILL THE BUFFER.
6866                                                                    ; 2 ON, 2 OFF IN 4 BIT MODE.
6867                                                                    ; 1 ON, 1 OFF IN 8 BIT MODE.
6868
6869 067250  012777  000001  113556            MOV     #SELPCS,@DSR    ;SAVE PCSAVE...
6870 067256  017737  113552  003146            MOV     @DSR,IPCSAV     ;
6871 067264  012701  136001                    MOV     #BM14+1,R1      ; 1ST OPCODE W/ COUNT=1.
6872 067270  013737  003240  067550            MOV     FREE,BM1PIX     ; BUFFER ADDRESS.
6873 067276  004737  026650            1$:     JSR     PC,CLRRW        ; INIT IN R/W MODE.
6874 067302  012704  000002                    MOV     #2,R4           ; INIT EXPECTED X...
6875 067306  005005                            CLR     R5              ;...AND Y.
6876 067310  010137  067546            2$:     MOV     R1,BM1PIX-2     ; SET 1ST/NEXT OPCODE...
6877 067314  010537  067544                    MOV     R5,BM1Y         ;...AND Y CO-ORD.
6878
6879 067320                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6880
6881 067322  012777  067532  113502            MOV     #BM1,@DPC       ;START, XCT BIT MAP MODE.
6882 067330  004737  027454                    JSR     PC,WAITF
6883 067334  027704  113476                    CMP     @DXR,R4         ;CHECK DXR = R4...
6884 067340  001003                            BNE     3$
6885 067342  027705  113472                    CMP     @DYR,R5
6886 067346  001410                            BEQ     4$
6887 067350                            3$:     HRDERR  BMF,XYERR       ;BM1 FAILS.
6888 067370                            4$:     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6889
6890 067372                                    LOOPTO  2$              ; REPEAT...
     067372  004737  027646                    JSR     PC,LOOP
     067376  103744                            BCS     2$
6891
6892 067400  013737  003234  027700            MOV     TEMP2,LOOPK     ;RESET LOOPER.
6893 067406  032701  000777                    BIT     #777,R1         ;DONE 512. PIXELS?
```

```
 6894 067412  001416                        BEQ     5$              ;...IF YES, GO SEE IF ALL DONE.
 6895 067414  005201                        INC     R1              ; NO, INCREMENT RUN LENGTH.
 6896 067416  062704  000002                ADD     #2,R4           ; NO, INCR EXPECTED X...
 6897 067422  062705  000002                ADD     #2,R5           ;...AND Y...
 6898 067426  032701  000777                BIT     #777,R1         ;COUNT OVERFLOW?
 6899 067432  001326                        BNE     2$              ;CONTINUE IF NOT.
 6900 067434  020127  137000                CMP     R1,#BM18        ;INTO BM18?
 6901 067440  001011                        BNE     6$              ;BR IF NOT.
 6902 067442  012701  136000                MOV     #BM14,R1        ;IF YES, DO BM14 W/ COUNT=0
 6903 067446  000720                        BR      2$              ;...AND CONTINUE.
 6904 067450  020127  137000          5$:   CMP     R1,#BM18        ;HERE IF ACTUAL COUNT=0.
 6905 067454  001406                        BEQ     10$             ;DONE IF ON BM18.
 6906 067456  012701  137001                MOV     #BM18+1,R1      ;GO TO BM18
 6907 067462  000705                        BR      1$              ;...AND CONTUNUE.
 6908 067464  012701  137000          6$:   MOV     #BM18+0,R1      ;DO BM1.8 W/ COUNT=0
 6909 067470  000707                        BR      2$              ;...AND CONTUNUE.
 6910
 6911 067472  004737  027514          10$:  JSR     PC,PAUSE1
 6912 067476  004737  030072                JSR     PC,TSTEND
 6913 067502                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
 6914                                ;
 6915                                ; ON ERROR, SHOW THE FAILING BIT MAP OPCODE.
 6916                                ;
 6917 067506                        20$:    PRINTX  #BMX,R1         ; EXTENSION SHOWS BAD OPCODE.
 6918 067530  000207                        RTS     PC
 6919                                ;
 6920                                ; DISPLAY CODE FOR BIT MAP MODE 1.
 6921                                ;
 6922 067532  164000  164000  164000 BM1:    DNOP, DNOP, DNOP        ; *** WRITE MODE (176024) ***
 6923 067540  114000                        APNT
 6924 067542  000000                BM1X:   0
 6925 067544  000000                BM1Y:   0
 6926 067546  136000                        BM14!0                  ; BIT MAP MODE...
 6927 067550  000000                BM1PIX: 0                       ;...PIXEL BUFFER ADDRESS.
 6928 067552  164000  164000  164000       DNOP, DNOP, DNOP        ; *** READ MODE ( 176040) ***
 6929 067560  173000                        STOPN
 6930
 6931 067562                                END.TEST

                    ;*****************************************************************************
                    ;*
                    ;*      END TEST 26
                    ;*
                    ;*****************************************************************************
```

```
6933
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 27   -   BIT MAP (0)
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 10.)

6934 067630  005737  002502                    TST     DPUMOD          ; DPU ONLY ??
6935 067634  001402                            BEQ     BM0TST          ; NO, EXECUTE.
6936 067636                            1$:     EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
6937                                    ;
6938                                    ; INITIALIZE AND FILL THE BUFFER.
6939                                    ;
6940 067642  004737  027702            BM0TST: JSR     PC,TSTGO        ;TITLE
6941 067646  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
6942 067652  004737  026650                    JSR     PC,CLRRW        ;INIT IN R/W MODE.
6943 067656  012700  000377                    MOV     #377,R0         ; DATA FOR 2 OR 4 PIXELS.
6944 067662  004737  027014                    JSR     PC,FILL1        ; FILL THE BUFFER.
6945 067666  013737  003240  070564            MOV     FREE,BM0P1+12   ; 1ST FRAME(S) USE BUFFER.
6946 067674  005037  070610                    CLR     BM0P2+12        ; 2ND USES ABS 0.
6947                                    ;
6948                                    ; FIRST TEST THE RESOLUTION BIT ON A 32 SQUARE ARRAY.
6949                                    ;
6950 067700  013737  027700  003234    1$:     MOV     LOOPK,TEMP2     ; SAVE A COPY OF THE LOOPER.
6951 067706  012701  134000                    MOV     #BM04,R1        ; MODE 0, 512 RESOLUTION.
6952 067712  010137  070562                    MOV     R1,BM0P1+10
6953 067716  005037  070556                    CLR     BM0P1+4
6954 067722  005037  070560                    CLR     BM0P1+6         ; ORIGIN 0,0.
6955 067726  012704  000100                    MOV     #100,R4
6956 067732  010405                          MOV     R4,R5
6957
6958 067734                            2$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6959 067736  012777  070552  113066            MOV     #BM0P1,@DPC
6960 067744  004737  070456                    JSR     PC,BM0CHK       ; ERROR CHECK.
6961 067750                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6962 067752                                    LOOPTO  2$              ; REPEAT...
     067752  004737  027646                    JSR     PC,LOOP
     067756  103766                            BCS     2$
6963
6964 067760  013737  003234  027700          MOV     TEMP2,LOOPK     ;...THEN RESTORE LOOPER.
6965 067766  012701  134100                    MOV     #BM04!R256,R1   ; SHIFT TO 256 RESOLUTION.
6966 067772  010137  070562                    MOV     R1,BM0P1+10
6967 067776  062737  000040  070556          ADD     #40,BM0P1+4
6968 070004  062737  000040  070560          ADD     #40,BM0P1+6
6969 070012  062704  000140                    ADD     #140,R4
6970 070016  010405                          MOV     R4,R5
6971
6972 070020                            3$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6973 070022  012777  070552  113002            MOV     #BM0P1,@DPC
6974 070030  004737  070456                    JSR     PC,BM0CHK       ; ERROR CHECK.
6975 070034                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6976 070036                                    LOOPTO  3$              ; REPEAT...
     070036  004737  027646                    JSR     PC,LOOP
     070042  103766                            BCS     3$
6977
```

```
6978 070044  013737  003234  027700          MOV     TEMP2,LOOPK     ;...AND RESTORE.
6979                                 ;
6980                                 ; NOW CHECK THE EXPANSION OF A 32 SQUARE ARRAY.
6981                                 ;
6982 070052  012701  134000          4$:     MOV     #BM04,R1        ; 4 BIT, 32 SQUARE, NO EXP.
6983 070056  000402                          SKP2
6984 070060  012701  135000          5$:     MOV     #BM08,R1        ; 8 BIT, 32 SQUARE, NO EXP.
6985 070064  005037  070556                  CLR     BMOP1+4         ; 1ST BLOCK AT 0,0
6986 070070  005037  070560                  CLR     BMOP1+6
6987 070074  012704  000100                  MOV     #100,R4         ; SET EXP X AND Y.
6988 070100  010405                          MOV     R4,R5
6989 070102  010137  070562                  MOV     R1,BMOP1+10     ; SET OPCODE.
6990 070106  004737  026650                  JSR     PC,CLRRW
6991
6992 070112                          6$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6993 070114  012777  070552  112710          MOV     #BMOP1,@DPC
6994 070122  004737  070456                  JSR     PC,BMOCHK       ; ERROR CHECK.
6995 070126                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6996 070130                                  LOOPTO  6$              ; REPEAT...
     070130  004737  027646                  JSR     PC,LOOP
     070134  103766                          BCS     6$
6997
6998 070136  013737  003234  027700          MOV     TEMP2,LOOPK
6999 070144  062737  000040  070556          ADD     #40,BMOP1+4
7000 070152  062737  000040  070560          ADD     #40,BMOP1+6
7001 070160  062704  000140                  ADD     #140,R4
7002 070164  010405                          MOV     R4,R5
7003 070166  042701  000777                  BIC     #777,R1
7004 070172  052701  000010                  BIS     #EX2,R1         ; EXPAND X2.
7005 070176  010137  070562                  MOV     R1,BMOP1+10
7006
7007 070202                          7$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7008 070204  005277  112622                  INC     @DPC
7009 070210  004737  070456                  JSR     PC,BMOCHK       ; ERROR CHECK.
7010 070214                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7011 070216                                  LOOPTO  7$
     070216  004737  027646                  JSR     PC,LOOP
     070222  103767                          BCS     7$
7012
7013 070224  013737  003234  027700          MOV     TEMP2,LOOPK
7014 070232  062737  000100  070556          ADD     #100,BMOP1+4
7015 070240  062737  000100  070560          ADD     #100,BMOP1+6
7016 070246  062704  000300                  ADD     #300,R4
7017 070252  010405                          MOV     R4,R5
7018 070254  042701  000777                  BIC     #777,R1
7019 070260  052701  000020                  BIS     #EX4,R1         ; EXPAND X4.
7020 070264  010137  070562                  MOV     R1,BMOP1+10
7021
7022 070270                          8$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7023 070272  005277  112534                  INC     @DPC
7024 070276  004737  070456                  JSR     PC,BMOCHK       ; ERROR CHECK.
7025 070302                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7026 070304                                  LOOPTO  8$              ; REPEAT
     070304  004737  027646                  JSR     PC,LOOP
     070310  103767                          BCS     8$
7027
7028 070312  013737  003234  027700          MOV     TEMP2,LOOPK
```

```
7029 070320  020127  135020              CMP     R1,#BM08!EX4    ; 8 BIT MODE DONE ??
7030 070324  001255                      BNE     5$              ; NO, GO 'ROUND.
7031                                ;
7032                                ; NOW TEST THE 64, 128, AND 256 SQUARE SIZES BY DISPLAYING
7033                                ; THE CONTENTS OF CPU MEMORY STARTING AT LOC 0.
7034                                ;
7035 070326  012701  134001        MAPMEM: MOV     #BM04!M64,R1    ; 4 BIT, 64 SQ.
7036 070332  012704  000200                MOV     #200,R4
7037 070336  010405                        MOV     R4,R5           ; EXP X,Y.
7038 070340  010137  070606        1$:     MOV     R1,BMOP2+10     ; SET OPCODE.
7039
7040 070344                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7041 070346  012777  070576  112456        MOV     #BMOP2,@DPC
7042 070354  004737  027454                JSR     PC,WAITF
7043 070360  103375                        BCC     .-4             ; BIG ONES WILL OUTLAST WAITF TIMER.
7044 070362  004737  070462                JSR     PC,BMOCHK+4     ; ERROR CHECK 200,200.
7045                                                                ;       400,400 (128 SQ).
7046                                                                ;       1000,1000 (256 SQ).
7047 070366                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7048
7049 070370  004737  027522                JSR     PC,PAUS.5
7050 070374  032701  001000                BIT     #1000,R1        ; 8 BIT MODE ??
7051 070400  001010                        BNE     2$              ; YES.
7052 070402  052701  001000                BIS     #1000,R1        ; NO, SHIFT TO 8 BIT MODE.
7053 070406  000754                        BR      1$              ; *** OR NOP ***
7054 070410  023727  070606  135003        CMP     BMOP2+10,#BM08!M256
7055 070416  001401                        BEQ     2$              ; QUIT AT 8 BIT, 256 SQ. ...
7056 070420  000747                        BR      1$              ;... OR WE RUN OUT OF MEMORY.
7057 070422  042701  001000        2$:     BIC     #1000,R1        ; RESET 4 BIT MODE...
7058 070426  005201                        INC     R1              ;...SET NEXT SIZE...
7059 070430  006304                        ASL     R4
7060 070432  006305                        ASL     R5              ;...AND EXP X,Y.
7061 070434  032701  000004                BIT     #4,R1           ; DONE THRU 256 SQ. ??
7062 070440  001737                        BEQ     1$              ; NO, GO 'ROUND.
7063
7064 070442  004737  027514        10$:    JSR     PC,PAUSE1       ; ALL DONE.
7065 070446  004737  030072                JSR     PC,TSTEND
7066 070452                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
7067
7068 070456  004737  027454        BMOCHK: JSR     PC,WAITF        ; WAIT FOR DONE...
7069 070462  027704  112350                CMP     @DXR,R4         ;...AND TEST X...
7070 070466  001003                        BNE     1$
7071 070470  027705  112344                CMP     @DYR,R5         ;...AND Y.
7072 070474  001413                        BEQ     2$
7073 070476                        1$:
7074 070476  012737  070526  022300        MOV     #3$,EXTA        ; SET ERROR EXTENSION ADDRESS.
7075 070504                                HRDERR  BMF,XYERR
7076 070524                        2$:
7077 070524  000207                        RTS     PC
7078 070526                        3$:     PRINTX  #BMX,R1         ; EXTENSION SHOWS BAD OPCODE.
7079 070550  000207                        RTS     PC
7080                                ;
7081                                ; DISPLAY CODE FOR BMO TESTS.
7082                                ;
7083 070552  164000        BMOP1:  DNOP
7084 070554  114000  000000  000000        APNT, 0, 0
7085 070562  134000                        BM04!0                  ; 32 SQUARE, NO EXP.
```

```
7086 070564  000000                          0                          ; BUFFER ADDRESS.
7087 070566  164000  173000                  DNOP, STOPN
7088 070572  160000  070554                  DJMP, BM0P1+2
7089
7090 070576  170140                  BM0P2:  CLRMEM
7091 070600  114000  000000  000000          APNT, 0, 0
7092 070606  134000                          BM04!0
7093 070610  000000                          0
7094 070612  164000  173000                  DNOP, STOPN
7095 070616  160000  070576                  DJMP, BM0P2
7096
7097 070622                                  END.TEST

                     ;*************************************************************************
                     ;*
                     ;*      END TEST 27
                     ;*
                     ;*************************************************************************
```

```
7099
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 28   -   IMAGE MEMORY CLEAR-SET
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 10.)

7100 070704  005737  002502            TST     DPUMOD          ;DPU ONLY ??
7101 070710  001402                    BEQ     CLRTST          ;NO, EXECUTE.
7102 070712                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
7103                            ;
7104                            ; "CLRMEM" TO ZERO, EXECUTE PIXEL-READ-BACK, AND TEST DATA.
7105                            ; "SETMEM" TO 1'S, AND REPEAT.
7106                            ; DISPLAY MODE (INTERLACE/NON-INTERLACE) IS TRANSPARENT.
7107                            ;
7108                            ; IF SHORT MODE, CYCLE X FOR Y = 0 AND 2 (ODD VS EVEN),
7109                            ; THEN CYCLE Y HOLDING X AT 0.
7110                            ; REPEAT EACH PIXEL READ 10 TIMES AFTER QUICK PASS.
7111                            ; IF LONG MODE, CYCLE X FOR EVERY Y (512 X 512 ADDRESSES),
7112                            ; READING EACH PIXEL ONLY ONCE.
7113                            ;
7114 070716  004737  027702    CLRTST: JSR     PC,TSTGO        ;TITLE
7115 070722  004737  030414            JSR     PC,CONFIG       ; CONFIGURE VSV SYSTEM.
7116 070726  004737  027054            JSR     PC,LUMIN        ; BLAST LUT.
7117 070732                            BGNSUB
7118 070734  005037  072552            CLR     IMC.CH
7119 070740  013701  072552    CLRT1:  MOV     IMC.CH,R1
7120 070744  006301                    ASL     R1
7121 070746  016101  003256            MOV     MEMTAB(R1),R1
7122 070752  001002                    BNE     .+6             ; BR IF MEMORY ON THIS CHANNEL
7123 070754  000137  071254            JMP     11$
7124 070760  010137  003072            MOV     R1,PDF
7125 070764  005101                    COM     R1
7126 070766  010137  003074            MOV     R1,PDM
7127 070772  013701  072552            MOV     IMC.CH,R1
7128 070776  000301                    SWAB    R1
7129 071000                            BISW3   #RDWRT,R1,CLR.ZA        ; INSERT CHANNEL OF MEMORY TO TEST
     071000  010146                    MOV     R1,-(SP)
     071002  052716  176074            BIS     #RDWRT,(SP)
     071006  012637  071572            MOV     (SP)+,CLR.ZA
7130 071012  005737  002504            TST     IMTMOD
7131 071016  001403                    BEQ     .+10            ;SKIP NEXT IF SHORT MODE...
7132 071020  012737  000001  027700    MOV     #1,LOOPK        ;...OTHERWISE, SET ONCE ONLY.
7133 071026  013737  027700  003234    MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
7134 071034  005003                    CLR     R3              ;R3 = PATTERN (0'S FIRST).
7135 071036  012777  071562  111766    MOV     #CLR.Z,@DPC     ;XCT "CLEAR MEM TO 0".
7136 071044  000402                    SKP2
7137 071046  005277  111760    1$:     INC     @DPC            ;RESUME, XCT "SETMEM TO 1'S".
7138 071052  004737  027454            JSR     PC,WAITF        ;WAIT 'TIL DONE
7139 071056  005001                    CLR     R1              ;INIT X ADDRESS...
7140 071060  005002                    CLR     R2              ;...AND Y ADDRESS.
7141
7142 071062                    2$:
7143 071062  010277  111752            MOV     R2,@DYR         ;SET Y ADDRESS.
7144 071066  004737  027454            JSR     PC,WAITF
```

```
7145 071072                              3$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7146 071074  010177  111736                      MOV     R1,@DXR         ;NOW READ-BACK PIXEL DATA...
7147 071100  004737  027454                      JSR     PC,WAITF
7148 071104  004737  023524                      JSR     PC,ERRCHK
7149 071110  017704  111720                      MOV     @DSR,R4         ;...TO R4.
7150 071114  100015                              BPL     10$             ;BR IF DPU HUNG (PRB FAILURE).
7151 071116  043704  003074                      BIC     PDM,R4          ;OK, JUST KEEP THE DATA BITS.
7152 071122  020304                              CMP     R3,R4           ;IS DATA RIGHT ??
7153 071124  001423                              BEQ     4$              ;BR IF WRONG.
7154 071126                                      HRDERR  IMDI,IMDERR     ;DATA INCORRECT.
7155 071146  000412                              BR      4$
7156 071150  004737  026756              10$:    JSR     PC,RELEAS       ;RELEASE "HUNG" DPU...
7157 071154                                      HRDERR  PRBHNG          ;...REPORT ERROR AND QUIT.
7158 071174                              4$:     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7159
7160 071176  013737  003234  027700      5$:     MOV     TEMP2,LOOPK     ;RESET LOOPER.
7161 071204  020227  000004              6$:     CMP     R2,#4           ;IF "SHORT" MODE, HOLD X...
7162 071210  103006                              BHIS    8$              ;...AT 0 FOR Y = 4 TO MAX.
7163
7164 071212  020127  001776              7$:     CMP     R1,#MAXX
7165 071216  001403                              BEQ     8$              ;BR IF AT MAX X.
7166 071220  062701  000002                      ADD     #2,R1           ;OTHERWISE, BUMP X...
7167 071224  000722                              BR      3$              ;...AND CONTINUE.
7168 071226  020237  003250              8$:     CMP     R2,YMAX
7169 071232  001404                              BEQ     9$              ;DONE IF AT MAX Y.
7170 071234  005001                              CLR     R1              ;OTHERWISE, RESET X...
7171 071236  062702  000002                      ADD     #2,R2           ;...BUMP Y...
7172 071242  000707                              BR      2$              ;...AND CONTINUE.
7173 071244  005103                      9$:     COM     R3              ;COMPLIMENT DATA PATTERN.
7174 071246  043703  003074                      BIC     PDM,R3
7175 071252  001275                              BNE     1$              ;RESTART 1'S PATTERN.
7176
7177 071254                              11$:
7178 071254  005237  072552                      INC     IMC.CH          ; DO NEXT CHANNEL
7179 071260  023727  072552  000004              CMP     IMC.CH,#4
7180 071266  103624                              BLO     CLRT1           ; LOOP THRU ALL 4 CHANNELS
7181 071270                                      ENDSUB
7182
7183 071272                              CLTNEW: BGNSUB
7184 071274  052737  000006  072576              BIS     #6,IMC.IR               ;LET READ SEE ALL BITS
7185 071302  012737  071612  071610              MOV     #CLMTAB,CLMPTR          ;START AT BEGINNING OF PATTERN TABLE
7186 071310  017700  000274              1$:     MOV     @CLMPTR,R0              ;GET PATTERN TO CLEAR TO
7187 071314  100515                              BMI     8$                      ;NEGATIVE IF DONE ALL
7188 071316  042737  001777  071720              BIC     #1777,CLR.D2            ;CLEAR OUT OLD PIXEL PATTERN
7189 071324  050037  071720                      BIS     R0,CLR.D2               ;INSERT THE NEW ONE
7190 071330  005700                              TST     R0                      ;IS NEW ONE 0?
7191                                     ;       BNE     2$                      ;IF NO, GO DO "SETMEM" FUNCTION.
7192 071332  000404                              BR      2$                      ;DO "SETMEM" TO 0 UNTIL NEW MICROCODE READY
7193 071334  012777  071704  111470              MOV     #CLR.Z2,@DPC            ;IF YES, DO THE "CLRMEM"
7194 071342  000403                              BR      3$
7195 071344  012777  071720  111460      2$:     MOV     #CLR.D2,@DPC            ;"CLEAR" TO THE PATTERN
7196 071352  004737  027454              3$:     JSR     PC,WAITF                ;WAIT FOR DONE
7197
7198 071356  005037  072552                      CLR     IMC.CH                  ;START CHECKING AT CHANNEL 0.
7199 071362  013700  072552              4$:     MOV     IMC.CH,R0               ;GET CHANNEL TO CHECK
7200 071366  006300                              ASL     R0                      ;MAKE A WORD INDEX
7201 071370  016001  003256                      MOV     MEMTAB(R0),R1           ;GET MASK FOR VALID PIXEL BITS
```

```
7202 071374  001445                          BEQ     7$                      ;IF 0, NO CHANNEL HERE. GO TO NEXT.
7203 071376  010137  003072                  MOV     R1,PDF                  ;SAVE VALID MASK
7204 071402  005101                          COM     R1                      ;COMPLEMENT IT.
7205 071404  057701  000200                  BIS     @CLMPTR,R1              ;INSERT THE PATTERN
7206 071410  006201                          ASR     R1                      ;SHIFT IT DOWN TO BITS <7:0>
7207 071412  006201                          ASR     R1
7208 071414  010146                          MOV     R1,-(SP)                ;PUT PATTERN IN LOW BYTE,
7209 071416  110166  000001                  MOVB    R1,1(SP)                ;AND HIGH BYTE.
7210 071422  012601                          MOV     (SP)+,R1                ;PUT IT BACK INTO R1.
7211 071424  012700  000400                  MOV     #256.,R0                ;GET COUNT OF NUMBER OF WORDS TO FILL (256.)
7212 071430  013702  003240                  MOV     FREE,R2                 ;GET ADDRESS TO PUT TEST DATA.
7213 071434  010122                5$:       MOV     R1,(R2)+                ;FILL THE TEST BUFFER
7214 071436  005300                          DEC     R0
7215 071440  001375                          BNE     5$
7216 071442  013737  003240  072554          MOV     FREE,IMC.TA             ;TELL CHECK ROUTINE WHERE IT IS.
7217 071450  010237  072606                  MOV     R2,IMC.RA               ;... AND WHERE TO PUT THE MEM DATA READ.
7218 071454  005037  072574                  CLR     IMC.Y                   ;START AT Y = 0.
7219
7220 071460                6$:       BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7221 071462  004737  071736                  JSR     PC,IMCHK                ;READ AND CHECK A LINE OF MEMORY DATA.
7222 071466                          ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7223
7224 071470  023737  072574  003250          CMP     IMC.Y,YMAX              ;DONE LAST LINE?
7225 071476  001404                          BEQ     7$                      ;MOVE ON TO NEXT CHANNEL IF YES.
7226 071500  062737  000002  072574          ADD     #2,IMC.Y                ;IF NO, BUMP Y & GO FOR MORE.
7227 071506  000764                          BR      6$
7228 071510  005237  072552        7$:       INC     IMC.CH                  ;INCREMENT THE CHANNEL NUMBER.
7229 071514  023727  072552  000004          CMP     IMC.CH,#4               ;DONE LAST CHANNEL?
7230 071522  002717                          BLT     4$                              ;IF NOT, GO FOR MORE.
7231 071524  062737  000002  071610          ADD     #2,CLMPTR               ;IF YES, MOVE ON TO NEXT PATTERN
7232 071532  005737  002504                  TST     IMTMOD          ; SHORT TEST?
7233 071536  001264                          BNE     1$              ; NO.
7234 071540  023727  071610  071626          CMP     CLMPTR,#CLMSHT  ; YES. DONE?
7235 071546  001260                          BNE     1$              ; NO.
7236                                                                 ; YES.
7237
7238 071550                8$:
7239 071550                          ENDSUB
7240 071552  004737  030072                  JSR     PC,TSTEND
7241 071556                          EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
7242
7243                       ;CLR.Z:  DNOP,CLRMEM,STOPN       ;CLEAR IMAGE MEMORY.
7244 071562  176000  176400  177000 CLR.Z:   PROTEC!CH0,PROTEC!CH1,PROTEC!CH2,PROTEC!CH3
7245 071572  176074                CLR.ZA:  RDWRT
7246 071574  116000  170100  173000         APNT!LO,SETMEM,STOPN    ;CLEAR MEMORY THIS WAY UNTIL NEW MICROCODE READY
7247 071602  117774  170100  173000 CLR.D:  APNT!ALL,SETMEM,STOPN   ;SET IMAGE MEMORY TO 1'S.
7248
7249
7250                                       .MACRO  CLP.D   CPAT
7251                                       <CPAT*BIT6>!<-CPAT-1&17*BIT2>
7252                                       .ENDM
7253 071610  071612                CLMPTR:  .WORD   CLMTAB
7254
7255 071612  000000                CLMTAB:  0
7256 071614                                 CLP.D   17
7257 071616                                 CLP.D   10
7258 071620                                 CLP.D   4
```

```
7259 071622                                        CLP.D   2
7260 071624                                        CLP.D   1
7261 071626                                CLMSHT: CLP.D   14
7262 071630                                        CLP.D   11
7263 071632                                        CLP.D   12
7264 071634                                        CLP.D   3
7265 071636                                        CLP.D   5
7266 071640                                        CLP.D   6
7267 071642                                        CLP.D   7
7268 071644  001774                                1774
7269 071646  177777                                -1
7270 071650  000000  000000  000000                0,0,0,0,0,0
7271 071664  000000  000000  000000                0,0,0,0,0,0,0,0
7272
7273                                       ;DISPLAY FILES FOR CLEARING MEMORY:
7274 071704  176074  176474  177074       CLR.Z2: RDWRT!CH0,RDWRT!CH1,RDWRT!CH2,RDWRT!CH3
7275 071714  170140  173000                       CLRMEM,STOPN
7276
7277 071720  117774                       CLR.D2: APNT!ALL
7278 071722  176074  176474  177074               RDWRT!CH0,RDWRT!CH1,RDWRT!CH2,RDWRT!CH3
7279 071732  170100  173000                       SETMEM,STOPN
7280
7281
7282                                       ;
7283                                       ; SUBROUTINE - READ & VERIFY 1 LINE OF IMAGE MEMORY.
7284                                       ;
7285                                       ;ENTER WITH:
7286                                       ;       IMC.CH = MEMORY CHANNEL TO READ (0, 1, 2, OR 3)
7287                                       ;       IMC.Y  = Y POSITION OF LINE TO BE READ
7288                                       ;       IMC.RA = ADDRESS OF 512.-BYTE BUFFER TO READ DATA INTO
7289                                       ;       IMC.TA = ADDRESS OF 512.-BYTE BUFFER TO TEST AGAINST
7290                                       ;                (DATA MUST INCLUDE 1'S IN NON-EXISTENT BIT POSITIONS TO
7291                                       ;                 CATCH BAD DRIVERS & DBUS LINES)
7292                                       ;
7293 071736  013700  072552       IMCHK:  MOV     IMC.CH,R0               ;GET CHANNEL #
7294 071742  000300                       SWAB    R0                      ;... INTO HIGH BYTE OF R0.
7295 071744  042737  001400  072566       BIC     #BIT9!BIT8,IMC.ME       ;CLEAR OLD CHANNEL # FROM ENABLING INSTR.
7296 071752  050037  072566               BIS     R0,IMC.ME               ;SET IN THE NEW CHANNEL
7297 071756  013737  072510  072512       MOV     IMCTRY,IMCRCT
7298 071764  012777  072556  111040 IMCHKR: MOV     #IMC.SR,@DPC            ;START THE READ.
7299 071772  004737  027454               JSR     PC,WAITF                ;... WAIT FOR IT TO FINISH.
7300 071776  013702  072606               MOV     IMC.RA,R2
7301 072002  062702  001000               ADD     #512.,R2                ;R2 GETS FINAL ADDRESS
7302 072006  013703  072576               MOV     IMC.IR,R3               ;R3 GETS WHAT DSR SHOULD BE.
7303 072012  027702  111014               CMP     @DPC,R2                 ;IS DPC CORRECT?
7304 072016  001003                       BNE     1$
7305 072020  027703  111010               CMP     @DSR,R3                 ;IS DSR CORRECT?
7306 072024  001410                       BEQ     2$
7307 072026                        1$:
7308 072026                               HRDERR  IMRF,PCSERR             ;PRINT IF NOT.
7309 072046  005004                2$:    CLR     R4                              ;DXR SHOULD BE 0.
7310 072050  013705  072574               MOV     IMC.Y,R5
7311 072054  062705  000002               ADD     #2,R5                   ;DYR SHOULD BE Y+2
7312 072060  027704  110752               CMP     @DXR,R4                 ;TEST DXR
7313 072064  001003                       BNE     3$
7314 072066  027705  110746               CMP     @DYR,R5                 ;TEST DYR
7315 072072  001410                       BEQ     4$
```

```
7316 072074                                3$:
7317 072074                                        HRDERR  IMRF,XYERR
7318 072114  012701  000000                4$:     MOV     #0,R1           ;R1 = X ADDRESS UNDER TEST.
7319 072120  013704  072554                        MOV     IMC.TA,R4       ;R4 POINTS TO EXPECTED DATA
7320 072124  013705  072606                        MOV     IMC.RA,R5       ;R5 POINTS TO RECEIVED DATA
7321 072130  022425                        5$:     CMP     (R4)+,(R5)+     ;CHECK THE DATA
7322 072132  001006                                BNE     10$
7323 072134  062701  000004                6$:     ADD     #4,R1           ;BUMP THE X ADDRESS
7324 072140  020127  002000                        CMP     R1,#MAXX+2
7325 072144  002771                                BLT     5$
7326 072146  000471                                BR      9$                      ; EXIT WHEN DONE WITH LINE.
7327
7328 072150  005337  072512                10$:    DEC     IMCRCT
7329 072154  100402                                BMI     .+6
7330 072156  000137  071764                        JMP     IMCHKR
7331 072162  010137  072504                        MOV     R1,IMCBDA       ;GET POSITION OF BAD DATA
7332 072166  126465  177776  177776              CMPB    -2(R4),-2(R5)   ;LOW BYTE BAD?
7333 072174  001440                                BEQ     12$             ;BR IF NOT.
7334 072176  116437  177776  023056              MOVB    -2(R4),GDDAT    ;IF YES, GET THE DATA
7335 072204  116537  177776  023060              MOVB    -2(R5),BADDAT
7336 072212  006337  023056                11$:    ASL     GDDAT           ;SHIFT IT TO LOOK LIKE DBUS DATA
7337 072216  006337  023056                        ASL     GDDAT
7338 072222  006337  023060                        ASL     BADDAT
7339 072226  006337  023060                        ASL     BADDAT
7340 072232  042737  176003  023056              BIC     #^C1774,GDDAT
7341 072240  042737  176003  023060              BIC     #^C1774,BADDAT
7342 072246                                        HRDERR  IMDI,IMCERR
7343 072266  032737  000002  072504              BIT     #2,IMCBDA               ;DONE HIGH BYTE?
7344 072274  001317                                BNE     6$                              ;IF YES, GO BACK TO MAIN LOOP.
7345 072276  062737  000002  072504  12$:        ADD     #2,IMCBDA
7346 072304  126465  177777  177777              CMPB    -1(R4),-1(R5)           ;COMPARE HIGH BYTES
7347 072312  001710                                BEQ     6$
7348 072314  116437  177777  023056              MOVB    -1(R4),GDDAT
7349 072322  116537  177777  023060              MOVB    -1(R5),BADDAT
7350 072330  000730                                BR      11$
7351
7352 072332  000207                        9$:     RTS     PC
7353
7354 072334                                BGNMSG  IMCERR
7355 072334                                        PRINTX  TNAM
7356 072354  013737  072504  072506              MOV     IMCBDA,IMCRAM
7357 072362  006237  072506                        ASR     IMCRAM
7358 072366                                        PRINTX  #IMDIXC,IMCBDA,IMC.Y,PDF,GDDAT,BADDAT
7359 072432                                        PRINTX  #IMDIX2,IMCRAM,IMC.CH
7360 072462                                        PRINTX  #NULCR                  ;PRINT BLANK LINE
7361 072502                                ENDMSG
7362
7363 072504  000000                        IMCBDA: .WORD   0                       ;X POSITION FOR PRINTOUT
7364 072506  000000                        IMCRAM: .WORD   0                       ;SUSPECT RAM
7365 072510  000000                        IMCTRY: .WORD   0
7366 072512  000000                        IMCRCT: .WORD   0
7367
7368 072514    040    040    104          IMRF:   .ASCIZ  /  DMA PIXEL READBACK FAILURE/
7369                                               .EVEN
7370
7371         163000                                IMR8=163000
7372 072552  000000                        IMC.CH: 0
```

```
 7373 072554  000000                    IMC.TA: 0                       ;ADDRESS OF TEST BUFFER
 7374 072556  176000  176400  177000    IMC.SR: PROTEC!CH0,PROTEC!CH1,PROTEC!CH2,PROTEC!CH3
 7375 072566  176074                    IMC.ME: RDWRT                   ;ENABLE SELECTED MEMORY
 7376 072570  114000  000000                    APNT,0                  ;START AT X=0
 7377 072574  000000                    IMC.Y:  0                       ;Y
 7378 072576  163006                    IMC.IR: IMR8!6                  ;READ, 2 8-BIT PIXELS/WORD, ALL BITS
 7379 072600  001000  000001                    512.,1                  ;1 FULL LINE
 7380 072604  000400                            256.                    ;WORD COUNT = 256.
 7381 072606  160000                    IMC.RA: .WORD   160000          ;READ BUFFER
 7382 072610  173000                            STOPN                   ;(SHOULD STOP BY ITSELF)
 7383 072612                                    END.TEST

                                        ;*************************************************************************
                                        ;*
                                        ;*      END TEST 28
                                        ;*
                                        ;*************************************************************************
```

```
7385
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 29   -   IMAGE MEMORY INTERLACE
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 10.)

7386 072674  005737  002502                    TST     DPUMOD          ; DPU ONLY?
7387 072700  001402                            BEQ     INLAC           ; NO. EXECUTE.
7388 072702                                    EXIT    TST     ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
7389
7390
7391                                    ;
7392                                    ; CHECK MEMORY INTERLACE / NON- INTERLACE BY WRITING A POINT AT 0,0.
7393                                    ; IF MEM IS INTERLACED, THE PIXEL DATA AT 0,2 SHOULD BE (NON-ZERO).
7394                                    ; IF MEM IS NON-INTERLACED, THE PIXEL DATA AT 0,2 SHOULD BE (ZERO).
                                        ;
7395 072706                            INLAC:  COMBEG
     072706  004737  027702                    JSR     PC,TSTGO        ; TITLE.
     072712  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
7396 072716  004737  030414                    JSR     PC,CONFIG       ; CONFIGURE VSV SYSTEM.
7397 072722  005005                    1$:     CLR     R5              ; INIT CHAN PTR.
7398 072724  004737  026650                    JSR     PC,CLRRW        ; CLEAR ALL MEMORIES.
7399
7400 072730                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7401 072732  010504                    2$:     MOV     R5,R4           ; GET MEMORY TABLE ENTRY FOR THIS CHAN.
7402 072734  006304                            ASL     R4              ;
7403 072736  016401  003256                    MOV     MEMTAB(R4),R1   ; (SAVE IT IN R1 & R3). IS THERE A MEM?
7404 072742  010103                            MOV     R1,R3           ;
7405 072744  001450                            BEQ     4$              ; NO.
7406 072746  010504                            MOV     R5,R4           ; GET CHAN IN BITS 8,9.
7407 072750  000304                            SWAB    R4              ;
7408 072752  052704  176034                    BIS     #WRT,R4         ; SET WRITE INST FOR CURRENT CHAN -
7409 072756  010437  073134                    MOV     R4,11$          ; - IN DISPLAY FILE.
7410 072762                                    DPSTART #10$            ; YES.
     072762  012777  073124  110042            MOV     #10$,@DPC       ; START THE DPU.
     072770  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
7411 072774  032737  020000  003270            BIT     #BIT13,SYCTAB   ; IS MEM NON-INTERLACED?
7412 073002  001401                            BEQ     3$              ; YES. (R1 = EXPECTED FOR NON-INTERLACED).
7413 073004  005001                            CLR     R1              ; NO. ADJUST R1 TO EXPECT FOR INTERLACED.
7414 073006  012777  000002  110024    3$:     MOV     #2,@DYR         ; PIXEL READ (POINT 0,2) INTO R2.
7415 073014  004737  027454                    JSR     PC,WAITF        ;
7416 073020  012777  000000  110006            MOV     #0,@DSR         ;
7417 073026  017702  110002                    MOV     @DSR,R2         ;
7418 073032  042702  176003                    BIC     #176003,R2      ;
7419 073036  074203                            XOR     R2,R3           ; AND PIXEL READ WITH MEM TAB VAL.
7420 073040  040302                            BIC     R3,R2           ;
7421 073042                                    IFERROR INLE,INLEX
     073042  020102                            CMP     R1,R2           ; OK?
     073044  001410                            BEQ     64$             ; YES.
     073046                                    HRDERR  INLE,INLEX,     ; NO.
     073066                          64$:
7422 073066  005205                    4$:     INC     R5              ; NEXT CHAN.
7423 073070  020527  000004                    CMP     R5,#4           ; DONE?
7424 073074  001316                            BNE     2$              ; NO.
7425                                                                    ; YES.
```

```
 7426 073076                                        ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 7427 073100                                        COMEND  1$
      073100  004737  027646                        JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
      073104  103401                                BCS     65$
      073106  000402                                BR      66$

      073110  000137  072722            65$:        JMP     1$

      073114  004737  030072            66$:        JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
 7428 073120                                        EXIT    TST     ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
 7429
 7430                                               ;
 7431                                               ; DISPLAY FILE.
 7432                                               ;
 7433 073124  176000  176400  177000   10$:        PROTEC!CH0,PROTEC!CH1,PROTEC!CH2,PROTEC!CH3
 7434 073134  176034                    11$:        WRT!CH0
 7435 073136  117774  040000  000000                APNT!ALL,I!0,0
 7436 073144  173000                                STOPN
 7437
 7438 073146                                        END.TEST

                                                   ;*************************************************************************
                                                   ;*
                                                   ;*      END TEST 29
                                                   ;*
                                                   ;*************************************************************************
```

```
7440
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 30   -   IMAGE MEMORY PATTERNS
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 1.)

7441 073226  005737  002502                    TST     DPUMOD          ;DPU ONLY ??
7442 073232  001402                            BEQ     PATTST          ;NO, EXECUTE.
7443 073234                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
7444                                    ;
7445                                    ; WRITE, THEN READ-BACK AND TEST PIXEL DATA PATTERNS.
7446                                    ; USE FOUR DATA PATTERNS PER PIXEL AS FOLLOWS:
7447                                    ;       1.  00000000    FOUR 2 BIT 0'S.
7448                                    ;       2.  01010101    FOUR 2 BIT 1'S
7449                                    ;       3.  10101010    FOUR 2 BIT 2'S
7450                                    ;       4.  11111111    FOUR 2 BIT 3'S.
7451                                    ;
7452                                    ; DISPLAY MODE (INTERLACE/NON-INTERLACE) IS TRANSPARENT.
7453                                    ; IF SHORT MODE, CYCLE X FOR Y = 0 AND 2 (ODD VS EVEN),
7454                                    ; THEN CYCLE Y HOLDING X AT 0.
7455                                    ; READ EACH PATTERN 10 TIMES AFTER QUICK PASS.
7456                                    ; IF LONG MODE, CYCLE X FOR EACH Y (512 X 512 ADDRESSES),
7457                                    ; READING EACH PATTERN ONLY ONCE.
7458                                    ;
7459 073240  004737  027702            PATTST: JSR     PC,TSTGO        ;TITLE
7460 073244  004737  030414                    JSR     PC,CONFIG       ; CONFIGURE VSV SYSTEM.
7461 073250  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
7462
7463 073254                                    BGNSUB
7464 073256  005037  072552                    CLR     IMC.CH          ;START AT CHANNEL 0
7465 073262  013701  072552            PATT1:  MOV     IMC.CH,R1
7466 073266  006301                            ASL     R1
7467 073270  016101  003256                    MOV     MEMTAB(R1),R1
7468 073274  001002                            BNE     .+6             ; BR IF MEMORY ON THIS CHANNEL.
7469 073276  000137  073644                    JMP     11$
7470 073302  010137  003072                    MOV     R1,PDF
7471 073306  005101                            COM     R1
7472 073310  010137  003074                    MOV     R1,PDM
7473 073314  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
7474 073322  013701  072552                    MOV     IMC.CH,R1
7475 073326  000301                            SWAB    R1
7476 073330                                    BISW3   #RDWRT,R1,CLR.ZA
     073330  010146                            MOV     R1,-(SP)
     073332  052716  176074                    BIS     #RDWRT,(SP)
     073336  012637  071572                    MOV     (SP)+,CLR.ZA
7477 073342                            DPSTART #CLR.Z
     073342  012777  071562  107462            MOV     #CLR.Z,@DPC     ; START THE DPU.
     073350  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
7478 073354  005001                            CLR     R1              ;R1 = X POINTER.
7479 073356  005002                            CLR     R2              ;R2 = Y POINTER.
7480 073360  012737  040000  074264            MOV     #I,WRT.X        ;INIT DISPLAY CODE X...
7481 073366  005037  074266                    CLR     WRT.Y           ;...AND Y.
7482 073372  012705  074250            1$:     MOV     #PATS,R5        ;PATTERN POINTER => R5.
7483 073376  012503                            MOV     (R5)+,R3        ;1ST PATTERN...
```

```
7484 073400  043703  003074          2$:   BIC     PDM,R3          ;...NEXT PATTERN => R3.
7485 073404  042737  001777  074262        BIC     #1777,WRT.D     ;SET NEXT PIXEL PATTERN...
7486 073412  050337  074262                BIS     R3,WRT.D        ;...IN DISPLAY CODE.
7487
7488 073416                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7489
7490 073420  012777  074262  107404        MOV     #WRT.D,@DPC     ;WRITE NEXT PATTERN, THIS PIXEL.
7491 073426  004737  027454                JSR     PC,WAITF
7492 073432  010177  107400                MOV     R1,@DXR         ;SET THE X ADDRESS.
7493 073436  004737  027454                JSR     PC,WAITF
7494 073442  010277  107372          3$:   MOV     R2,@DYR         ;NOW READ-BACK PIXEL DATA...
7495 073446  004737  027454                JSR     PC,WAITF
7496 073452  017704  107356                MOV     @DSR,R4         ;... TO R4.
7497 073456  043704  003074                BIC     PDM,R4          ;OK, STRIP ANY CRAP.
7498 073462  020304                        CMP     R3,R4           ;AND TEST THE DATA.
7499 073464  001410                        BEQ     9$              ;BR IF OK
7500 073466                                HRDERR  IMDI,IMDERR     ;DATA INCORRECT ERROR.
7501 073506                          9$:   LOOPTO  3$              ; REPEAT...
     073506  004737  027646                JSR     PC,LOOP
     073512  103753                        BCS     3$
7502 073514  013737  003234  027700        MOV     TEMP2,LOOPK     ;RESET LOOPER.
7503 073522                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7504
7505 073524  013737  003234  027700  4$:   MOV     TEMP2,LOOPK     ;RESET LOOPER.
7506 073532  012503                        MOV     (R5)+,R3        ;GET NEXT PATTERN...
7507 073534  100321                        BPL     2$              ;...BR IF NOT TERMINATOR.
7508 073536  020227  000004          5$:   CMP     R2,#4           ;HOLD X...
7509 073542  103011                        BHIS    7$              ;...AT 0, FOR Y 4 TO MAX.
7510
7511 073544  020127  001776          6$:   CMP     R1,#MAXX
7512 073550  001406                        BEQ     7$              ;BR IF AT MAX X.
7513 073552  062701  000002                ADD     #2,R1           ;OTHERWISE, BUMP X...
7514 073556  062737  000002  074264        ADD     #2,WRT.X
7515 073564  000702                        BR      1$              ;...AND CONTINUE.
7516 073566  020237  003250          7$:   CMP     R2,YMAX
7517 073572  001424                        BEQ     11$             ;ALL DONE IF MAX Y.
7518 073574  005001                        CLR     R1              ;OTHERWISE, RESET X...
7519 073576  012737  040000  074264        MOV     #I,WRT.X
7520 073604  062702  000002                ADD     #2,R2           ;...BUMP Y...
7521 073610  062737  000002  074266        ADD     #2,WRT.Y
7522 073616  000665                        BR      1$              ;...AND CONTINUE.
7523
7524 073620  004737  026756         10$:   JSR     PC,RELEAS       ;RELEASE "HUNG" DPU...
7525 073624                                HRDERR  PRBHNG          ;...REPORT ERROR AND QUIT.
7526 073644                         11$:
7527 073644  005237  072552                INC     IMC.CH
7528 073650  023727  072552  000004        CMP     IMC.CH,#4
7529 073656  103601                        BLO     PATT1
7530 073660                                ENDSUB
7531
7532 073662                         MPTNEW: BGNSUB
7533 073664  013700  003240                MOV     FREE,R0         ;START THE "WRITE BUFFER" AT 1ST FREE LOCATION
7534 073670  012701  000045                MOV     #256./MPLEN+MPLEN,R1  ;SETUP # OF TIMES TO REPEAT PATTERN TABLE
7535 073674  012702  000011          1$:   MOV     #MPLEN,R2       ;GET LENGTH OF PATTERN TABLE
7536 073700  012703  074306                MOV     #MPAT2,R3       ;...AND ITS ADDRESS
7537 073704  012320                  2$:   MOV     (R3)+,(R0)+     ;COPY PATTERN TABLE INTO WRITE BUFFER
7538 073706  005302                        DEC     R2
```

```
7539 073710  001375                        BNE     2$
7540 073712  005301                        DEC     R1                      ;DO IT ENOUGH TIMES FOR 1 SCAN LINE, ALL PATTERNS
7541 073714  001367                        BNE     1$
7542 073716  010037  074332                MOV     R0,MPCBAS       ;NEXT FREE LOCATION IS BASE OF "COMPARE BUFFER"
7543 073722  005037  072552                CLR     IMC.CH          ;START TESTING AT CHANNEL 0.
7544
7545 073726  004737  026650          3$:   JSR     PC,CLRRW        ;CLEAR ALL MEMORIES
7546 073732  013700  072552                MOV     IMC.CH,R0       ;GET CHANNEL NUMBER
7547 073736  000300                        SWAB    R0                      ;...INTO BITS <9:8>
7548 073740  042737  001400  074344        BIC     #CH3,MPT.WE     ;CLEAR OLD CHANNEL FROM "WRITE-ENABLE" INSTRUCTION
7549 073746  050037  074344                BIS     R0,MPT.WE       ;...AND INSERT THE CURRENT CHANNEL
7550 073752  013700  072552                MOV     IMC.CH,R0       ;GET THE CHANNEL AGAIN
7551 073756  006300                        ASL     R0                      ;...AND MAKE IT A WORD INDEX
7552 073760  016001  003256                MOV     MEMTAB(R0),R1   ;GET THE MEMORY PARAMETER (PIXEL MASK)
7553 073764  001516                        BEQ     8$                      ;IF ZERO, NO MEMORY ON THIS CHANNEL.
7554 073766  010137  003072                MOV     R1,PDF          ;SAVE PIXEL MASK (FOR ERROR REPORTS)
7555 073772  006201                        ASR     R1                      ;GET PIX MASK INTO LOW BYTE
7556 073774  006201                        ASR     R1
7557 073776  005101                        COM     R1                      ;COMPLEMENT IT, TO ALLOW SETTING OF UNUSED
7558 074000  010146                        MOV     R1,-(SP)        ;...BITS IN COMPARE WORDS
7559 074002  110166  000001                MOVB    R1,1(SP)        ;BOTH BYTES.
7560 074006  012601                        MOV     (SP)+,R1
7561 074010  013700  074332                MOV     MPCBAS,R0       ;START AT BEGINNING OF COMPARE BUFFER
7562 074014  012702  000412                MOV     #256.+MPLEN+1,R2
7563 074020  013703  003240                MOV     FREE,R3         ;AND AT WRITE BUFFER
7564 074024  012310          4$:   MOV     (R3)+,(R0)      ;COPY WRITE BUFFER INTO COMPARE BUFFER,
7565 074026  050120                        BIS     R1,(R0)+        ;...SETTING UNUSED BITS.
7566 074030  005302                        DEC     R2
7567 074032  001374                        BNE     4$
7568 074034  010037  072606                MOV     R0,IMC.RA       ;MAKE NEXT FREE ADDRESS THE START OF READ BUFFER.
7569
7570 074040  005037  072574                CLR     IMC.Y           ;START TESTING AT FIRST LINE (Y=0)
7571 074044  013737  072574  074352  5$:   MOV     IMC.Y,MPT.WY    ;TELL WRITER WHERE TO WRITE
7572 074052  005037  074330                CLR     MPOFF           ;CLEAR THE PATTERN OFFSET
7573 074056  013737  003240  074356  6$:   MOV     FREE,MPT.WB     ;GET BASE OF WRITE BUFFER
7574 074064  063737  074330  074356        ADD     MPOFF,MPT.WB    ;...AND ADD THE OFFSET
7575
7576 074072                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7577
7578 074074  012777  074334  106730        MOV     #MPT.ST,@DPC    ;THEN WRITE DATA FROM THERE.
7579 074102  004737  027454                JSR     PC,WAITF
7580 074106  013737  074332  072554        MOV     MPCBAS,IMC.TA   ;GET BASE OF COMPARE BUFFER
7581 074114  063737  074330  072554        ADD     MPOFF,IMC.TA    ;...AND ADD THE OFFSET TO IT.
7582 074122  004737  071736                JSR     PC,IMCHK        ;READ THE LINE AND COMPARE IT TO COMPARE BUFFER
7583 074126                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7584
7585 074130  005737  003230                TST     INTLAC          ;INTERLACED MODE?
7586 074134  001014                        BNE     7$                      ;IF YES, SKIP AHEAD.
7587 074136  032737  000002  072574        BIT     #2,IMC.Y
7588 074144  001010                        BNE     7$
7589 074146  062737  000002  072574        ADD     #2,IMC.Y        ;IF NONINTERLACED, NEXT LINE SHOULD READ THE SAME
7590 074154  004737  071736                JSR     PC,IMCHK        ;CHECK FOR IT.
7591 074160  162737  000002  072574        SUB     #2,IMC.Y
7592 074166  062737  000002  074330  7$:   ADD     #2,MPOFF        ;BUMP THE OFFSET
7593 074174  023727  074330  000022        CMP     MPOFF,#MPLEN*2  ;DONE ALL PATTERNS?
7594 074202  002725                        BLT     6$                      ;IF NOT, LOOP FOR MORE ON THIS SAME LINE.
7595 074204  062737  000002  072574        ADD     #2,IMC.Y        ;IF DONE ALL, BUMP Y
```

```
7596 074212  023737  072574  003250          CMP     IMC.Y,YMAX      ;DONE LAST LINE?
7597 074220  101711                          BLOS    5$                      ;IF NOT, GO FOR MORE
7598 074222  005237  072552          8$:     INC     IMC.CH          ;BUMP THE CHANNEL NUMBER.
7599 074226  023727  072552  000004          CMP     IMC.CH,#4       ;DONE ALL CHANNELS?
7600 074234  002634                          BLT     3$                      ;IF NOT, DO THEM.
7601 074236                                   ENDSUB
7602
7603 074240  004737  030072                  JSR     PC,TSTEND
7604 074244                                   EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
7605 074250  000000                  PATS:   0                       ;PATRN 1 (00000000)
7606 074252  000524                          0524                    ;PATRN 2 (01010101)
7607 074254  001250                          1250                    ;PATRN 3 (10101010)
7608 074256  001774                          1774                    ;PATRN 4 (11111111)
7609 074260  177777                          -1                      ;LIST TERMINATOR.
7610
7611 074262  116000                  WRT.D:  APNT!LO                 ;PIXEL DATA SET ON THE FLY.
7612 074264  040000                  WRT.X:  I!0                     ;LIKEWISE ADDRESSES.
7613 074266  000000                  WRT.Y:  0
7614 074270  164000  164000  164000          DNOP,DNOP,DNOP          ;IN CASE WE WANT TO CHANGE...
7615 074276  164000  164000  164000          DNOP,DNOP,DNOP          ;...MODE FOR P-R-B DEBUG.
7616 074304  173000                          STOPN                   ;STOP AND TEST THIS PIXEL.
7617
7618                                  ;++++++++++++++++++++++++++++++++++
7619                                          .MACRO  PAT.D MDX
7620                                  <MDX*BIT8>!<MDX*BIT4>!<-MDX-1&17*BIT12>!<-MDX-1&17>
7621                                          .ENDM
7622
7623 074306                  MPAT2:  PAT.D   0               ;PUT 4-BIT PATTERN INTO BITS <7:4> AND <11:8>,
7624 074310                          PAT.D   1               ;...AND ITS COMPLEMENT INTO <15:12> AND <3:0>
7625 074312                          PAT.D   2               ;USE FLOATING 1'S AND 0'S
7626 074314                          PAT.D   4
7627 074316                          PAT.D   10
7628 074320                          PAT.D   7
7629 074322                          PAT.D   13
7630 074324                          PAT.D   15
7631 074326                          PAT.D   16
7632         000011          MPLEN=.-MPAT2/2         ;USE ODD # OF WORDS TO MAXIMIZE CHANCES OF FINDING
7633                                  ;               DUAL-ADDRESSING PROBLEMS.
7634
7635 074330  000000          MPOFF:  0       ;OFFSET FOR WRITING AND COMPARING
7636 074332  000000          MPCBAS: 0               ;BASE ADDRESS OF COMPARE BUFFER
7637
7638                                  ;COMMANDS FOR WRITING THE PATTERN INTO THE LINE
7639 074334  176000  176400  177000  MPT.ST: PROTEC!CH0,PROTEC!CH1,PROTEC!CH2,PROTEC!CH3
7640 074344  176034          MPT.WE: WRT!CH0         ;ENABLE A CHANNEL FOR WRITING
7641 074346  114000  000000          APNT,0          ;START AT X=0
7642 074352  000000          MPT.WY: 0
7643 074354  137000                  BM18            ;USE BIT-MAP-1, WITH A COUNT OF 512. PIXELS
7644 074356  000000          MPT.WB: 0
7645 074360  173000                  STOPN
7646
7647                                  ;++++++++++++++++++++++++++++++++++++++++++++++++++
7648 074362                                  END.TEST

                                      ;*********************************************************************************
                                      ;*
                                      ;*      END TEST 30
```

```
                                ;*
                                ;*****************************************************************************
```

```
7650
                                ;*****************************************************************************
                                ;*
                                ;*      BEGIN TEST 31   -   SPARE
                                ;*
                                ;*****************************************************************************

                                ; (ITERATION COUNT = 1.)

7651 074422  005737  002502            TST     DPUMOD          ;DPU ONLY ??
7652 074426  001402                    BEQ     SPARTS          ;NO, EXECUTE.
7653 074430                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
7654
7655 074434  004737  027702    SPARTS: JSR     PC,TSTGO
7656 074440                            EXIT    TST     ; ***NOT YET CODED***
7657
7658                            ;++++++++++++++++++++
7659 074444                            END.TEST

                                ;*****************************************************************************
                                ;*
                                ;*      END TEST 31
                                ;*
                                ;*****************************************************************************
```

```
   7662
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 32   -   SYSTEM VERIFICATION DISPLAY
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 1.)

   7663 074532  005737  002502                  TST     DPUMOD          ;DPU ONLY ??
   7664 074536  001402                          BEQ     VIS1            ;NO, EXECUTE.
   7665 074540                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
   7666                                 ;
   7667                                 ; SMORGASBORD, ONE FRAME THAT HAS A BIT OF EVERYTHING.
   7668                                 ; FIRST, A DEC LOGO (SORT OF), THEN VIOLATE THE EDGES
   7669                                 ; AT LOWER LEFT AND UPPER RIGHT.
   7670                                 ;
   7671 074544  004737  027702          VIS1:   JSR     PC,TSTGO
   7672 074550  004737  026650                  JSR     PC,CLRRW        ; CLEAR.
   7673 074554  004737  105356                  JSR     PC,DBLINK       ; DISABLE BLINKING.
   7674 074560  004737  027054                  JSR     PC,LUMIN        ; BLAST THE LUT.
   7675
   7676 074564  012777  074600  106240  1$:     MOV     #LOGO,@DPC      ; START WITH THE LOGO.
   7677 074572  004737  027454                  JSR     PC,WAITF
   7678                                 ;       JSR     PC,PAUS.5
   7679 074576  000446                          BR      VIS2            ;...AND CONTINUE.
   7680
   7681 074600  114000  000540  000100  LOGO:   APNT, 540, 100
   7682 074606  152000  110302                  SETCB,ACAT
   7683 074612  103774                          CHAR!ALL
   7684 074614     104     040     111          .ASCIZ  /D I G I T A L/
   7685
   7686 074632  114000  000140  000140          APNT,140,140            ;EXCEED EDGES AT LOWER LEFT...
   7687 074640  110000                          LVEC
   7688 074642                                  LXY     I,-200,0
   7689 074646                                  LXY     I,0,-200
   7690 074652                                  LXY     I,200,0
   7691 074656                                  LXY     I,0,200
   7692 074662  114000  001640  001540          APNT,1640,1540          ;...AND AT UPPER RIGHT.
   7693 074670  110000                          LVEC
   7694 074672                                  LXY     I,200,0
   7695 074676                                  LXY     I,0,300
   7696 074702                                  LXY     I,-200,0
   7697 074706                                  LXY     I,0,-300
   7698 074712  173000                          STOPN
```

```
                                        ;
                                        ; NOW 1 LARGE AND 2 SMALL OCTAGONS.
                                        ;
074714  012777  074754  106110  VIS2:         MOV     #LOCT,@DPC      ;1 LARGE ONE...
074722  004737  027454                  JSR     PC,WAITF
074726  012777  075026  106076          MOV     #SOCT1,@DPC     ;...AND 2 SMALL ONES.
074734  004737  027454                  JSR     PC,WAITF
074740  012777  075040  106064          MOV     #SOCT2,@DPC
074746  004737  027454                  JSR     PC,WAITF
                                ;       JSR     PC,PAUS.5
074752  000456                          BR      VIS3            ;...AND CONTINUE.

074754  114000  000500  000002  LOCT:   APNT,500,2
074762  113774                          LVEC!ALL                ; 1 LARGE OCTAGON...
074764                                  LXY     I,576,0
074770                                  LXY     I,436,436
074774                                  LXY     I,0,576
075000                                  LXY     I,-436,436
075004                                  LXY     I,-576,0
075010                                  LXY     I,-436,-436
075014                                  LXY     I,0,-576
075020                                  LXY     I,436,-436
075024  173000                          STOPN

075026  114000  000220  000620  SOCT1:  APNT,220,620            ;...AND 2 SMALL ONES.
075034  160000  075046                  DJMP,.+10
075040  114000  001460  000620  SOCT2:  APNT,1460,620
075046  107774                          SVEC!ALL
075050                                  SXY     I,76,0
075052                                  SXY     I,56,56
075054                                  SXY     I,0,76
075056                                  SXY     I,-56,56
075060                                  SXY     I,-76,0
075062                                  SXY     I,-56,-56
075064                                  SXY     I,0,-76
075066                                  SXY     I,56,-56
075070  130000                          RPNT                    ;5 REL POINTS IN CENTER.
075072                                  SXY     U,20,50
075074                                  SXY     I,20,50         ;CENTER.
075076                                  SXY     I,20,0
075100                                  SXY     I,-20,20
075102                                  SXY     I,-20,-20
075104                                  SXY     I,20,-20
075106  173000                          STOPN
```

```
7745                                        ;
7746                                        ; NEXT, SHOW GRAPH/HISTO Y IN UPPER LEFT.
7747                                        ;
7748 075110  012737  011240  075262  VIS3:          MOV     #1240!GHIINH,GY+2 ;PLOT GRAPH Y STARTING AT 1240!GHIINH.
7749 075116  012777  075236  105706                MOV     #GRY,@DPC         ;START GRAPH Y.
7750 075124  000402                                SKP2
7751 075126  005277  105700          1$:           INC     @DPC
7752 075132  004737  027454                        JSR     PC,WAITF
7753 075136  042737  010000  075262                BIC     #GHIINH,GY+2
7754 075144  062737  000004  075262                ADD     #4,GY+2           ;INCR Y AMPLITUDE...
7755 075152  023727  075262  001604                CMP     GY+2,#1604
7756 075160  001362                                BNE     1$                ;LOOP UNTIL DONE.
7757
7758 075162  012737  051240  075310                MOV     #HST!1240!GHIINH,HY+2 ;NOW, HISTO Y STARTING AT 1240!GHIINH.
7759 075170  012777  075272  105634                MOV     #HSY,@DPC
7760 075176  000402                                SKP2
7761 075200  005277  105626          2$:           INC     @DPC
7762 075204  004737  027454                        JSR     PC,WAITF
7763 075210  042737  010000  075310                BIC     #GHIINH,HY+2
7764 075216  062737  000010  075310                ADD     #10,HY+2          ;INCR Y AMPLITUDE...
7765 075224  023727  075310  041550                CMP     HY+2,#HST!1550
7766 075232  001362                                BNE     2$                ;...AND LOOP TIL DONE.
7767
7768                                        ;       JSR     PC,PAUS.5
7769 075234  000431                                BR      VIS4              ;...AND CONTINUE.
7770
7771 075236  114000  000100  001240  GRY:    APNT,100,1240
7772 075244  113774                        LVEC!ALL                          ;DRAW BASE LINE.
7773 075246                                 LXY     I,340,0
7774 075252                                 LXY     U,-340,0
7775 075256  174104                        GXI!4                             ;X INCR = 4 FOR GRAPH.
7776 075260  127774  001240          GY:    GHY!ALL,0!1240
7777 075264  173000  160000  075260        STOPN,DJMP,GY
7778
7779 075272  150000  001240          HSY:    SETHB,1240                        ;HISTO BASE LINE.
7780 075276  114000  000140  001240        APNT,140,1240
7781 075304  174110                        GXI!10                            ;X INCR = 10 FOR HISTO.
7782 075306  127774  041240          HY:    GHY!ALL,HST!1240
7783 075312  173000  160000  075306        STOPN,DJMP,HY
```

```
                                        ;
                                        ; NEXT, GRAPH/HISTO X IN LOWER RIGHT.
                                        ;
7788 075320  012737  011340  075472  VIS4:         MOV     #1340!GHIINH,GX+2 ;PLOT GRAPH X STARTING AT 1340!GHIINH.
7789 075326  012777  075446  105476          MOV     #GRX,@DPC         ;START GRAPH X.
7790 075334  000402                          SKP2
7791 075336  005277  105470          1$:     INC     @DPC
7792 075342  004737  027454                  JSR     PC,WAITF
7793 075346  042737  010000  075472          BIC     #GHIINH,GX+2
7794 075354  062737  000004  075472          ADD     #4,GX+2           ;INCR X AMPLITUDE...
7795 075362  023727  075472  001704          CMP     GX+2,#1704
7796 075370  001362                          BNE     1$                ;LOOP UNTIL DONE.
7797
7798 075372  012737  051340  075520          MOV     #HST!1340!GHIINH,HX+2 ;NOW, HISTO X STARTING AT 1340!GHIINH.
7799 075400  012777  075502  105424          MOV     #HSX,@DPC
7800 075406  000402                          SKP2
7801 075410  005277  105416          2$:     INC     @DPC
7802 075414  004737  027454                  JSR     PC,WAITF
7803 075420  042737  010000  075520          BIC     #GHIINH,HX+2
7804 075426  062737  000010  075520          ADD     #10,HX+2          ;INCR X AMPLITUDE...
7805 075434  023727  075520  041650          CMP     HX+2,#HST!1650
7806 075442  001362                          BNE     2$                ;...AND LOOP TIL DONE.
7807
7808                                    ;     JSR     PC,PAUS.5
7809 075444  000431                          BR      VIS5              ;...AND CONTINUE.
7810
7811 075446  114000  001340  000100  GRX:    APNT,1340,100
7812 075454  113774                          LVEC!ALL                  ;DRAW BASE LINE.
7813 075456                                   LXY     I,0,340
7814 075462                                   LXY     U,0,-340
7815 075466  174104                          GYI!4                     ;Y INCR = 4 FOR GRAPH.
7816 075470  123774  001340          GX:     GHX!ALL,0!1340
7817 075474  173000  160000  075470          STOPN,DJMP,GX
7818
7819 075502  150000  001340          HSX:    SETHB,1340                ;HISTO BASE LINE.
7820 075506  114000  001340  000140          APNT,1340,140
7821 075514  174110                          GYI!10                    ;Y INCR = 10 FOR HISTO.
7822 075516  123774  041340          HX:     GHX!ALL,HST!1340
7823 075522  173000  160000  075516          STOPN,DJMP,HX
```

```
                                        ;
                                        ; NEXT SHOW BIT MAP MODE 1 AT BOTTOM CENTER,
                                        ; AND BIT MAP MODE 0 AT TOP CENTER.
                                        ;
075530  012700  177777        VIS5:     MOV     #-1,R0          ; SET PIXEL DATA PATTERN...
075534  004737  027020                  JSR     PC,FILL2        ;...AND FILL BUFFER WITH...
                                                                ;...ALTERNATE WORDS.

075540  013737  003240  075710          MOV     FREE,B148       ; SET BUFFER ADDRESS.
075546  012737  136340  075706          MOV     #BM14!340,B148-2 ; SET OPCODE = BM14 X 340
075554  012777  075666  105250          MOV     #B14,@DPC       ;XCT BM14, DRAWS A DASHED...
075562  004737  027454                  JSR     PC,WAITF        ;...LINE (4 ON, 4 OFF)...
                                                                ;...FOR 340 PIXELS.
075566  012737  137400  075706          MOV     #BM18!400,B148-2 ;SET OPCODE = BM18 X 400
075574  012777  075700  105230          MOV     #B18,@DPC       ;XCT BM18, DRAWS A DASHED...
075602  004737  027454                  JSR     PC,WAITF        ;...LINE (2 ON, 2 OFF)...
                                                                ;...FOR 400 PIXELS.

075606  012700  000377                  MOV     #377,R0
075612  004737  027014                  JSR     PC,FILL1

075616  013737  003240  075736          MOV     FREE,B048       ; SET BUFFER ADDRESS.
                                ;       MOV     #BM04!M64,B048-2 ; 4 BIT X 64 SQUARE.
075624  012737  134010  075734          MOV     #BM04!M32!EX2,B048-2 ; 4 BIT X 64 SQUARE.
075632  012777  075714  105172          MOV     #B04,@DPC       ; DO IT.
075640  004737  027454                  JSR     PC,WAITF
                                ;       MOV     #BM08!M64,B048-2 ; 8 BIT X 64 SQUARE.
075644  012737  135010  075734          MOV     #BM08!M32!EX2,B048-2 ; 8 BIT X 64 SQUARE.
075652  012777  075726  105152          MOV     #B08,@DPC       ; AND DO THAT TOO.
075660  004737  027454                  JSR     PC,WAITF

075664                          1$:     ;JSR    PC,PAUS.5
075664  000430                          BR      VIS5A           ;...AND CONTINUE.

075666  114000  000400  000240  B14:    APNT,400,240            ; 4 ON 4 OFF FOR 340 PIXELS.
075674  160000  075706                  DJMP, .+10
075700  114000  000300  000340  B18:    APNT,300,340            ; 2 ON 2 OFF FOR 400 PIXELS.
075706  136340                          BM14!340
075710  000000                  B148:   0
075712  173000                          STOPN

075714  114000  000540  001340  B04:    APNT, 540, 1340
075722  160000  075734                  DJMP, .+10
075726  114000  001040  001340  B08:    APNT, 1040, 1340
075734  134001                          BM04!M64
075736  000000                  B048:   0
075740  164000  164000                  DNOP, DNOP
075744  173000                          STOPN
```

```
                                        ;
                                        ; NEXT SHOW RUN-LENGTH MODE DIAGONAL BARS BETWEEN THE BM1 AND BM0 LINES.
                                        ;  USES 16 INTENSITY LEVELS, FROM 0 TO MAX.  THE 0 LEVEL RUN-LENGTH IS VARIED
                                        ;  FROM 4 TO 24(8) FOR 20 LINES OF DATA, WHILE THE OTHER RUN LENGTHS REMAIN
                                        ;  CONSTANT AT 14.  THE LINES ARE DRAWN FROM TOP TO BOTTOM (YDOWN=1).
                                        ;
075746  112737  000004  076027  VIS5A:        MOVB    #4,VRLF2+1      ;SET INITIAL COUNT FOR 0-INTENS AT 4
075754  012777  076016  105050          MOV     #VRLF,@DPC      ;START THE FILE
075762  000403                        BR      2$              ;SKIP OVER THE RESUME
075764  012777  000001  105040  1$:     MOV     #1,@DPC         ;RESUME THE DISPLAY
075772  004737  027454          2$:     JSR     PC,WAITF        ;WAIT FOR DONE
075776  123727  076027  000024          CMPB    VRLF2+1,#24     ;DONE 20(8) LINES?
076004  103003                        BHIS    3$                      ;FINISH IF YES.
076006  105237  076027                INCB    VRLF2+1         ;BUMP INITIAL COUNT IF NO.
076012  000764                        BR      1$                      ;...THEN XCT AGAIN.
076014                          3$:     ;JSR    PC,PAUS.5       ;PAUSE
076014  000430                        BR      VIS6            ;...THEN CONTINUE.
                                        ;
                                        ;DISPLAY FILE FOR VISUAL RUN-LENGTH MODE
076016  114000  000300  000320  VRLF:   APNT,300,320                ;START BETWEEN BM1 & BM0
076024  144040                  VRLF1:  RNLN!YDOWN              ;ENTER RUN-LENGTH MODE, GO DOWN THE SCREEN
076026     000     004          VRLF2:  .BYTE   0,4     ;0 INTENSITY, START AT LENGTH=4
076030  006020  006040  006060          .WORD   6020,6040,6060  ;INCREASING INTENSITY, LENGTH=14
076036  006100  006120  006140          .WORD   6100,6120,6140,6160     ;....AND MORE
076046  006200  006220  006240          .WORD   6200,6220,6240,6260     ;....AND MORE
076056  006300  006320  006340          .WORD   6300,6320,6340,6360     ;....AND MORE
076066  173000                        STOPN           ;STOP
076070  000000                        .WORD   0               ;COUNT = 0 FOR NEWLINE
076072  160000  076026                DJMP,VRLF2              ;THEN JUMP BACK FOR MORE
```

```
                                        ;
                                        ; FINALLY, AT DEAD-CENTER:
                                        ; A CIRCULAR SWEEP FROM 12 O'CLOCK THRU 360 DEGREES.
                                        ; INCREMENT INTENSITY/COLOR LEVEL SUCH THAT ALL LEVELS
                                        ; ARE DISPLAYED IN 360 DEGREES.
                                        ; 3 VECTORS PER LEVEL * 256 LEVELS = 768 VECTORS TOTAL.
                                        ; (OCTAL 3 * 400 = 1400)
                                        ; ACTUAL COLOR/INTENSITY SPECTRUM DEPENDS UPON HOW MANY
                                        ; PIXEL BITS ARE AVAILABLE, AND HOW THE LOOK-UP-RAM IS
                                        ; PROGRAMMED (GREY-SCALE BY DEFAULT, IF INSTALLED).
                                        ;       2 BITS <9:8> = 4 LEVELS.
                                        ;       4 BITS <9:6> = 16 LEVELS (4 PER QUADRANT).
                                        ;       6 BITS <9:4> = 64 LEVELS (LUT REQ'D).
                                        ;       8 BITS <9:2> = 256 LEVELS (LUT REQ'D).
                                        ;
076076  012737  112000  076536  VIS6:         MOV     #LVEC!LO,RBL    ;INIT DISPLAY FILE.
076104  012737  040000  076540                MOV     #I!0,RBX
076112  012737  000300  076542                MOV     #300,RBY
076120  012701  000003                        MOV     #3,R1           ; SET FOR 3 VECTORS PER LEVEL...
076124  012703  000004                        MOV     #4,R3           ;...AND INCR LEVEL BY "1".
076130  010102                        1$:     MOV     R1,R2           ; SCRATCH COPY LEVEL COUNTER.
076132  012777  076530  104672                MOV     #RBO,@DPC       ;START, QUAD 1.
076140  004737  027454                        JSR     PC,WAITF
076144  023727  076542  000000        2$:     CMP     RBY,#0          ;DELTA Y MIN ??
076152  001403                                BEQ     3$              ;DONE QUAD 1 IF SO.
076154  004737  076366                        JSR     PC,Q1
076160  000771                                BR      2$
076162  052737  020000  076542        3$:     BIS     #MXY,RBY        ;QUAD 2, DX POS, DY NEG.
076170  023727  076540  040000        4$:     CMP     RBX,#I!0        ;DELTA X MIN ??
076176  001403                                BEQ     5$              ;DONE QUAD 2 IF SO.
076200  004737  076424                        JSR     PC,Q2
076204  000771                                BR      4$
076206  052737  020000  076540        5$:     BIS     #MXY,RBX        ;QUAD 3, DX NEG, DY NEG.
076214  023727  076542  020000        6$:     CMP     RBY,#MXY!0      ;DELTA Y -MIN ??
076222  001403                                BEQ     7$              ;DONE QUAD 3 IF SO.
076224  004737  076366                        JSR     PC,Q3
076230  000771                                BR      6$
076232  042737  020000  076542        7$:     BIC     #MXY,RBY        ;QUAD 4, DX NEG, DY POS.
076240  023727  076540  060000        8$:     CMP     RBX,#I!MXY!0    ;DELTA X -MIN ??
076246  001403                                BEQ     9$              ;DONE QUAD 4 IF SO.
076250  004737  076424                        JSR     PC,Q4
076254  000771                                BR      8$

076256  012737  004000  003220        9$:     MOV     #4000,SHADLY    ; FINALLY, SET SHADING DELAY...
076264  012705  110224                        MOV     #NTSC8+2,R5     ;...AND CYCLE THRU 16 SHADES...
076270  012501                        10$:    MOV     (R5)+,R1        ;...OF EACH BASIC (NTSC) COLOR.
076272  001403                                BEQ     11$             ; BR WHEN ALL DONE.
076274  004737  027172                        JSR     PC,SHAD16
076300  000773                                BR      10$

076302  004737  104762                11$:    JSR     PC,OUTLIN       ; OUTLINE THE SCREEN.
076306  004737  027514                        JSR     PC,PAUSE1
076312  004737  027514                        JSR     PC,PAUSE1
076316  004737  027514                        JSR     PC,PAUSE1       ; PAUSE 3 SECONDS...
076322  004737  105330                        JSR     PC,EBLINK       ; ENABLE BLINKING.
076326  004737  027514                        JSR     PC,PAUSE1       ;
076332  004737  027514                        JSR     PC,PAUSE1       ; PAUSE 3 SECONS...
```

```
   7961 076336  004737  027514                    JSR     PC,PAUSE1
   7962 076342  004737  105356                    JSR     PC,DBLINK       ; DISABLE BLINKING.
   7963 076346  004737  027514                    JSR     PC,PAUSE1
   7964 076352  004737  027514                    JSR     PC,PAUSE1
   7965 076356  004737  030072          VISXIT:   JSR     PC,TSTEND
   7966 076362                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
7968 076366  013705  076540          Q1:Q3:  MOV     RBX,R5          ;QUAD 1 AND 3, INCR DX 'TIL...
7969 076372  042705  176000                  BIC     #^C1777,R5      ;...MAX, THEN DECR DY.
7970 076376  020527  000300                  CMP     R5,#300         ;DELTA X MAX ??
7971 076402  001404                          BEQ     1$
7972 076404  062737  000002  076540          ADD     #2,RBX          ;NO, INCR DELTA X.
7973 076412  000403                          SKP3
7974 076414  162737  000002  076542  1$:     SUB     #2,RBY          ;YES, DECR DELTA Y.
7975 076422  000416                          BR      QXIT            ;CHECK FOR LEVEL CHANGE AND EXIT
7976
7977 076424  013705  076542          Q2:Q4:  MOV     RBY,R5          ;QUAD 2 AND 4, INCR DY 'TIL...
7978 076430  042705  176000                  BIC     #^C1777,R5      ;...MAX, THEN DECR DX.
7979 076434  020527  000300                  CMP     R5,#300         ;DELTA Y MAX ??
7980 076440  001404                          BEQ     1$
7981 076442  062737  000002  076542          ADD     #2,RBY          ;NO, INCR DELTA Y.
7982 076450  000403                          SKP3
7983 076452  162737  000002  076540  1$:     SUB     #2,RBX          ;YES, DECR DELTA X.
7984
7985 076460  012700  000600          QXIT:   MOV     #600,R0         ; DELAY APPROX 3 MSEC...
7986 076464  005300                  1$:     DEC     R0              ;...BETWEEN VECTORS.
7987 076466  001376                          BNE     1$
7988 076470  005302                          DEC     R2              ; LEVEL CHANGE REQ'D ??
7989 076472  001011                          BNE     2$              ;NOT YET.
7990 076474  010102                          MOV     R1,R2           ;YES, RESET COUNTER...
7991 076476  060337  076536                  ADD     R3,RBL          ;...INCREMENT...
7992 076502  042737  176000  076536          BIC     #^C1777,RBL     ;...AND STRIP LEVEL...
7993 076510  052737  112000  076536          BIS     #LVEC!LO,RBL    ;...REPLACE OPCODE.
7994 076516  005277  104310          2$:     INC     @DPC            ;RESUME...
7995 076522  004737  027454                  JSR     PC,WAITF        ;...WAIT...
7996 076526  000207                          RTS     PC              ;...AND EXIT.
7997                                  ;
7998                                  ; DISPLAY FILE FOR CIRCULAR RAINBOW.
7999                                  ;
8000 076530  114000  000776  000736  RBO:     APNT,HAFX,HAFY60        ;SET ORIGIN AT CENTER.
8001 076536  112000                  RBL:    LVEC!LO                 ;LEVEL WILL INCREMENT.
8002 076540  040002                  RBX:    I!2                     ;DITTO X...
8003 076542  000300                  RBY:    300                     ;...AND Y.
8004 076544  164000  173000  164000          DNOP,STOPN,DNOP
8005 076552  160000  076530                  DJMP,RBO
8006
8007 076556                                  END.TEST

;*****************************************************************************
;*
;*      END TEST 32
;*
;*****************************************************************************
```

```
8009
                                ;*****************************************************************************
                                ;*
                                ;*      BEGIN TEST 33   -   JOY-STICK VERIFICATION (STAND-ALONE)
                                ;*
                                ;*****************************************************************************

                                ; (ITERATION COUNT = 1.)

8010 076656  005737  030070            TST     TESTK
8011 076662  001402                    BEQ     JSV             ; BR IF CLEAR TO RUN.
8012 076664                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
8013                            ;
8014                            ; THIS TEST WILL RUN ONLY IF IT IS SPECIFIED BY ITSELF, AND IF MANUAL
8015                            ; INTERVENTION IS ALLOWED (UAM=0), AND, OBVIOUSLY, ONLY IF A JOY-STICK
8016                            ; IS AVAILABLE.
8017                            ;
8018                            ; ONCE STARTED, THERE IS NO EXIT, SAVE <^C>.
8019                            ;
8020 076670  004737  027702    JSV:    JSR     PC,TSTGO        ; TITLE.
8021
8022 076674                            MANUAL
8023 076676                            BCOMPLETE       JSVA1
8024 076700                            PRINTF  #NOMAN
8025 076720                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
8026 076724                    JSVA1:  PRINTF  #SCOUT
8027 076744  004737  030414    JSVA:   JSR     PC,CONFIG       ; FIND THE CHANNELS
8028 076750  005037  077302            CLR     SELCHA          ; SET SELECTED CHANNEL TO 0 (DEFAULT).
8029 076754                            GMANID  MCHAN,SELCHA,0,1400,0,3,YES
8030 076774  012737  146072  100320    MOV     #CUON!CUIOFF,JCXA
8031 077002  053737  077302  100320    BIS     SELCHA,JCXA
8032 077010  012737  146016  100740    MOV     #CUIM,JCXB
8033 077016  053737  077302  100740    BIS     SELCHA,JCXB
8034 077024  012737  146013  101110    MOV     #CUIS,JCXC
8035 077032  053737  077302  101110    BIS     SELCHA,JCXC
8036 077040  012737  146100  101114    MOV     #CURD,JCXD
8037 077046  053737  077302  101114    BIS     SELCHA,JCXD
8038 077054  013737  077302  100306    MOV     SELCHA,JSPIXA
8039 077062  006237  100306            ASR     JSPIXA
8040 077066  006237  100306            ASR     JSPIXA
8041 077072  052737  175000  100306    BIS     #CUWT,JSPIXA
8042 077100  012737  146100  077152    MOV     #CURD,10$
8043 077106  053737  077302  077152    BIS     SELCHA,10$
8044 077114  013701  003026            MOV     UNITN,R1
8045 077120  013700  077302            MOV     SELCHA,R0       ; GET CHANNEL NO. (BITS 9-8)
8046 077124  000300                    SWAB    R0
8047 077126  006300                    ASL     R0              ; MAKE WORD OFFSET
8048 077130  005760  003270            TST     SYCTAB(R0)      ; IS THERE A JOYSTICK HERE?
8049 077134  100431                    BMI     3$
8050 077136  000414                    BR      2$
8051                            ;       TSTB    STIK(R1)        ;STICK AVAILABLE THIS UNIT ??
8052                            ;       BGT     JSTST           ;YES, GET ON WITH IT.
8053                                    ;*************************************************************
8054                                    ; BEQ   2$              ;NO, CAN'T RUN.
8055                            ;       240, 240
8056                                    ;*************************************************************
8057 077140  012777  077150  103664    MOV     #1$,@DPC        ;DON'T KNOW, LET'S FIND OUT.
```

```
 8058 077146  000403                    SKP3
 8059 077150  164000              1$:   DNOP                    ;TRY TO READ CURSOR POSITION.
 8060 077152  146100  173000      10$:  CURD,STOPN
 8061 077156  004737  027454            JSR     PC,WAITF
 8062 077162  103416                    BCS     3$              ;BR IF DPU RESPONDS TO STICK.
 8063                                    ;*************************************************
 8064                                    ; CLRB  STIK(R1)        ;OTHERWISE, SET FLAG = NO STICK.
 8065                             ;      240, 240
 8066                                    ;*************************************************
 8067 077164  004737  026756            JSR     PC,RELEAS       ;RELEASE "HUNG" DPU...
 8068 077170  006200              2$:   ASR     R0
 8069 077172                            PRINTF  #NOSTIK,R0      ;...TELL THE MAN...
 8070 077214  000137  076744            JMP     JSVA            ; KEEP TRYING (FOREVER) !!
 8071
 8072 077220                      3$:   ;       MOVB    #1,STIK(R1)     ;SET FLAG = STICK AVAILABLE...
 8073 077220  000431                    BR      JSTST           ;...AND START 'EM UP.
 8074
 8075 077222     045     101     040  NOSTIK: .ASCIZ  /%A -- NON-EXISTENT SYNC CHANNEL : %01/
 8076                                    .EVEN
 8077
 8078                                    ;
 8079                                    ; CHANNEL SELECTION QUESTION AND ANSWER.
 8080                                    ;
 8081 077270     103     110     101  MCHAN:  .ASCIZ  /CHANNEL /
 8082                                    .EVEN
 8083 077302  000000              SELCHA: 0                       ; USER SELECTED CHANNEL (DEFAULT=0).
```

```
                                        ;
                                        ; TURN ON THE CURSOR AND DISPLAY IT'S CO-ORDINATES ON SCREEN.
                                        ; ON "STOP" INTERRUPT, UPDATE READ-OUT IF STICK IS MOVING.
                                        ; ON "SWITCH" INTERRUPT, MARK THE CO-ORD WITH A LITTLE "X".
                                        ; ON "MATCH" INTERRUPT, MARK THE SPOT AS BEFORE, AND SAY "MATCH"
                                        ;
077304  004737  027054          JSTST:  JSR     PC,LUMIN        ; BLAST THE LUT.
077310  004737  026650                  JSR     PC,CLRRW        ;INIT IN READ/WRITE MODE.
077314  012700  177777                  MOV     #-1,R0
077320  004737  027014                  JSR     PC,FILL1        ; FILL PIXEL BUFFER.
077324  013737  003240  101062          MOV     FREE,BMXX       ; SET BUFFER POINTER.
077332  012777  077406  103512          MOV     #2$,@STPV       ;SET STOP...
077340  012777  077442  103506          MOV     #4$,@JSMV       ;...MATCH...
077346  012777  077512  103504          MOV     #5$,@JSSV       ;...AND SWITCH VECTORS.
077354  005037  003170                  CLR     SDXR            ;CLEAR OLD JSX,JSY CO-ORDS.
077360  005037  003172                  CLR     SDYR
077364                                  DPSTART #JSPIXA
077364  012777  100306  103440          MOV     #JSPIXA,@DPC    ; START THE DPU.
077372  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
077376  012777  100316  103426  1$:     MOV     #JSPIX,@DPC     ;START THE DISPLAY...
077404  000777                          BR      .               ;...AND WAIT FOR INTERRUPT.
                                        ;
077406                          2$:     BREAK                   ; ON "STOP", BREAK...
077410  004737  077752                  JSR     PC,GTCURS       ;GET & MASK THE CURSOR DATA
077414  023737  100004  003170          CMP     MKDXR,SDXR      ;...AND CHECK STICK MOTION.
077422  001103                          BNE     6$              ; BR IF STICK IS MOVING.
077424  023737  100006  003172          CMP     MKDYR,SDYR
077432  001077                          BNE     6$              ;       DITTO
077434  005277  103372                  INC     @DPC            ; OTHERWISE, JUST RESUME...
077440  000002                          RTI                     ;...AND DISMISS.
                                        ;
077442  052737  001777  100470  4$:     BIS     #1777,MATCH     ; "MATCH" -- ENABLE TEXT.
077450  017737  103362  003170          MOV     @DXR,SDXR       ;SEE THAT NO SIGN BIT SET ON MATCH
077456  100404                          BMI     41$             ;...IF SET, REPORT ERROR
077460  017737  103354  003172          MOV     @DYR,SDYR       ;AND ALSO FOR DYR
077466  100010                          BPL     42$             ;SKIP IF OK
077470                          41$:    HRDERR  DXYME,JXYPE             ;REPORT THE ERROR
077510  000433                  42$:    BR      54$             ;GO PROCESS THE INTERRUPT
077512  042737  001777  100470  5$:     BIC     #1777,MATCH     ; "SWITCH" -- DISABLE TEXT.
077520  004737  023524          51$:    JSR     PC,ERRCHK       ;CHECK FOR ERROR CODE IN CSR
077524  017737  103306  003170          MOV     @DXR,SDXR       ;GET X-COORDINATE & TEST SIGN
077532  100004                          BPL     52$             ;...ERROR IF NOT SET.
077534  017737  103300  003172          MOV     @DYR,SDYR       ;GET Y-COORDINATE & TEST SIGN
077542  100410                          BMI     53$             ;...OK IF SET.
077544                          52$:    HRDERR  JSDXYE,JXYPE            ;REPORT THE ERROR.
077564  042737  174000  003170  53$:     BIC     #174000,SDXR    ;CLEAR SIGN & STATUS ON SAVED X
077572  042737  174000  003172          BIC     #174000,SDYR    ;CLEAR SIGN & STATUS ON SAVED Y
077600                          54$:
077600  013737  100442  100360          MOV     MARK,EMK        ; ON EITHER...
077606  013737  100444  100362          MOV     MARK+2,EMK+2    ;...ERASE OLD MARKER...
077614  013737  003170  100442          MOV     SDXR,MARK
077622  013737  003172  100444          MOV     SDYR,MARK+2     ;...AND SET A NEW ONE.
077630  000411                          BR      62$             ;GOTO COMMON HANDLER.
                                        ;
077632  042737  001777  100470  6$:     BIC     #1777,MATCH     ;STICK MOVING, DISABLE TEXT.
077640  013737  100004  003170          MOV     MKDXR,SDXR
077646  013737  100006  003172          MOV     MKDYR,SDYR      ;SAVE OBSERVED CO-ORDS.
```

```
8140 077654                            62$:
8141 077654  013737  100414  100336            MOV     JSX,EX
8142 077662  013737  100416  100340            MOV     JSX+2,EX+2
8143 077670  013737  100434  100352            MOV     JSY,EY
8144 077676  013737  100436  100354            MOV     JSY+2,EY+2      ;ERASE OLD X/Y.
8145 077704  013701  003170                    MOV     SDXR,R1
8146 077710  004737  027324                    JSR     PC,CNVRT        ;CONVERT X TO ASCII...
8147 077714  010137  100414                    MOV     R1,JSX
8148 077720  010237  100416                    MOV     R2,JSX+2        ;...UPDATE READ-OUT.
8149 077724  013701  003172                    MOV     SDYR,R1
8150 077730  004737  027324                    JSR     PC,CNVRT        ;CONVERT Y TO ASCII...
8151 077734  010137  100434                    MOV     R1,JSY
8152 077740  010237  100436                    MOV     R2,JSY+2        ;...UPDATE READ-OUT.
8153 077744  012716  077376                    MOV     #1$,(SP)        ; SET PC TO RESTART...
8154 077750  000002                            RTI                     ;...AND DISMISS.
8155 077752  017737  103060  100004  GTCURS:   MOV     @DXR,MKDXR
8156 077760  017737  103054  100006            MOV     @DYR,MKDYR
8157 077766  042737  174000  100004            BIC     #174000,MKDXR
8158 077774  042737  174000  100006            BIC     #174000,MKDYR
8159 100002  000207                            RTS     PC
8160 100004  000000                    MKDXR:  0
8161 100006  000000                    MKDYR:  0
8162
8163                                    ;ERROR MESSAGES FOR JOYSTICK OPERATIONS:
8164 100010      040      040      127  JSWTEX: .ASCIZ  /  WAITED TOO LONG FOR J.S. STATUS RETRIEVAL/
8165 100064      040      040      104  JSDXYE: .ASCIZ  /  DXR OR DYR SIGN NOT SET ON J.S. RETRIEVAL/
8166 100140      040      040      104  DXYME:  .ASCIZ  /  DXR OR DYR SIGN SET ON J.S. MATCH INTERRUPT/
8167 100216      045      101      040  JXYPX:  .ASCIZ  /%A   DXR: %06%A  DYR: %06%A%N/
8168                                            .EVEN
8169 100254                            BGNMSG  JXYPE
8170 100254                                    PRINTX  #JXYPX,@DXR,@DYR
8171 100304                                    ENDMSG
8172
```

```
                                                                                    H 1
                                        ;
                                        ; DISPLAY CODE.
                                        ;
100306  175000  000100  000100  JSPIXA: CUWT,100,100
100314  173000                          STOPN

100316  164000                  JSPIX:  DNOP                    ;CURSOR ON, INT'S OFF.
100320  146072                  JCXA:   CUON!CUIOFF
100322  152000  110302                  SETCB,ACAT              ;ASCII BASE ADDRESS.
100326  114000  001620  001600          APNT,1620,1600
100334  102000                          CHAR!LO                 ;ERASE OLD X ...
100336     060     060     060  EX:     .ASCII  /0000/
100342  114000  001620  001540          APNT,1620,1540
100350  100000                          CHAR                    ;...AND Y NUMERICS...
100352     060     060     060  EY:     .ASCII  /0000/
100356  114000                          APNT
100360  000000  000000          EMK:    0,0
100364  104000                          SVEC                    ;...AND OLD MARKER.
100366                                  SXY     I,-10,-10
100370                                  SXY     U,20,0
100372                                  SXY     I,-20,20
100374                                  SXY     U,20,0
100376                                  SXY     I,-10,-10

100400  114000  001460  001600          APNT,1460,1600
100406  103774                          CHAR!ALL                ;DISPLAY NEW X...
100410     112     123     130          .ASCII  /JSX /
100414     060     060     060  JSX:    .ASCII  /0000/
100420  114000  001460  001540          APNT,1460,1540
100426  100000                          CHAR                    ;...AND Y...
100430     112     123     131          .ASCII  /JSY /
100434     060     060     060  JSY:    .ASCII  /0000/
100440  114000                          APNT
100442  000000  000000          MARK:   0,0                     ;...AND NEW MARKER.
100446  104000                          SVEC
100450                                  SXY     I,-10,-10
100452                                  SXY     U,20,0
100454                                  SXY     I,-20,20
100456                                  SXY     U,20,0
100460                                  SXY     I,-10,-10

100462  114000  001540  001500          APNT,1540,1500
100470  102000                  MATCH:  CHAR!LO                 ; "CHAR!ALL" IF MATCH INTERRUPT
100472     040     115     101          .ASCII  / MATCH/

100500  114000  000010  000716          APNT,10,HAFY60-20       ;NOW LABEL THE BOX VECTORS.
100506  103774                          CHAR!ALL
100510     060     060     060          .ASCII  /0000/
100514  114000  000260  000716          APNT,260,HAFY60-20
100522  103774                          CHAR!ALL
100524     060     062     065          .ASCII  /0250/
100530  114000  000530  000716          APNT,530,HAFY60-20
100536  103774                          CHAR!ALL
100540     060     065     062          .ASCII  /0520/
100544  114000  001116  000716          APNT,1256-140,HAFY60-20
100552  103774                          CHAR!ALL
100554     061     062     065          .ASCII  /1256/
```

```
                                                                    I 1
 8231 100560  114000  001366  000716          APNT,1526-140,HAFY60-20
 8232 100566  103774                          CHAR!ALL
 8233 100570     061     065     062          .ASCII  /1526/
 8234 100574  114000  001636  000716          APNT,1776-140,HAFY60-20
 8235 100602  103774                          CHAR!ALL
 8236 100604     061     067     067          .ASCII  /1776/
 8237 100610  114000  000716  000010          APNT,HAFX-60,10
 8238 100616  103774                          CHAR!ALL
 8239 100620     060     060     060          .ASCII  /0000/
 8240 100624  114000  000716  000250          APNT,HAFX-60,250
 8241 100632  103774                          CHAR!ALL
 8242 100634     060     062     064          .ASCII  /0240/
 8243 100640  114000  000716  000510          APNT,HAFX-60,510
 8244 100646  103774                          CHAR!ALL
 8245 100650     060     065     060          .ASCII  /0500/
 8246 100654  114000  000716  001136          APNT,HAFX-60,1176-40
 8247 100662  103774                          CHAR!ALL
 8248 100664     061     061     067          .ASCII  /1176/
 8249 100670  114000  000716  001376          APNT,HAFX-60,1436-40
 8250 100676  103774                          CHAR!ALL
 8251 100700     061     064     063          .ASCII  /1436/
 8252 100704  114000  000716  001636          APNT,HAFX-60,1676-40
 8253 100712  103774                          CHAR!ALL
 8254 100714     061     066     067          .ASCII  /1676/
 8255
 8256 100720  114000  001706  000010          APNT,1776-70,10          ;FINALLY...
 8257 100726  103774                          CHAR!ALL
 8258 100730     107     120                  .ASCII  /GP/             ;...MY INITIALS.
 8259
 8260 100732  164000  164000  164000  BOXES:  DNOP,DNOP,DNOP           ;*** FOR DEBUG ***
 8261 100740  146016                  JCXB:   CUIM                     ; MATCH ON, SWITCH OFF.
 8262 100742  114000  000000  000000          APNT,0,0
 8263 100750  113774                          LVEC!ALL                 ;AND DRAW NESTED BOXES.
 8264 100752                                   LXY     I,1776,0
 8265 100756                                   LXY     I,0,1676
 8266 100762                                   LXY     I,-1776,0
 8267 100766                                   LXY     I,0,-1676
 8268 100772  114000  000250  000240          APNT,250,240
 8269 101000  110000                          LVEC
 8270 101002                                   LXY     I,1256,0
 8271 101006                                   LXY     I,0,1176
 8272 101012                                   LXY     I,-1256,0
 8273 101016                                   LXY     I,0,-1176
 8274 101022  114000  000520  000500          APNT,520,500
 8275 101030  110000                          LVEC
 8276 101032                                   LXY     I,536,0
 8277 101036                                   LXY     I,0,476
 8278 101042                                   LXY     I,-536,0
 8279 101046                                   LXY     I,0,-476
 8280 101052  114000  000676  000636          APNT,HAFX-64.,HAFY60-64.
 8281 101060  134001                          BM04!M64                 ; A 64 SQUARE BIT MAP AT CENTER.
 8282 101062  000000                  BMXX:   0                        ; POINTS TO FREE CORE BUFFER.
 8283 101064  114000  000756  000736          APNT,HAFX-20,HAFY60
 8284 101072  106000                          SVEC!LO                  ; CROSS AT CENTER (BLK ON WHITE)
 8285 101074                                   SXY     I,40,0
 8286 101076                                   SXY     U,-20,20
 8287 101100                                   SXY     I,0,-40
```

```
 8288
 8289 101102  164000  164000  164000          DNOP,DNOP,DNOP          ;*** FOR DEBUG ***
 8290 101110  146013  164003          JCXC:   CUIS,SYNC+2             ;SWITCH ON, MATCH OFF.
 8291 101114  146100  173400          JCXD:   CURD,STOPI              ;READ JSX/JSY, STOP, INTERRUPT.
 8292 101120  164000  164000  164000          DNOP,DNOP,DNOP          ;*** FOR DEBUG ***
 8293 101126  160000  100732                  DJMP,BOXES              ;LOOP
 8294
 8295 101132                                  END.TEST

                                     ;*************************************************************************
                                     ;*
                                     ;*      END TEST 33
                                     ;*
                                     ;*************************************************************************
```

```
8297
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 34   -   SELECTED DISPLAYS (STAND-ALONE)
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 1.)

8298 101224  005737  030070            TST     TESTK
8299 101230  001420                    BEQ     SELDIS          ; BR IF OK TO RUN.
8300 101232                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
8301                            ;
8302                            ; THIS TEST CONSISTS OF A SELECTION OF DISPLAYS INDIVIDUALLY
8303                            ; SELECTABLE BY THE USER AS FOLLOWS:
8304                            ;
8305                            ; DISPLAY DISPATCH TABLE.
8306                            ;
8307 101236  000000            DTAB:   0       ; 0 = "TYPE THIS"
8308 101240  102262                    TBON    ; 1 = TURN BLINKING ON
8309 101242  102320                    TBOFF   ; 2 = TURN BLINKING OFF
8310 101244  102356                    BARS    ; 3 = COLOR BARS
8311 101246  102556                    DOTS    ; 4 = 7 X 7 DOTS
8312 101250  102702                    HATCH   ; 5 = 7 X 7 CROSS-HATCH
8313 101252  103066                    PERIM   ; 6 = PERIMETER OUTLINE
8314 101254  103106                    COLID   ; 7 = BASIC COLOR ID
8315 101256  103360                    GUNID   ; 8 = GUNS ID
8316 101260  104526                    SCRWHT  ; 9 = ALL WHITE SCREEN
8317 101262  104620                    BLOOM   ;10 = "BLOOM" TEST (ALT. WHITE SCREEN & PERIMETER OUTLINE)
8318 101264  104534                    SCRRED  ;11 = ALL RED SCREEN
8319 101266  104542                    SCRBLU  ;12 = ALL BLUE SCREEN
8320 101270  104550                    SCRGRN  ;13 = ALL GREEN SCREEN
8321 101272                    DTABE:
8322         000015            DTHIL=<DTABE-DTAB>/2-1
8323                            ;
8324                            ; IT WILL RUN ONLY IF IT IS SPECIFIED BY ITSELF, AND IF MANUAL
8325                            ; INTERVENTION IS ALLOWED (UAM=0).
8326                            ;
8327                            ; ONCE STARTED, THERE IS NO EXIT, SAVE <^C>.
8328                            ;
8329                            ;
8330                            ; CHANGE MODES VIA KEYBOARD:
8331                            ;
8332                            ;       SELECT DISPLAY (D) N<CR>
8333                            ;
8334                            ;               WHERE N = DISPLAY NUMBER (SEE ABOVE).
8335                            ;
8336 101272                    SELDIS: COMBEG
     101272  004737  027702            JSR     PC,TSTGO        ; TITLE.
     101276  004737  026074            JSR     PC,DPRESET      ; DO SOFT INIT.
8337 101302  004737  027054            JSR     PC,LUMIN        ; BLAST LUT.
8338 101306  004737  102320            JSR     PC,TBOFF        ; START OUT WITH BLINKING TURNED OFF.
8339
8343 101312                            MANUAL
8344 101314                            BCOMPLETE       IDLEX
8345 101316                            PRINTF  #NOMAN
8346 101336                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
                                        ;
                                        ; IDLE LOOP.  WAIT FOR KBD ENTRY TO CHANGE MODES.
                                        ;
101342                          IDLEX:  PRINTF  #SCOUT
101362                          IDLE:   GMANID  SDLIST,DISSEL,D,-1,0,DTHIL,YES
101402  000410                          BR      IDLEB
101404                          IDLEA:  GMANID  SDASK,DISSEL,D,-1,0,DTHIL,YES
101424  005077  101404         IDLEB:  CLR     @DSR    ;STOP THE DPU (IE., FOR BLOOM)
                                ;***B   JSR     PC,WAITF
                                ;***B   JSR     PC,WAITF
                                ;***B   JSR     PC,WAITF
                                ;***B   JSR     PC,WAITF        ;WAIT FOR STOP
101430  012701  000050                  MOV     #40.,R1         ;SET UP A TIMEOUT COUNTER.      ;***B
101434  004737  027454         1$:     JSR     PC,WAITF        ;WAIT FOR THE STOP BIT.         ;***B
101440  103402                          BCS     2$              ;BR IF STOPPED.                 ;***B
101442  005301                          DEC     R1              ;NOT STOPPED. BUMP COUNTER.     ;***B
101444  001373                          BNE     1$              ;LOOP IF NOT STOPPED.           ;***B
101446                          2$:
101446  013700  101470                  MOV     DISSEL,R0
101452  001743                          BEQ     IDLE
101454  005037  101470                  CLR     DISSEL
101460  006300                          ASL     R0
101462  004770  101236                  JSR     PC,@DTAB(R0)
101466  000746                          BR      IDLEA

101470  000000                  DISSEL: 0                       ; DISPLAY CURRENTLY SELECTED.

101472    040     040     060   SDLIST: .ASCII  /  0 = "TYPE THIS"/
101513    015     012                   .BYTE   15,12
101515    040     040     061           .ASCII  /  1 = TURN BLINKING ON/
101543    015     012                   .BYTE   15,12
101545    040     040     062           .ASCII  /  2 = TURN BLINKING OFF/
101574    015     012                   .BYTE   15,12
101576    040     040     063           .ASCII  /  3 = COLOR BARS/
101616    015     012                   .BYTE   15,12
101620    040     040     064           .ASCII  /  4 = 7 X 7 DOTS/
101640    015     012                   .BYTE   15,12
101642    040     040     065           .ASCII  /  5 = 7 X 7 CROSS-HATCH/
101671    015     012                   .BYTE   15,12
101673    040     040     066           .ASCII  /  6 = PERIMETER OUTLINE/
101722    015     012                   .BYTE   15,12
101724    040     040     067           .ASCII  /  7 = BASIC COLOR ID/
101750    015     012                   .BYTE   15,12
101752    040     040     070           .ASCII  /  8 = GUNS ID/
101767    015     012                   .BYTE   15,12
101771    040     040     071           .ASCII  /  9 = ALL WHITE SCREEN/
102017    015     012                   .BYTE   15,12
102021    040     061     060           .ASCII  / 10 = ALTERNATING WHITE SCREEN & PERIMETER OUTLINE/
102103    015     012                   .BYTE   15,12
102105    040     061     061           .ASCII  / 11 = ALL RED SCREEN/
102131    015     012                   .BYTE   15,12
102133    040     061     062           .ASCII  / 12 = ALL BLUE SCREEN/
102160    015     012                   .BYTE   15,12
102162    040     061     063           .ASCII  / 13 = ALL GREEN SCREEN/
102210    015     012                   .BYTE   15,12
102212    015     012           SDASK:  .BYTE   15,12
```

```
 8407 102214      050     102     114            .ASCII  /(BLINKING IS /
 8408 102231      117     106     106     SDO:   .ASCII  /OFF),   /
 8409 102241      123     105     114            .ASCIZ  /SELECT DISPLAY /
 8410                                            .EVEN
 8411
```

```
8413                                        ;
8414                                        ; TURN BLINKING ON.
8415                                        ;
8416 102262  004737  105330                 TBON:   JSR     PC,EBLINK       ; ENABLE BLINKING.
8417 102266  112737  000116  102232                 MOVB    #'N,SDO+1       ; ADJUST PROMPT MSG.
8418 102274  112737  000051  102233                 MOVB    #'),SDO+2
8419 102302  112737  000054  102234                 MOVB    #'.,SDO+3
8420 102310  112737  000040  102235                 MOVB    #' ,SDO+4
8421 102316  000207                                RTS     PC
8422
8423
8424
8425
8426                                        ;
8427                                        ; TURN BLINKING OFF.
8428                                        ;
8429 102320  004737  105356                 TBOFF:  JSR     PC,DBLINK       ; DISABLE BLINKING.
8430 102324  112737  000106  102232                 MOVB    #'F,SDO+1       ; ADJUST PROMPT MSG.
8431 102332  112737  000106  102233                 MOVB    #'F,SDO+2
8432 102340  112737  000051  102234                 MOVB    #'),SDO+3
8433 102346  112737  000054  102235                 MOVB    #'.,SDO+4
8434 102354  000207                                RTS     PC
```

```
                                        ;
                                        ; COLOR BAR GENERATOR.
                                        ; NUMBER AND COLOR OF BARS DEPENDANT ON PIXEL SIZE.
                                        ; ROUTINE DEFAULTS TO 8 BAR NTSC RAINBOW (IF LUT INSTALLED),
                                        ; OR GRAY-SCALE SHADES DEPENDANT ON PIXEL SIZE AS FOLLOWS:
                                        ;       2 BITS <9:8> = 4 COLORS/SHADES.
                                        ;       4 BITS <9:6> = 16 COLORS/SHADES.
                                        ;       6 BITS <9:4> = 64 COLORS/SHADES.
                                        ;       8 BITS <9:2> = 256 COLORS/SHADES.
                                        ;
8446 102356  004737  026650            BARS:   JSR     PC,CLRRW        ;INIT DISPLAY.
8447 102362  004737  027066                    JSR     PC,NTSC         ; NTSC COLORS (IF LUT AVAILABLE)
8448 102366  012737  112000  102466            MOV     #LVEC!LO,BV     ;INIT DISPLAY CODE AT LVL 0...
8449 102374  012701  000004                    MOV     #4,R1           ;...AND INCREMENT BY LVL 1.
8450 102400  012702  000400            1$:     MOV     #256.,R2        ; 256 PAIRS = 512 TOTAL.
8451 102404  012777  102460  100420            MOV     #BGEN,@DPC
8452 102412  000402                            SKP2
8453 102414  005277  100412            2$:     INC     @DPC
8454 102420  004737  027454                    JSR     PC,WAITF
8455 102424  005302                            DEC     R2
8456 102426  001411                            BEQ     3$              ;BR WHEN FINISHED.
8457 102430  060137  102466                    ADD     R1,BV           ; OTHERWISE, INCREMENT...
8458 102434  042737  176000  102466            BIC     #^C1777,BV      ;...AND STRIP THE LEVEL...
8459 102442  052737  112000  102466            BIS     #LVEC!LO,BV     ;...AND REPLACE OPCODE.
8460 102450  000761                            BR      2$              ;...AND LOOP.
8461
8462 102452  004737  104762            3$:     JSR     PC,OUTLIN       ; OUTLINE THE SCREEN.
8463 102456  000207                            RTS     PC              ; RETURN TO IDLE LOOP.
8464
8465 102460  114000  000000  000000    BGEN:   APNT, 0, 0
8466 102466  112000                    BV:     LVEC!LO
8467 102470  040000  001776                    I!0, MAXY               ; UP...
8468 102474  000002  000000                    2, 0                    ;...AND OVER...
8469 102500  040000  021776                    I!0, MXY!MAXY           ;...DOWN...
8470 102504  000002  000000                    2, 0                    ;...AND OVER AGAIN.
8471 102510  173000                            STOPN
8472 102512  160000  102466                    DJMP, BV
```

```
 8474                                   ;
 8475                                   ; 7 X 7 POINT TABLE FOR DOTS AND CROSS-HATCH.
 8476                                   ;
 8477 102516                            XTBL:                   ;X POINTS...
 8478 102516  000104                    Y50TBL: 34.*2           ;...AND 50 HZ Y POINTS.
 8479 102520  000330                            108.*2          ;DELTA = 74.
 8480 102522  000554                            182.*2
 8481 102524  001000                            256.*2
 8482 102526  001224                            330.*2
 8483 102530  001450                            404.*2
 8484 102532  001674                            478.*2
 8485 102534  177777                            -1
 8486 102536  000110                    Y60TBL: 36.*2           ; 60 HZ Y POINTS.
 8487 102540  000320                            104.*2          ;DELTA = 68.
 8488 102542  000530                            172.*2
 8489 102544  000740                            240.*2
 8490 102546  001150                            308.*2
 8491 102550  001360                            376.*2
 8492 102552  001570                            444.*2
 8493 102554  177777                            -1
```

```
                                        ;
                                        ; DOT GENERATOR.
                                        ;
8498 102556  004737  026650      DOTS:  JSR     PC,CLRRW        ;INIT DISPLAY.
8499 102562  012702  102536             MOV     #Y60TBL,R2      ;ASSUME 60 HZ Y'S.
8500 102566  023727  003246  000074     CMP     HZ,#60.
8501 102574  001402                     BEQ     1$
8502 102576  012702  102516             MOV     #Y50TBL,R2      ;WRONG, USE 50 HZ Y'S.
8503
8504 102602  012237  102664      1$:    MOV     (R2)+,DGY       ;SET Y POINT.
8505 102606  100416                     BMI     3$              ;BR WHEN ALL DOTS DONE.
8506 102610  012701  102516             MOV     #XTBL,R1
8507 102614  012137  102662      2$:    MOV     (R1)+,DGX       ;SET X POINT.
8508 102620  100770                     BMI     1$              ;X DONE, GET NEXT Y.
8509 102622  052737  040000  102662     BIS     #I,DGX          ;SET I BIT.
8510 102630  012777  102660  100174     MOV     #DGEN,@DPC
8511 102636  004737  027454             JSR     PC,WAITF
8512 102642  000764                     BR      2$              ;LOOP
8513
8514 102644  012737  000001  105070 3$:  MOV     #1,OUTFLG       ; SAY WE WANT DOTS ON THE OUTLINE
8515 102652  004737  104762             JSR     PC,OUTLIN       ; OUTLINE THE SCREEN.
8516 102656  000207                     RTS     PC              ; RETURN TO IDLE LOOP.
8517
8518 102660  117774             DGEN:  APNT!ALL
8519 102662  040000             DGX:   I!0
8520 102664  000000             DGY:   0
8521 102666  173000                    STOPN           ; *** NOP TO ENLARGE THE DOTS ***
8522 102670  130000                    RPNT
8526 102700  173000                    STOPN
```

```
8528                                    ;
8529                                    ; CROSS-HATCH GENERATOR.
8530                                    ;
8531 102702  004737  026650            HATCH:  JSR     PC,CLRRW        ;INIT DISPLAY.
8532 102706  005037  103052                    CLR     CHX
8533 102712  012737  041776  103060            MOV     #I!MAXX,CHV     ;SET FOR HORIZ VECTORS.
8534 102720  005037  103062                    CLR     CHV+2
8535 102724  012701  102536                    MOV     #Y60TBL,R1      ;ASSUME 60 HZ Y'S.
8536 102730  023727  003246  000074            CMP     HZ,#60.
8537 102736  001402                          BEQ     1$
8538 102740  012701  102516                    MOV     #Y50TBL,R1      ;WRONG AGAIN.
8539
8540 102744  012137  103054            1$:     MOV     (R1)+,CHY       ;SET NEXT Y.
8541 102750  100406                          BMI     2$              ;BR AT END OF Y'S.
8542 102752  012777  103050  100052            MOV     #HGEN,@DPC
8543 102760  004737  027454                    JSR     PC,WAITF
8544 102764  000767                          BR      1$              ;LOOP FOR 7 HORIZ VECTORS.
8545
8546 102766  005037  103054            2$:     CLR     CHY
8547 102772  012737  040000  103060            MOV     #I!0,CHV        ;SET FOR VERT VECTORS.
8548 103000  012737  001776  103062            MOV     #MAXY,CHV+2
8549 103006  012701  102516                    MOV     #XTBL,R1
8550 103012  012137  103052            3$:     MOV     (R1)+,CHX       ;SET NEXT X.
8551 103016  100406                          BMI     4$              ;BR AT END OF X'S.
8552 103020  012777  103050  100004            MOV     #HGEN,@DPC
8553 103026  004737  027454                    JSR     PC,WAITF
8554 103032  000767                          BR      3$              ;LOOP FOR 7 VERT VECTORS.
8555 103034  012737  000001  105070  4$:     MOV     #1,OUTFLG       ; SAY WE WANT DOTS ON THE OUTLINE.
8556 103042  004737  104762                    JSR     PC,OUTLIN       ; OUTLINE THE SCREEN.
8557 103046  000207                          RTS     PC              ; RETURN TO IDLE LOOP.
8558
8559 103050  114000                    HGEN:   APNT
8560 103052  000000                    CHX:    0
8561 103054  000000                    CHY:    0
8562 103056  113774                            LVEC!ALL
8563 103060  041776  000000            CHV:    I!MAXX,0
8564 103064  173000                            STOPN
```

```
8566                                    ;
8567                                    ; DISPLAY A SQUARE, OUTLINING THE PERIMETER OF THE SCREEN.
8568                                    ;
8569 103066  004737  026650            PERIM:  JSR     PC,CLRRW        ; CLEAR FOR READ/WRITE.
8570 103072  012737  000001  105070            MOV     #1,OUTFLG       ; SAY WE WANT DOTS ON THE OUTLINE
8571 103100  004737  104762                    JSR     PC,OUTLIN       ; OUTLINE THE SCREEN.
8572 103104  000207                           RTS     PC              ; RETURN TO IDLE LOOP.
```

```
                                        ;
                                        ; DISPLAY THE FOUR BASIC COLORS (BLUE, RED, LT GREEN, DK GREEN)
                                        ; AND DISPLAY THE NAME OF EACH UNDERNEATH.
                                        ; EACH COLOR BAND WILL BE 1/4 SCREEN WIDTH.
                                        ;
103106  004737  026650          COLID:  JSR     PC,CLRRW        ; CLEAR FOR READ/WRITE.
103112  012737  112000  103240          MOV     #LVEC!LO,11$
103120  012701  000100                  MOV     #100,R1         ; START WITH BLUE.
103124  050137  103240                  BIS     R1,11$          ;
103130  012702  000400                  MOV     #400,R2         ; (R2 DIVIDES SCREEN BY 4 & CHANGES COLORS).
103134  012777  103232  077670          MOV     #10$,@DPC       ; START THE DPU.
103142  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
103146  000405                  3$      BR      3$
                                        ;
103150                          2$:     ;
103150  052777  000001  077654          BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
103156  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
103162  005302                  3$:     DEC     R2              ; ALL DONE?
103164  001412                          BEQ     4$              ; YES.
103166  032702  000077                  BIT     #77,R2          ; NO. TIME TO CHANGE COLOR?
103172  001366                          BNE     2$              ; NO.
103174  006301                          ASL     R1              ; YES.
103176  042737  001700  103240          BIC     #1700,11$       ;
103204  050137  103240                  BIS     R1,11$          ;
103210  000757                          BR      2$              ;
                                        ;                       ; NOW PRINT COLOR ID'S.
103212                          4$:     ;
103212  012777  103270  077612          MOV     #12$,@DPC       ; START THE DPU.
103220  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
103224  004737  104762                  JSR     PC,OUTLIN       ; OUTLINE THE SCREEN.
103230  000207                          RTS     PC              ; RETURN TO IDLE LOOP.
                                        ;
                                        ; DISPLAY CODE.
                                        ;
103232  114000  000000  000000  10$:     APNT,0,0
103240  112000                  11$:     LVEC!LO
103242  040000  001776                  I!0,MAXY
103246  000002  000000                  2,0
103252  040000  021776                  I!0,MXY!MAXY
103256  000002  000000                  2,0
103262  173000                          STOPN
103264  160000  103240                  DJMP,11$
103270  114000  000004  000004  12$:     APNT,4,4
103276  152000  110302                  SETCB,ACAT
103302  103774                          CHAR!ALL
103304     102     114     125          .ASCIZ  /BLUE        RED        LSB-GRN    MSB-GRN /
                                        .EVEN
103356  173000                          STOPN
```

```
                                        ;
                                        ; GUNS ID DISPLAY
                                        ; THIS DISPLAY IS SIMILAR TO THE COLOR BARS DISPLAY, EXCEPT:
                                        ; 1.  THE BARS ARE HORIZONTAL.
                                        ; 2.  THE GUNS ACTIVATED FOR EACH BAR ARE PRINTED INSIDE EACH BAR.
                                        ;
103360  004737  026650          GUNID:  JSR     PC,CLRRW        ; CLEAR FOR READ/WRITE.
103364  012777  103636  077440          MOV     #10$,@DPC       ; START THE DPU.
103372  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
103376  012704  112000                  MOV     #LVEC!LO,R4     ; INIT VECTOR CONTROL.
103402  012702  000020                  MOV     #16.,R2         ; INIT BAR COUNT.
103406  012703  000017          1$:     MOV     #15.,R3         ; INIT VECTORS PER BAR -
103412  023727  003246  000074          CMP     HZ,#60.         ; - 15 FOR 60HZ, 16 FOR 50HZ.
103420  001402                          BEQ     2$
103422  012703  000020                  MOV     #16.,R3
103426  010437  103646          2$:     MOV     R4,11$          ; MODIFY DISPLAY FILE VECTOR CONTROL.
103432                          3$:
103432  052777  000001  077372          BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
103440  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
103444  005303                          DEC     R3              ; THIS BAR COMPLETED?
103446  001371                          BNE     3$              ; NO.
103450  062704  000100                  ADD     #100,R4         ; YES. UPDATE VECTOR CONTROL.
103454  005302                          DEC     R2              ; ALL BARS COMPLETED?
103456  001353                          BNE     1$              ; NO.
103460  012704  000004                  MOV     #4,R4           ; YES. INIT MSG POSITION CONTROL.
103464  012701  103776                  MOV     #30$,R1         ; INIT MESSAGE POINTER.
103470  012702  000020                  MOV     #16.,R2         ; INIT MESSAGE COUNTER.
103474  012737  103774  103720          MOV     #CHAR!ALL,23$   ; INIT DISPLAY FILE INTENSITY CONTROL.
103502  012777  103676  077322          MOV     #20$,@DPC       ; START THE DPU.
103510  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
103514  010437  103712          4$:     MOV     R4,22$          ; MODIFY DISPLAY FILE MSG POSITION CONTROL.
103520  012700  103722                  MOV     #24$,R0         ; MODIFY DISPLAY FILE MESSAGE.
103524  112120                  5$:     MOVB    (R1)+,(R0)+
103526  001376                          BNE     5$
103530  112760  000040  177777          MOVB    #' ,-1(R0)
103536  112720  000040          6$:     MOVB    #' ,(R0)+
103542  020027  103770                  CMP     R0,#25$
103546  001373                          BNE     6$
103550  105060  177777                  CLRB    -1(R0)
103554  052777  000001  077250          BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
103562  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
103566  062704  000074                  ADD     #74,R4          ; UPDATE MSG POSITION CONTROL.
103572  023727  003250  001776          CMP     YMAX,#MAXY50    ; ARE WE RUNNING 50HZ?
103600  001002                          BNE     60$             ; BR IF NO
103602  062704  000004                  ADD     #4,R4           ; YES -- NEED TO BUMP UP 2 MORE PIXELS.
103606  005302                  60$:    DEC     R2              ; ALL MSGS DONE?
103610  001407                          BEQ     7$              ; YES.
103612  020227  000010                  CMP     R2,#10          ; TIME TO FLIP INTENSITY?
103616  001336                          BNE     4$              ; NOT YET.
103620  012737  102000  103720          MOV     #CHAR!LO,23$    ; YES. CHANGE INTENSITY FROM WHITE TO BLACK.
103626  000732                          BR      4$

103630  004737  104762          7$:     JSR     PC,OUTLIN       ; OUTLINE THE SCREEN.
103634  000207                          RTS     PC              ; RETURN TO IDLE LOOP.


                                        ;
                                        ; DISPLAY FILES.
```

```
                                                                I 2
8671                                    ;
8672 103636  114000  000000  000000  10$:    APNT,0,0
8673 103644  173000                        STOPN
8674 103646  112000                  11$:   LVEC!LO                  ; THE COLOR BARS...
8675 103650  041776  000000                 I!MAXX,0
8676 103654  000000  000002                 0,2
8677 103660  061776  000000                 I!MXY!MAXX,0
8678 103664  000000  000002                 0,2
8679 103670  173000                        STOPN
8680 103672  160000  103646                 DJMP,11$
8681
8682 103676  114000  000000  000000  20$:    APNT,0,0
8683 103704  173000                        STOPN
8684 103706  114000                  21$:   APNT                     ; THE IDENTIFICATION MESSAGES.
8685 103710  000004                        4
8686 103712  000004                  22$:   4
8687 103714  152000  110302                 SETCB,ACAT
8688 103720  103774                  23$:   CHAR!ALL
8689 103722     130     130     130  24$:   .ASCIZ  /XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX/
8690 103770  173000                  25$:   STOPN
8691 103772  160000  103706                 DJMP,21$
8692
8693
8694                                    ;
8695                                    ; MESSAGES FOR DISPLAY.
8696                                    ;
8697 103776      050     060     060  30$:   .ASCIZ  /(0000)/
8698 104005      050     060     060        .ASCIZ  /(0001)  BLUE/
8699 104022      050     060     060        .ASCIZ  /(0010)  RED/
8700 104036      050     060     060        .ASCIZ  /(0011)  BLUE, RED/
8701 104060      050     060     061        .ASCIZ  /(0100)  LSB-GRN/
8702 104100      050     060     061        .ASCIZ  /(0101)  BLUE, LSB-GRN/
8703 104126      050     060     061        .ASCIZ  /(0110)  RED, LSB-GRN/
8704 104153      050     060     061        .ASCIZ  /(0111)  BLUE, RED, LSB-GRN/
8705 104206      050     061     060        .ASCIZ  /(1000)  MSB-GRN/
8706 104226      050     061     060        .ASCIZ  /(1001)  BLUE, MSB-GRN/
8707 104254      050     061     060        .ASCIZ  /(1010)  RED, MSB-GRN/
8708 104301      050     061     060        .ASCIZ  /(1011)  BLUE, RED, MSB-GRN/
8709 104334      050     061     061        .ASCIZ  /(1100)  LSB-GRN, MSB-GRN/
8710 104365      050     061     061        .ASCIZ  /(1101)  BLUE, LSB-GRN, MSB-GRN/
8711 104424      050     061     061        .ASCIZ  /(1110)  RED, LSB-GRN, MSB-GRN/
8712 104462      050     061     061        .ASCIZ  /(1111)  BLUE, RED, LSB-GRN, MSB-GRN/
8713                                        .EVEN
```

```
                                        ;
                                        ; DISPLAYS FOR FILLING SCREEN WITH SOLID COLOR
                                        ;
104526  012746  003774          SCRWHT: MOV #ALL,-(SP)
104532  000410                          BR      SCRCOM          ; GO TO COMMON SECTION

104534  012746  002200          SCRRED: MOV     #L0!200,-(SP)   ; GET RED ONTO STACK,
104540  000405                          BR      SCRCOM          ; GO TO COMMON

104542  012746  002100          SCRBLU: MOV     #L0!100,-(SP)   ; GET BLUE ONTO STACK,
104546  000402                          BR      SCRCOM          ; GO TO COMMON

104550  012746  003400          SCRGRN: MOV     #L0!1400,-(SP)  ; GET FULL GREEN ONTO STACK,

104554  004737  026650          SCRCOM: JSR     PC,CLRRW        ;CLEAR FOR READ/WRITE
104560  004737  027066                  JSR     PC,NTSC         ;SETUP COLOR TABLE (IF AVAILABLE)
104564  012737  114000  104612        MOV     #APNT,10$       ;INIT PIXEL DATA SELECTION
104572  052637  104612                BIS     (SP)+,10$       ;INSERT THE COLOR
104576  012777  104612  076226        MOV     #10$,@DPC       ;START THE DPU
104604  004737  027454                JSR     PC,WAITF        ;WAIT FOR DONE
104610  000207                        RTS     PC              ;RETURN TO IDLE

104612  117774          10$:    APNT!ALL
104614  170100                  SETMEM          ;SET THE MEMORY
104616  173000                  STOPN
```

```
                                        ;
                                        ; 'BLOOM'' TEST DISPLAY
                                        ; THIS DISPLAY ALTERNATELY DISPLAYS A FULL WHITE SCREEN AND A PERIMETER OUTLINE.
                                        ; THE DISPLAY CHANGES ONCE EVERY 2 SECONDS.
                                        ;
104620  004737  025402          BLOOM:  JSR     PC,DPINIT       ;INIT THE DPU
104624  004737  102320                  JSR     PC,TBOFF        ;TURN BLINKING OFF
104630  004737  027066                  JSR     PC,NTSC         ;SET UP THE COLOR TABLE (IF AVAILABLE)
104634  013737  003250  104730          MOV     YMAX,12$        ;SET UP THE MAX Y VALUES, BASED ON FREQUENCY.
104642  013737  003250  104740          MOV     YMAX,13$
104650  052737  020000  104740          BIS     #MXY,13$
104656  012777  104666  076146          MOV     #10$,@DPC       ;START THE DISPLAY & LET IT RUN.
104664  000207                          RTS     PC              ;RETURN TO THE IDLE LOOP

                                        ;
                                        ; DISPLAY FILE:
                                        ;
104666  176034                  10$:    WRT!CH0         ;SET ALL CHANNELS TO WRITE-ONLY, SWITCH-ENABLED.
104670  176434                          WRT!CH1
104672  177034                          WRT!CH2
104674  177434                          WRT!CH3
                                ;LOOP TO HERE:
104676  117774  000000  000000  11$:    APNT!ALL,0,0    ;SET WHITE PIXEL DATA
104704  170100                          SETMEM          ;SET MEMORIES TO ALL WHITE
104706  170200                          SWTCH           ;SWITCH TO READ-ONLY TO DISPLAY WHITE SCREEN
104710  164170                          DNOP+120.       ;SHOW IT FOR 2 SECONDS.
104712  170200                          SWTCH           ;SWITCH TO WRITE-ONLY MODE.
104714  170140                          CLRMEM          ;CLEAR MEMORIES.
104716  117774  000000  000000          APNT!ALL,0,0    ;START PERIMETER OUTLINE AT <0,0>
104724  110000                          LVEC            ;ENTER LONG-VECTOR MODE.
104726  040000                          I!0             ;DRAW LEFT EDGE ...
104730  001776                  12$:    MAXY            ;... TO TOP OF VISIBLE AREA.
104732  041776                          I!MAXX          ;DRAW TOP EDGE ...
104734  000000                          0               ;... TO RIGHT SIDE.
104736  040000                          I!0             ;DRAW RIGHT EDGE ...
104740  021776                  13$:    MXY!MAXY        ;... TO BOTTOM.
104742  061776                          I!MXY!MAXX      ;DRAW BOTTOM EDGE ...
104744  000000                          0               ;... BACK TO <0,0>
104746  170200                          SWTCH           ;SWITCH TO READ-ONLY TO DISPLAY PERIMETER
104750  164170                          DNOP+120.       ;SHOW IT FOR 2 SECONDS.
104752  170200                          SWTCH           ;SWITCH BACK TO WRITE-ONLY.
104754  160000  104676                  DJMP,11$        ;LOOP BACK TO DO IT ALL OVER AGAIN.


;*******************************************************************************
;*
;*      END TEST 34
;*
;*******************************************************************************
```

```
                                        ;
                                        ;       SOME MISCELLANEOUS DISPLAY SUBROUTINES.
                                        ;
                                        ;
                                        ; SUBROUTINE TO OUTLINE THE PERIMETER OF THE SCREEN.
                                        ;
105762  013737  003250  105050  OUTLIN: MOV     YMAX,11$        ; SET Y COORDINATES AS PER HZ.
105770  013737  003250  105060          MOV     YMAX,12$
105776  052737  020000  105060          BIS     #MXY,12$
105004  012777  105036  076020          MOV     #10$,@DPC       ; START THE DPU.
105012  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
105016  005737  105070                  TST     OUTFLG          ; MAKE IT DOTTED (3 DOTS PER SIDE)?
105022  001404                          BEQ     1$              ;BR IF NO
105024  004737  105072                  JSR     PC,OUTDOT       ;YES -- GO DO IT
105030  005037  105070                  CLR     OUTFLG          ;THEN CLEAR THE FLAG
105034                          1$:
105034  000207                          RTS     PC
                                        ;
                                        ; DISPLAY CODE.
                                        ;
105036  114000  000000  000000  10$:    APNT,0,0
105044  113774                          LVEC!ALL
105046  040000                          I!0
105050  001776                  11$:    MAXY
105052  041776                          I!MAXX
105054  000000                          0
105056  040000                          I!0
105060  021776                  12$:    MXY!MAXY
105062  061776                          I!MXY!MAXX
105064  000000                          0
105066  173000                          STOPN
                                        ;ROUTINE TO MAKE 3 DOTS IN EACH LINE OF THE PERIMETER:
105070  000000                  OUTFLG: 0                       ;FLAG FOR ENABLING
105072  013737  003250  105162  OUTDOT: MOV     YMAX,OUTDT1     ; SET UP THE DISPLAY FILE
105100  013746  003250                  MOV     YMAX,-(SP)      ; ... WITH FREQ.- DEPENDENT Y DATA.
105104  062716  000002                  ADD     #2,(SP)
105110  006216                          ASR     (SP)
105112  006216                          ASR     (SP)
105114  012637  105302                  MOV     (SP)+,OUTDY1
105120  012777  105134  075704          MOV     #OUTDTF,@DPC    ; START THE DPU.
105126  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
105132  000207                          RTS     PC
                                        ; DISPLAY FILE FOR OUTLINE DOTS:
105134  116000  000000  000000  OUTDTF: APNT!LO,0,0             ; BEGIN AT 0,0 WITH BLACK
105142  160001  105246                  DJMS,OUTDFX             ; DO DOT AT 400,0
105146  160001  105246                  DJMS,OUTDFX             ; ...AND AT 1000,0
105152  160001  105246                  DJMS,OUTDFX             ; ... AND AT 1400,0.
105156  116000                          APNT!LO
105160  000000                          0
105162  001676                  OUTDT1: MAXY60                  ; GETS YMAX
105164  160001  105246                  DJMS, OUTDFX            ; DO DOT AT 400,YMAX
105170  160001  105246                  DJMS, OUTDFX            ; ... AND AT 1000, YMAX
105174  160001  105246                  DJMS, OUTDFX            ; ... AND AT 1400,YMAX
```

```
 8841 105200  116000  000000  000000          APNT!L0,0,0             ; GO AGAIN TO 0,0
 8842 105206  160001  105276                  DJMS, OUTDFY            ; DO A DOT AT 0,YMAX+2/4
 8843 105212  160001  105276                  DJMS, OUTDFY            ; ... AND AT 0, 2*<YMAX+2/4>
 8844 105216  160001  105276                  DJMS, OUTDFY            ; ... AND AT 0, 3*<YMAX+2/4>
 8845 105222  116000  001776  000000          APNT!L0,MAXX,0
 8846 105230  160001  105276                  DJMS, OUTDFY            ; DO A DOT AT 0,YMAX+2/4
 8847 105234  160001  105276                  DJMS, OUTDFY            ; ... AND AT 0, 2*<YMAX+2/4>
 8848 105240  160001  105276                  DJMS, OUTDFY            ; ... AND AT 0, 3*<YMAX+2/4>
 8849 105244  173000                          STOPN
 8850
 8851 105246  110000                OUTDFX:   LVEC
 8852 105250  000376  000000                  MAXX+2/4-2,0            ; SPACE OVER +400-2
 8853 105254  060016  000000                  I!M!14.,0               ; MAKE 8 BLANK PIXELS
 8854 105260  000022  000000                  18.,0                   ; GO TO BEYOND THE TARGET POINT
 8855 105264  040016  000000                  I!14.,0                 ; MAKE 8 BLANK PIXELS
 8856 105270  020020  000000                  M!16.,0                 ; GO BACK TO TARGET
 8857 105274  165000                          DPOP                    ; RETURN
 8858
 8859 105276  112000                OUTDFY:   LVEC!L0
 8860 105300  000000                          0
 8861 105302  000360                OUTDY1:   MAXY60+2/4              ; GETS YMAX+2/4
 8862 105304  000000  020020                  0,M!16.           ; BACK UP
 8863 105310  040000  000040                  I!0,32.           ; ZAP 17 PIXELS
 8864 105314  000000  020020                  0,M!16.           ; BACK UP
 8865 105320  113774                          LVEC!WHITE
 8866 105322  040000  000000                  I!0,0             ; WRITE A DOT
 8867 105326  165000                          DPOP
 8868
 8869
 8870
 8871                                ;
 8872                                ; SUBROUTINE TO ENABLE BLINKING (ALL CHANS).
 8873                                ;
 8874 105330                         EBLINK:
      105330  012777  105344  075474          MOV     #10$,@DPC       ; START THE DPU.
      105336  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 8875 105342  000207                          RTS     PC
 8876
 8877                                ;
 8878                                ; DISPLAY CODE.
 8879                                ;
 8880 105344  175401                10$:      BLINK
 8881 105346  175501                          BLINK+100
 8882 105350  175601                          BLINK+200
 8883 105352  175701                          BLINK+300
 8884 105354  173000                          STOPN
 8885
 8886
 8887
 8888                                ;
 8889                                ; SUBROUTINE TO DISABLE BLINKING (ALL CHANS).
 8890                                ;
 8891 105356                         DBLINK:
      105356  012777  105372  075446          MOV     #10$,@DPC       ; START THE DPU.
      105364  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 8892 105370  000207                          RTS     PC
 8893
```

```
 8894
 8895                                   ;
 8896                                   ; DISPLAY CODE.
 8897 105372  175400                   ;
 8898 105374  175500                   10$:    NBLINK
 8899 105376  175600                           NBLINK+100
 8900 105400  175700                           NBLINK+200
 8901 105402  173000                           NBLINK+300
                                               STOPN
```

```
8903
                                ;*****************************************************************************
                                ;*
                                ;*      BEGIN TEST 35   -   SYSTEM CONFIGURATION TYPEOUT (STAND-ALONE)
                                ;*
                                ;*****************************************************************************

                                ; (ITERATION COUNT = 1.)

8904
8905 105510  005737  030070            TST     TESTK           ; STAND ALONE ENTRY?
8906 105514  001402                    BEQ     SCONTY          ; YES.
8907 105516                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
8908
8909                            ;
8910                            ; THIS "TEST" COMPUTES THE VSV SYSTEM CONFIGURATION AND TYPES IT ON
8911                            ; THE TTY, AFTER WHICH IT WAITS FOR THE USER TO "CONTROL-C"
8912                            ; BACK TO THE MONITOR (DS).
8913                            ;
8914 105522  013701  002012    SCONTY: MOV     L$UNIT,R1       ; GET NUMBER OF LAST UNIT
8915 105526  162701  000001            SUB     #1,R1
8916 105532  006301                    ASL     R1              ; MAKE A WORD OFFSET
8917 105534  032761  040000  003302    BIT     #BIT14,ERTABL(R1)  ; WAS LAST UNIT PREVIOUSLY DROPPED?
8918 105542  001402                    BEQ     .+6             ; BR IF NOT.
8919 105544  000137  106440            JMP     16$             ; GO TO END IF YES -- WE FINISHED.
8920 105550  004737  027702            JSR     PC,TSTGO
8921 105554  005037  030070            CLR     TESTK           ; CLEAR TEST COUNTER SO WE CAN REENTER.
8922 105560  004737  025402            JSR     PC,DPINIT       ; INITIALIZE THE DPU
8923 105564  013701  002012            MOV     L$UNIT,R1
8924 105570  162701  000001            SUB     #1,R1
8925 105574  006301                    ASL     R1
8926 105576  032761  040000  003302    BIT     #BIT14,ERTABL(R1)
8927 105604  001402                    BEQ     .+6
8928 105606  000137  106414            JMP     15$
8929 105612  004737  030414            JSR     PC,CONFIG
8930 105616                            PRINTF  #NULCR
8931 105636  012701  003256            MOV     #MEMTAB,R1      ; INIT MEM TABLE PTR.
8932 105642  012102            1$:     MOV     (R1)+,R2        ; GET A MEM TAB ENTRY.  END OF TABLE?
8933 105644  100467                    BMI     6$              ; YES. (AND R1 NOW PTS TO SYNC CHAN TABLE.)
8934 105646  001775                    BEQ     1$              ; NO. (BUT THERE'S NO MEMORY THERE EITHER.)
8935 105650  005003                    CLR     R3              ; INIT BIT COUNTER.
8936 105652  012704  000010            MOV     #8.,R4          ; INIT SHIFT COUNTER.
8937 105656  005005                    CLR     R5              ; INIT NON-STDD FLAG.
8938 105660  006302                    ASL     R2              ; POSITION THE MEMORY BITS.
8939 105662  006302                    ASL     R2              ;
8940 105664  006302                    ASL     R2              ;
8941 105666  006302                    ASL     R2              ;
8942 105670  006302                    ASL     R2              ;
8943 105672  006302            2$:     ASL     R2              ; SHIFT. GOT A ONE?
8944 105674  100001                    BPL     3$              ; NO.
8945 105676  005203                    INC     R3              ; YES. COUNT IT.
8946 105700  005304            3$:     DEC     R4              ; DONE ENOUGH SHIFTING?
8947 105702  001373                    BNE     2$              ; NO.
8948 105704  016102  177776            MOV     -2(R1),R2       ; YES. NOW CHECK IF ALL ONE BITS ARE -
8949 105710  010305                    MOV     R3,R5           ; - CONSECUTIVE, STARTING AT BIT 9 -
8950 105712  006302                    ASL     R2              ; - (IF NOT, WE HAVE A NON-STANDARD -
8951 105714  006302                    ASL     R2              ; - MEMORY.)
```

```
8952 105716  006302                    ASL     R2              ;
8953 105720  006302                    ASL     R2              ;
8954 105722  006302                    ASL     R2              ;
8955 105724  006302            4$:     ASL     R2              ;
8956 105726  100002                    BPL     5$              ;
8957 105730  005305                    DEC     R5              ;
8958 105732  001374                    BNE     4$              ;
8959 105734  010102            5$:     MOV     R1,R2           ; GET NUMBER OF THIS MEM IN R2.
8960 105736  162702  003260            SUB     #MEMTAB+2,R2    ;
8961 105742  006202                    ASR     R2              ;
8962 105744  016104  177776            MOV     -2(R1),R4       ; GET TABLE ENTRY FOR THIS MEM IN R4.
8963 105750                            PRINTF  #SCMEM,R2,R4,R3 ; PRINT DATA FOR THIS MEM.
8964 105776  005705                    TST     R5              ; NON-STANDARD?
8965 106000  001720                    BEQ     1$              ; NO. NEXT MEM.
8966 106002                            PRINTF  #SCNSTD         ; YES. TELL THE WORLD!.
8967 106022  000707                    BR      1$              ; NEXT MEM.
8968
8969 106024  012102            6$:     MOV     (R1)+,R2        ; GET SYNC CHAN TABLE ENTRY. END OF TABLE?
8970 106026  100376                    BPL     6$              ; NO. (BUT NO SYNC CCHAN THERE).
8971 106030  020227  177777            CMP     R2,#-1          ; MAYBE. IS IT EOT OR A SYNC CHAN?
8972 106034  001443                    BEQ     8$              ; EOT.
8973 106036  010103                    MOV     R1,R3           ; SYNC CHAN. GET ITS NUMBER IN R3.
8974 106040  162703  003272            SUB     #SYCTAB+2,R3    ;
8975 106044  006203                    ASR     R3              ;
8976 106046                            PRINTF  #SCSYC,R3,R2    ; PRINT DATA FOR THIS SYNC CHAN.
8977 106072  032702  020000            BIT     #BIT13,R2       ; INTERLACED?
8978 106076  001011                    BNE     7$              ; YES.
8979 106100                            PRINTF  #SCNINT         ; NO.
8980 106120  000741                    BR      6$              ; NEXT SYNC CHAN.
8981
8982 106122            7$:     PRINTF  #SCINT          ;
8983 106142  000730            BR      6$              ; NEXT SYNC CHAN.
8984
8985 106144  005737  002512    8$:     TST     MFGFLG          ; MFG MODE?
8986 106150  001521                    BEQ     15$             ; NO.
8987 106152  013702  003022            MOV     MFGM0,R2        ; GET MFG MASTER FOR MEM 0.
8988 106156  005003                    CLR     R3              ; INIT BIT COUNTER.
8989 106160  012704  000010            MOV     #8.,R4          ; INIT SHIFT COUNTER.
8990 106164  005005                    CLR     R5              ; INIT NON-STDD FLAG.
8991 106166  006302                    ASL     R2              ; POSITION THE MEMORY BITS.
8992 106170  006302                    ASL     R2              ;
8993 106172  006302                    ASL     R2              ;
8994 106174  006302                    ASL     R2              ;
8995 106176  006302                    ASL     R2              ;
8996 106200  006302            9$:     ASL     R2              ; SHIFT. GOT A ONE?
8997 106202  100001                    BPL     10$             ; NO.
8998 106204  005203                    INC     R3              ; YES. COUNT IT.
8999 106206  005304            10$:    DEC     R4              ; DONE ENOUGH SHIFTING?
9000 106210  001373                    BNE     9$              ; NO.
9001 106212  016102  177776            MOV     -2(R1),R2       ; YES. NOW CHECK IF ALL ONE BITS ARE -
9002 106216  010305                    MOV     R3,R5           ; - CONSECUTIVE, STARTING AT BIT 9 -
9003 106220  006302                    ASL     R2              ; - (IF NOT, WE HAVE A NON-STANDARD -
9004 106222  006302                    ASL     R2              ; - MEMORY.)
9005 106224  006302                    ASL     R2              ;
9006 106226  006302                    ASL     R2              ;
9007 106230  006302                    ASL     R2              ;
9008 106232  006302            11$:    ASL     R2              ;
```

```
9009 106234  100002                    BPL     12$             ;
9010 106236  005305                    DEC     R5              ;
9011 106240  001374                    BNE     11$             ;
9012 106242  013704  003022      12$:   MOV     MFGM0,R4        ; GET TABLE ENTRY FOR THIS MEM IN R4.
9013 106246                            PRINTF  #SCMMM0,R4,R3   ; PRINT DATA FOR THIS MEM.
9014 106272  005705                    TST     R5              ; NON-STANDARD?
9015 106274  001410                    BEQ     13$             ;
9016 106276                            PRINTF  #SCNSTD         ; YES. TELL THE WORLD!.
9017 106316  013702  003024      13$:   MOV     MFGS0,R2        ; GET MFG MASTER SYNC CHAN 0 IN R2.
9018 106322                            PRINTF  #SCMMS0,R2      ; PRINT DATA FOR THIS SYNC CHAN.
9019 106344  032702  020000            BIT     #BIT13,R2       ; INTERLACED?
9020 106350  001011                    BNE     14$             ; YES.
9021 106352                            PRINTF  #SCNINT         ; NO.
9022 106372  000410                    BR      15$             ;
9023
9024 106374                    14$:    PRINTF  #SCINT          ;
9025 106414  013701  002012    15$:    MOV     L$UNIT,R1       ; GET NUMBER OF UNITS
9026 106420  162701  000001            SUB     #1,R1
9027 106424  006301                    ASL     R1              ; MAKE A WORD OFFSET
9028 106426  005761  003302            TST     ERTABL(R1)      ; DID LAST UNIT RUN?
9029 106432  001002                    BNE     16$             ; BR IF YES
9030 106434                    EXIT    TST                     ; NO -- EXIT TO DO ANOTHER.
9031 106440                    16$:    PRINTF  #SCOUT          ;
9032 106460                    17$:    BREAK                   ; WAIT FOR USER TO <^C>.
9033 106462  000776                    BR      17$             ;
9034
9035 106464      045     116     045   SCMEM:  .ASCIZ  /%N%AMEMORY:    %D1%A,  MEMTAB VAL: %06%A,  %D1%A BITS/
9036 106552      045     101     054   SCNSTD: .ASCIZ  /%A,  (NON-STD)/
9037 106571      045     116     045   SCSYC:  .ASCIZ  /%N%ASYNC CHAN: %01%A,  SYCTAB VAL: %06/
9038 106640      045     101     054   SCINT:  .ASCIZ  /%A,  INTERLACED/
9039 106660      045     101     054   SCNINT: .ASCIZ  /%A,  NON-INTERLACED/
9040 106704      045     116     045   SCMMM0: .ASCIZ  /%N%AMFG MASTER MEM 0,  MEMTAB VAL: %06%A,  %D1%A BITS/
9041 106772      045     116     045   SCMMS0: .ASCIZ  /%N%AMFG SYNC CHAN 0,  SYCTAB VAL: %06/
9042 107040      045     116     045   SCOUT:  .ASCIZ  /%N%N%A   (<^C> TO GET BACK TO 'DR>')%N/
9043                                            .EVEN
9044
9045 107110                            END.TEST

                                ;*******************************************************************************
                                ;*
                                ;*      END TEST 35
                                ;*
                                ;*******************************************************************************
```

```
9047                                    .SBTTL
9048                                    .SBTTL  SUPERVISORS DISPATCH TABLE
9049
9050                                    ;++
9051                                    ; THIS TABLE HOLDS THE STARTING ADDRESS OF EACH TEST
9052                                    ; FOR THE SUPERVISOR'S DISPATCHER.
9053                                    ;--
9054
```

```
                                        .SBTTL  *** LUMINANCE (GREY-SCALE) AND COLOR DATA TABLES
                                        ;
                                        ; TABLE OF 256 (8 BITS) LUMINANCE VALUES TO PRODUCE
                                        ; A LINEAR 12 BIT GREY-SCALE.
                                        ;
107222                                  LUMTBL:
107222  000000  000400  001000  0,      400,    1000,   1
107232  001400  000401  002000  1400,   401,    2000,   1001
107242  002400  000002  003000  2400,   2,      3000,   402
107252  003400  002401  004000  3400,   2401,   4000,   3
107262  003001  002020  005000  3001,   2020,   5000,   2420
107272  000004  005400  003002  4,      5400,   3002,   6000
107302  001040  006400  002403  1040,   6400,   2403,   5
107312  003003  002040  007400  3003,   2040,   7400,   5020    ; LEVEL 37
107322  002404  000060  000006  2404,   60,     6,      1441
107332  003004  002023  001042  3004,   2023,   1042,   2405
107342  000007  004440  002024  7,      4440,   2024,   1061
107352  000100  000010  000462  100,    10,     462,    2025
107362  001044  002407  000011  1044,   2407,   11,     501
107372  007440  001063  000120  7440,   1063,   120,    12
107402  000502  003500  001064  502,    3500,   1064,   2411
107412  005407  000503  002030  5407,   503,    2030,   1065    ; LEVEL 77
107422  000140  000014  000504  140,    14,     504,    2031
107432  001066  002413  000015  1066,   2413,   15,     505
107442  007444  002540  000160  7444,   2540,   160,    16
107452  003014  003504  002541  3014,   3504,   2541,   125
107462  000017  002160  002034  17,     2160,   2034,   200
107472  000126  003142  002161  126,    3142,   2161,   3506
107502  001072  001600  007066  1072,   1600,   7066,   2162
107512  002036  000220  001601  2036,   220,    1601,   3126    ; LEVEL 137
107522  002163  005161  002563  2163,   5161,   2563,   131
107532  004600  000513  005162  4600,   513,    5162,   240
107542  000132  003146  002165  132,    3146,   2165,   1222
107552  004220  004074  004602  4220,   4074,   4602,   2166
107562  003513  000260  004075  3513,   260,    4075,   3132
107572  003640  001152  002477  3640,   1152,   2477,   3223
107602  000607  003567  000300  607,    3567,   300,    210
107612  005476  002243  005241  5476,   2243,   5241,   1154    ; LEVEL 177
107622  004152  004660  003153  4152,   4660,   3153,   2172
107632  003517  000320  004117  3517,   320,    4117,   646
107642  002173  001230  002555  2173,   1230,   2555,   157
107652  000647  007624  000340  647,    7624,   340,    2574
107662  005554  002321  002213  5554,   2321,   2213,   2703
107672  000267  003265  004612  267,    3265,   4612,   3647
107702  000360  001741  003266  360,    1741,   3266,   4631
107712  002177  005213  005703  2177,   5213,   5703,   1634    ; LEVEL 237
107722  000671  003651  005176  671,    3651,   5176,   272
107732  001617  006266  001362  1617,   6266,   1362,   4360
107742  000273  007232  000655  273,    7232,   655,    3653
107752  001237  004235  004743  1237,   4235,   4743,   1256
107762  007615  001312  005237  7615,   1312,   5237,   3345
107772  000731  007616  002712  731,    7616,   2712,   1747
110002  006235  002331  003656  6235,   2331,   3656,   4364
110012  000277  003275  002332  277,    3275,   2332,   5330    ; LEVEL 277
110022  002714  005712  003276  2714,   5712,   3276,   7745
110032  001370  004366  007364  1370,   4366,   7364,   3277
110042  002334  002371  004367  2334,   2371,   4367,   5714
```

```
 9114 110052  004751  006276  001372  4751,   6276,   1372,   2717
 9115 110062  005715  004752  006277  5715,   4752,   6277,   1373
 9116 110072  004371  007367  004753  4371,   7367,   4753,   2337
 9117 110102  001374  004372  005717  1374,   4372,   5717,   4754
 9118 110112  007752  001375  004373  7752,   1375,   4373,   1757    ; LEVEL 337
 9119 110122  004755  007753  005337  4755,   7753,   5337,   4374
 9120 110132  007372  003357  006355  7372,   3357,   6355,   7754
 9121 110142  005374  002776  005774  5374,   2776,   5774,   7337
 9122 110152  006356  005357  004376  6356,   5357,   4376,   7374
 9123 110162  004776  007756  006757  4776,   7756,   6757,   4377
 9124 110172  007375  006376  007757  7375,   6376,   7757,   2777
 9125 110202  006776  005777  006376  6776,   5777,   6376,   6377
 9126 110212  007776  006777  007377  7776,   6777,   7377,   7777    ; LEVEL 377
 9127                                 ;
 9128                                 ; THE BASIC (NTSC) COLORS, PLUS BLACK AND WHITE.
 9129                                 ;
 9130         007400                  BLU=    17*BIT8
 9131         000017                  RED=    17*BIT0
 9132         000360                  GRN=    17*BIT4
 9133
 9134 110222  000000                  NTSC8:  0               ; BLACK (NO COLOR AT ALL).
 9135 110224  007400                          BLU
 9136 110226  000017                          RED
 9137 110230  007417                          RED+BLU         ; MAGENTA
 9138 110232  000360                          GRN
 9139 110234  007760                          GRN+BLU         ; CYAN
 9140 110236  000377                          GRN+RED         ; YELLOW
 9141 110240  007777                          GRN+RED+BLU     ; WHITE
 9142                                 ;
 9143                                 ; A 16 LEVEL VARIATION OF THE ABOVE TO EXPERIMENT WITH.
 9144                                 ;
 9145 110242  000000                  EXPRM:  0
 9146 110244  000043                          43              ; BROWN (3RD LEVEL YELLOW).
 9147 110246  000000                          0
 9148 110250  000017                          RED             ; RED
 9149 110252  000000                          0
 9150 110254  000117                          117             ; ORANGE (RED + HI GREEN)
 9151 110256  000000                          0
 9152 110260  000377                          RED+GRN         ; YELLOW
 9153 110262  000000                          0
 9154 110264  000360                          GRN             ; GREEN
 9155 110266  000000                          0
 9156 110270  007400                          BLU             ; BLUE
 9157 110272  000000                          0
 9158 110274  007417                          RED+BLU         ; VIOLET (MAGENTA)
 9159 110276  000000                          0
 9160 110300  007777                          BLU+RED+GRN     ; WHITE.
```

```
                                .SBTTL  *** ASCII CHARACTER ADDRESS TABLE AND SUBPIX
                                ;
                                ;THE FOLLOWING 128. WORDS CONTAIN THE CHARACTER SUBPIX
                                ;STARTING ADDRESSES.  0-37 AND 140-177 ARE NULL FOR NOW.
                                ;       (EXCEPT FOR 12 AND 15)
                                ;
110302  110702                  ACAT:   ..NULL          ;0-37 ARE NULL (NLISTED).
110402                                  $SVPC=.         ;SAVE PC, BACK UP AND...
110326                                  .=ACAT+<12*2>
110326  110704                          ..LF            ;...INSERT LINE FEED...
110334                                  .=ACAT+<15*2>
110334  110714                          ..CR            ;...AND CARRIAGE RETURN ADDRESSES.
110402                                  .=$SVPC         ;RESTORE PC

110402  110730                          ..SPC           ;SPACE
110404  110736                          ..EXC           ;EXCLAM
110406  110754                          ..QUO           ;DOUBLE QUOTE.
110410  110772                          ..NUM           ;NUMBER SIGN
110412  111020                          ..DOL           ;DOLLAR
110414  111052                          ..PCT           ;PER CENT
110416  111112                          ..AND           ;AMPERSAND
110420  111152                          ..QUO1          ;SINGLE QUOTE
110422  111164                          ..LPAR          ;LFT PAREN
110424  111202                          ..RPAR          ;RT PAREN
110426  111220                          ..AST           ;ASTERISK
110430  111242                          ..PLS           ;PLUS
110432  111260                          ..CMA           ;COMMA
110434  111272                          ..MNS           ;MINUS
110436  111304                          ..DOT           ;PERIOD
110440  111314                          ..SLSH          ;SLASH
110442  111330                          ..00
110444  111362                          ..11
110446  111402                          ..22
110450  111434                          ..33
110452  111474                          ..44
110454  111514                          ..55
110456  111544                          ..66
110460  111576                          ..77
110462  111616                          ..88
110464  111666                          ..99
110466  111716                          ..COLN          ;COLON
110470  111734                          ..SEMI          ;SEMI COLON
110472  111752                          ..LANG          ;LFT ANGLE
110474  111766                          ..EQ            ;EQUALS
110476  112004                          ..RANG          ;RT ANGLE
110500  112016                          ..QUES          ;QUESTION
110502  112044                          ..ATS           ;AT SIGN
110504  112100                          ..AA
110506  112124                          ..BB
110510  112160                          ..CC
110512  112206                          ..DD
110514  112230                          ..EE
110516  112236                          ..FF
110520  112254                          ..GG
110522  112272                          ..HH
110524  112312                          ..II
```

```
    9224 110526  112334                    ..JJ
    9225 110530  112354                    ..KK
    9226 110532  112372                    ..LL
    9227 110534  112406                    ..MM
    9228 110536  112424                    ..NN
    9229 110540  112444                    ..OO
    9230 110542  112474                    ..PP
    9231 110544  112516                    ..QQ
    9232 110546  112532                    ..RR
    9233 110550  112546                    ..SS
    9234 110552  112604                    ..TT
    9235 110554  112622                    ..UU
    9236 110556  112644                    ..VV
    9237 110560  112664                    ..WW
    9238 110562  112704                    ..XX
    9239 110564  112730                    ..YY
    9240 110566  112754                    ..ZZ
    9241 110570  112776                    ..LBRK          ;LFT BRACKET
    9242 110572  113014                    ..BSLH          ;BACK SLASH
    9243 110574  113032                    ..RBRK          ;RT BRACKET
    9244 110576  113050                    ..CRT           ;CARROT
    9245 110600  113064                    ..USC           ;UNDERSCORE
    9246 110602  110702                    ..NULL          ;140-177 ARE NULL (NLISTED).
```

```
                                ; THESE ARE THE CHAR SUBPIX ROUTINES.
                                ; EACH IS BUILT ON A 5 X 7 DOT MATRIX (ALA VT05).
                                ; <CR> WILL INVOKE A "STOP".  CPU MUST CALCULATE THE START
                                ; CO-ORDS FOR NEXT LINE, STUFF IN "..CRX AND ..CRX+2" AND
                                ; RESUME, TO XCT PSUEDO-CRLF.
                                ;
                                ; THE FOLLOWING MACRO (SXY4) APPLIES A 4X SCALE FACTOR AT
                                ; ASSEMBLY TIME TO YIELD A CHAR SIZE OF 30 X 40 (OCTAL)
                                ; WHICH PRODUCES A MEDIUM SIZE CHAR FONT.
                                ;
                                        .MACRO  SXY4 A,B,C
                                        SXY     A,B*4,C*4
                                        .ENDM   SXY4

110702  165000          ..NULL: DPOP                    ;ALL UNDEFINED CHARS EXIT HERE.

110704  164000          ..LF:   DNOP                    ;DPOP TO NULL THE <LF>.
110706  104000                  SVEC
110712  165000                  DPOP

110714  164000          ..CR:   DNOP                    ;DPOP TO NULL THE <CRLF>.
110716  173000                  STOPN                   ;WAIT FOR NEW X AND Y...
110720  114000                  APNT                    ;...OK, EXECUTE THE <CRLF>
110722  000000  000000  ..CRX:  .WORD   0,0
110726  165000                  DPOP

110730  104000          ..SPC:  SVEC
110734  165000                  DPOP
110736  104000          ..EXC:  SVEC
110752  165000                  DPOP
110754  104000          ..QUO:  SVEC
110770  165000                  DPOP
110772  104000          ..NUM:  SVEC
111016  165000                  DPOP
111020  104000          ..DOL:  SVEC
111050  165000                  DPOP
111052  104000          ..PCT:  SVEC
111110  165000                  DPOP
111112  104000          ..AND:  SVEC
111150  165000                  DPOP
111152  104000          ..QUO1: SVEC
111162  165000                  DPOP
111164  104000          ..LPAR: SVEC
111200  165000                  DPOP
111202  104000          ..RPAR: SVEC
111216  165000                  DPOP
111220  104000          ..AST:  SVEC
111240  165000                  DPOP
111242  104000          ..PLS:  SVEC
111256  165000                  DPOP
111260  104000          ..CMA:  SVEC
111270  165000                  DPOP
111272  104000          ..MNS:  SVEC
111302  165000                  DPOP
111304  104000          ..DOT:  SVEC
111312  165000                  DPOP
111314  104000          ..SLSH: SVEC
```

```
 9407 111326  165000                          DPOP
 9408 111330  104000                ..00:     SVEC
 9420 111360  165000                          DPOP
 9421 111362  104000                ..11:     SVEC
 9428 111400  165000                          DPOP
 9429 111402  104000                ..22:     SVEC
 9441 111432  165000                          DPOP
 9442 111434  104000                ..33:     SVEC
 9457 111472  165000                          DPOP
 9458 111474  104000                ..44:     SVEC
 9465 111512  165000                          DPOP
 9466 111514  104000                ..55:     SVEC
 9477 111542  165000                          DPOP
 9478 111544  104000                ..66:     SVEC
 9490 111574  165000                          DPOP
 9491 111576  104000                ..77:     SVEC
 9498 111614  165000                          DPOP
 9499 111616  104000                ..88:     SVEC
 9518 111664  165000                          DPOP
 9519 111666  104000                ..99:     SVEC
 9530 111714  165000                          DPOP
 9531 111716  104000                ..COLN:   SVEC
 9537 111732  165000                          DPOP
 9538 111734  104000                ..SEMI:   SVEC
 9544 111750  165000                          DPOP
 9545 111752  104000                ..LANG:   SVEC
 9550 111764  165000                          DPOP
 9551 111766  104000                ..EQ:     SVEC
 9557 112002  165000                          DPOP
 9558 112004  104000                ..RANG:   SVEC
 9562 112014  165000                          DPOP
 9563 112016  104000                ..QUES:   SVEC
 9573 112042  165000                          DPOP
 9574 112044  104000                ..ATS:    SVEC
 9587 112076  165000                          DPOP
 9588 112100  104000                ..AA:     SVEC
 9597 112122  165000                          DPOP
 9598 112124  104000                ..BB:     SVEC
 9611 112156  165000                          DPOP
 9612 112160  104000                ..CC:     SVEC
 9622 112204  165000                          DPOP
 9623 112206  104000                ..DD:     SVEC
 9631 112226  165000                          DPOP
 9632 112230  104000                ..EE:     SVEC
 9635 112236  104000                ..FF:     SVEC
 9641 112252  165000                          DPOP
 9642 112254  104000                ..GG:     SVEC
 9647 112266  160000  112160                  .WORD   DJMP,..CC       ;FINISH USING ''C''.
 9648 112272  104000                ..HH:     SVEC
 9655 112310  165000                          DPOP
 9656 112312  104000                ..II:     SVEC
 9664 112332  165000                          DPOP
 9665 112334  104000                ..JJ:     SVEC
 9672 112352  165000                          DPOP
 9673 112354  104000                ..KK:     SVEC
 9679 112370  165000                          DPOP
 9680 112372  104000                ..LL:     SVEC
```

```
 9685 112404  165000                        DPOP
 9686 112406  104000              ..MM:     SVEC
 9692 112422  165000                        DPOP
 9693 112424  104000              ..NN:     SVEC
 9700 112442  165000                        DPOP
 9701 112444  104000              ..OO:     SVEC
 9712 112472  165000                        DPOP
 9713 112474  104000              ..PP:     SVEC
 9721 112514  165000                        DPOP
 9722 112516  104000              ..QQ:     SVEC
 9726 112526  160000  112444                .WORD   DJMP,..OO       ;FINISH USING 'O'.
 9727 112532  104000              ..RR:     SVEC
 9731 112542  160000  112474                .WORD   DJMP,..PP       ;FINISH USING 'P'.
 9732 112546  104000              ..SS:     SVEC
 9746 112602  165000                        DPOP
 9747 112604  104000              ..TT:     SVEC
 9753 112620  165000                        DPOP
 9754 112622  104000              ..UU:     SVEC
 9762 112642  165000                        DPOP
 9763 112644  104000              ..VV:     SVEC
 9770 112662  165000                        DPOP
 9771 112664  104000              ..WW:     SVEC
 9778 112702  165000                        DPOP
 9779 112704  104000              ..XX:     SVEC
 9788 112726  165000                        DPOP
 9789 112730  104000              ..YY:     SVEC
 9798 112752  165000                        DPOP
 9799 112754  104000              ..ZZ:     SVEC
 9807 112774  165000                        DPOP
 9808 112776  104000              ..LBRK:   SVEC
 9814 113012  165000                        DPOP
 9815 113014  104000              ..BSLH:   SVEC
```

```
   9822 113030  165000                        DPOP
   9823 113032  104000              ..RBRK:   SVEC
   9829 113046  165000                        DPOP
   9830 113050  104000              ..CRT:    SVEC
   9835 113062  165000                        DPOP
   9836 113064  104000              ..USC:    SVEC
   9839 113072  165000                        DPOP
```

```
                                .SBTTL  *** DEVELOPMENT VERSIONS CONTROL

                                ; THE FOLLOWING NUMBERS GET CHANGED AT EACH NEW ASSEMBLY, THEREFORE:
                                ; IF THESE NUMBERS DON'T MATCH WHAT'S IN CORE (OCTALLY) THEN THIS
                                ; LISTING DOESN'T MATCH THE LOADED PROGRAM!

113074  021557                  DDMMY:  09071.
113076  002165                  HHMM:   1141.


                                .SBTTL  *** PATCH AREA
                                .SBTTL

                                ;
                                ; FINALLY A GENEROUS PATCH AREA.
                                ;
                                ; AND AN ADJUSTMENT TO ACCOUNT FOR THE "LASTAD BIT7" HACK
                                ; DESCRIBED IN "SUPPRG.MEM" (FOR REV C).
                                ;
113100                          PATCH:  .BLKW   16.

                                ;       .BLKB   <<.+400>&^C377>-.
                                ;       .=      .
        113400                  .=.!377+1
113404                          L$LAST::
```

```
 9873                                   .SBTTL  DEFAULT HARDWARE P-TABLE
 9874
 9875                                   ;++
 9876                                   ; THE DEFAULT HARDWARE P-TABLE CONTAINS DEFAULT VALUES OF
 9877                                   ; THE TEST-DEVICE PARAMETERS.  THE STRUCTURE OF THIS TABLE
 9878                                   ; IS IDENTICAL TO THE STRUCTURE OF THE RUN-TIME P-TABLE.
 9879                                   ;--
 9882 113410  172000                           .WORD   172000          ; 1ST (OF 4) REGISTER(S).
 9883                                                                  ;       ( 7 IF LUT INSTALLED ).
 9884 113412  000320                           .WORD   320             ; 1ST (OF 4) VECTOR(S).
 9885                                                                  ;       ( 5 IF LUT INSTALLED ).
 9886 113414  000200                           .WORD   PRI04           ; INTERRUPT PRIORITY.
 9887 113416  000000                           .WORD   0               ; LUT AVAILABLE (IF NONZERO)
 9888 113420  000000                           .WORD   0               ; SYNC FREQUENCY: 0 = 60HZ
 9889                                                                  ;                 NZ = 50HZ
 9892 113426  172010                           .WORD   172010          ; 1ST (OF 4) REGISTER(S).
 9893                                                                  ;       ( 7 IF LUT INSTALLED ).
 9894 113430  000340                           .WORD   340             ; 1ST (OF 4) VECTOR(S).
 9895                                                                  ;       ( 5 IF LUT INSTALLED ).
 9896 113432  000200                           .WORD   PRI04           ; INTERRUPT PRIORITY.
 9897 113434  000000                           .WORD   0               ; LUT AVAILABLE (IF NONZERO)
 9898 113436  000000                           .WORD   0               ; SYNC FREQUENCY: 0 = 60HZ
 9899                                                                  ;                 NZ = 50HZ
 9902                                   .SBTTL  ****** TH - TH - TH - TH - THAT'S ALL FOLKS ! ******
 9903 113440  000001                   PEND:   .END
```

SYMBOL TABLE

SEQ 0249

```
ABP      055140          BM1      067532          CMIVB    015430          CSMT1    060314          C$EDIT= 000003
ABPX     055146          BM1PIX   067550          CMNOMI   015371          CSMT10   062244          C$ERDF= 000055
ABPY     055150          BM1TST   067214          CMPE     016224          CSMT11   062364          C$ERHR= 000056
ACAT     110302          BM1X     067542          CNVRT    027324          CSMT12   062450          C$ERRO= 000060
ADR    = 000020 G        BM1Y     067544          COLID    103106          CSMT13   062656          C$ERSF= 000054
ALL    = 003774 G        BM14   = 136000 G        CONFIG   030414          CSMT14   063020          C$ERSO= 000057
ALLB   = 003774 G        BM18   = 137000 G        CONMEM   030414          CSMT15   063306          C$ESCA= 000010
ALLNB  = 003374 G        BOE    = 000400 G        CONSYC   030760          CSMT2    060426          C$ESEG= 000005
APF      005047          BOXES    100732          CRDAX    013734          CSMT3    060540          C$ESUB= 000003
APNT   = 114000 G        BRINIT   007030          CRDAY    014017          CSMT4    060672          C$ETST= 000001
APTST    054566          BV       102466          CRWRE    014102          CSMT5    061024          C$EXIT= 000032
ASSEMB= 000010           B04      075714          CRWRO    013664          CSMT6    061202          C$GETB= 000026
AUXSEG= 000001 G         B048     075736          CRWRZ    013613          CSMT7    061320          C$GETW= 000027
A1     = 001776          B08      075726          CSCHAN   065140          CSMT8    061434          C$GMAN= 000043
BADDAT   023060          B14      075666          CSCH.1   065146          CSMT9    062070          C$GPHR= 000042
BARS     102356          B148     075710          CSCH.6   065142          CSMXIT   065016          C$GPLO= 000030
BGEN     102460          B18      075700          CSCH.8   065144          CSPCE    010411          C$GPRI= 000040
BIT0   = 000001 G        CBAER    021422 G        CSDSE    010351          CSRCHK   022440          C$INIT= 000011
BIT00  = 000001 G        CBAERA   013507          CSFLE    010451          CSRER    021166 G        C$INLP= 000020
BIT01  = 000002 G        CBCHK    022624          CSIFC    014630          CSRERA   013165          C$MANI= 000050
BIT02  = 000004 G        CESWE    014326          CSMEND   064776          CSRETR   065150          C$MEM  = 000031
BIT03  = 000010 G        CEZWE    014255          CSMF1A   065200          CSRGEN   014543          C$MSG  = 000023
BIT04  = 000020 G        CGRPH  = 020000 G        CSMF1B   065206          CSRGSN   014461          C$OPEN= 000034
BIT05  = 000040 G        CHAR   = 100000 G        CSMF2A   065212          CSXPE    010651          C$PNTB= 000014
BIT06  = 000100 G        CHDISP   054122          CSMF2B   065214          CSXRE    010611          C$PNTF= 000017
BIT07  = 000200 G        CHIE   = 010000 G        CSMF2C   065216          CT       054070          C$PNTS= 000016
BIT08  = 000400 G        CHPROT= 000200 G         CSMF2D   065222          CTAB     003256 G        C$PNTX= 000015
BIT09  = 001000 G        CHRF     005475          CSMF3A   065230          CTABE    003302 G        C$QIO  = 000377
BIT1   = 000002 G        CHRFX    005521          CSMF3B   065236          CTABM    003256 G        C$RDBU= 000007
BIT10  = 002000 G        CHRTST   053374          CSMF3C   065240          CTABS    003270 G        C$REFG= 000047
BIT11  = 004000 G        CHSEG    054066          CSMF3D   065252          CT.ASC   054104          C$RESE= 000033
BIT12  = 010000 G        CHV      103060          CSMF4A   065300          CT.BAS   054116          C$REVI= 000003
BIT13  = 020000 G        CHX      103052          CSMF4B   065320          CT.RET   054106          C$RFLA= 000021
BIT14  = 040000 G        CHY      103054          CSMF4C   065314          CT.SUB   054110          C$RPT  = 000025
BIT15  = 100000 G        CH0    = 000000 G        CSMF4D   065334          CUIM   = 146016 G        C$SEFG= 000046
BIT2   = 000004 G        CH1    = 000400 G        CSMF4E   065372          CUIOFF= 146012 G         C$SPRI= 000041
BIT3   = 000010 G        CH2    = 001000 G        CSMF4F   065420          CUIS   = 146013 G        C$SVEC= 000037
BIT4   = 000020 G        CH3    = 001400 G        CSMF4G   065436          CUOFF  = 146040 G        C$TPRI= 000013
BIT5   = 000040 G        CKDROP   030354          CSMF4H   065454          CUON   = 146060 G        DAVTO  = 144003 G
BIT6   = 000100 G        CKEMAX   030254          CSMF4I   065470          CURD   = 146100 G        DBLINK   105356
BIT7   = 000200 G        CLFLGS= 000012 G         CSMF4J   065346          CUWT   = 175000 G        DBLPIX= 000100 G
BIT8   = 000400 G        CLMFCN   026754          CSMF4K   065370          CVC    = 000040 G        DDMMY    113074
BIT9   = 001000 G        CLMPTR   071610          CSMF4L   065324          C$AU   = 000052          DEVCNT   032726
BLINK  = 175401 G        CLMSHT   071626          CSMF4U   065270          C$AUTO= 000061           DEVDRO   034004
BLOOM    104620          CLMTAB   071612          CSMF4V   065254          C$BRK  = 000022          DEVNRD   033723
BLU    = 007400          CLRMEM= 170140 G         CSMF4X   065326          C$BSEG= 000004           DEVNXR   033641
BLUC   = 000004 G        CLRRW    026650          CSMF4Y   065330          C$BSUB= 000002           DEVONL   033571
BLUEG  = 002100 G        CLRRW1   026640          CSMITR   060204          C$CEFG= 000045           DEVSUM   033534
BMF      005431          CLRTST   070716          CSMLOO   060210          C$CLCK= 000062           DFPTBL   002224 G
BMX      005453          CLRT1    070740          CSMNEX   064756          C$CLEA= 000012           DGEN     102660
BMXX     101062          CLRW     026632          CSMPE    010551          C$CLOS= 000035           DGX      102662
BMOCHK   070456          CLR.D    071602          CSMPTB   065026          C$CLP1= 000006           DGY      102664
BMOP1    070552          CLR.D2   071720          CSMPTE   065136          C$CVEC= 000036           DIAGMC= 000000
BMOP2    070576          CLR.Z    071562          CSMPX    065022          C$DCLN= 000044           DISSEL   101470
BMOTST   067642          CLR.ZA   071572          CSMPY    065024          C$DODU= 000051           DJM      026274
BM04   = 134000 G        CLR.Z2   071704          CSMRE    010511          C$DRPT= 000024           DJMP   = 160000 G
BM08   = 135000 G        CLTNEW   071272          CSMTST   060164          C$DU   = 000053          DJMS   = 160001 G
```

```
DJS     026326          ECMPE4  052452          FCOORD  016377          GHYTST  066450          HUE     003222 G
DNOP  = 164000 G        ECRSV1  052466          FERCM   024030          GHYTYP  067104          HX      075516
DOTS    102556          ECRSV2  052472          FHST  = 060000 G        GOLD  = 003200 G        HY      075306
DPC     003032 G        ECRSV3  052502          FILL1   027014          GRN   = 000360          HZ      003246 G
DPCER   020732 G        ECSEQ1  052512          FILL2   027020          GRNC  = 000010 G        I     = 040000 G
DPCERA  012643          ECSEQ2  052530          FLGCHK  022410          GRN1  = 002400 G        IAUX    003160 G
DPDSE   007611          ECSEQ3  052576          FLGER   021134 G        GRN2  = 003000 G        IBE   = 010000 G
DPINIT  025402          ECSEQ4  052630          FLGERA  013122          GRN3  = 003400 G        ICBAE   007476
DPOP  = 165000 G        ECSEQ5  052654          FNOCSR  023772          GRX     075446          ICBASE  003162 G
DPRESE  026074          ECSEQ6  052676          FNOINT  004315          GRY     075236          ICSR    003152 G
DPRI    003050 G        ECSEQ7  052726          FORCER  003020          GTCURS  077752          ICSRE   007156
DPSINI  025540          ECSQ7B  052750          FORCET  032776          GTJSSW= 040000 G        IDLE    101362
DPUMOD  002502 G        ECSTO   052754          FORCSI= 000040 G        GUNID   103360          IDLEA   101404
DPUSAV  026360          EF.CON= 000036 G        FREE    003240 G        GX      075470          IDLEB   101424
DRDYTO= 150003 G        EF.NEW= 000035 G        FRESIZ  003242 G        GXI   = 174100 G        IDLEX   101342
DRR     003034 G        EF.PWR= 000034 G        FSYCHA  016500          GY      075260          IDPC    003126 G
DSBINT  026176          EF.RES= 000037 G        FUMI    004206          GYI   = 174100 G        IDPCE   007100
DSCSE   010111          EF.STA= 000040 G        FUSI    003752          G$CNTO= 000200          IDSR    003130 G
DSDPE   007651          EGGBL = 003500 G        FUSWI   004251          G$DELM= 000372          IDSRE   007127
DSDXE   007711          EMAXDU  030151          FUTO    004066          G$DISP= 000003          IDU   = 000040 G
DSDYE   007751          EMK     100360          F$AU  = 000015          G$EXCP= 000400          IDXR    003132 G
DSFLE   010051          EN    = 006654          F$AUTO= 000020          G$HILI= 000002          IDYR    003134 G
DSMPE   010211          ENAINT  026114          F$BGN = 000040          G$LOLI= 000001          IER   = 020000 G
DSMRE   010151          ENECHK= 000100 G        F$CLEA= 000007          G$NO  = 000000          IFAULT  004356
DSPCE   010011          ENVIRN  031476          F$DU  = 000016          G$OFFS= 000400          IFLAGS  003150 G
DSR     003034 G        ERCM    024041          F$END = 000041          G$OFSI= 000376          IHBAE   007447
DSRCHK  022332          ERCNAM  024264          F$HARD= 000004          G$PRMA= 000001          IHBASE  003156 G
DSRER   020764 G        ERCOD   024074 G        F$HW  = 000013          G$PRMD= 000002          IIRCHK  023422
DSRERA  012706          ERCODS  047466          F$INIT= 000006          G$PRML= 000000          IIRCNC  023406
DSTP    026242          ERCTBL  024266          F$JMP = 000050          G$RADA= 000140          IIRCNF  023434
DSXPE   010311          ERICHK  023516          F$MOD = 000000          G$RADB= 000000          IIRNFC  023452
DSXRE   010251          ERR   = 100000 G        F$MSG = 000011          G$RADD= 000040          ILDR    003140 G
DTAB    101236          ERRCHK  023524          F$PROT= 000021          G$RADL= 000120          ILMR    003142 G
DTABE   101272          ERRCK1  023526          F$PWR = 000017          G$RADO= 000020          ILSR    003136 G
DTHIL = 000015          ERRK    030130          F$RPT = 000012          G$XFER= 000004          IMAIN   003154 G
DTO     026210          ERRVEC= 000004 G        F$SEG = 000003          G$YES = 000010          IMCBDA  072504
DUADR   035704          ERTABE  003502          F$SOFT= 000005          HAFX  = 000776 G        IMCERR  072334 G
DUADRE  021454 G        ERTABL  003302          F$SRV = 000010          HAFY  = 000776 G        IMCHK   071736
DUAD12  017105          ESF     004612          F$SUB = 000002          HAFY50= 000776 G        IMCHKR  071764
DUFLG   003224 G        ESUM    030132          F$SW  = 000014          HAFY60= 000736 G        IMCRAM  072506
DUMMY   003076          EVL   = 000004 G        F$TEST= 000001          HATCH   102702          IMCRCT  072512
DX    = 000002 G        EX      100336          GCOFF = 000100 G        HBAER   021370 G        IMCTRY  072510
DXR     003036 G        EXPGOT  017051          GDBAD   022736          HBAERA  013444          IMC.CH  072552
DXRER   021016 G        EXPGT2  016322          GDDAT   023056 G        HBCHK   022526          IMC.IR  072576
DXRERA  012751          EXPR    027100          GERRMA  002516 G        HCOPY = 176051 G        IMC.ME  072566
DXYME   100140          EXPRC2  022074 G        GETCUR  065474          HCPY  = 000001 G        IMC.RA  072606
DYI   = 000002 G        EXPREC  020700 G        GETJSE  065542          HGEN    103050          IMC.SR  072556
DYNI  = 000004 G        EXPRM   110242          GHIINH= 010000 G        HHMM    113076          IMC.TA  072554
DYR     003040 G        EXSM2 = 000050 G        GHX   = 120000 G        HOE   = 100000 G        IMC.Y   072574
DYRER   021050 G        EXSM4 = 000060 G        GHXF    005314          HPM1    002310          IMDERR  020506 G
DYRERA  013014          EXTA    022300          GHXT    066334          HPM2    002340          IMDI    005641
EBLINK  105330          EXTEND  022276          GHXTST  065676          HPM3    002370          IMDIX   005677
ECE     013552          EX2   = 000010 G        GHXTYP  066332          HPM4    002420          IMDIXC  006101
ECISVC  052416          EX4   = 000020 G        GHXYX   005374          HPM5    002450          IMDIX2  006010
ECMPE1  052420          EY      100352          GHY   = 124000 G        HST   = 040000 G        IMPME   007277
ECMPE2  052424          E$END = 002100          GHYF    005344          HSX     075502          IMPWPE  007234
ECMPE3  052436          E$LOAD= 000035          GHYT    067106          HSY     075272          IMREAD= 162000 G
```

```
IMRF    072514          JMSPOP  044222          LDCP  = 175000 G      L$EXP1  002046 G      L10031  021452
IMRRE   007205          JMWD  = 000004 G        LDECC = 175400 G      L$EXP4  002064 G      L10032  021506
IMR8  = 163000          JMWE  = 000010 G        LDI   = 002000 G      L$EXP5  002066 G      L10033  021646
IMTMOD  002504 G        JPDPCE  007525          LDINC   006662        L$HARD  002240 G      L10034  021760
INCCBA  044014          JPPCSE  007557          LDJSS = 146000 G      L$HIME  002120 G      L10035  022072
INCERK  030216          JSDXYE  100064          LDR     003044 G      L$HPCP  002016 G      L10036  022130
INCHBA  043544          JSEWRO  014211          LDUN    026602        L$HPTP  002022 G      L10037  022176
INCREG  041232          JSEWRZ  014145          LERRMA  002514 G      L$HW    002224 G      L10040  022224
INLAC   072706          JSF     004566          LINTER  020610 G      L$ICP   002104 G      L10041  022274
INLE    017207          JSLCKO= 000010 G        LINTX   006540        L$INIT  032014 G      L10042  023322
INLEX   022226 G        JSMIES= 010000 G        LMR     003046 G      L$LADP  002026 G      L10043  024262
INLEXF  017235          JSMV    003054 G        LOCT    074754        L$LAST  113404 G      L10044  025740
INTCPC  026112          JSPIX   100316          LOE   = 040000 G      L$LOAD  002100 G      L10046  032724
INTERR  020226 G        JSPIXA  100306          LOGO    074600        L$LUN   002074 G      L10047  033106
INTFLA  026107          JSSIES= 004000 G        LONGX   005121        L$MREV  002050 G      L10050  033214
INTLAC  003230 G        JSSV    003060 G        LOOP    027646        L$NAME  002000 G      L10051  033216
INTMAS  026106          JSTST   077304          LOOPFL  023054 G      L$PRIO  002042 G      L10052  033270
INTVEC  026110          JSV     076670          LOOPK   027700        L$PROT  032004 G      L10053  033532
INTX    004400          JSVA    076744          LOT   = 000010 G      L$PRT   002112 G      L10054  034272
IOKCKI= 000200          JSVA1   076724          LREAD = 010000 G      L$REPP  002062 G      L10055  035042
IOKERR= 000004          JSWD  = 000001 G        LRWERR  020642 G      L$REV   002010 G      L10056  035620
IOKJSM= 000002          JSWE  = 000002 G        LRWX    006707        L$RPT   033272 G      L10057  041150
IOKJSS= 000010          JSWTEX  100010          LSET  = 010000 G      L$SOFT  002522 G      L10060  042136
IOKSTP= 000001          JSX     100414          LSR     003042 G      L$SPC   002056 G      L10061  042564
IPCSAV  003146 G        JSY     100434          LUMIN   027054        L$SPCP  002020 G      L10062  043446
IREGCK  023066          JS0   = 000000 G        LUMTBL  107222        L$SPTP  002024 G      L10063  043716
ISDSR   003144 G        JS1   = 000400 G        LUTAV   003216 G      L$STA   002030 G      L10064  044156
ISR   = 000100 G        JS2   = 001000 G        LUTSAV  026606        L$SW    002502 G      L10065  044652
IXE   = 004000 G        JS3   = 001400 G        LUTV    003062 G      L$TEST  002114 G      L10066  045366
IXPME   007355          JXYPE   100254 G        LV      055606        L$TIML  002014 G      L10067  046036
IXPWPE  007404          JXYPX   100216          LVEC  = 110000 G      L$UNIT  002012 G      L10070  046766
IXRRE   007326          J$JMP = 000167          LVF     005073        L0    = 002000 G      L10071  047334
I$AU  = 000041          KIPAR0= 172340          LVTST   055224        L10000  002236        L10072  052774
I$AUTO= 000041          KIPAR1= 172342          LVX     055616        L10001  002500        L10073  053326
I$CLN = 000041          KIPAR2= 172344          LVY     055620        L10002  002520        L10074  054514
I$DU  = 000041          KIPAR3= 172346          L$ACP   002110 G      L10003  003020        L10075  055154
I$HRD = 000041          KIPAR4= 172350          L$APT   002036 G      L10004  020062        L10076  055624
I$INIT= 000041          KIPAR5= 172352          L$AU    033012 G      L10005  020124        L10077  056314
I$MOD = 000041          KIPAR6= 172354          L$AUT   002070 G      L10006  020166        L10100  057014
I$MSG = 000041          KIPAR7= 172356          L$AUTO  033216 G      L10007  020224        L10101  060062
I$PROT= 000040          KIPDR0= 172300          L$CCP   002106 G      L10010  020504        L10102  065610
I$PTAB= 000041          KIPDR1= 172302          L$CLEA  033220 G      L10011  020606        L10103  066362
I$PWR = 000041          KIPDR2= 172304          L$CO    002032 G      L10012  020640        L10104  067134
I$RPT = 000041          KIPDR3= 172306          L$DEPO  002011 G      L10013  020676        L10105  067562
I$SEG = 000041          KIPDR4= 172310          L$DESC  002122 G      L10014  020730        L10106  070622
I$SETU= 000041          KIPDR5= 172312          L$DESP  002076 G      L10015  020762        L10107  072612
I$SFT = 000041          KIPDR6= 172314          L$DEVP  002060 G      L10016  021014        L10110  071270
I$SRV = 000041          KIPDR7= 172316          L$DISP  107114 G      L10017  021046        L10111  071550
I$SUB = 000041          KLUDGE  031572          L$DLY   002116 G      L10020  021100        L10112  072502
I$TST = 000041          KTBKGD  031322          L$DTP   002040 G      L10021  021132        L10113  073146
JCWD  = 000020 G        KTFLG   003244 G        L$DTYP  002034 G      L10022  021164        L10114  074362
JCWE  = 000040 G        KTINIT  031656          L$DU    033110 G      L10023  021216        L10115  073660
JCXA    100320          KTOFF   031306          L$DUT   002072 G      L10024  021250        L10116  074236
JCXB    100740          KTON    031270          L$DVTY  002162 G      L10025  021302        L10117  074444
JCXC    101110          K100US  027564          L$EF    002052 G      L10026  021334        L10120  076556
JCXD    101114          LAIER   006555          L$ENVI  002044 G      L10027  021366        L10121  101132
JENOCL  014376          LAIEW   006617          L$ETP   002102 G      L10030  021420        L10122  100304
```

```
L10123  104760          MPT.ST  074334          OPTI    003064 G        PTINER  023324          SCBASE  003212 G
L10124  107110          MPT.WB  074356          OUTDFX  105246          PUNIT   032730          SCFLG   003214 G
L10125  113410          MPT.WE  074344          OUTDFY  105276          PVEC2 = 000004 G        SCIDE   017355
L10126  113426          MPT.WY  074352          OUTDOT  105072          QVP     003030 G        SCINT   106640
L10127  113422          MPXPE   011551          OUTDTF  105134          QXIT    076460          SCM     017534
L10131  113440          MPXRE   011511          OUTDT1  105162          Q1      076366          SCME    017416
L255  = 003774 G        MRCSE   011051          OUTDY1  105302          Q2      076424          SCMEM   106464
M     = 020000 G        MRDSE   010711          OUTFLG  105070          Q3      076366          SCMFG   017461
MAGEN = 002300 G        MRFLE   011011          OUTLIN  104762          Q4      076424          SCMMMO  106704
MAPMEM  070326          MRMPE   011111          O$APTS= 000000          RBL     076536          SCMMSO  106772
MARK    100442          MRPCE   010751          O$AU  = 000001          RBO     076530          SCNINT  106660
MATCH   100470          MRRER   021220 G        O$BGNR= 000001          RBX     076540          SCNSTD  106552
MATFLE  016017          MRRERA  013230          O$BGNS= 000001          RBY     076542          SCONTY  105522
MATIRE  015663          MRXPE   011211          O$DU  = 000001          RDDXY   065520          SCOUT   107040
MATPCS  015505          MRXRE   011151          O$ERRT= 000000          RDJSE   065566          SCPRE   017326
MATPOS  015611          MSX   = 020000 G        O$GNSW= 000001          RDWRT = 176074 G        SCRBLU  104542
MATUI   016064          MSY   = 000100 G        O$POIN= 000001          RDWRT1= 176076 G        SCRCOM  104554
MATXYE  015547          MXY   = 020000 G        O$SETU= 000000          READ  = 176050 G        SCRGRN  104550
MAXX  = 001776 G        M128  = 000002 G        PASRPT  032240          RED   = 000017          SCRRED  104534
MAXY  = 001776 G        M256  = 000003 G        PATCH   113100          REDC  = 000020 G        SCRWHT  104526
MAXY50= 001776 G        M32   = 000000 G        PATS    074250          REDG  = 002200 G        SCS     017632
MAXY60= 001676 G        M64   = 000001 G        PATTST  073240          REGERR  023144 G        SCSR    003202 G
MCHAN   077270          NBLINK= 175400 G        PATT1   073262          REGNAM  023062          SCSYC   106571
MDV0  = 000000 G        NEWPAS  032174          PAUSE1  027514          RELEAS  026756          SDASK   102212
MDV1  = 000001 G        NLRI    006306          PAUS.5  027522          RELF    004746          SDLIST  101472
MDV2  = 000002 G        NLWI    006362          PCSCHK  022360          RELFX   004773          SDO     102231
MDV3  = 000003 G        NOCMI   015263          PCSER   021102 G        RELTST  053050          SDPC    003164 G
MEMFLG  003252 G        NOCSR   024003          PCSERA  013057          RELT1   053322          SDSR    003166 G
MEMTAB  003256 G        NODEV   003226 G        PCSERR  020064 G        RESET1  034340          SDX     027312
MENDPC  016141          NOINIT  004434          PCSX    003564          RESTRS  031436          SDXR    003170 G
MEREG   015733          NOINTR  004321          PDBITS  003066 G        RGE     022664          SDY     027270
MFGFLG  002512 G        NOITS   002510 G        PDF     003072 G        RJF     004534          SDYR    003172 G
MFGMO   003022          NOMAN   017735          PDI     003070 G        RLMOD   057062          SELCBA= 000007 G
MFGSO   003024          NOMAT   016267          PDM     003074 G        RLMOD1  057076          SELCHA  077302
MKDXR   100004          NOMST   015334          PDXI    004642          RLMXIT  057676          SELCSR= 000003 G
MKDYR   100006          NOSTIK  077222          PEND    113440          RLNEWL  057346          SELDIS  101272
MMCHK   022466          NOSWI   014777          PERIM   103066          RLSKIP= 000400 G        SELDSR= 000000 G
MMMAC   044744          NOSWPI  014712          PINTRA= 000001 G        RLXIE   016552          SELFLG= 000002 G
MMMACA  045016          NOTOT   047336          PINTRB= 000002 G        RLXYE   020170 G        SELHBA= 000005 G
MMMACE  012511          NSI     004014          PIXRBK= 000000 G        RLX2IE  016613          SELMPM= 000004 G
MMMAE   021650 G        NSINIT  006760          PNT   = 001000 G        RLYDE   016715          SELMRR= 000004 G
MMMNXE  012600          NTO     004135          PNTCOO  022132 G        RLYUE   016663          SELPCS= 000001 G
MMVEC = 000250          NTSC    027066          PNTSYC  022200 G        RLY2DE  017010          SELXPM= 000006 G
MNOKT   045316          NTSC8   110222          PRBH    005574          RLY2UE  016751          SELXRR= 000006 G
MPAT2   074306          NUL     016546          PRBHNG  005556          RND8  = 000400 G        SEQERR= 120003 G
MPCBAS  074332          NULCR   016547          PRGSIZ= 174561          RNLN  = 144000 G        SETCB = 152000 G
MPCSE   011411          NXM     027566          PRI   = 002000 G        RPD     056310          SETCSR= 000013 G
MPDSE   011251          NXME  = 104003 G        PRIOO = 000000 G        RPF     005175          SETDSR= 000010 G
MPE   = 114003 G        NXR     003504          PRIO1 = 000040 G        RPNT  = 130000 G        SETFLG= 000012 G
MPFLE   011351          NXRERR  020032 G        PRIO2 = 000100 G        RPT     056300          SETHB = 150000 G
MPLEN = 000011          NXRX    003543          PRIO3 = 000140 G        RPTST   055676          SETMEM= 170100 G
MPMER   021252 G        NXTU    032206          PRIO4 = 000200 G        RSTPOS= 100000 G        SETMPM= 000015 G
MPMERA  013273          ODDSKP= 000200 G        PRIO5 = 000240 G        RSVDOP= 124003 G        SETMRR= 000014 G
MPMRE   011451          OPCF    004676          PRIO6 = 000300 G        R256  = 000100 G        SETU    032272
MPOFF   074330          OPCFX   004724          PRIO7 = 000340 G        SAUX    003210 G        SETXPM= 000017 G
MPPCE   011311          OPCOD   042632          PROTEC= 176000 G        SAVED   031462          SETXRR= 000016 G
MPTNEW  073662          OPC1    043440          PRSYCH  065152          SAVERS  031412          SFLAGS  003200 G
```

```
SFPTBL  002502 G    SWCHON= 000002 G    T$$AU = 010047      UAM    = 000200 G    XPXRE   012451
SHADLY  003220 G    SWE    = 000010 G    T$$AUT= 010051      ULRI     006437      XRCSE   011751
SHAD16  027172      SWIBV    015036      T$$CLE= 010052      ULWI     006477      XRDSE   011611
SHBASE  003206 G    SWNOCU   015205      T$$DAT= 010131      UMI      004212      XRFLE   011711
SHRTX   005250      SWNOHL   015112      T$$DU = 010050      UNITN    003026 G    XRMPE   012051
SIFLAG  023064      SWTCH = 170200 G    T$$HAR= 010001      USI      003756      XRMRE   012011
SIMSG   023012      SYCFLG   003254 G    T$$HW = 010000      USWI     004255      XRPCE   011651
SINIT = 100000 G    SYCTAB   003270 G    T$$INI= 010046      US100    027530      XRRER   021304 G
SINIT1  035114      SYNC   = 164001 G    T$$MSG= 010122      UTO      004072      XRRERA  013336
SIOFF = 171000 G    SYNCTO= 140003 G    T$$PC = 000002      VIOLET= 003300 G    XRXPE   012111
SION  = 171400 G    SYSCON   021510 G    T$$PRO= 010045      VISXIT   076356      XTBL    102516
SKP0  = 000400      S$LSYM= 010000      T$$PTA= 010130      VIS1     074544      XXCOMM  003236 G
SKP1  = 000401      TBOFF    102320      T$$RPT= 010053      VIS2     074714      XXDP   = 003726
SKP2  = 000402      TBON     102262      T$$SEG= 010000      VIS3     075110      XX.ENT= 025570
SKP3  = 000403      TEMP1    003232 G    T$$SOF= 010003      VIS4     075320      XX.EXI= 026072
SMAIN   003204 G    TEMP2    003234 G    T$$SUB= 010116      VIS5     075530      XYERR   020126 G
SOCT1   075026      TESTK    030070      T$$SW = 010002      VIS5A    075746      XYX     003657
SOCT2   075040      TIDFLG   002506 G    T$$TES= 010124      VIS6     076076      X$ALWA= 000000
SPARTS  074434      TINERR   022775      T1       034054 G    VRLF     076016      X$FALS= 000040
SPCSAV  003176 G    TN     = 000043      T10      044160 G    VRLF1    076024      X$OFFS= 000400
SPM1    002600      TNAM     030066      T11      044654 G    VRLF2    076026      X$TRUE= 000020
SPM2    002630      TNUM     030064      T12      045370 G    VSHGE    026010      X2INC   057752
SPM4    002660      TOTV     003056 G    T13      046040 G    VSHNG    025720 G    X2INCA  057762
SPM5    002710      TOTVST   047050      T14      046770 G    VSHUNG   025742      X2INCB  057772
SPM6    002740      TSTEND   030072      T15      047422 G    WAITF    027454      YA      067124
SPM7    002770      TSTGO    027702      T16      052776 G    WE     = 000010 G    YDN     060016
SR0   = 177572      T$ARGC= 000001      T17      053330 G    WECC   = 160000 G    YDOWN = 000040 G
SR1   = 177574      T$CODE= 000052      T18      054516 G    WHITE  = 003774 G    YELLOW= 003600 G
SR2   = 177576      T$ERRN= 006346      T19      055156 G    WJSS   = 042000 G    YI      066340
SR3   = 172516      T$EXCP= 000000      T2       034274 G    WRT    = 176034 G    YMAX    003250 G
SSDSR   003174 G    T$FLAG= 000040      T20      055626 G    WRTCPX= 140000 G    YUP     060002
SSF     004503      T$FREE= 113440      T21      056316 G    WRTCPY= 140000 G    Y2DN    060046
SSTST   042210      T$GMAN= 000000      T22      057016 G    WRTCSR= 000013 G    Y2UP    060032
SST1    042516      T$HILI= 000015      T23      060064 G    WRTJSS= 042000 G    Y50TBL  102516
SST2    042526      T$LAST= 000001      T24      065612 G    WRTMSR= 040000 G    Y60TBL  102536
SST3    042550      T$LOLI= 000000      T25      066364 G    WRT.D    074262      $SVPC = 110402
STOP  = 172000 G    T$LSYM= 010000      T26      067136 G    WRT.X    074264      .DX    = 002000
STOPI = 173400 G    T$LTNO= 000043      T27      067564 G    WRT.Y    074266      .DY    = 000000
STOPN = 173000 G    T$NEST= 177777      T28      070624 G    WRT1   = 176036 G    .I     = 000000
STPTST  046110      T$NS0 = 000001      T28.1    070732      XA       066352      .NT    = 000037
STPV    003052 G    T$NS1 = 000011      T28.2    071272      XI       067112      ..AA    112100
STP1    046742      T$NS2 = 000003      T29      072614 G    XINC     057722      ..AND   111112
STP2    046746      T$PCNT= 000000      T3       035044 G    XINCA    057732      ..AST   111220
STP3    046752      T$PTAB= 010130      T30      073150 G    XINCB    057742      ..ATS   112044
STP4    046760      T$PTHV= 000002      T30.1    073254      XMCHK    022566      ..BB    112124
STRSET  034122      T$PTNU= 000002      T30.2    073662      XMMAC    045456      ..BSLH  113014
SUPVSR= 026050      T$SAVL= 177777      T31      074364 G    XMMACA   045530      ..CC    112160
SVCGBL= 000000      T$SEGL= 177777      T32      074446 G    XMMACE   012545      ..CMA   111260
SVCINS= 177777      T$SEK0= 010000      T33      076560 G    XMMAE    021762 G    ..COLN  111716
SVCSUB= 177777      T$SIZE= 000016      T34      101134 G    XPCSE    012311      ..CR    110714
SVCTAG= 177777      T$SUBN= 000000      T35      105404 G    XPDSE    012151      ..CRT   113050
SVCTST= 177777      T$TAGL= 177777      T4       035622 G    XPFLE    012251      ..CRX   110722
SVD     057010      T$TAGN= 010132      T5       041152 G    XPMER    021336 G    ..DD    112206
SVEC  = 104000 G    T$TEMP= 000044      T6       042140 G    XPMERA   013401      ..DOL   111020
SVF     005221      T$TEST= 000043      T7       042566 G    XPMPE    012411      ..DOT   111304
SVT     057000      T$TSTM= 177777      T8       043450 G    XPMRE    012351      ..EE    112230
SVTST   056364      T$TSTS= 000001      T9       043720 G    XPPCE    012211      ..EQ    111766
```

```
..EXC   110736          ..LPAR  111164          ..QQ    112516          ..SPC   110730          ..00    111330
..FF    112236          ..MM    112406          ..QUES  112016          ..SS    112546          ..11    111362
..GG    112254          ..MNS   111272          ..QUO   110754          ..TT    112604          ..22    111402
..HH    112272          ..NN    112424          ..QUO1  111152          ..USC   113064          ..33    111434
..II    112312          ..NULL  110702          ..RANG  112004          ..UU    112622          ..44    111474
..JJ    112334          ..NUM   110772          ..RBRK  113032          ..VV    112644          ..55    111514
..KK    112354          ..OO    112444          ..RPAR  111202          ..WW    112664          ..66    111544
..LANG  111752          ..PCT   111052          ..RR    112532          ..XX    112704          ..77    111576
..LBRK  112776          ..PLS   111242          ..SEMI  111734          ..YY    112730          ..88    111616
..LF    110704          ..PP    112474          ..SLSH  111314          ..ZZ    112754          ..99    111666
..LL    112372

. ABS.  113440      000
        000000      001
ERRORS DETECTED:  0

VIRTUAL MEMORY USED:  34896 WORDS  ( 137 PAGES)
DYNAMIC MEMORY:  20346 WORDS  ( 78 PAGES)
ELAPSED TIME:  00:36:46
CVVSA.BIN,CVVSA.LST/-SP=SVC33.MLB/ML,CVVSAB0.MAC/EN:AMA:ABS/DS:GBL
```