VSV11

VSV11 CSS DIAG
CVVSAA0

AH-E769A-MC
FICHE 1 OF 2

AUG 1981
COPYRIGHT© 1981
MADE IN USA

DEC-11-HVSAA-A-D

VSV11 VSV11 CSS DIAG AH-E769A-MC AUG 1981
CVVSAA0 FICHE 2 OF 2 COPYRIGHT@ 1981
MADE IN USA

.REM%

IDENTIFICATION

PRODUCT CODE: AC-E768A-MC

PRODUCT NAME: CVVSAA0 VSV11 CSS DIAG

MAINTAINER: CSS PRODUCTS

AUTHORS: DON MACOMBER
GUS PASQUANTONIO
BILL WEISKE

DATE: 23 FEB 81

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

NO RESPONSIBILITY IS ASSUMED FOR THE USE OR RELIABILITY OF SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1981 BY DIGITAL EQUIPMENT CORPORATION

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

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-NUMBERED 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.

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.

6.2.23 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 SELFCTED 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 UPPFR RIGH[illegible].
3. THREE OCTAGONS USING LONG VECTORS (LARGE O[illegible]),
   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 TO[illegible]
7. A CIRCULAR VECTOR SWEEP AT CENTER, SHOWING
   COLOR/INTENSITY LEVELS.
8. A PERIMETER OUTLINE OF THE DISPLAY.
9. STATIC FOR 2 SECONDS, THEN BLINKING FOR 2 S[illegible]ONDS.

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.

6.3.2 TEST 34. SELECTED DISPLAYS

THIS IS A PATTERN GENERATOR FOR COLOR MONITOR CONVERGENCE SET-UP. IT INCLUDES THE FOLLOWING PATTERNS SELECTABLE BY THE USER AT WILL.

0	"TYPE THIS" (ON CONSOLE)
1	TURN ON BLINKING
2	TURN OFF BLINKING
3	COLOR BARS
4	7 X 7 DOTS
5	7X7 CROSSHATCH
6	PERIMETER OUTLINE
7	BASIC COLOR ID
8	GUNS ID
9	ALL WHITE SCREEN
10	ALTERNATING WHITE SCREEN & PERIMETER OUTLINE
11	ALL RED SCREEN
12	ALL BLUE SCREEN
13	ALL GREEN SCREEN

NOTE: SELECTIONS 4 THROUGH 6 ALSO INCLUDE A PERIMETER OUTLINE OF THE SCREEN, IN WHICH EACH SIDE OF THE OUTLINE INCLUDES 3 "DOTS". SELECTIONS 3, 7 AND 8 INCLUDE A SOLID WHITE PERIMETER OUTLINE.

6.3.3 TEST 35. SYSTEM CONFIGURATION TYPE-OUT

PRINT ON THE CONSOLE THE CONFIGURATION OF THE VSV11 SYSTEM.

7.0 PROGRAM LISTING

PROGRAM LISTING FOLLOWS:

```
.NLIST  BEX
.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

                                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.
```

PROGRAM HEADER

```
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  106760                  .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$HIME::                 ;PTR. TO HIGH MEM
      002120  000000                             .WORD   0
  931 002122                                     DESCRIPT <**** VSV11-VS11 DIAGNOSTIC ****>
      002122                              L$DESC::
      002122     052      052      052           .ASCIZ  /**** VSV11-VS11 DIAGNOSTIC ****/
                                                 .EVEN
  932 002162                                     DEVTYP  <VSV11-VS11, AND DISPLAY MONITOR>
      002162                              L$DVTYP::
      002162     126      123      126           .ASCIZ  /VSV11-VS11, AND DISPLAY MONITOR/
                                                 .EVEN
  933
  934         000001                      SVCGBL= 1                ; SHOVE EVERYTHING BACK TO THE RIGHT.
  935         000001                      SVCINS= 1
  936         177777                      SVCGBL=-1                ; KILL ALL SVC STUFF.
  937         177777                      SVCTST=-1
  938         177777                      SVCSUB=-1
  939         177777                      SVCTAG=-1
  940         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      /
002450                          .EVEN
002500                          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
```

SEQ 0026

```
                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
1051
```

```
                                .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 THF 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          MDV'==  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>.
1145
1146
1147    134000          BM04==  134000                  ; MODE 0 WITH 4 BIT PIXELS.
1148    135000          BM08==  135000                  ; MODE 0 WITH 8 BIT PIXELS.
1149    000000          M32==   0                       ;  32 SQUARE PIXEL ARRAY.
1150    000001          M64==   1                       ;  64 SQUARE.
1151    000002          M128==  2                       ; 128 SQUARE.
1152    000003          M256==  3                       ; 256 SQUARE.
1153    000010          EX2==   10                      ; EXPAND BY 2, NO SMOOTHING.
1154    000020          EX4==   20                      ; EXPAND BY 4, NO SMOOTHING.
1155    000050          EXSM2== 50                      ; EXPAND AND SMOOTH BY 2.
1156    000060          EXSM4== 60                      ; EXPAND AND SMOOTH BY 4.
1157    000100          R256==  100                     ; 256 X 256 RESOLUTION.
1158    000400          RND8==  400                     ; ROUND FOR 8-BIT-WIDE IMAGE MEMORY
1159    000200          ODDSKP== 200                    ; SKIP ODD-NUMBERED LINES ON SCREEN (FOR NON-INTERLACED) ;@@@I
1160
1161    144000          RNLN==  144000                  ;OP-CODE FOR RUN-LENGTH MODE
1162    000100          DBLPIX==        BIT6            ; DOUBLE THE PIXEL COUNT
1163    000040          YDOWN== BIT5                    ; DISPLAY LINES TOP-TO-BOTTOM
1164    000400          RLSKIP== BIT8                   ;SKIP LINES, WITHOUT THROWING THE DATA AWAY
1165                                                    ;   (FOR RUN-LENGTH MODE ONLY)
1166                                                    ;   SKIPS EVEN LINES IF START ON EVEN, & VICE-VERSA
1167    162000          IMREAD== 162000
```

```
                                ;CONTROL AND STATUS INSTRUCTIONS.
                                ;
146040          CUOFF== 146040          ;CURSOR (JOY-STICK) VIDEO OFF.
146060          CUON==  146060          ;CURSOR VIDEO ON.
146100          CURD==  146100          ;READ JSX,JSY => DXR,DYR.
146016          CUIM==  146016          ;INTERRUPT ON MATCH (SWITCH DISABLED).
146013          CUIS==  146013          ;INTERRUPT ON SWITCH (MATCH DISABLED).
146012          CUIOFF== 146012         ;INTERRUPTS (BOTH) OFF.
175000          CUWT==  175000          ; CURSOR WRITE.
175000          LDCP==  175000          ; LOAD CURSOR POSITION (-CUWT)
146000          LDJSS== 146000          ;LOAD JOYSTICK STATUS DISPLAY INSTRUCTION
175400          LDECC== 175400          ;LOAD EXTENDED CURSOR CONTROL
000002          SWCHON== BIT1           ; SOFT SWITCH ON.
175401          BLINK== 175401          ; BLINK.
175400          NBLINK==175400          ; NO BLINK.

000002          JSWE == BIT1            ; JOYSTICK SWITCH-ENABLE WRITE ENABLE
000001          JSWD == BIT0            ; JOYSTICK SWITCH-ENABLE WRITE DATA
000010          JMWE == BIT3            ; CURSOR MATCH-ENABLE WRITE ENABLE
000004          JMWD == BIT2            ; CURSOR MATCH-ENABLE WRITE DATA
000040          JCWE == BIT5            ; CROSSHAIR-INTENSITY WRITE-ENABLE
000020          JCWD == BIT4            ; CROSSHAIR-INTENSITY WRITE DATA


150000          SETHB== 150000          ;SET GRAPH/HISTO BASE LINE.
152000          SETCB== 152000          ;SET CHARACTER BASE ADDRESS.

160000          DJMP==  160000          ;DISPLAY JUMP.
160001          DJMS==  160001          ;DISPLAY JUMP-TO-SUBROUTINE
164000          DNOP==  164000          ;DISPLAY NO-OP.
165000          DPOP==  165000          ;DPOP = RTS FROM CHARACTER SUB-PIX.
164001          SYNC==  DNOP+1          ;PROCEED IN SYNC = DNOP + N <8:0>.
                                        ;WAITS FOR N OCCURRENCES OF VERTICAL...
                                        ;...SYNC START, THEN PROCEEDS...
                                        ;...(16.6MS @ 60HZ OR 20MS @ 50HZ).
000001          AUXSEG== BIT0           ;SPECIFY AUX. SEGMENT IN DISPATCH ADDRESSES.    @@@I

170200          SWTCH== 170200          ;SWITCH READ/WRITE STATUS.
170140          CLRMEM== 170140         ;CLEAR IMAGE MEMORY TO 0'S.
170100          SETMEM== 170100         ;SET IMAGE MEMORY TO CURRENT PIX DATA.

172000          STOP==  172000          ;DISPLAY STOP.
171000          SIOFF== 171000          ;STOP INTERRUPT DISABLE.
171400          SION==  171400          ;STOP INTERRUPT ENABLE.
173000          STOPN== STOP!SIOFF      ;STOP, DO NOT INTERRUPT.
173400          STOPI== STOP!SION       ;STOP AND INTERRUPT.

174100          GXI==   174100          ;SET X INCREMENT <5:0> FOR GRAPH/HST Y.
174100          GYI==   GXI             ;SET Y INCREMENT <5:0> FOR GRAPH/HST X.

000001          HCPY==  BIT0            ;ENABLE HARD-COPY.
000010          SWE==   BIT3            ;ENABLE SWITCH RD/WRT.
176000          PROTEC== 176000         ;PROTECT MEMORY (OFF).
176050          READ==  176040!SWE      ;READ MEM (DISPLAY), SWITCH ENABLED.
176034          WRT==   176024.SWE      ;WRITE MEM (1'S AND 0'S), SWITCH ENABLED
176036          WRT1==  176026!SWE      ;WRITE MEM (1'S ONLY), SWITCH ENABLED.
176074          RDWRT== READ!WRT        ;READ, WRITE ALL DURING RETRACE.
```

```
    1226        176076          RDWRT1== READ.WRT1      ;READ, WRITE 1'S DURING RETRACE.
    1227        176051          HCOPY== READ.HCPY       ;READ MEM => HARD COPY DEVICE.
    1228
    1229        000000          CH0==   0               ;CHANNEL SELECT BITS...
    1230        000400          CH1==   BIT8            ;...FOR MEMORY CONTROL...
    1231        001000          CH2==   BIT9            ;...INSTRUCTIONS (176XXX).
    1232        001400          CH3==   BIT8.BIT9
    1233
    1234        000000          JS0==   CH0             ;JOY-STICK UNIT SELECT BITS...
    1235        000400          JS1==   CH1             ;...FOR CURSOR CONTROL...
    1236        001000          JS2==   CH2             ;...INSTRUCTIONS (146XXX).
    1237        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 ;@@@I
140000                          WRTCPX== 3*BIT14                ;WRITE CURSOR POSITION X-COORDINATE -- <15:14>=11
040000                          GTJSSW=- BIT14                  ;GET JOYSTICK SWITCH INTERRUPT COORDINATES
```

```
                                ;'WRITE" FUNCTIONS FOR DYR (BITS 15-14):
                                ;  PIXEL READBACK IS THE SAME - 00
040000                          WRTMSR== 1*BIT14                ;WRITE MEMORY STATUS REGISTER BA -- <15:14>=01
100000                          SINIT== 2*BIT14                 ;PERFORM "SOFT-INIT" -- <15:14>=10
140000                          WRTCPY== 3*BIT14                ;WRITE CURSOR POSITION Y-COORDINATE -- <15:14>=11

                                ;CSR BIT DEFINITIONS:
000040                          FORCSI== BIT5           ;FORCE STOP INTERRUPT
000100                          ENECHK== BIT6           ;ENABLE ERROR CHECKING
000200                          CHPROT== BIT7           ;CHANNEL PROTECT
                                ;@@@I
                                ;FLAGS REGISTER BIT POSITIONS:
000001                          PINTRA== BIT0           ;PENDING INTERRUPT, CHANNEL A
000002                          PINTRB== BIT1           ;PENDING INTERRUPT, CHANNEL B
000004                          PVEC2== BIT2            ;VECTOR BIT 2 FOR INTERRUPT
000010                          JSLCKO== BIT3           ;JOYSTICK INTERRUPT LOCKOUT

                                ; ERROR CODE DEFINITIONS.

100000                          ERR==   100000          ; ERROR (COMPOSITE).
104003                          NXME==  ERR!4000!SELCSR ; NONEXISTANT MEMORY ERROR.
114003                          MPE==   ERR!14000!SELCSR ; MEMORY PROTECTION ERROR.
124003                          RSVDOP== ERR!24000!SELCSR ; RESERVED OPCODE, RESERVED OPERATION.
120003                          SEQERR== ERR!20000!SELCSR ; SEQUENCE ERROR.
140003                          SYNCTO== ERR.40000!SELCSR ; SYNC TIMEOUT FROM IMAGE MEMORY.
144003                          DAVTO== ERR.44000.SELCSR ; DATA AVAILABLE TIMEOUT (PIXEL READBACK).
150003                          DRDYTO== ERR!50000!SELCSR ; DATA READY TIMEOUT (JOYSTICK STATUS).

                                ;DXR, DYR BITS

010000                          CHIE    ==BIT12                 ; CROSSHAIR INTENSITY ENABLE STATUS (1=ON)
010000                          JSMIES  ==BIT12                 ; JOYSTICK MATCH INTERRUPT ENABLE STATE
004000                          JSSIES  ==BIT11                 ; JOYSTICK SWITCH INTERRUPT ENABLE STATE
042000                          WJSS    ==BIT14.BIT10           ; WRITE JOYSTICK STATUS REGISTER
160000                          WECC    ==160000                ; WRITE EXTENDED CURSOR CONTROL


                                ;
                                ;SOME HANDY -11 OPDEF'S.
                                ;
000403                          SKP3=   BR+3            ;SKIP NEXT 3 WORDS.
000402                          SKP2=   BR+2            ;SKIP NEXT 2 WORDS.
000401                          SKP1=   BR+1            ;SKIP NEXT WORD.
000400                          SKP0=   BR+0            ;SAME AS A NOP.

                                ;
                                ; SOME GENERAL EQUATES.
                                ;

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
```

```
1390                            ;
1391                            ; MACROS TO STANDARDIZE THE INITIALIZATION OF EACH TEST.
1392                            ; PROVIDES AN ASCIZ TITLE STRING, AND POINTER FOR
1393                            ; SUBSEQUENT PRINTING (IF REQ'D) VIA 'PRINTF TNAM,TNUM'' CALL.
1394                            ; ALSO INITS A SEQUENTIAL ERROR NUMBER SEQUENCE, AND
1395                            ; SETS AN ITERATION COUNT FOR THE CURRENT TEST.
1396                            ;
1397                                    .MACRO  BEGIN.TEST TEXT,LK,?TAG1,?TAG2
1398                            .RADIX 10
1399                                    TSTITLE \TN+1,<TEXT>,<LK>
1400                            .RADIX
1401                            .NLIST
1402                            .LIST MC
1403                            .LIST
1404                                    BGNTST
1405                                    TN=T$TESTNUM            ;TEST NUMBER
1406                                    EN=TN*100.              ;INIT ERROR NUMBER SEQUENCE.
1407                                    .IIF B <LK>, MOV #1,LOOPK       ;DEFAULT = 1.
1408                                    .IIF NB <LK>, MOV #LK,LOOPK
1409                                    MOV     #TN,TNUM        ;SET TEST NUMBER...
1410                                    MOV     #TAG1,TNAM      ;...AND NAME POINTER.
1411                                    BR      TAG2            ;SKIP OVER THE ASCII.
1412                            ;TAG1:  .ASCIZ  \%N%ATEST %D2%A: TEXT\
1413                            TAG1:   .ASCIZ  \%A  TEXT\
1414                                    .EVEN
1415                            TAG2:
1416                                    .ENDM   BEGIN.TEST
1417
1418                            ;
1419                            ; THIS MACRO IS USED BY MACRO 'BEGIN.TEST'' MACRO, ABOVE.
1420                            ;
1421                                    .MACRO  TSTITLE A,ASCI,I
1422                            .SBTTL  *** TEST 'A'    ASCI
1423                            .NLIST
1424                            .LIST ME
1425                            .LIST
1426
1427                            ;*****************************************************************************
1428                            ;*
1429                            ;*      BEGIN TEST 'A'  -   ASCI
1430                            ;*
1431                            ;*****************************************************************************
1432
1433                            ; (ITERATION COUNT - I)
1434
1435                            .NLIST
1436                            .NLIST ME
1437                            .LIST
1438                                    .ENDM   TSTITLE
1439
1440                            ;
1441                            ; MACROS TO STANDARDIZE THE END OF EACH TEST.
1442                            ;
1443
1444                                    .MACRO  END.TEST
1445                            .NLIST
1446                            .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.
```

```
1504                                    .NLIST
1505                                    .NLIST ME
1506                                    .LIST
1507                                            EN=EN+1
1508                                            JSR     PC,INCERK
1509                                            ERRHRD  EN,ADDR,PNTR
1510                                            JSR     PC,CKEMAX
1511                                            .IF     NB,CK
1512                                            CKLOOP
1513                                            .ENDC
1514                                            .ENDM   HRDERR
1515
1516
1517                                    ;
1518                                    ; MACRO - DO SOME COMMON STUFF AT THE BEGINNING OF A TEST.
1519                                    ;
1520                                            .MACRO  COMBEG
1521                                    .NLIST
1522                                    .LIST ME
1523                                    .LIST
1524                                            JSR     PC,TSTGO        ; TITLE.
1525                                            JSR     PC,DPRESET      ; DO SOFT INIT.
1526                                    .NLIST ME
1527                                            .ENDM   COMBEG
1528
1529                                    ;
1530                                    ; MACRO - DO SOME COMMON STUFF AT THE END OF A TEST.
1531                                    ;
1532                                            .MACRO  COMEND  TAG,?L,?X
1533                                    .NLIST
1534                                    .LIST ME
1535                                    .LIST
1536                                            JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
1537                                            BCS     L
1538                                            BR      X
1539
1540                                    L:      JMP     TAG
1541
1542                                    X:      JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
1543                                    .NLIST
1544                                    .NLIST ME
1545                                    .LIST
1546                                            .ENDM   COMEND
1547
1548                                    ;
1549                                    ; MACRO TO CONTROL ITERATION LOOPS.
1550                                    ;
1551                                            .MACRO  LOOPTO  TAG
1552                                    .NLIST
1553                                    .LIST ME
1554                                    .LIST
1555                                            JSR     PC,LOOP
1556                                            BCS     TAG
1557                                    .NLIST
1558                                    .NLIST ME
1559                                    .LIST
1560                                            .ENDM   LOOPTO
```

```
1561
1562                            ;
1563                            ; MACRO - WRITE FROM SPEC'D DATA SOURCE INTO SPEC'D DSR ACCESS REGISTER.
1564                            ; ONLY DSR, FLG, CSR, MRR, MPM, XRR, XPM ARE VALID ARGUMENTS FOR 'DSRREG'.
1565                            ; R0=WORK.
1566                            ;
1567                                    .MACRO  WTDSRA  DATSRC,DSRREG
1568                            .NLIST
1569                            .LIST MEB
1570                            .LIST
1571                                    .NTYPE  .NT,DATSRC
1572                                    .IF     EQ,.NT&^O77-^O27
1573                                    MOV     DATSRC&^C17!SET'DSRREG,@DSR
1574                                    .IFF
1575                                    MOV     #SET'DSRREG',R0 ; WRITE DATSRC INTO DSRREG .
1576                                    BIS     DATSRC,R0
1577                                    MOV     R0,@DSR
1578                                    .ENDC
1579                            .NLIST
1580                            .NLIST MEB
1581                            .LIST
1582                                    .ENDM   WTDSRA
1583
1584                            ;
1585                            ; MACRO - READ FROM SPEC'D DSR ACCESS REGISTER INTO SPEC'D DATA DESTINATION.
1586                            ; ONLY DSR, PCS, FLG, CSR, MRR, MPM, XRR, XPM, HBA, CBA ARE VALID
1587                            ; 'DSRREG'' ARGUMENTS.
1588                            ; R0=WORK.
1589                            ;
1590                                    .MACRO  RDDSRA  DSRREG,DATDST
1591                            .NLIST
1592                            .LIST ME
1593                            .LIST
1594                                    MOV     #SEL'DSRREG',@DSR       ; READ DSRREG INTO DATDST .
1595                                    MOV     @DSR,DATDST
1596                            .NLIST
1597                            .NLIST ME
1598                            .LIST
1599                                    .ENDM   RDDSRA
1600
1601                            ;
1602                            ; MACRO - IF R1-R2, BRANCH TO OK, ELSE CALL HARDWARE ERROR.
1603                            ;
1604                                    .MACRO  IFERROR ARG1,ARG2,?OK$,CK
1605                            .NLIST
1606                            .LIST ME
1607                            .LIST
1608                                    CMP     R1,R2           ; OK?
1609                                    BEQ     OK$             ; YES.
1610                                    HRDERR  ARG1,ARG2,CK    ; NO.
1611                            .NLIST
1612                            .LIST ME
1613                            .LIST
1614                            OK$:
1615                            .NLIST ME
1616                                    .ENDM   IFERROR
1617
```

```
1618                            ;
1619                            ; MACRO - IF R1=R2 AND R3=R4, BRANCH TO OK, ELSE CALL HARDWARE ERROR.
1620                            ;
1621                                    .MACRO  IFERRX2 ARG1,ARG2,?OK$,?ERR,CK
1622                            .NLIST
1623                            .LIST ME
1624                            .LIST
1625                                    CMP     R1,R2           ; 1ST ITEM OK?
1626                                    BNE     ERR             ; NO.
1627                                    CMP     R3,R4           ; 2ND ITEM OK?
1628                                    BEQ     OK$             ; YES.
1629                            ERR:    HRDERR  ARG1,ARG2,CK    ; NO.
1630                            .NLIST
1631                            .LIST ME
1632                            .LIST
1633                            OK$:
1634                            .NLIST ME
1635                                    .ENDM   IFERRX2
1636
1637                            ;
1638                            ; MACRO - SOFTWARE INITIALIZE THE DPU.  THIS DIFFERS FROM 'DPINIT'
1639                            ; IN TWO WAYS.  FIRST, IT IS ASSUMED THAT THE DPU IS STOPPED AND
1640                            ; NOT HUNG UP.  SECOND, NO CHECKING IS DONE BY THIS CODE.
1641                            ;
1642                                    .MACRO  INITDP
1643                            .NLIST
1644                            .LIST ME
1645                            .LIST
1646                                    JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
1647                            .NLIST
1648                            .NLIST ME
1649                            .LIST
1650                                    .ENDM   INITDP
1651
1652
1653                                    .MACRO  DPSTART ADRS
1654                            .NLIST
1655                            .LIST ME
1656                            .LIST
1657                                    MOV     ADRS,@DPC       ; START THE DPU.
1658                                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
1659                            .NLIST
1660                            .NLIST ME
1661                            .LIST
1662                                    .ENDM   DPSTART
1663
1664
1665
1666                                    .MACRO  DPCONT
1667                            .NLIST
1668                            .LIST ME
1669                            .LIST
1670                                    BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
1671                                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
1672                            .NLIST
1673                            .NLIST ME
1674                            .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
```

```
1732
1733                                    ;
1734                                    ; MACRO TO FORCE IN-LINE RETURN FROM REAL OR IMAGINED INTERUPT.
1735                                    ;
1736                                            .MACRO  FORRTI
1737                                    .NLIST
1738                                    .LIST ME
1739                                    .LIST
1740                                            MOV     #.+6,(SP)       ; FORCE RTI RETURN TO -
1741                                            RTI                     ; - NEXT LOC.
1742                                    .NLIST
1743                                    .NLIST ME
1744                                    .LIST
1745                                            .ENDM   FORRTI
```

USER PATCHABLE FLAGS

```
                                .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.)

C03024  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
```

```
1825                                    ;
1826                                    ; STORAGE FOR DEVICE REGISTERS
1827                                    ;
1828 003076  000000  100000  000000     DUMMY:  0,100000,0,0    ;DUMMY DEVICE REGISTERS...
1829 003106  000000  000000  000000             0,0,0,0,0,0,0,0
1830                                                            ;...FOR MULTI-UNIT CHECKOUT.
1831
1832 003126  000000                     IDPC::  0               ;SAVED DPC ON ANY INTERRUPT.
1833 003130  000000                     IDSR::  0               ;       DITTO DSR
1834 003132  000000                     IDXR::  0               ;       DITTO DXR
1835 003134  000000                     IDYR::  0               ;       DITTO DYR
1836 003136  000000                     ILSR::  0               ;       DITTO LUT CSR
1837 003140  000000                     ILDR::  0               ;       DITTO LUT DATA
1838 003142  000000                     ILMR::  0               ;       DITTO LUT MAINT.
1839
1840                                    ;SAVED "INTERNAL" REGISTERS ON AN INTERRUPT (IN ORDER OF SELECT CODE):
1841 003144  000000                     ISDSR:: 0               ;SAVED "SELECTED" DSR
1842 003146  000000                     IPCSAV::        0       ;SAVED PCSAVE
1843 003150  000000                     IFLAGS::        0       ;SAVED FLAGS
1844 003152  000000                     ICSR::  0               ;SAVED CSR
1845 003154  000000                     IMAIN:: 0               ;SAVED MAIN SEG. MMGT.
1846 003156  000000                     IHBASE::        0       ;SAVE HISTOGRAM BASE
1847 003160  000000                     IAUX::  0               ;SAVED AUX. SEG MMGT.
1848 003162  000000                     ICBASE::        0       ;SAVED CHARACTER BASE
1849
1850 003164  000000                     SDPC::  0               ;SAVED DPC AT ANY OTHER TIME.
1851 003166  000000                     SDSR::  0               ;       DITTO DSR
1852 003170  000000                     SDXR::  0               ;       DITTO DXR
1853 003172  000000                     SDYR::  0               ;       DITTO DYR
1854                                    ;SAVED INTERNAL REGISTERS AT OTHER TIMES
1855 003174  000000                     SSDSR:: 0               ;SAVED "SELECTED" DSR
1856 003176  000000                     SPCSAV::        0       ;SAVED PCSAVE
1857 003200  000000                     SFLAGS::        0       ;SAVED FLAGS
1858 003202  000000                     SCSR::  0               ;SAVED CSR
1859 003204  000000                     SMAIN:: 0               ;SAVED MAIN SEG. MMGT.
1860 003206  000000                     SHBASE::        0       ;SAVE HISTOGRAM BASE
1861 003210  000000                     SAUX::  0               ;SAVED AUX. SEG MMGT.
1862 003212  000000                     SCBASE::        0       ;SAVED CHARACTER BASE
1863                                    ;
1864                                    ; AND FINALLY SOME MISCELLANEOUS REGISTERS.
1865                                    ;
1866 003214  000000                     SCFLG:: 0               ; SYS CONFIG FLAG (0=1ST TIME, NZ=SUBSEQUENT).
1867 003216  000000                     LUTAV:: 0               ; LUT AVAILABLE FLAG (NZ = USE LUT).
1868 003220  000000                     SHADLY:: 0              ; SHADING DELAY COUNTER (OPTIONAL).
1869 003222  110070                     HUE::   NTSC8+2         ; CURRENT COLOR POINTER (OPTIONAL).
1870
1871                                    ;STIK:: .BYTE   -1, -1  ; 8 JOY-STICK AVAILABLE FLAGS...
1872                                    ;       .BYTE   -1, -1  ;... -1 - I DUNNO...
1873                                    ;       .BYTE   -1, -1  ;...  0 - NO...
1874                                    ;       .BYTE   -1, -1  ;... +1 - YES.
1875
1876 003224  000000                     DUFLG:: 0               ;"DROPPED UNIT" FLAG.  INHIBITS DPU...
1877                                                            ;...CODE IN "CLEAN-UP".
1878 003226  000000                     NODEV:: 0               ;FLAG TO SAY NO DEVICE.
1879
1880 003230  000001                     INTLAC:: 1              ; INTERLACED MODE
1881 003232  000000                     TEMP1:: 0               ;SOME TEMP LOCATIONS.
```

```
1882 003234  000000                TEMP2:: 0
1883 003236  000000                XXCOMM::        0                       ; XXDP+ COMM BLOCK POINTER.
1884 003240  000000                FREE::  0                       ; 1ST FREE MEMORY ADDRESS...
1885 003242  000000                FRESIZ::        0                       ;...AND SIZE (IN WORDS).
1886 003244  000000                KTFLG:: 0                               ; KT11, MEM AVAIL FLAG -
1887                                                                       ; - 0 = <24K OR NO KT -
1888                                                                       ; - NZ = >24K AND KT.
1889 003246  000074                HZ::    60.                     ; LINE FREQUENCY.
1890 003250  000000                YMAX::  0                       ; MAX Y ACCORDING TO LINE FREQENCY.
1891 003252  000000                MEMFLG:: 0                      ; MEM FLAGS (1'S IN BITS 0-3 REFLECT MEMS AVAIL).
1892 003254  000000                SYCFLG:: 0                      ; SYNC CHAN FLAGS (1'S IN BITS 0-3 REFLECT SYNC CHANS AVAIL)
1893 003256                        CTAB::                          ; CONFIGURATION TABLES.
1894 003256                        MEMTAB::
1895 003256  000000                CTABM:: 0                       ; MEMORY CONFIG WORK.
1896 003260  000000                        0
1897 003262  000000                        0
1898 003264  000000                        0
1899 003266  177777                        -1                      ; END OF MEM TABLE.
1900 003270                        SYCTAB::
1901 003270  000000                CTABS:: 0                       ; SYNC CHAN CONFIG WORK.
1902 003272  000000                        0
1903 003274  000000                        0
1904 003276  000000                        0
1905 003300  177777                        -1                      ; END OF SYNC CHAN TABLE.
1906 003302                        CTABE::                         ; END OF CONFIG WORK AREA.
1907
1908                               ;ERROR STATISTICS TABLE (1 WORD PER UNIT), 64 UNITS MAX:
1909                               ;
1910                               ;       0       =       UNIT NOT TESTED
1911                               ;       100000  =       UNIT ONLINE, NO ERRORS
1912                               ;       10XXXX  =       UNIT ONLINE, ENCOUNTERED XXXX ERRORS
1913                               ;       160000  -       UNIT DROPPED, NON-EXISTENT DEVICE REGISTER
1914                               ;       160001  -       UNIT DROPPED, NOT IDLE AT START
1915                               ;       14XXXX          UNIT DROPPED, ENCOUNTERED XXXX EPRORS
1916                               ;
1917 003302                        ERTABL: .BLKW   64.
1918 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, PELOC(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/
```

```
1977 006101  045  116  045  IMDIXC: .ASCII  /%N%A  ADDRESS (X,Y): %04%A,%04/
1978 006137  045  116  045          .ASCIZ  /%N%A  FIELD: %04%A  EXP'D: %04%A  REC'D: %04/
1979 006214  045  116  045          .ASCIZ  /%N%A  SUSPECT RAM GROUP %01%A,  IMAGE MEMORY CHANNEL: %01/
1980 006306  040  040  114  NLRI:   .ASCIZ  /  LUT READ INTERRUPT EXPECTED, NOT RECEIVED/
1981 006362  040  040  114  NLWI:   .ASCIZ  /  LUT WRITE INTERRUPT EXPECTED, NOT RECEIVED/
1982 006437  040  040  125  ULRI:   .ASCIZ  /  UNEXPECTED LUT READ INTERRUPT/
1983 006477  040  040  125  ULWI:   .ASCIZ  /  UNEXPECTED LUT WRITE INTERRUPT/
1984 006540  045  101  040  LINTX:  .ASCIZ  /%A  CSR: %06/
1985 006555  040  040  114  LAIER:  .ASCIZ  /  LUT AUTO-INCR INCORRECT ON READ/
1986 006617  040  040  114  LAIEW:  .ASCIZ  /  LUT AUTO-INCR INCORRECT ON WRITE/
1987 006662  040  040  114  LDINC:  .ASCIZ  /  LUT DATA INCORRECT/
1988 006707  045  101  040  LRWX:   .ASCIZ  /%A  CSR: %06%A  EXP'D: %06%A  REC'D: %06/
1989 006760  040  040  042  NSINIT: .ASCIZ  /  ''SOFT-INIT'' DIDN'T INITIALIZE THE DPU/
1990 007030  040  040  042  BRINIT: .ASCIZ  /  ''BUS-RESET'' DIDN'T INITIALIZE THE DPU/
1991 007100  040  040  104  IDPCE:  .ASCIZ  #  DPC READ/WRITE ERROR#
1992 007127  040  040  104  IDSRE:  .ASCIZ  #  DSR READ/WRITE ERROR#
1993 007156  040  040  103  ICSRE:  .ASCIZ  #  CSR READ/WRITE ERROR#
1994 007205  040  040  115  IMRRE:  .ASCIZ  #  MRR READ/WRITE ERROR#
1995 007234  040  040  115  IMPWPE: .ASCIZ  #  MPM WRITE/MWP (HBASE) READ ERROR#
1996 007277  040  040  115  IMPME:  .ASCIZ  #  MPM READ/WRITE ERROR#
1997 007326  040  040  130  IXRRE:  .ASCIZ  #  XRR READ/WRITE ERROR#
1998 007355  040  040  130  IXPME:  .ASCIZ  #  XPM READ/WRITE ERROR#
1999 007404  040  040  130  IXPWPE: .ASCIZ  #  XPM WRITE/XWP (HBASE) READ ERROR#
2000 007447  040  040  110  IHBAE:  .ASCIZ  #  HBA READ/WRITE ERROR#
2001 007476  040  040  103  ICBAE:  .ASCIZ  #  CBA READ/WRITE ERROR#
2002 007525  040  040  104  JPDPCE: .ASCIZ  #  DJMS/DPOP FAILURE (DPC)#
2003 007557  040  040  104  JPPCSE: .ASCIZ  #  DJMS/DPOP FAILURE (PCS)#
2004 007611  040  040  104  DPDSE:  .ASCIZ  /  DPC(W),DSR(R) DUAL ADRS FAULT/
2005 007651  040  040  104  DSDPE:  .ASCIZ  /  DSR(W),DPC(R) DUAL ADRS FAULT/
2006 007711  040  040  104  DSDXE:  .ASCIZ  /  DSR(W),DXR(R) DUAL ADRS FAULT/
2007 007751  040  040  104  DSDYE:  .ASCIZ  /  DSR(W),DYR(R) DUAL ADRS FAULT/
2008 010011  040  040  104  DSPCE:  .ASCIZ  /  DSR(W),PCS(R) DUAL ADRS FAULT/
2009 010051  040  040  104  DSFLE:  .ASCIZ  /  DSR(W),FLG(R) DUAL ADRS FAULT/
2010 010111  040  040  104  DSCSE:  .ASCIZ  /  DSR(W),CSR(R) DUAL ADRS FAULT/
2011 010151  040  040  104  DSMRE:  .ASCIZ  /  DSR(W),MRR(R) DUAL ADRS FAULT/
2012 010211  040  040  104  DSMPE:  .ASCIZ  /  DSR(W),MPM(R) DUAL ADRS FAULT/
2013 010251  040  040  104  DSXRE:  .ASCIZ  /  DSR(W),XRR(R) DUAL ADRS FAULT/
2014 010311  040  040  104  DSXPE:  .ASCIZ  /  DSR(W),XPM(R) DUAL ADRS FAULT/
2015 010351  040  040  103  CSDSE:  .ASCIZ  /  CSR(W),DSR(R) DUAL ADRS FAULT/
2016 010411  040  040  103  CSPCE:  .ASCIZ  /  CSR(W),PCS(R) DUAL ADRS FAULT/
2017 010451  040  040  103  CSFLE:  .ASCIZ  /  CSR(W),FLG(R) DUAL ADRS FAULT/
2018 010511  040  040  103  CSMRE:  .ASCIZ  /  CSR(W),MRR(R) DUAL ADRS FAULT/
2019 010551  040  040  103  CSMPE:  .ASCIZ  /  CSR(W),MPM(R) DUAL ADRS FAULT/
2020 010611  040  040  103  CSXRE:  .ASCIZ  /  CSR(W),XRR(R) DUAL ADRS FAULT/
2021 010651  040  040  103  CSXPE:  .ASCIZ  /  CSR(W),XPM(R) DUAL ADRS FAULT/
2022 010711  040  040  115  MRDSE:  .ASCIZ  /  MRR(W),DSR(R) DUAL ADRS FAULT/
2023 010751  040  040  115  MRPCE:  .ASCIZ  /  MRR(W),PCS(R) DUAL ADRS FAULT/
2024 011011  040  040  115  MRFLE:  .ASCIZ  /  MRR(W),FLG(R) DUAL ADRS FAULT/
2025 011051  040  040  115  MRCSE:  .ASCIZ  /  MRR(W),CSR(R) DUAL ADRS FAULT/
2026 011111  040  040  115  MRMPE:  .ASCIZ  /  MRR(W),MPM(R) DUAL ADRS FAULT/
2027 011151  040  040  115  MRXRE:  .ASCIZ  /  MRR(W),XRR(R) DUAL ADRS FAULT/
2028 011211  040  040  115  MRXPE:  .ASCIZ  /  MRR(W),XPM(R) DUAL ADRS FAULT/
2029 011251  040  040  115  MPDSE:  .ASCIZ  /  MPM(W),DSR(R) DUAL ADRS FAULT/
2030 011311  040  040  115  MPPCE:  .ASCIZ  /  MPM(W),PCS(R) DUAL ADRS FAULT/
2031 011351  040  040  115  MPFLE:  .ASCIZ  /  MPM(W),FLG(R) DUAL ADRS FAULT/
2032 011411  040  040  115  MPCSE:  .ASCIZ  /  MPM(W),CSR(R) DUAL ADRS FAULT/
2033 011451  040  040  115  MPMRE:  .ASCIZ  /  MPM(W),MRR(R) DUAL ADRS FAULT/
```

```
2034 011511  040  040  115  MPXRE:  .ASCIZ  /  MPM(W),XRR(R) DUAL ADRS FAULT/
2035 011551  040  040  115  MPXPE:  .ASCIZ  /  MPM(W),XPM(R) DUAL ADRS FAULT/
2036 011611  040  040  130  XRDSE:  .ASCIZ  /  XRR(W),DSR(R) DUAL ADRS FAULT/
2037 011651  040  040  130  XRPCE:  .ASCIZ  /  XRR(W),PCS(R) DUAL ADRS FAULT/
2038 011711  040  040  130  XRFLE:  .ASCIZ  /  XRR(W),FLG(R) DUAL ADRS FAULT/
2039 011751  040  040  130  XRCSE:  .ASCIZ  /  XRR(W),CSR(R) DUAL ADRS FAULT/
2040 012011  040  040  130  XRMRE:  .ASCIZ  /  XRR(W),MRR(R) DUAL ADRS FAULT/
2041 012051  040  040  130  XRMPE:  .ASCIZ  /  XRR(W),MPM(R) DUAL ADRS FAULT/
2042 012111  040  040  130  XRXPE:  .ASCIZ  /  XRR(W),XPM(R) DUAL ADRS FAULT/
2043 012151  040  040  130  XPDSE:  .ASCIZ  /  XPM(W),DSR(R) DUAL ADRS FAULT/
2044 012211  040  040  130  XPPCE:  .ASCIZ  /  XPM(W),PCS(R) DUAL ADRS FAULT/
2045 012251  040  040  130  XPFLE:  .ASCIZ  /  XPM(W),FLG(R) DUAL ADRS FAULT/
2046 012311  040  040  130  XPCSE:  .ASCIZ  /  XPM(W),CSR(R) DUAL ADRS FAULT/
2047 012351  040  040  130  XPMRE:  .ASCIZ  /  XPM(W),MRR(R) DUAL ADRS FAULT/
2048 012411  040  040  130  XPMPE:  .ASCIZ  /  XPM(W),MPM(R) DUAL ADRS FAULT/
2049 012451  040  040  130  XPXRE:  .ASCIZ  /  XPM(W),XRR(R) DUAL ADRS FAULT/
2050 012511  040  040  115  MMMACE: .ASCIZ  /  MAIN MEM MGT ACCESS FAULT/
2051 012545  040  040  101  XMMACE: .ASCIZ  /  AUX MEM MGT ACCESS FAULT/
2052 012600  040  040  115  MMMNXE: .ASCIZ  /  MAIN MEM MGT NON-EXIST MEM FAULT/
2053 012643  045  101  040  DPCERA: .ASCIZ  /%A (DPC)  EXP'D: %06%A, REC'D: %06/
2054 012706  045  101  040  DSRERA: .ASCIZ  /%A (DSR)  EXP'D: %06%A, REC'D: %06/
2055 012751  045  101  040  DXRERA: .ASCIZ  /%A (DXR)  EXP'D: %06%A, REC'D: %06/
2056 013014  045  101  040  DYRERA: .ASCIZ  /%A (DYR)  EXP'D: %06%A, REC'D: %06/
2057 013057  045  101  040  PCSERA: .ASCIZ  /%A (PCS)  EXP'D: %06%A, REC'D: %06/
2058 013122  045  101  040  FLGERA: .ASCIZ  /%A (FLG)  EXP'D: %06%A, REC'D: %06/
2059 013165  045  101  040  CSRERA: .ASCIZ  /%A (CSR)  EXP'D: %06%A, REC'D: %06/
2060 013230  045  101  040  MRRERA: .ASCIZ  /%A (MRR)  EXP'D: %06%A, REC'D: %06/
2061 013273  045  101  040  MPMERA: .ASCIZ  /%A (MPM)  EXP'D: %06%A, REC'D: %06/
2062 013336  045  101  040  XRRERA: .ASCIZ  /%A (XRR)  EXP'D: %06%A, REC'D: %06/
2063 013401  045  101  040  XPMERA: .ASCIZ  /%A (XPM)  EXP'D: %06%A, REC'D: %06/
2064 013444  045  101  040  HBAERA: .ASCIZ  /%A (HBA)  EXP'D: %06%A, REC'D: %06/
2065 013507  045  101  040  CBAERA: .ASCIZ  /%A (CBA)  EXP'D: %06%A, REC'D: %06/
2066 013552  040  040  104  ECE:    .ASCIZ  /  DIDN'T GET EXPECTED ERROR CODE/
2067                         ;MORE ERROR MESSAGES --
2068
2069 013613  040  040  103  CRWRZ: .ASCIZ  ''  CURSOR REGISTER WRITE/READ ZEROS ERROR''
2070 013664  040  040  103  CRWRO: .ASCIZ  ''  CURSOR REGISTER WRITE/READ ONES ERROR''
2071 013734  040  040  103  CRDAX: .ASCIZ  ''  CURSOR REGISTER DUAL ADDRESS - WRITE Y CHANGED X''
2072 014017  040  040  103  CRDAY: .ASCIZ  ''  CURSOR REGISTER DUAL ADDRESS - WRITE X CHANGED Y''
2073 014102  040  040  103  CRWRE: .ASCIZ  ''  CURSOR REGISTER WRITE/READ ERROR''
2074 014145  040  040  112  JSEWRZ: .ASCIZ  ''  JOYSTICK ENABLES WRT/RD 0'S ERROR''
2075 014211  040  040  112  JSEWRO: .ASCIZ  ''  JOYSTICK ENABLES WRT/RD 1'S ERROR''
2076 014255  040  040  112  CEZWE: .ASCIZ  ''  JOYSTICK ENABLES ZERO-THEN-WRITE ERROR''
2077 014326  040  040  112  CESWE: .ASCIZ  ''  JOYSTICK ENABLES SET-THEN-WRITE ERROR''
2078 014376  040  040  112  JENOCL: .ASCIZ  ''  J.S. CHANNEL DECODE ERROR - ENABLES DIDN'T CLEAR''
2079 014461  040  040  122  CSRGSNE: .ASCIZ ''  REGISTER ERROR AFTER J.S. SWITCH SET, NO ENABLE''
2080 014543  040  040  122  CSRGENS: .ASCIZ ''  REGISTER ERROR AFTER J.S. SW ENABLE SET, NO SWITCH''
2081 014630  040  040  111  CSIFC:  .ASCIZ  ''  INTERNAL REG. ERROR AFTER CLEARING J.S. SW INTR''
2082 014712  040  040  105  NOSWPI: .ASCIZ  ''  ERROR IN FLAGS REG. -- SHOULD SHOW PENDING SW INTR''
2083 014777  040  040  106  NOSWI:  .ASCIZ  ''  FAULTY J.S. SWITCH INTERRUPT''
2084 015036  040  040  111  SWIBV:  .ASCIZ  ''  INCORRECT VECTOR ON J.S. SWITCH INTERRUPT''
2085 015112  040  040  123  SWNOHLT: .ASCIZ ''  SWITCH INTR REQ DIDN'T HALT DISPLAY PROCESSING - BAD DPC''
2086 015205  040  040  106  SWNOCU: .ASCIZ  ''  FAULTY CURSOR RETRIEVAL ON J.S. SW INTR REQ''
2087
2088 015263  040  040  115  NOCMI:  .ASCIZ  ''  MATCH INTERRUPT FAULTY OR NOT RECEIVED''
2089 015334  040  040  116  NOMST:  .ASCIZ  ''  NO STOP ON MATCH INTERRUPT''
2090 015371  040  040  115  CMNOMI: .ASCIZ  ''  MATCH BUT NO MATCH INTERRUPT''
```

```
2091 015430  040  040  111  CMIVB:  .ASCIZ  ''  INCORRECT VECTOR ON CURSOR MATCH INTERRUPT''
2092 015505  040  040  127  MATPCS: .ASCIZ  ''  WRONG DPC OR DSR FOR MATCH STOP''
2093 015547  040  040  127  MATXYE: .ASCIZ  ''  WRONG DXR OR DYR FOR MATCH STOP''
2094 015611  040  040  127  MATPOS: .ASCIZ  ''  WRONG DXR OR DYR AFTER POSITION-RESTORE''
2095 015663  040  040  111  MATIRE: .ASCIZ  ''  INTERNAL REGISTER INCORRECT FOR MATCH''
2096 015733  040  040  111  MEREG:  .ASCIZ  ''  INTERNAL REGISTER INCORRECT AFTER SETUP FOR MATCH''
2097 016017  040  040  106  MATFLE: .ASCIZ  ''  FLAGS REGISTER INCORRECT FOR MATCH''
2098 016064  040  040  125  MATUI:  .ASCIZ  ''  UNEXPECTED INTERRUPT ON RESUME AFTER MATCH''
2099 016141  040  040  104  MENDPCS: .ASCIZ ''  DPC OR DSR WRONG ON AFTER RESUME FOLLOWING MATCH''
2100 016224  040  040  103  CMPE:   .ASCIZ  ''  CURSOR MATCHED AT WRONG POSITION''
2101 016267  040  040  116  NOMAT:  .ASCIZ  ''  NO CURSOR MATCH DETECTED''
2102 016322  045  116  045  EXPGT2: .ASCIZ  /%N%A EXP'D: %06%A, %06%N%A REC'D: %06%A, %06/
2103 016377  045  116  045  FCOORD: .ASCII  /%N%A CURSOR  (X,Y)  - %06%A, %06/
2104 016437  045  116  045          .ASCIZ  /%N%A REC'D DXR, DYR = %06%A, %06/
2105 016500  045  116  045  FSYCHAN:         .ASCIZ  '%N%A  SYNC GEN/JOYSTICK CHANNEL = %01''
2106
2107
2108 016546  000            NUL:    .ASCIZ  //
2109 016547  045  116  000  NULCR:  .ASCIZ  /%N/
2110 016552  040  040  122  RLXIE:  .ASCIZ  /  RUN-LENGTH X-INCREMENT FAILURE/
2111 016613  040  040  122  RLX2IE: .ASCIZ  /  RUN-LENGTH X-DOUBLE INCREMENT FAILURE/
2112 016663  040  040  122  RLYUE:  .ASCIZ  /  RUN-LENGTH Y-UP FAILURE/
2113 016715  040  040  122  RLYDE:  .ASCIZ  /  RUN-LENGTH Y-DOWN FAILURE/
2114 016751  040  040  122  RLY2UE: .ASCIZ  /  RUN-LENGTH Y-SKIP UP FAILURE/
2115 017010  040  040  122  RLY2DE: .ASCIZ  /  RUN-LENGTH Y-SKIP DOWN FAILURE/
2116 017051  045  101  040  EXPGOT: .ASCIZ  /%A EXP'D: %06%A, REC'D: %06/
2117 017105  045  101  040  DUAD12: .ASCIZ  /%A REG(W) WRITTEN TO: %06%A REG(R) READ; EXP'D: %06%A, REC'D: %06/
2118 017207  040  040  115  INLE:   .ASCIZ  /  MEM INTERLACE ERROR/
2119 017235  045  101  040  INLEXF: .ASCIZ  /%A  PIXEL READBACK EXP'D: %06%A, REC'D: %06%A, CHAN: %01/
2120 017326  040  040  120  SCPRE:  .ASCIZ  /  PIXEL READBACK ERROR/
2121 017355  040  040  123  SCIDE:  .ASCIZ  /  SYNC CHAN INTERLACE DIFFERENCE/
2122 017416  040  040  103  SCME:   .ASCIZ  /  CONFIG DOESN'T MATCH MFG. MASTER/
2123 017461  045  101  040  SCMFG:  .ASCIZ  /%A  MFG MEM 0: %06%A, MFG SYNC CHAN 0: %06/
2124 017534  045  116  045  SCM:    .ASCII  /%N%A  MEMORYS:/
2125 017552  045  101  040          .ASCII  /%A  0 = %06/
2126 017565  045  101  054          .ASCII  /%A,  1 = %06/
2127 017601  045  101  054          .ASCII  /%A,  2 = %06/
2128 017615  045  101  054          .ASCIZ  /%A,  3 = %06/
2129 017632  045  116  045  SCS:    .ASCII  /%N%A  SYNC CHANS:/
2130 017653  045  101  040          .ASCII  /%A  0 = %06/
2131 017666  045  101  054          .ASCII  /%A,  1 = %06/
2132 017702  045  101  054          .ASCII  /%A,  2 = %06/
2133 017716  045  101  054          .ASCIZ  /%A,  3 = %06%N/
2134 017735  045  116  045  NOMAN:  .ASCIZ  /%N%A**** MANUAL INTERVENTION NOT ALLOWED -- ABORTING ****%N/
2135                              .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$:     BITB    #IOKSTP,INTFLAG ; WAS IT AN UNEXPECTED ''STOP'' INTERRUPT?
020334  001411                          BEQ     2$              ; 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  072350                  MOV     R1,IMCRAM               ;TRANSLATE X ADDR...
020512  006237  072350                  ASR     IMCRAM          ;...TO RAM NUMBER.
```

```
2194 020516                                     PRINTX  #IMDIX,R1,R2,PDF,R3,R4
2195 020552                                     PRINTX  #IMDIX2,IMCRAM,IMC.CH
2196 020602  004737  022276                     JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
2197 020606                                     ENDMSG
2198
2199 020610                             BGNMSG  LINTER          ; LUT INTERRUPT ERROR.
2200 020610                                     PRINTX  #LINTX,@LSR     ; SHOW CSR.
2201 020634  004737  022276                     JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
2202 020640                                     ENDMSG
2203
2204 020642                             BGNMSG  LRWERR          ; LUT READ/WRITE ERROR.
2205 020642                                     PRINTX  #LRWX,@LSR,R1,R2 ; CSR, EXP'D, REC'D.
2206 020672  004737  022276                     JSR     PC,EXTEND       ; PRINT EXTENSION IF REQUIRED.
2207 020676                                     ENDMSG
2208
2209 020700                             BGNMSG  EXPREC
2210 020700                                     PRINTX  #EXPGOT,R1,R2           ; R1=EXP'D, R2=REC'D.
2211 020724  004737  022276                     JSR     PC,EXTEND               ; PRINT EXTENSION IF REQUIRED.
2212 020730                             ENDMSG
2213
2214 020732                             BGNMSG  DPCER
2215 020732                                     PRINTX  #DPCERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2216 020756  004737  022276                     JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2217 020762                             ENDMSG
2218
2219 020764                             BGNMSG  DSRER
2220 020764                                     PRINTX  #DSRERA,R1,R2           ; R1=EXPECTED, R2-RECEIVED.
2221 021010  004737  022276                     JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2222 021014                             ENDMSG
2223
2224 021016                             BGNMSG  DXRER
2225 021016                                     PRINTX  #DXRERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2226 021042  004737  022276                     JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2227 021046                             ENDMSG
2228
2229 021050                             BGNMSG  DYRER
2230 021050                                     PRINTX  #DYRERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2231 021074  004737  022276                     JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2232 021100                             ENDMSG
2233
2234 021102                             BGNMSG  PCSER
2235 021102                                     PRINTX  #PCSERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2236 021126  004737  022276                     JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2237 021132                             ENDMSG
2238
2239 021134                             BGNMSG  FLGER
2240 021134                                     PRINTX  #FLGERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2241 021160  004737  022276                     JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2242 021164                             ENDMSG
2243
2244 021166                             BGNMSG  CSRER
2245 021166                                     PRINTX  #CSRERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2246 021212  004737  022276                     JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2247 021216                             ENDMSG
2248
2249 021220                             BGNMSG  MRRER
2250 021220                                     PRINTX  #MRRERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
```

```
2251 021244  004737  022276                 JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2252 021250                         ENDMSG
2253
2254 021252                         BGNMSG  MPMER
2255 021252                                 PRINTX  #MPMERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2256 021276  004737  022276                 JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2257 021302                         ENDMSG
2258
2259 021304                         BGNMSG  XRRER
2260 021304                                 PRINTX  #XRRERA,R1,R2           ; R1-EXPECTED, R2=RECEIVED.
2261 021330  004737  022276                 JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2262 021334                         ENDMSG
2263
2264 021336                         BGNMSG  XPMER
2265 021336                                 PRINTX  #XPMERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2266 021362  004737  022276                 JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2267 021366                         ENDMSG
2268
2269 021370                         BGNMSG  HBAER
2270 021370                                 PRINTX  #HBAERA,R1,R2           ; R1=EXPECTED, R2=RECEIVED.
2271 021414  004737  022276                 JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2272 021420                         ENDMSG
2273
2274 021422                         BGNMSG  CBAER
2275 021422                                 PRINTX  #CBAERA,R1,R2           ; R1=EXPECTED, R2-RECEIVED.
2276 021446  004737  022276                 JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2277 021452                         ENDMSG
2278
2279 021454                         BGNMSG  DUADRE
2280 021454                                 PRINTX  #DUAD12,R3,R1,R2        ; R3=WRITTEN, R1=EXP'D, R2=REC'D.
2281 021502  004737  022276                 JSR     PC,EXTEND               ; PRINT EXTENSION IF REQ'D.
2282 021506                         ENDMSG
2283
2284 021510                         BGNMSG  SYSCON
2285 021510  005737  002512                 TST     MFGFLG                  ; MFG MODE?
2286 021514  001414                         BEQ     1$                      ; NO.
2287 021516                                 PRINTX  #SCMFG,MFGM0,MFGS0
2288 021546                         1$:     PRINTX  #SCM,MEMTAB,MEMTAB+2,MEMTAB+4,MEMTAB+6
2289 021606                                 PRINTX  #SCS,SYCTAB,SYCTAB+2,SYCTAB+4,SYCTAB+6
2290 021646                         ENDMSG
2291
2292 021650                         BGNMSG  MMMAE
2293 021650                                 PRINTX  #MRRERA,R3,R4
2294 021674                                 PRINTX  #NULCR
2295 021714                                 PRINTX  #DPCERA,R1,R2
2296 021740                                 PRINTX  #NULCR                  ;PRINT BLANK LINE
2297 021760                         ENDMSG
2298
2299
2300 021762                         BGNMSG  XMMAE
2301 021762                                 PRINTX  #XRRERA,R3,R4
2302 022006                                 PRINTX  #NULCR
2303 022026                                 PRINTX  #DPCERA,R1,R2
2304 022052                                 PRINTX  #NULCR                  ;PRINT BLANK LINE
2305 022072                         ENDMSG
2306
2307
```

```
2308 022074                                BGNMSG  EXPRC2
2309 022074                                        PRINTX  #EXPGT2,R1,R3,R2,R4     ;R1,R3 = EXP'D, R2,R4 - REC'D
2310 022124  004737  022276                        JSR     PC,EXTEND               ; PRINT EXTENSION, IF REQ'D.
2311 022130                                        ENDMSG
2312
2313 022132                                BGNMSG  PNTCOOR
2314 022132                                        PRINTX  #FCOORD,CSMF4X,CSMF4Y,SDXR,SDYR
2315 022172  004737  022276                        JSR     PC,EXTEND
2316 022176                                        ENDMSG
2317
2318 022200                                BGNMSG  PNTSYCH
2319 022200                                        PRINTX  #FSYCHAN,CSCHAN
2320 022224                                        ENDMSG
2321
2322 022226                                BGNMSG  INLEX
2323 022226                                        PRINTX  #INLEXF,R1,R2,R5
2324 022254                                        PRINTX  #NULCR          ;PRINT BLANK LINE
2325 022274                                ENDMSG
2326                                       ;
2327                                       ; THIS ROUTINE APPENDS A UNIQUE EXTENSION (IF REQUIRED)
2328                                       ; TO ANY OF THE ABOVE ERROR SIGNATURES.
2329                                       ;
2330 022276  005727                        EXTEND: TST     (PC)+
2331 022300  000000                        EXTA:   0                       ; 0 = NO EXTENSION.
2332 022302  001402                                BEQ     1$
2333 022304  004777  177770                        JSR     PC,@EXTA        ; APPEND EXTENSION TEXT.
2334 022310                                1$:     PRINTX  #NULCR          ; PRINT A BLANK LINE
2335 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: %06%A  REC'D: %06/
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     SIFLA           ;SOFT INIT?
023260  001410                          BEQ     2$              ;BR IF NO
023262                                  PRINTX  #SIMSG          ;PRINT IF YES
023302                          2$:     PRINTX  #NULCR          ;PRINT BLANK LINE
023322                                  ENDMSG
```

```
2394 023324                                PTINER:
2395 023324                                        PRINTX  #TINERR         ;PRINT ''TEST:''
2396 023344                                        PRINTX  TNAM            ;PRINT NAME OF TEST
2397 023364                                        PRINTX  #NULCR          ;PRINT BLANK LINE
2398 023404  000207                                RTS     PC
2399
2400                                       ;ROUTINES TO CHECK FOR THE INITIALIZED STATE OF INTERNAL REGISTERS:
2401                                       ;    IIRCHK  CHECKS THEM ALL.
2402                                       ;    IIRCNF  CHECKS ALL BUT THE FLAGS REGISTER.
2403                                       ;    IIRCNC  CHECKS ALL BUT THE CSR
2404                                       ;    IIRNFC  CHECKS ALL BUT FLAGS AND CSR.
2405                                       ;
2406 023406  012737  100000  023056  IIRCNC: MOV     #CHAR,GDDAT     ;SEE IF FLAGS HAS ''CHAR'' MODE ONLY.
2407 023414  004737  022410                JSR     PC,FLGCHK
2408 023420  000414                        BR      IIRNFC          ;BYPASS CSR CHECK.
2409 023422  012737  100000  023056  IIRCHK: MOV     #CHAR,GDDAT     ;SEE IF FLAGS HAS ''CHAR'' MODE ONLY.
2410 023430  004737  022410                JSR     PC,FLGCHK
2411 023434  004737  023524  IIRCNF: JSR     PC,ERRCHK       ;SPECIFICALLY CHECK FOR AN ERROR CODE.
2412 023440  012737  000003  023056        MOV     #SELCSR,GDDAT   ;WE SHOULD ONLY SEE SELECT CODE IN CSR.
2413 023446  004737  022440                JSR     PC,CSRCHK       ;
2414 023452  012737  000001  023056  IIRNFC: MOV     #1,GDDAT        ;EXPECT TO SEE ''EMPTY'' STATUS IN PCSAVE.
2415 023460  004737  022360                JSR     PC,PCSCHK       ;...GO CHECK IT.
2416 023464  005037  023056                CLR     GDDAT           ;REMAINING REGISTERS S/B CLEAR.
2417 023470  004737  022466                JSR     PC,MMCHK        ;...MAIN MEM MGMT
2418 023474  004737  022526                JSR     PC,HBCHK        ;...HISTOGRAM BASE
2419 023500  004737  022566                JSR     PC,XMCHK        ;...AUX MEM MGMT
2420 023504  004737  022624                JSR     PC,CBCHK        ;...CHARACTER BASE.
2421 023510  005037  023064                CLR     SIFLAG          ;CLEAR SOFT-INIT FLAG.
2422 023514  000207                        RTS     PC
2423
2424                                       ;ERRCHK: CHECKS THE CSR FOR AN ERROR CODE & REPORTS IT.
2425 023516  012746  000001        ERICHK: MOV     #1,-(SP)        ; SET FLAG TO INHIBIT USE OF 'DFERR''
2426 023522  000401                        BR      ERRCK1
2427 023524  005046                ERRCHK: CLR     -(SP)           ; CLEAR FLAG
2428 023526  012777  000003  157300  ERRCK1: MOV     #SELCSR,@DSR    ;SELECT THE CSR.
2429 023534  017746  157274                MOV     @DSR,-(SP)      ;READ IT.
2430 023540  012777  000000  157266        MOV     #SELDSR,@DSR    ;SELECT REAL DSR SO PEOPLE WON'T GET MIXED UP..
2431 023546  011637  023060                MOV     (SP),BADDAT     ;STORE CSR CONTENTS.
2432 023552  042716  177760                BIC     #^C17,(SP)      ;EXTRACT THE SELECT.WE FIELD.
2433 023556  022627  000003                CMP     (SP)+,#SELCSR   ;WE SHOULD SEE THE CSR REG SELECT CODE.
2434 023562  001424                        BEQ     1$
2435 023564  005716                        TST     (SP)            ; SHOULD WE DO DFERR?
2436 023566  001011                        BNE     20$             ; BR IF NO
2437 023570                                DFERR   NOCSR           ;IF NOT, SAY WE CAN'T READ IT.
2438 023610  000466                        BR      2$              ; GO TO EXIT
2439 023612                        20$:    PRINTX  #FNOCSR         ; PRINT MESSAGE ''CAN'T READ CSR''
2440 023632  000455                        BR      2$              ; GO TO EXIT
2441
2442 023634  005737  023060        1$:     TST     BADDAT          ;TEST THE ERROR BIT.
2443 023640  100052                        BPL     2$              ;OK IF CLEAR; JUST EXIT.
2444 023642  000337  023060                SWAB    BADDAT          ;IF SET, GET ERROR CODE
2445 023646  006237  023060                ASR     BADDAT
2446 023652  006237  023060                ASR     BADDAT          ;...INTO BITS<4:1>.
2447 023656  042737  177741  023060        BIC     #^C36,BADDAT    ;ZAP OUT JUNK.
2448 023664  013746  023060                MOV     BADDAT,-(SP)    ;SAVE ERROR CODE * 2 ON STACK
2449 023670  062716  024266                ADD     #ERCTBL,(SP)    ;GET TO THE MESSAGE ADDRESS TABLE.
2450 023674  013637  024264                MOV     @(SP)+,ERCNAM   ;GET THE ADDRESS OF THE REG NAME MESSAGE.
```

```
2451 023700  005716                        TST     (SP)            ; USE 'DFERR'?
2452 023702  001011                        BNE     11$             ; BR IF NO
2453 023704                                HRDERR  ERCM,ERCOD      ;REPORT THE BAD NEWS.
2454 023724  000420                        BR      2$
2455 023726                         11$:   PRINTX  #FERCM          ; PRINT ''CSR ERROR CODE DETECTED =''
2456 023746                                PRINTX  ERCNAM          ; ... AND THE ERROR NAME.
2457 023766  005726                 2$:    TST     (SP)+
2458 023770  000207                        RTS     PC
2459
2460 023772     045     116     045 FNOCSR: .ASCII  /%N%A  ***/
2461 024003     040     040     103 NOCSR:  .ASCIZ  /  CAN'T READ THE CSR/
2462 024030     045     116     045 FERCM:  .ASCII  /%N%A  ***/
2463 024041     040     040     103 ERCM:   .ASCIZ  /  CSR ERROR CODE REC'D - /
2464                                        .EVEN
2465 024074                         BGNMSG ERCOD
2466 024074                                PRINTX  ERCNAM
2467 024114  023727  023060  000026        CMP     BADDAT,#13*2    ; IS IT 'DBUS DATA R/W ERROR'?
2468 024122  001014                        BNE     1$              ;BR IF NO
2469 024124                                PRINTX  #GDBAD,@DXR,@DYR  ; YES -- DXR - EXP'D, DYR - REC'D
2470 024154                         1$:    PRINTX  #TINERR
2471 024174                                PRINTX  TNAM
2472 024214  005737  023064                TST     SIFLAG          ;SOFT INIT?
2473 024220  001410                        BEQ     2$              ;BR IF NO
2474 024222                                PRINTX  #SIMSG          ;PRINT IF YES
2475 024242                         2$:    PRINTX  #NULCR          ;PRINT BLANK LINE
2476 024262                                ENDMSG
2477 024264  000000                 ERCNAM: 0
2478 024266  024326  024365  024430 ERCTBL: 77$,1$,2$,3$,4$,5$,6$,7$
2479 024306  024735  025004  025067        10$,11$,12$,13$,14$,15$,16$,17$
2480
2481                                ;ERROR CODE EXPLANATIONS:
2482 024326     045     101     056 77$:    .ASCIZ  /%A... 0:  (ILLEGAL ERROR CODE)/
2483 024365     045     101     056 1$:     .ASCIZ  /%A... 1:  NXM TIMEOUT ON DMA X'FER/
2484 024430     045     101     056 2$:     .ASCIZ  /%A... 2:  (ILLEGAL ERROR CODE)/
2485 024467     045     101     056 3$:     .ASCIZ  /%A... 3:  MEMORY PROTECTION ERROR/
2486 024531     045     101     056 4$:     .ASCIZ  /%A... 4:  SEQUENCE ERROR/
2487 024562     045     101     056 5$:     .ASCIZ  /%A... 5:  RESERVED OPERATION/
2488 024617     045     101     056 6$:     .ASCIZ  /%A... 6:  VECTOR ENDPOINT MISMATCH/
2489 024662     045     101     056 7$:     .ASCIZ  /%A... 7:  U-PROC EXECUTED UNUSED MICROWORD/
2490 024735     045     101     056 10$:    .ASCIZ  /%A... 10:  IMAGE MEMORY ''SYNC'' TIMEOUT/
2491 025004     045     101     056 11$:    .ASCIZ  /%A... 11:  PIXEL READBACK 'DATA AVAILABLE'' TIMEOUT/
2492 025067     045     101     056 12$:    .ASCIZ  /%A... 12:  J.S. STATUS 'DATA READY'' TIMEOUT/
2493 025143     045     101     056 13$:    .ASCIZ  /%A... 13:  DBUS DATA READ-WRITE ERROR/
2494 025211     045     101     056 14$:    .ASCIZ  /%A... 14:  DBUS SIGNAL HUNG/
2495 025245     045     101     056 15$:    .ASCIZ  /%A... 15:  VBUS SIGNAL HUNG/
2496 025301     045     101     056 16$:    .ASCIZ  /%A... 16:  (ILLEGAL ERROR CODE)/
2497 025341     045     101     056 17$:    .ASCIZ  /%A... 17:  (ILLEGAL ERROR CODE)/
2498                                        .EVEN
2499
2500                                ;ROUTINE TO DO A ''SOFT'' INIT ON THE DPU:
2501                                        .ENABL  LSB
2502 025402  004737  031412         DPINIT: JSR     PC,SAVERS       ; SAVE R0-R5.
2503 025406  012737  000001  023064        MOV     #1,SIFLAG       ; SAY DOING SOFT INIT
2504 025414  012777  000000  155412        MOV     #0,@DSR         ;SELECT THE REAL DSR
2505 025422  005777  155406                TST     @DSR            ;ARE WE STOPPED?
2506 025426  100423                        BMI     11$             ;BR IF YES
2507 025430  004737  026756                JSR     PC,RELEAS       ;SEE IF WE'RE HUNG.
```

```
2508 025434  017700  155372                    MOV     @DPC,R0
2509 025440  057700  155370                    BIS     @DSR,R0
2510 025444  057700  155366                    BIS     @DXR,R0
2511 025450  057700  155364                    BIS     @DYR,R0
2512 025454  001010                           BNE     11$
2513 025456                                    DFERR   VSHUNG,VSHNG    ; REPORT HUNG, ALL REGISTERS 0
2514 025476                            11$:
2515 025476  012777  040000  155332           MOV     #GTJSSW,@DXR    ;CLEAR PENDING INTERRUPT
2516 025504  004737  027454                    JSR     PC,WAITF
2517 025510  012703  000004                    MOV     #4,R3           ;SET COUNTER FOR # OF SWITCHES TO CLEAR
2518 025514  012704  160000                    MOV     #WECC,R4        ;SET UP OP TO CLEAR SWITCHES
2519 025520  010477  155314            12$:    MOV     R4,@DYR         ;CLEAR A SWITCH
2520 025524  004737  027454                    JSR     PC,WAITF
2521 025530  062704  000400                    ADD     #CH1,R4
2522 025534  005303                           DEC     R3
2523 025536  001370                           BNE     12$
2524 025540  012777  040000  155270  DPSINI:  MOV     #GTJSSW,@DXR    ;CLEAR PENDING INTERRUPT
2525 025546  004737  027454                    JSR     PC,WAITF
2526 025552  012777  100000  155260           MOV     #100000,@DYR    ;DO THE INIT.
2527 025560  004737  027454                    JSR     PC,WAITF        ;WAIT FOR A COMPLETE
2528 025564  103402                           BCS     1$              ;OK IF STOPPED.
2529 025566  004737  026756                    JSR     PC,RELEAS       ;TRY TO UNHANG.
2530 025572  005002                    1$:     CLR     R2              ;EXPECT DPC TO BE 0.
2531 025574  012703  100000                    MOV     #BIT15,R3       ;EXPECT DSR TO BE 100000.
2532 025600  005777  155226                    TST     @DPC            ;TEST THE DPC, S/B -0.
2533 025604  001003                           BNE     2$              ;ERROR IF NOT 0.
2534 025606  027703  155222                    CMP     @DSR,R3         ;CHECK DSR.
2535 025612  001410                           BEQ     3$
2536 025614                            2$:     DFERR   NSINIT,PCSERR   ;REPORT ERROR IN DPC OR DSR.
2537 025634  012737  000001  023064  3$:      MOV     #1,SIFLAG       ;SAY DOING SOFT INIT.
2538 025642  004737  023422                    JSR     PC,IIRCHK       ;CHECK THE INTERNAL REGISTERS
2539 025646  005004                           CLR     R4              ; DXR AND DYR SHOULD BE 0.
2540 025650  005005                           CLR     R5              ;
2541 025652  005777  155160                    TST     @DXR
2542 025656  001003                           BNE     4$
2543 025660  005777  155154                    TST     @DYR
2544 025664  001410                           BEQ     5$
2545 025666                            4$:     DFERR   NSINIT,XYERR
2546 025706  004737  031436            5$:     JSR     PC,RESTRS       ; RESTORE R0-R5.
2547 025712  005037  023064                    CLR     SIFLAG
2548 025716  000207                           RTS     PC              ;RETURN TO CALLER
2549                                           .DSABL  LSB
2550
2551 025720                            BGNMSG  VSHNG
2552 025720                                    PRINTX  #VSHGE
2553 025740                                    ENDMSG
2554
2555 025742     040     040     104   VSHUNG: .ASCIZ  /  DPU HUNG -- ALL REGISTERS READ ZERO/
2556 026010     045     101     040   VSHGE:  .ASCIZ  /%A   BAD M7064 OR NO B-CLOCK OR VBLANK FROM M7061%N/
2557                                           .EVEN
2558
2559
2560
2561 026074                            DPRESET: BGNSEG         ; <<<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
2562 026076  004737  025402                    JSR     PC,DPINIT       ; DO SOFT INIT.
2563 026102                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
2564 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  026526                  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.
```

```
2623 026232  152737  000004  026107          BISB    #IOKERR,INTFLAG ;NO. SET THE ERROR FLAG.
2624 026240  000447                          BR      DPUSAV          ;GO FINISH UP.
2625
2626 026242  013737  003052  026110  DSTP:   MOV     STPV,INTVEC     ;SAVE THE VECTOR
2627 026250  105037  026107                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2628 026254  132737  000001  026106          BITB    #IOKSTP,INTMASK ;EXPECTING STOP INTERRUPT?
2629 026262  001036                          BNE     DPUSAV          ;BR IF YES
2630 026264  152737  000001  026107          BISB    #IOKSTP,INTFLAG ;NO. SET THE ERROR FLAG.
2631 026272  000432                          BR      DPUSAV          ;GO FINISH UP.
2632
2633 026274  013737  003054  026110  DJM:    MOV     JSMV,INTVEC     ;SAVE THE VECTOR
2634 026302  105037  026107                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2635 026306  132737  000002  026106          BITB    #IOKJSM,INTMASK ;EXPECTING MATCH INTERRUPT?
2636 026314  001021                          BNE     DPUSAV          ;BR IF YES
2637 026316  152737  000002  026107          BISB    #IOKJSM,INTFLAG ;NO. SET THE ERROR FLAG.
2638 026324  000415                          BR      DPUSAV          ;GO FINISH UP.
2639
2640 026326  013737  003060  026110  DJS:    MOV     JSSV,INTVEC     ;SAVE THE VECTOR
2641 026334  105037  026107                  CLRB    INTFLAG         ;CLEAR FLAG TO SAY WE GOT INTERRUPT
2642 026340  132737  000010  026106          BITB    #IOKJSS,INTMASK ;EXPECTING SWITCH INTERRUPT?
2643 026346  001004                          BNE     DPUSAV          ;BR IF YES
2644 026350  152737  000010  026107          BISB    #IOKJSS,INTFLAG ;NO. SET THE ERROR FLAG.
2645 026356  000400                          BR      DPUSAV          ;GO FINISH UP.
2646
2647
2648 026360  011637  026112  DPUSAV: MOV     (SP),INTCPC     ;SAVE CPU PC.
2649 026364  010146                  MOV     R1,-(SP)        ;SAVE R1
2650 026366  013701  003034                  MOV     DSR,R1          ;GET DSR ADDRESS INTO R1
2651 026372  017737  154434  003126          MOV     @DPC,IDPC       ;SAVE DISPLAY PC...
2652 026400  011137  003130                  MOV     (R1),IDSR       ;...STATUS...
2653 026404  017737  154426  003132          MOV     @DXR,IDXR       ;...X POSITION...
2654 026412  017737  154422  003134          MOV     @DYR,IDYR       ;...Y POSITION...
2655 026420  012711  000001                  MOV     #SELPCS,(R1)    ;SAVE PCSAVE...
2656 026424  011137  003146                  MOV     (R1),IPCSAV     ;
2657 026430  012711  000002                  MOV     #SELFLG,(R1)    ;SAVE FLAGS...
2658 026434  011137  003150                  MOV     (R1),IFLAGS     ;
2659 026440  012711  000003                  MOV     #SELCSR,(R1)    ;SAVE CSR...
2660 026444  011137  003152                  MOV     (R1),ICSR       ;
2661 026450  012711  000004                  MOV     #SELMRR,(R1)    ;SAVE MAIN MEM-MGMT..
2662 026454  011137  003154                  MOV     (R1),IMAIN      ;
2663 026460  012711  000005                  MOV     #SELHBA,(R1)    ;SAVE HISTO. BASE...
2664 026464  011137  003156                  MOV     (R1),IHBASE     ;
2665 026470  012711  000006                  MOV     #SELXRR,(R1)    ;SAVE AUX. MEM-MGMT..
2666 026474  011137  003160                  MOV     (R1),IAUX       ;
2667 026500  012711  000007                  MOV     #SELCBA,(R1)    ;SAVE CHAR. BASE...
2668 026504  011137  003162                  MOV     (R1),ICBASE     ;
2669 026510  012711  000000                  MOV     #SELDSR,(R1)    ;END UP AT DSR...
2670 026514  011137  003144                  MOV     (R1),ISDSR      ;...AND SAVE IT.
2671                         ;               BITB    #IOKERR,INTFLAG ; DID WE GET AN ERROR INTERRUPT?
2672 026520  105737  026107                  TSTB    INTFLAG         ; ANY ERROR FLAGS SET?
2673 026524  001424                          BEQ     1$              ; BR IF NO
2674 026526  105737  026106                  TSTB    INTMASK         ; YES. IS CHECKING TURNED OFF?
2675 026532  100421                          BMI     1$              ; BR IF YES
2676 026534  013746  022300                  MOV     EXTA,-(SP)      ; SAVE ERROR EXTENSION ADDRESS.
2677 026540  012737  023324  022300          MOV     #PTINER,EXTA    ; SET EXTENSION TO PRINT TEST IN ERROR.
2678 026546  013701  026112                  MOV     INTCPC,R1       ;GET CPU PC.
2679 026552  004737  030216                  JSR     PC,INCERR       ;INCREMENT THE ERROR COUNTS
```

```
2680 026556                                        ERRDF   0,IFAULT,INTERR ;REPORT ERROR...
2681 026566  012637  022300                        MOV     (SP)+,EXTA      ;RESTORE EXTA
2682 026572  004737  030254                        JSR     PC,CKEMAX
2683
2684 026576  012601                          1$:   MOV     (SP)+,R1        ;RESTORE R1
2685 026600  000002                                RTI                     ;...AND DISMISS.
2686                                         ;
2687                                         ; AND A SIMILAR ONE FOR THE LUT INTERRUPTS.
2688                                         ;
2689 026602  000137                          LDUN:  JMP     @(SP)+
2690 026604  026606                                 LUTSAV
2691 026606  017737  154230  003136          LUTSAV: MOV     @LSR,ILSR       ; SAVE LUT REGISTERS.
2692 026614  017737  154224  003140                  MOV     @LDR,ILDR
2693 026622  017737  154220  003142                  MOV     @LMR,ILMR
2694 026630  000002                                 RTI                     ; AND DISMISS.
```

```
2696                                    ;
2697                                    ; SUBROUTINE TO INITIALIZE THE CURRENT UNIT DISPLAY.
2698                                    ; CALL "CLRW" TO INIT WITH IM'S IN WRITE MODE.
2699                                    ; CALL "CLRRW" TO INIT WITH IM'S IN WRITE-IN-RETRACE MODE.
2700                                    ; CALL "CLRRW1" TO INIT WITH IM'S IN WRITE-IN-RETRACE, 1'S-ONLY MODE.
2701                                    ;
2702                                            .ENABL  LSB
2703
2704
2705
2706                                    ;THE OPERATIONS ARE PERFORMED VIA PROGRAMMED I/O SO THAT EXECUTION OF A DISPLAY
2707                                    ; FILE DOES NOT HAVE TO TAKE PLACE.
2708 026632  005037  026754            CLRW:   CLR     CLMFCN          ;SAY CLEAR TO "WRITE-ONLY" MODE.
2709 026636  000407                            BR      1$
2710 026640  012737  176076  026754   CLRRW1: MOV     #RDWRT1,CLMFCN  ;SAY CLEAR TO READ/WRITE-1'S-ONLY.
2711 026646  000403                            BR      1$
2712 026650  012737  176074  026754   CLRRW:  MOV     #RDWRT,CLMFCN   ;SAY CLEAR TO READ/WRITE
2713                                    ;1$:    JSR     PC,DPINIT       ;DO INIT (CLEARS MEM, WRITE MODE, DISABLES J.S.) ;@@@I
2714 026656  012777  100000  154154   1$:     MOV     #BIT15,@DYR     ; DO SOFT INIT TO CLEAR MEMORIES
2715 026664  004737  027454                    JSR     PC,WAITF
2716 026670  005737  026754                    TST     CLMFCN          ;ANY THING ELSE TO DO?
2717 026674  001426                            BEQ     3$                      ;EXIT IF NOT.
2718 026676  042737  176000  026754           BIC     #176000,CLMFCN  ;ZAP OUT OP-CODE.
2719 026704  052737  040000  026754           BIS     #WRTMSR,CLMFCN  ;INSERT CODE FOR WRITING INTO DYR.
2720
2721 026712                            2$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
2722
2723 026714  013777  026754  154116           MOV     CLMFCN,@DYR     ;WRITE MEMORY STATUS-BA REGISTER
2724 026722  004737  027454                    JSR     PC,WAITF        ;WAIT FOR NOT BUSY.
2725 026726  004737  023524                    JSR     PC,ERRCHK       ;CHECK FOR ERROR
2726 026732                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
2727
2728 026734  062737  000400  026754           ADD     #CH1,CLMFCN     ;BUMP THE CHANNEL #.
2729 026742  032737  001400  026754           BIT     #CH3,CLMFCN     ;DONE ALL CHANNELS?
2730 026750  001360                            BNE     2$                      ;LOOP IF NOT.
2731 026752  000207                    3$:     RTS     PC                      ;RETURN WHEN DONE ALL 3 CHANNELS
2732 026754  000000                    CLMFCN: 0               ;FUNCTION FOR CLEARING MEMORY.
2733                                            .DSABL  LSB
2734                                    ;
2735                                    ; ANOTHER TO RELEASE THE DPU.
2736                                    ; EXECUTE "EXT STOP" IF HE'S RUNNING, "BUS-INIT" IF HE'S HUNG.
2737                                    ;
2738 026756  012777  000000  154050   RELEAS: MOV     #SELDSR,@DSR    ;MAKE SURE WE SELECT REAL DSR
2739 026764  005777  154044                    TST     @DSR            ;EXIT IF DPU IS STOPPED.
2740 026770  100410                            BMI     1$              ;EXIT IF HE'S FREE.
2741 026772  005077  154036                    CLR     @DSR            ;XCT "EXTERNAL STOP".
2742 026776  004737  027454                    JSR     PC,WAITF
2743 027002  103403                            BCS     1$              ;EXIT IF STOP WORKED.
2744                                                                    ;IT DIDN'T, XCT BUS-INIT...
2745 027004                                    BRESET                          ; BUS RESET.
2746 027006  004737  027454                    JSR     PC,WAITF        ;...AND WAIT AGAIN.
2747 027012  000207                    1$:     RTS     PC
2748                                    ;
2749                                    ; SUBROUTINE TO FILL ALL OF FREE CORE WITH PIXEL DATA
2750                                    ; FOR BIT MAP FUNCTIONS.  (R0) - DATA PATTERN.
2751                                    ; REGISTERS ARE NOT SAVED.
2752                                    ;
```

```
2753 027014  010001                FILL1:  MOV     R0,R1           ; FILL WITH (R0).
2754 027016  000402                        SKP2
2755 027020  010001                FILL2:  MOV     R0,R1
2756 027022  005101                        COM     R1              ; FILL WITH ALTERNATE WORDS.
2757 027024  013702  003240               MOV     FREE,R2         ; POINTER.
2758 027030  013703  003242               MOV     FRESIZ,R3       ; WORD COUNT.
2759 027034  042703  100003               BIC     #100003,R3              ; INSURE WC IS EVEN.
2760 027040  010022                1$:     MOV     R0,(R2)+        ; 1ST WORD...
2761 027042  010122                        MOV     R1,(R2)+        ;...NEXT WORD.
2762 027044  162703  000002               SUB     #2,R3
2763 027050  003373                        BGT     1$
2764 027052  000207                        RTS     PC
```

```
2821                                    ;
2822                                    ; SUBROUTINE TO BLAST THE LUT RAM WITH COLOR OR GREY-SCALE DATA.
2823                                    ;
2824 027054  012701  107066        LUMIN:  MOV     #LUMTBL,R1      ; GREY-SCALE TABLE ADDRESS.
2825 027060  012746  000001                MOV     #1,-(SP)        ; LUT WORDS / SHADE.
2826 027064  000404                        SKP2+2
2827 027066  012701  110066        NTSC:   MOV     #NTSC8,R1       ; BASIC 8 COLOR TABLE ADDRESS.
2828 027072  012746  000040                MOV     #32.,-(SP)      ; LUT WORDS / COLOR.
2829 027076  000404                        SKP2+2
2830 027100  012701  110106        EXPR:   MOV     #EXPRM,R1       ; EXPERIMENTAL 16 COLOR TABLE.
2831 027104  012746  000020                MOV     #16.,-(SP)      ; LUT WORDS / COLOR.
2832
2833 027110  005737  003216                TST     LUTAV
2834 027114  001424                        BEQ     3$              ; RETURN IF NO LUT INSTALLED.
2835 027116  005077  153720                CLR     @LSR            ; ADDR 0, NO INTERRUPTS.
2836 027122  052777  000100  153716        BIS     #GCOFF,@LMR     ; GAMMA CORRECTION OFF.
2837 027130  011600                1$:     MOV     (SP),R0         ; SET ITERATION K.
2838 027132  012177  153706                MOV     (R1)+,@LDR      ; BLAST THE RAM.
2839 027136  005777  153700        2$:     TST     @LSR            ; WAIT FOR READY.
2840 027142  100375                        BPL     .-4
2841 027144  105777  153672                TSTB    @LSR            ; LAST ADDRESS DONE ??
2842 027150  001406                        BEQ     3$              ; EXIT IF SO.
2843 027152  005300                        DEC     R0
2844 027154  001765                        BEQ     1$              ; GET NEXT COLOR.
2845 027156  016177  177776  153660        MOV     -2(R1),@LDR     ; REPEAT THIS COLOR.
2846 027164  000764                        BR      2$
2847 027166  005726                3$:     TST     (SP)+           ; FIX THE STACK...
2848 027170  000207                        RTS     PC              ;...AND RETURN.
2849                                    ;
2850                                    ; BLAST 16 SHADES OF THE BASIC COLOR IN (R1).
2851                                    ; LOCATION "SHADLY" SET BY CALLER TO PRODUCE A PLEASING
2852                                    ; VISUAL EFFECT.
2853                                    ;
2854 027172  005737  003216        SHAD16: TST     LUTAV
2855 027176  001433                        BEQ     3$              ; RETURN IF NO LUT INSTALLED.
2856 027200                                BREAK
2857 027202  005077  153634                CLR     @LSR            ; ADDR 0, NO INTERRUPTS.
2858 027206  052777  000100  153632        BIS     #GCOFF,@LMR     ; GAMMA CORRECT OFF.
2859 027214  042701  177356                BIC     #^C421,R1       ; SET DIMMEST SHADE THIS COLOR.
2860 027220  005002                        CLR     R2              ; 1ST SHADE IS ALWAYS BLACK.
2861 027222  012703  000020        1$:     MOV     #16.,R3         ; 16 LUT WORDS / SHADE.
2862 027226  010277  153612        2$:     MOV     R2,@LDR         ; WRITE DATA.
2863 027232  005777  153604                TST     @LSR
2864 027236  100375                        BPL     .-4
2865 027240  013700  003220                MOV     SHADLY,R0       ; DELAY.
2866 027244  005300                        DEC     R0
2867 027246  001376                        BNE     .-2
2868 027250  105777  153566                TSTB    @LSR            ; LUT RAM FULL ??
2869 027254  001404                        BEQ     3$              ; EXIT IF SO.
2870 027256  005303                        DEC     R3              ; THIS SHADE DONE ??
2871 027260  001362                        BNE     2$              ; NOT YET, LOOP.
2872 027262  060102                        ADD     R1,R2           ; YES, INCREMENT THE SHADE...
2873 027264  000756                        BR      1$              ;...AND CONTINUE.
2874 027266  000207                3$:     RTS     PC
```

```
2876                                    ;
2877                                    ; SUBROUTINES TO TRANSLATE AN ABSOLUTE X OR Y TO SHORT FORM
2878                                    ; DELTA X OR Y.  USED IN SVEC AND RPNT TESTS.
2879                                    ; INPUT R0 = ABS CO-ORD, IN THE RANGE (1000 +/-76).
2880                                    ; OUTPUT R0 = DX OR DY CORRECTLY POSITIONED AND SIGNED.
2881                                    ;
2882 027270  020027  001000            SDY:    CMP     R0,#1000
2883 027274  103003                             BHIS    .+10            ;SKP3 IF POS DELTA.
2884 027276  005400                             NEG     R0              ;MAKE NEG DELTA.
2885 027300  062700  000100                     ADD     #100,R0         ;SET SIGN BIT.
2886 027304  042700  177600                     BIC     #^C177,R0       ;R0 = +/-DY IN <6:0>.
2887 027310  000207                             RTS     PC
2888
2889 027312  004737  027270            SDX:    JSR     PC,SDY          ;MAKE A DY...
2890 027316  000300                             SWAB    R0              ;...MOVE TO HI BYTE.
2891 027320  006200                             ASR     R0              ;R0 = +/-DX IN <13:7>.
2892 027322  000207                             RTS     PC
2893                                    ;
2894                                    ; SUBROUTINE TO CONVERT 12 BIT OCTAL TO ASCII FOR DISPLAY.
2895                                    ; R1 = OCTAL X OR Y CO-ORD AT ENTRY.
2896                                    ; R1,,R2 = 4 DIGIT ASCII EQUIVALENT AT EXIT.
2897                                    ; R0, R1, AND R2 NOT SAVED AT ENTRY.
2898                                    ;
2899 027324  012737  030060  003232    CNVRT:  MOV     #'00,TEMP1      ;ASCII 00
2900 027332  013737  003232  003234            MOV     TEMP1,TEMP2     ;HERE TOO.
2901 027340  042701  170000                    BIC     #^C7777,R1      ;STRIP ANY CRAP BITS.
2902 027344  010102                            MOV     R1,R2
2903 027346  000302                            SWAB    R2
2904 027350  006202                            ASR     R2
2905 027352  042702  177770                    BIC     #^C7,R2
2906 027356  150237  003232                    BISB    R2,TEMP1        ;1ST (HI ORDER) DIGIT.
2907 027362  010102                            MOV     R1,R2
2908 027364  012700  000006                    MOV     #6,R0
2909 027370  006202                    1$:     ASR     R2              ;SHIFT OFF 6 BITS.
2910 027372  005300                             DEC     R0
2911 027374  001375                               BNE   1$
2912 027376  042702  177770                    BIC     #^C7,R2
2913 027402  150237  003233                    BISB    R2,TEMP1+1      ;2ND DIGIT.
2914 027406  010102                            MOV     R1,R2
2915 027410  012700  000003                    MOV     #3,R0
2916 027414  006202                    2$:     ASR     R2              ;SHIFT OFF 3 BITS.
2917 027416  005300                             DEC     R0
2918 027420  001375                               BNE   2$
2919 027422  042702  177770                    BIC     #^C7,R2
2920 027426  150237  003234                    BISB    R2,TEMP2        ;3RD DIGIT.
2921 027432  042701  177770                    BIC     #^C7,R1
2922 027436  150137  003235                    BISB    R1,TEMP2+1      ;4TH (AND LAST) DIGIT.
2923 027442  013701  003232                    MOV     TEMP1,R1        ;HI HALF => R1.
2924 027446  013702  003234                    MOV     TEMP2,R2        ;LO HALF => R2.
2925 027452  000207                            RTS     PC              ;AND EXIT.
```

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

```
2970                                    ;
2971                                    ; ROUTINE TO TEST FOR A NEXM IN THE RANGE (R1) THRU (R2).
2972                                    ; ON RETURN, IF "C" = 1, (R1) = NEXM ADDRESS.
2973                                    ;               "C" = 0, ALL ADDRESSES OK.
2974                                    ;
2975                                    ;CALL:  MOV ADR1,R1
2976                                    ;       MOV ADR2,R2
2977                                    ;       JSR PC,NXM
2978                                    ;       RETURN          ;TEST "C" AND PROCEED.
2979                                    ;
2980 027566  012737  027620  000004    NXM:    MOV     #2$,@#4         ; SET BUSERR VECTOR.
2981 027574  012737  000200  000006            MOV     #PRI04,@#6
2982 027602  005003                            CLR     R3              ;FLAG.
2983 027604  005711                    1$:     TST     (R1)            ;TEST THE ADDRESS(ES).
2984                                                                    ;IF ANY TRAP, CONTINUE AT 2$.
2985 027606  020102                            CMP     R1,R2           ;OTHERWISE, CONTINUE HERE.
2986 027610  001407                            BEQ     3$              ;BR IF FINISHED (NO NEXM'S).
2987 027612  062701  000002                    ADD     #2,R1           ;SET NEXT ADDRESS...
2988 027616  000772                            BR      1$              ;...AND CONTINUE.
2989
2990 027620  005103                    2$:     COM     R3              ;GOT ONE, SET FLAG...
2991 027622  012716  027630                    MOV     #3$,(SP)
2992 027626  000002                            RTI                     ;...AND DISMISS INTERRUPT...
2993 027630                            3$:     CLRVEC  #4              ;...AND GIVE BACK THE VECTOR.
2994 027636  005703                            TST     R3              ;DID WE CATCH ONE ??
2995 027640  001401                            BEQ     .+4             ;NO, "C" = 0, SKIP NEXT.
2996 027642  000261                            SEC                     ;YES, "C" = 1, (R1) - NEXM ADDR.
2997 027644  000207                            RTS     PC
2998                                    ;
2999                                    ; SUBROUTINE TO EXECUTE TEST ITERATIONS.
3000                                    ; EXIT WITH "C" SET IF LOOPS ALLOWED AND LOOP COUNT NON-ZERO.
3001                                    ; LOOP COUNTER IS SET BY "BEGIN.TEST" MACRO.
3002                                    ;
3003                                    ; CALL: LOOPTO  ARG
3004                                    ;
3005 027646                            LOOP:
3006 027646  005737  002510                    TST     NOITS           ; ITERATIONS INHIBITED?
3007 027652  001006                            BNE     1$              ; YES.
3008 027654  005737  003030                    TST     QVP             ; NO.
3009 027660  100403                            BMI     1$              ;LOOPS DISALLOWED IN QUICK PASS.
3010 027662  005337  027700                    DEC     LOOPK           ; BUMP LOOP COUNTER.
3011 027666  001002                            BNE     2$
3012 027670  000241                    1$:     CLC                     ;LOOP DISALLOWED, OR DONE.
3013 027672  000401                            SKP1
3014 027674  000261                    2$:     SEC                     ;LOOP ENABLED.
3015 027676  000207                            RTS     PC
3016 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     SFTFLAG           ; 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:
                                        ;
```

```
3075 030216  005237  030130          INCERK: INC     ERRK            ; INCREMENT LOCAL ERROR COUNT
3076 030222  010046                          MOV     R0,-(SP)        ; SAVE R0
3077 030224  013700  003026                  MOV     UNITN,R0        ; GET UNIT NUMBER,
3078 030230  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET.
3079 030232  062700  003302                  ADD     #ERTABL,R0      ; R0 GETS ADDRESS OF ERROR TABLE ENTRY.
3080 030236  005210                          INC     (R0)            ; INCREMENT THE DEVICE ERROR COUNT
3081 030240  032710  007777                  BIT     #7777,(R0)      ; DID WE OVERFLOW THE FIELD?
3082 030244  001001                          BNE     1$              ; BR IF NO.
3083 030246  005310                          DEC     (R0)            ; YES -- BACK IT UP TO 7777.
3084 030250  012600                  1$:     MOV     (SP)+,R0        ; RESTORE R0
3085 030252  000207                          RTS     PC              ; RETURN TO CALLER.
3086
3087 030254  010046                  CKEMAX: MOV     R0,-(SP)        ; SAVE R0
3088 030256  013700  003026                  MOV     UNITN,R0        ; GET UNIT NUMBER
3089 030262  006300                          ASL     R0              ; ... AND MAKE IT A WORD OFFSET
3090 030264  016000  003302                  MOV     ERTABL(R0),R0   ; GET ERROR TABLE ENTRY
3091 030270  042700  170000                  BIC     #170000,R0      ; EXTRACT ERROR COUNT FIELD
3092 030274  020037  002516                  CMP     R0,GERRMAX      ; IS GLOBAL LIMIT EXCEEDED FOR THIS UNIT?
3093 030300  103004                          BHIS    1$              ; BR IF YES
3094 030302  023737  030130  002514          CMP     ERRK,LERRMAX    ; IS LOCAL LIMIT EXCEEDED FOR THIS TEST?
3095 030310  103417                          BLO     2$              ; BR IF NO
3096 030312                          1$:     RFLAGS  R0              ; GET OPERATOR FLAGS
3097 030314  032700  000040                  BIT     #IDU,R0         ; IS DROPPING INHIBITED?
3098 030320  001013                          BNE     2$              ; BR IF YES.
3099 030322  012737  177777  003224          MOV     #-1,DUFLG       ; NO -- DROP THE UNIT
3100 030330                                  ERRDF   4,EMAXDU
3101 030340                                  DODU    UNITN
3102 030346                                  DOCLN
3103 030350  012600                  2$:     MOV     (SP)+,R0        ; RESTORE R0
3104 030352  000207                          RTS     PC              ; RETURN TO CALLER
3105
3106 030354  010046                  CKDROP: MOV     R0,-(SP)
3107 030356                                  RFLAGS  R0
3108 030360  032700  000040                  BIT     #IDU,R0
3109 030364  001011                          BNE     1$
3110 030366  012600                          MOV     (SP)+,R0
3111 030370  012737  177777  003224          MOV     #-1,DUFLG
3112 030376                                  DODU    UNITN
3113 030404                                  DOCLN                   ;ABORT THE PASS
3114 030406  000401                          BR      2$
3115 030410  012600                  1$:     MOV     (SP)+,R0
3116 030412  000207                  2$:     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      ;
```

```
3175 030740  005721                    7$:     TST     (R1)+           ;
3176 030742  001403                            BEQ     8$              ;
3177 030744  052737  000020  003252            BIS     #BIT4,MEMFLG    ;
3178 030752  006237  003252            8$:     ASR     MEMFLG          ;
3179 030756  103370                            BCC     7$              ;
3180 030760  004737  025402            CONSYC: JSR     PC,DPINIT
3181 030764  005003                            CLR     R3
3182 030766  005005                            CLR     R5              ; INIT SYNC CHAN TABLE PTR.
3183 030770  012777  000013  152036    1$:     MOV     #13,@DSR        ; CLEAR CSR.
3184 030776  005065  003270                    CLR     CTABS(R5)       ; CLEAR TABLE ENTRY.
3185 031002  010501                            MOV     R5,R1           ; READ CURSOR STATUS.
3186 031004  006201                            ASR     R1
3187 031006  000301                            SWAB    R1
3188 031010  052701  042100                    BIS     #42100,R1
3189 031014  010177  152016                    MOV     R1,@DXR
3190 031020  004737  027454                    JSR     PC,WAITF
3191 031024  012777  000003  152002          MOV     #3,@DSR         ; GOT A DATA READY ERROR?
3192 031032  017702  151776                    MOV     @DSR,R2
3193 031036  042702  003777                    BIC     #3777,R2
3194 031042  020227  150000                    CMP     R2,#150000
3195 031046  001412                            BEQ     2$              ; YES. (NO CHAN HERE).
3196 031050  052765  100000  003270          BIS     #BIT15,CTABS(R5) ; NO. SET CHAN EXISTS IN TABLE.
3197 031056  017702  151756                    MOV     @DYR,R2         ; GET INTERLACE BIT.
3198 031062  042702  157777                    BIC     #157777,R2
3199 031066  050265  003270                    BIS     R2,CTABS(R5)    ; SET IT IN TABLE.
3200 031072  050203                            BIS     R2,R3           ; REMEMBER IT IN R3.
3201 031074  062705  000002            2$:     ADD     #2,R5           ; NEXT SYNC CHAN.
3202 031100  020527  000010                    CMP     R5,#10          ; ALL SYNC CHANS CHECKED?
3203 031104  001331                            BNE     1$              ; NO.
3204 031106  012705  003270                    MOV     #CTABS,R5       ; YES. ALL INTERLACE BITS THE SAME?
3205 031112  010337  003230                    MOV     R3,INTLAC       ; REMEMBER INTERLACE/NON-INTERLACE STATUS
3206 031116  052703  100000                    BIS     #BIT15,R3       ; (R3 HAS LAST INTERLACE BIT FOUND).
3207 031122  005715                    3$:     TST     (R5)            ; GOT AN ENTRY IN TABLE?
3208 031124  001402                            BEQ     4$              ; NO.
3209 031126  021503                            CMP     (R5),R3         ; YES. INTERLACE BIT SAME?
3210 031130  001006                            BNE     5$              ; NO.
3211 031132  062705  000002            4$:     ADD     #2,R5           ; YES. NEXT TABLE ENTRY.
3212 031136  020527  003300                    CMP     R5,#CTABS+10    ; DONE?
3213 031142  001367                            BNE     3$              ; NO.
3214 031144  000410                            BR      6$              ; YES.
3215
3216 031146                            5$:     HRDERR  SCIDE,SYSCON
3217
3218 031166  005737  002512            6$:     TST     MFGFLG          ; MANUFACTURING FLAG SET?
3219 031172  001420                            BEQ     9$              ; NO.
3220 031174  023737  003256  003022            CMP     MEMTAB,MFGM0    ; YES. NEW MEM 0 = MFG MEM 0 MASTER?
3221 031202  001004                            BNE     7$              ; NO.
3222 031204  023737  003270  003024            CMP     SYCTAB,MFGS0    ; NEW SYNC CHAN 0 = MFG MASTER SYNC CHAN 0?
3223 031212  001410                            BEQ     9$              ; YES.
3224 031214                            7$:     HRDERR  SCME,SYSCON     ; NO.
3225
3226 031234  012737  000010  003254    9$:     MOV     #BIT3,SYCFLG    ; SET ONE FLAG TO INDICATE SYNC CHANS AVAIL.
3227 031242  012701  003270                    MOV     #SYCTAB,R1      ;
3228 031246  005721                    10$:    TST     (R1)+           ;
3229 031250  001403                            BEQ     11$             ;
3230 031252  052737  000020  003254            BIS     #BIT4,SYCFLG    ;
3231 031260  006237  003254            11$:    ASR     SYCFLG          ;
```

GLOBAL SUBROUTINES SECTION

```
3232 031264  103370                    BCC     10$             ;
3233 031266  000207                    RTS     PC
3234
3235
3236
```

```
                                        ;
                                        ; SUBROUTINE - ENABLE MEM MGT.
                                        ;
031270  005737  003244          KTON:   TST     KTFLG           ; GOT KT?
031274  001403                          BEQ     1$              ; NO.
031276  012737  000001  177572          MOV     #1,SR0          ; YES. ENABLE KT11.
031304  000207                  1$:     RTS     PC


                                        ;
                                        ; SUBROUTINE - DISABLE MEM MGT.
                                        ;
031306                          KTOFF:  ;TST    KTFLG           ; GOT KT11?
                                        ;BEQ    1$              ; NO.
031306  000240                          NOP
031310  000240                          NOP
031312  012737  000000  177572          MOV     #0,SR0          ; DISABLE KT.
031320  000207                  1$:     RTS     PC
```

```
                                        ;
                                        ; SUBROUTINE - FILL ALL MEMORY OVER 28K WITH 'STOPN'.
                                        ; (NOTE - ENABLE KT BEFORE CALLING THIS).
                                        ;
031322  005737  003244          KTBKGD: TST     KTFLG           ; GOT KT?
031326  001430                          BEQ     9$              ; NO. GET OUT.
031330  004737  031270                  JSR     PC,KTON         ; YES. ENABLE KT.
031334  012700  173000                  MOV     #STOPN,R0       ; FILL ALL MEM (>28K) WITH BACKGROUND.
031340  012737  001600  172354          MOV     #1600,@#KIPAR6
031346  012701  140000          1$:     MOV     #140000,R1
031352  012702  150000                  MOV     #150000,R2
031356  010021                  2$:     MOV     R0,(R1)+
031360  020102                          CMP     R1,R2
031362  001375                          BNE     2$
031364  023737  172354  003244          CMP     @#KIPAR6,KTFLG
031372  001404                          BEQ     3$
031374  062737  000200  172354          ADD     #200,@#KIPAR6
031402  000761                          BR      1$

031404  004737  031306          3$:     JSR     PC,KTOFF        ; DISABLE KT.
031410  000207                  9$:     RTS     PC
```

```
3280                                    ;
3281                                    ; SUBROUTINE TO SAVE R0-R5.
3282                                    ;
3283 031412  010037  031462    SAVERS: MOV     R0,SAVED
3284 031416  012700  031464            MOV     #SAVED+2,R0
3285 031422  010120                    MOV     R1,(R0)+
3286 031424  010220                    MOV     R2,(R0)+
3287 031426  010320                    MOV     R3,(R0)+
3288 031430  010420                    MOV     R4,(R0)+
3289 031432  010520                    MOV     R5,(R0)+
3290 031434  000207                    RTS     PC
3291
3292                                    ;
3293                                    ; SUBROUTINE TO RESTORE R0-R5.
3294                                    ;
3295 031436  012700  031464    RESTRS: MOV     #SAVED+2,R0
3296 031442  012001                    MOV     (R0)+,R1
3297 031444  012002                    MOV     (R0)+,R2
3298 031446  012003                    MOV     (R0)+,R3
3299 031450  012004                    MOV     (R0)+,R4
3300 031452  012005                    MOV     (R0)+,R5
3301 031454  013700  031462            MOV     SAVED,R0
3302 031460  000207                    RTS     PC
3303
3304
3305
3306 031462  000000    SAVED:  0
3307 031464  000000            0
3308 031466  000000            0
3309 031470  000000            0
3310 031472  000000            0
3311 031474  000000            0
```

```
3313                                        ;
3314                                        ; SUBROUTINE TO SET-UP VARIOUS ENVIRONMENTAL PARAMETERS.
3315                                        ;
3316 031476                                 ENVIRN: MEMORY  R0
3317 031500  010037  003240                         MOV     R0,FREE         ; GET 1ST FREE ADDRESS...
3318 031504  062737  000002  003240                 ADD     #2,FREE
3319 031512  011037  003242                         MOV     (R0),FRESIZ     ;...AND WORD COUNT.
3320 031516  162737  000004  003242                 SUB     #4,FRESIZ
3321 031524  013702  002012                         MOV     L$UNIT,R2       ; GET NUMBER OF UNITS
3322 031530  162737  000007  003242  10$:          SUB     #7,FRESIZ       ; TAKE AWAY 7 WORDS PER UNIT
3323 031536  005302                               DEC     R2
3324 031540  001373                               BNE     10$
3325
3326 031542  012737  000010  027564               MOV     #8.,K100US      ; ASSUME LSI FOR 100US TIMER.
3327 031550                                        READBUS
3328 031552                                        BCOMPLETE 1$
3329 031554  012737  000040  027564               MOV     #32.,K100US     ; NOT LSI -- CHANGE TIMER.
3330 031562                                 1$:     ;MOV    #60.,HZ         ; ASSUME 60 HZ SYSTEM.
3331                                        ;       MOV     #MAXY60,YMAX    ;
3332                                        ;       CLOCK   L,R0
3333                                        ;       BNCOMPLETE 2$
3334                                        ;       CLR     @0(R0)          ; INSURE CLOCK IS OFF.
3335                                        ;       MOV     6(R0),HZ        ; SET ACTUAL FREQUENCY.
3336                                        ;       CMP     HZ,#60.         ;
3337                                        ;       BEQ     2$              ;
3338                                        ;       MOV     #MAXY50,YMAX    ;
3339 031562  004737  031656                 2$:     JSR     PC,KTINIT       ; INIT KT11 MEM MGT.
3340 031566  000240  000240                         240,240
3341                                        ;*****************************************************************
3342                                        ; NOW THE FOLLOWING CODE PROVIDES A KEY RESTART FOR
3343                                        ; THE DIAGNOSTIC SUPERVISOR (DRS), AND FOR XXDP+ MONITOR.
3344                                        ; IF EITHER ONE CHANGES, THIS WILL PROBABLY TURN TO SHIT ..
3345                                        ; BUT FOR NOW -- IT'S CONVENIENT FOR DEBUGGING PURPOSES.
3346                                        ;
3347                                        ; THE FOLLOWING OFFSETS APPLY TO DRS REV D.
3348           025570                       XX.ENTRY= 25570         ; DRS (REV D) ENTRY FROM XXDP+.
3349           026050                       SUPVSR=   26050         ; DRS (REV D) SUPERVISOR RESTART.
3350           026072                       XX.EXIT=  26072         ; DRS (REV D) EXIT TO XXDP+.
3351           003726                       XXDP=      3726         ; XXDP  RESTART ADDRESS (XX3726).
3352                                        ;
3353 031572  104042                        KLUDGE: EMT     42              ; XXDP+ COMM BLOCK POINTER -> R0
3354 031574  010037  003236                        MOV     R0,XXCOMM       ; SAVE IT.
3355 031600  042700  007777                        BIC     #7777,R0        ; MASK TO 2K PAGE...
3356 031604  162700  027000                        SUB     #27000,R0       ;...MINUS DRS SIZE (5.75K)...
3357 031610  062700  026050                        ADD     #SUPVSR,R0      ;...PLUS DRS RESTART OFFSET...
3358 031614  010037  000206                        MOV     R0,@#206        ;... - DRS RESTART ADDRESS.
3359 031620  012737  000137  000204               MOV     #137,@#204      ; LOC 204 = JMP DRS-RESTART.
3360 031626  013700  003236                        MOV     XXCOMM,R0       ; GET POINTER AGAIN.
3361 031632  042700  007777                        BIC     #7777,R0        ; MASK TO 2K PAGE...
3362 031636  062700  003726                        ADD     #XXDP,R0        ;...PLUS RESTART OFFSET...
3363 031642  010037  000212                        MOV     R0,@#212        ;... = XXDP+ RESTART ADDRESS.
3364 031646  012737  000137  000210               MOV     #137,@#210      ; LOC 210 = JMP XXDP+
3365 031654  000207                               RTS     PC
3366                                        ;*****************************************************************
3367
```

```
3369                                    ;
3370                                    ; SUBROUTINE - IF OVER 28K & KT11 AVAIL, INIT KT11.
3371                                    ;
3372 031656  005037  003244            KTINIT: CLR     KTFLG
3373 031662  023727  002120  001577            CMP     L$HIME,#1577    ; GOT ENOUGH MEMORY (>28K)?
3374 031670  101444                            BLOS    9$              ; NO.
3375 031672  013700  000004                    MOV     @#ERRVEC,R0     ; SAVE OLD ERR VEC PTR.
3376 031676  012737  031770  000004            MOV     #2$,@#ERRVEC    ; SET ERR VEC PTR.
3377 031704  005737  177572                    TST     @#SR0           ; GOT KT11?
3378 031710  000240                            NOP                     ;  (TRAP IF NO).
3379 031712  013737  002120  003244            MOV     L$HIME,KTFLG    ; YES. SET KT FLAG.
3380 031720  042737  000177  003244            BIC     #177,KTFLG      ;
3381 031726  010037  000004                    MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
3382 031732  005000                            CLR     R0              ; R0 = AR DATA.
3383 031734  012701  172340                    MOV     #KIPAR0,R1      ; R1 = KI REGS PTR.
3384 031740  012761  077406  177740    1$:     MOV     #77406,-40(R1)  ; SET DESCRIPTOR REG.
3385 031746  010021                            MOV     R0,(R1)+        ; SET KIPAR REG.
3386 031750  062700  000200                    ADD     #200,R0         ; BUMP AR DATA BY ''4K''.
3387 031754  020027  002000                    CMP     R0,#2000        ; AT ''I/O''?
3388 031760  001367                            BNE     1$              ; NO.
3389 031762  012741  177600                    MOV     #177600,-(R1)   ; YES. SET KTPAR7 FOR I/O.
3390 031766  000405                            BR      9$
3391
3392 031770                            2$:     FORRTI
     031770  012716  031776                    MOV     #.+6,(SP)       ; FORCE RTI RETURN TO -
     031774  000002                            RTI                     ; - NEXT LOC.
3393 031776  010037  000004                    MOV     R0,@#ERRVEC     ; RESTORE OLD ERR VEC PTR.
3394
3395 032002  000207                    9$:     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$
```

```
3454 032234                                        DOCLN                   ;ABORT, NO MORE UNITS.
3455 032236  000240                        11$:    NOP
3456 032240                                PASRPT:
3457 032240  023727  002012  000001               CMP     L$UNIT,#1       ;HOW MANY UNITS SELECTED?
3458 032246  101752                               BLOS    NEWPAS          ;BR IF ONLY 1
3459 032250  005737  032726                       TST     DEVCNT          ;ARE ANY STILL RUNNING?
3460 032254  001747                               BEQ     NEWPAS          ;BR IF NO
3461 032256                                        RFLAGS  R0
3462 032260  032700  000100                       BIT     #ISR,R0         ;SHOULD WE PRINT STATISTICS
3463 032264  001343                               BNE     NEWPAS          ;BR IF NO
3464
3465 032266                                        DORPT
3466 032270  000741                               BR      NEWPAS
3467 032272                                10$:
3468
3469 032272                                SETU:   GPHARD  UNITN,R0        ;GET UNIT N P-TABLE POINTER.
3470 032300                                        BNCOMPLETE NXTU         ;BR IF UNIT NOT AVAILABLE.
3471 032302  005037  003224                       CLR     DUFLG           ;CLEAR 'DROPPED'' FLAG.
3472 032306  005237  032726                       INC     DEVCNT
3473 032312  012001                               MOV     (R0)+,R1        ; GET 1ST REGISTER ADDRESS.
3474 032314  012702  003032                       MOV     #DPC,R2
3475 032320  010122                        20$:    MOV     R1,(R2)+
3476 032322  062701  000002                       ADD     #2,R1
3477 032326  020227  003046                       CMP     R2,#LMR
3478 032332  101772                               BLOS    20$
3479
3480
3481 032334  012001                               MOV     (R0)+,R1        ; GET 1ST VECTOR ADDRESS.
3482 032336  011002                               MOV     (R0),R2         ; GET INTERRUPT PRIORITY
3483 032340  012037  003050                       MOV     (R0)+,DPRI      ; SET INTERRUPT PRIORITY.
3484 032344  010137  003052                       MOV     R1,STPV         ; SET STOP VECTOR POINTER...
3485 032350  012721  026242                       MOV     #DSTP,(R1)+     ;...VECTOR...
3486 032354  010221                               MOV     R2,(R1)+        ;...AND PRIORITY.
3487 032356  010137  003054                       MOV     R1,JSMV         ; DITTO JOY-STICK MATCH.
3488 032362  012721  026274                       MOV     #DJM,(R1)+
3489 032366  010221                               MOV     R2,(R1)+
3490 032370  010137  003056                       MOV     R1,TOTV         ; DITTO TIME-OUT.
3491 032374  012721  026210                       MOV     #DTO,(R1)+
3492 032400  010221                               MOV     R2,(R1)+
3493 032402  010137  003060                       MOV     R1,JSSV         · DITTO JOY-STICK SWITCH.
3494 032406  012721  026326                       MOV     #DJS,(R1)+
3495 032412  010221                               MOV     R2,(R1)+
3496 032414  010137  003062                       MOV     R1,LUTV         ; DITTO LUT DONE.
3497 032420  012721  026602                       MOV     #LDUN,(R1)+
3498 032424  010221                               MOV     R2,(R1)+
3499 032426  012037  003216                       MOV     (R0)+,LUTAV     ; GET 'LUT AVAILABLE'' ENTRY
3500 032432  012737  001676  003250               MOV     #MAXY60,YMAX    ; ASSUME 60HZ FREQUENCY.
3501 032440  012737  000074  003246               MOV     #60.,HZ         ;
3502 032446  005720                               TST     (R0)+           ; WHAT IS SELECTED FREQUENCY?
3503 032450  001406                               BEQ     21$             ; BR IF 60HZ
3504 032452  012737  000062  003246               MOV     #50.,HZ         ; SAY 50 HZ
3505 032460  012737  001776  003250               MOV     #MAXY50,YMAX    ; AND SET UP FOR 512 VISIBLE LINES
3506 032466                                21$:
3507 032466  005037  003064                       CLR     OPTI
3508 032472  005737  002502                       TST     DPUMOD          ; DPU ONLY?
3509 032476  001003                               BNE     1$              ; SKIP NEXT IF DPU-ONLY.
3510 032500  012737  040000  003064               MOV     #I,OPTI         ;SET OPTIONAL I(NTENSIFY) BIT.
```

```
3511 032506                             1$:
3512                                    ;       TST     QVP             ;1ST PASS ??
3513                                    ;       BEQ     5$              ;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.
```

SEQ 0081

```
3572 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
```

```
3678 033532                                    ENDRPT          ; UNUSED.
3679
3680
3681 033534    045     116     045  DEVSUM: .ASCIZ  /%N%ADEVICE STATUS SUMMARY:%N/
3682 033571    045     101     040  DEVONL: .ASCIZ  /%A  UNIT %D3%A   ONLINE,  ERRORS - %D%N/
3683 033641    045     101     040  DEVNXR: .ASCIZ  /%A  UNIT %D3%A   DROPPED, NON-EXISTENT REGISTER%N/
3684 033723    045     101     040  DEVNRD: .ASCIZ  /%A  UNIT %D3%A   DROPPED, NOT READY AT STARTUP%N/
3685 034004    045     101     040  DEVDRO: .ASCIZ  /%A  UNIT %D3%A   DROPPED, ERRORS - %D%N/
3686                                           .EVEN
```

```
3688                                    .SBTTL
3689                                    .SBTTL  HARDWARE TEST SECTION.
3690                                    .SBTTL
3691                                    .SBTTL  TEST #          DESCRIPTION
3692                                    .SBTTL  ------          -----------
3693                                    .SBTTL
3694            000000                  TN=0
3695                                    .NLIST  MC,ME
```

```
3697
                                        ;*****************************************************************
                                        ;*
                                        ;*      BEGIN TEST 1    -   STATIC RESET
                                        ;*
                                        ;*****************************************************************

                                        ; (ITERATION COUNT = 1.)

3698                                    ;
3699                                    ; EXECUTE A BUS RESET INSTRUCTION AND CHECK THE STATE OF THE DPU.
3700                                    ; THIS TEST IS ALMOST IDENTICAL TO THE "SOFT-INIT" ROUTINE (DPINIT:)
3701                                    ; USED TO INIT THE DPU AT THE START OF MANY OF THE TESTS IN THIS
3702                                    ; PROGRAM.  THE DIFFERENCES ARE THAT INSTEAD OF A SOFTWARE DPU INIT,
3703                                    ; A BUS RESET IS USED, AND THAT THE ERROR MESSAGE READS "BUS RESET",
3704                                    ; INSTEAD OF "SOFT-INIT".
3705                                    ;
3706 034122  004737  027702            STRSET: JSR     PC,TSTGO        ; TITLE.
3707
3708 034126                             1$:
3709 034126                                     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3710 034130                                     BRESET                          ; BUS RESET.
3711 034132  004737  027454                     JSR     PC,WAITF        ;WAIT FOR A COMPLETE
3712 034136  005002                             CLR     R2              ;EXPECT DPC TO BE 0.
3713 034140  012703  100000                     MOV     #BIT15,R3       ;EXPECT DSR TO BE 100000.
3714 034144  005777  146662                     TST     @DPC            ;TEST THE DPC, S/B -0.
3715 034150  001003                             BNE     2$              ;ERROR IF NOT 0.
3716 034152  027703  146656                     CMP     @DSR,R3         ;CHECK DSR.
3717 034156  001410                             BEQ     3$
3718 034160                             2$:     DFERR   BRINIT,PCSERR   ;REPORT ERROR IN DPC OR DSR.
3719 034200  004737  023422             3$:     JSR     PC,IIRCHK       ;CHECK THE INTERNAL REGISTERS
3720 034204  005004                             CLR     R4
3721 034206  005005                             CLR     R5              ;DXR AND DYR SHOULD BE 0.
3722 034210  005777  146622                     TST     @DXR
3723 034214  001003                             BNE     4$
3724 034216  005777  146616                     TST     @DYR
3725 034222  001410                             BEQ     5$
3726 034224                             4$:     DFERR   BRINIT,XYERR
3727
3728 034244                             5$:     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3729 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  030C72             65$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
3730 034266                                     EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
3731
3732 034272                                     END.TEST

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

```
;*** TEST 1      STATIC RESET
```

```
3734
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 2    -   RESET ONES
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 1.)

3735                                    ;
3736                                    ; SET ALL APPLICABLE REGISTERS TO ONES, CHECK THAT ONES GOT SET, THEN
3737                                    ; EXECUTE A BUS RESET INSTRUCTION AND CHECK THE STATE OF THE DPU.
3738                                    ; THIS IS THE SAME AS THE "STATIC RESET" TEST ABOVE, EXCEPT THAT
3739                                    ; THE REGISTERS ARE SET TO ONES FIRST.
3740                                    ; R1-EXPECTED, R2=RECEIVED, R0,R3-WORK REGS.
3741                                    ;
3742 034340  004737  027702            RESET1: JSR     PC,TSTGO        ; TITLE.
3743 034344                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3744
3745 034346  012701  177763                    MOV     #177763,R1
3746 034352                                    WTDSRA  R1,CSR
     034352  012700  000013                    MOV     #SETCSR,R0      ; WRITE R1 INTO CSR .
     034356  050100                            BIS     R1,R0
     034360  010077  146450                    MOV     R0,@DSR
3747 034364                                    RDDSRA  CSR,R2
     034364  012777  000003  146442            MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     034372  017702  146436                    MOV     @DSR,R2
3748 034376                                    IFERROR ICSRE,EXPREC,,CKLOOP
     034376  020102                            CMP     R1,R2           ; OK?
     034400  001411                            BEQ     64$             ; YES.
     034402                                    HRDERR  ICSRE,EXPREC,CKLOOP     ; NO.
     034424                             64$:
3749
3750 034424                                    WTDSRA  #177760,MRR
     034424  012777  177774  146402            MOV     #177760&^C17!SETMRR,@DSR
3751 034432                                    WTDSRA  #170000,MPM
     034432  012777  170015  146374            MOV     #170000&^C17!SETMPM,@DSR
3752 034440                                    RDDSRA  MRR,R2
     034440  012777  000004  146366            MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
     034446  017702  146362                    MOV     @DSR,R2
3753 034452  010203                            MOV     R2,R3
3754 034454  042702  170000                    BIC     #170000,R2      ; MRR OK?
3755 034460  012701  007777                    MOV     #7777,R1
3756 034464                                    IFERROR IMRRE,EXPREC,,CKLOOP
     034464  020102                            CMP     R1,R2           ; OK?
     034466  001411                            BEQ     65$             ; YES.
     034470                                    HRDERR  IMRRE,EXPREC,CKLOOP     ; NO.
     034512                             65$:
3757 034512  010302                            MOV     R3,R2           ; MPM OK?
3758 034514  042702  007777                    BIC     #7777,R2
3? ~34520  012701  170000                    MOV     #170000,R1
.       524                                    IFERROR IMPME,EXPREC,,CKLOOP
        524  020102                            CMP     R1,R2           ; OK?
       .526  001411                            BEQ     66$             ; YES.
     034530                                    HRDERR  IMPME,EXPREC,CKLOOP     ; NO.
     034552                             66$:
3761
```

```
3762 034552
     034552  012777  177776  146254          WTDSRA  #177760,XRR
                                             MOV     #177760&^C17!SETXRR,@DSR
3763 034560
     034560  012777  170017  146246          WTDSRA  #170000,XPM
                                             MOV     #170000&^C17!SETXPM,@DSR
3764 034566
     034566  012777  000006  146240          RDDSRA  XRR,R2
                                             MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
     034574  017702  146234                  MOV     @DSR,R2
3765 034600  010203                          MOV     R2,R3
3766 034602  042702  170000                  BIC     #170000,R2      ; XRR OK?
3767 034606  012701  007777                  MOV     #7777,R1
3768 034612                                  IFERROR IXRRE,EXPREC,,CKLOOP
     034612  020102                          CMP     R1,R2           ; OK?
     034614  001411                          BEQ     67$             ; YES.
     034616                                  HRDERR  IXRRE,EXPREC,CKLOOP     ; NO.
     034640                          67$:
3769 034640  010302                          MOV     R3,R2           ; XPM OK?
3770 034642  042702  007777                  BIC     #7777,R2
3771 034646  012701  170000                  MOV     #170000,R1
3772 034652                                  IFERROR IXPME,EXPREC,,CKLOOP
     034652  020102                          CMP     R1,R2           ; OK?
     034654  001411                          BEQ     68$             ; YES.
     034656                                  HRDERR  IXPME,EXPREC,CKLOOP     ; NO.
     034700                          68$:
3773
3774 034700                          10$:    BRESET                          ; BUS RESET.
3775 034702  004737  027454                  JSR     PC,WAITF        ;WAIT FOR A COMPLETE
3776 034706  005002                          CLR     R2              ;EXPECT DPC TO BE 0.
3777 034710  012703  100000                  MOV     #BIT15,R3       ;EXPECT DSR TO BE 100000.
3778 034714  005777  146112                  TST     @DPC            ;TEST THE DPC, S/B =0.
3779 034720  001003                          BNE     11$             ;ERROR IF NOT 0.
3780 034722  027703  146106                  CMP     @DSR,R3         ;CHECK DSR.
3781 034726  001410                          BEQ     12$
3782 034730                          11$:    DFERR   BRINIT,PCSERR   ;REPORT ERROR IN DPC OR DSR.
3783 034750  004737  023422          12$:    JSR     PC,IIRCHK       ;CHECK THE INTERNAL REGISTERS
3784 034754  005004                          CLR     R4
3785 034756  005005                          CLR     R5              ;DXR AND DYR SHOULD BE 0
3786 034760  005777  146052                  TST     @DXR
3787 034764  001003                          BNE     13$
3788 034766  005777  146046                  TST     @DYR
3789 034772  001410                          BEQ     14$
3790 034774                          13$:    DFERR   BRINIT,XYERR
3791
3792 035014                          14$:    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3793
3794 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.
3795 035036                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
3796
3797 035042                                  END.TEST

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

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

```
3799
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 3    -   SOFT INIT ONES
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 10.)

3800                                    ;
3801                                    ; SET ALL APPLICABLE REGISTERS TO ONES, CHECK THAT ONES GOT SET, THEN
3802                                    ; EXECUTE SOFT INIT AND CHECK THE STATE OF THE DPU.
3803                                    ; THIS IS THE SAME AS THE 'RESET ONES' TEST ABOVE, EXCEPT THAT
3804                                    ; A SOFT INIT IS EXECUTED INSTEAD OF A RESET INSTRUCTION.
3805                                    ; R1=EXPECTED, R2=RECEIVED, R0,R3=WORK REGS.
3806                                    ;
3807 035114  004737  027702            SINIT1: JSR     PC,TSTGO        ; TITLE.
3808 035120                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3809
3810 035122  012701  177763                    MOV     #177763,R1
3811 035126                                    WTDSRA  R1,CSR
      035126  012700  000013                   MOV     #SETCSR,R0      ; WRITE R1 INTO CSR .
      035132  050100                           BIS     R1,R0
      035134  010077  145674                   MOV     R0,@DSR
3812 035140                                    RDDSRA  CSR,R2
      035140  012777  000003  145666           MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      035146  017702  145662                   MOV     @DSR,R2
3813 035152                                    IFERROR ICSRE,EXPREC,,CKLOOP
      035152  020102                           CMP     R1,R2           ; OK?
      035154  001411                           BEQ     64$             ; YES.
      035156                                    HRDERR  ICSRE,EXPREC,CKLOOP     ; NO.
      035200                            64$:
3814
3815 035200                                    WTDSRA  #177760,MRR
      035200  012777  177774  145626           MOV     #177760&^C17!SETMRR,@DSR
3816 035206                                    WTDSRA  #170000,MPM
      035206  012777  170015  145620           MOV     #170000&^C17!SETMPM,@DSR
3817 035214                                    RDDSRA  MRR,R2
      035214  012777  000004  145612           MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
      035222  017702  145606                   MOV     @DSR,R2
3818 035226  010203                           MOV     R2,R3
3819 035230  042702  170000                   BIC     #170000,R2      ; MRR OK?
3820 035234  012701  007777                   MOV     #7777,R1
3821 035240                                    IFERROR IMRRE,EXPREC,,CKLOOP
      035240  020102                           CMP     R1,R2           ; OK?
      035242  001411                           BEQ     65$             ; YES.
      035244                                    HRDERR  IMRRE,EXPREC,CKLOOP     ; NO.
      035266                            65$:
3822 035266  010302                           MOV     R3,R2           ; MPM OK?
3823 035270  042702  007777                   BIC     #7777,R2
3824 035274  012701  170000                   MOV     #170000,R1
3825 035300                                    IFERROR IMPME,EXPREC,,CKLOOP
      035300  020102                           CMP     R1,R2           ; OK?
      035302  001411                           BEQ     66$             ; YES.
      035304                                    HRDERR  IMPME,EXPREC,CKLOOP     ; NO.
      035326                            66$:
3826
```

```
3827 035326                                        WTDSRA  #177760,XRR
     035326  012777  177776  145500                MOV     #177760&^C17!SETXRR,@DSR
3828 035334                                        WTDSRA  #170000,XPM
     035334  012777  170017  145472                MOV     #170000&^C17!SETXPM,@DSR
3829 035342                                        RDDSRA  XRR,R2
     035342  012777  000006  145464                MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
     035350  017702  145460                        MOV     @DSR,R2
3830 035354  010203                                MOV     R2,R3
3831 035356  042702  170000                        BIC     #170000,R2      ; XRR OK?
3832 035362  012701  007777                        MOV     #7777,R1
3833 035366                                        IFERROR IXRRE,EXPREC,,CKLOOP
     035366  020102                                CMP     R1,R2           ; OK?
     035370  001411                                BEQ     67$             ; YES.
     035372                                        HRDERR  IXRRE,EXPREC,CKLOOP     ; NO.
     035414                             67$:
3834 035414  010302                                MOV     R3,R2           ; MPM OK?
3835 035416  042702  007777                        BIC     #7777,R2
3836 035422  012701  170000                        MOV     #170000,R1
3837 035426                                        IFERROR IXPME,EXPREC
     035426  020102                                CMP     R1,R2           ; OK?
     035430  001410                                BEQ     68$             ; YES.
     035432                                        HRDERR  IXPME,EXPREC,   ; NO.
     035452                             68$:
3838
3839
3840 035452  012777  100000  145360                MOV     #100000,@DYR    ;DO SOFT INIT.
3841 035460  004737  027454                        JSR     PC,WAITF        ;WAIT FOR STOP
3842 035464  005002                                CLR     R2              ;EXPECT DPC TO BE 0.
3843 035466  012703  100000                        MOV     #BIT15,R3       ;EXPECT DSR TO BE 100000.
3844 035472  005777  145334                        TST     @DPC            ;TEST THE DPC, S/B =0.
3845 035476  001003                                BNE     11$             ;ERROR IF NOT 0.
3846 035500  027703  145330                        CMP     @DSR,R3         ;CHECK DSR.
3847 035504  001410                                BEQ     12$
3848 035506                             11$:       DFERR   NSINIT,PCSERR   ;REPORT ERROR IN DPC OR DSR.
3849 035526  004737  023422             12$:       JSR     PC,IIRCHK       ;CHECK THE INTERNAL REGISTERS
3850 035532  005004                                CLR     R4
3851 035534  005005                                CLR     R5              ;DXR AND DYR SHOULD BE 0.
3852 035536  005777  145274                        TST     @DXR
3853 035542  001003                                BNE     13$
3854 035544  005777  145270                        TST     @DYR
3855 035550  001410                                BEQ     14$
3856 035552                             13$:       DFERR   NSINIT,XYERR
3857
3858 035572                             14$:       ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3859
3860 035574                                        COMEND  1$
     035574  004737  027646                        JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
     035600  103401                                BCS     69$
     035602  000402                                BR      70$

     035604  000137  035120             69$:       JMP     1$

     035610  004737  030072             70$:       JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
3861 035614                                        EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
3862
3863 035620                                        END.TEST
```

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

```
3865
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 4    -   REGISTERS UNIQUE ADDRESS
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 10.)

3866                                    ;
3867                                    ; ALL REGISTERS ARE SET TO THEIR NORMAL STATIC STATE VIA A SOFT INIT.
3868                                    ; EACH APPLICABLE REGISTER, IN TURN, IS SET TO ITS CAPABILITY FOR ONES,
3869                                    ; AND THEN ALL OF THE OTHER REGISTERS ARE CHECKED THAT THEY REMAINED
3870                                    ; AT THEIR STATIC STATE.
3871                                    ;
3872 035704  004737  027702            DUADR:  JSR     PC,TSTGO        ; PRINT TITLE.
3873 035710                             1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3874 035712                                     INITDP
      035712  004737  025402                    JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
3875 035716  012703  177760                    MOV     #177760,R3
3876 035722  010377  145106                    MOV     R3,@DSR         ; WRITE DSR.
3877 035726  012701  000000                    MOV     #0,R1
3878 035732  017702  145074                    MOV     @DPC,R2         ; READ DPC INTO R2.
3879 035736                                     IFERROR DSDPE,DUADRE
      035736  020102                            CMP     R1,R2           ; OK?
      035740  001410                            BEQ     64$             ; YES.
      035742                                    HRDERR  DSDPE,DUADRE,   ; NO.
      035762                            64$:
3880 035762  017702  145050                    MOV     @DXR,R2         ; READ DXR INTO R2.
3881 035766                                     IFERROR DSDXE,DUADRE
      035766  020102                            CMP     R1,R2           ; OK?
      035770  001410                            BEQ     65$             ; YES.
      035772                                    HRDERR  DSDXE,DUADRE,   ; NO.
      036012                            65$:
3882 036012  017702  145022                    MOV     @DYR,R2         ; READ DYR INTO R2.
3883 036016                                     IFERROR DSDYE,DUADRE
      036016  020102                            CMP     R1,R2           ; OK?
      036020  001410                            BEQ     66$             ; YES.
      036022                                    HRDERR  DSDYE,DUADRE,   ; NO.
      036042                            66$:
3884 036042                             ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3885 036044                             BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3886 036046                             INITDP
      036046  004737  025402            JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
3887 036052  012703  177760            MOV     #177760,R3
3888 036056                             WTDSRA  R3,DSR
      036056  012700  000010            MOV     #SETDSR,R0      ; WRITE R3 INTO DSR .
      036062  050300                    BIS     R3,R0
      036064  010077  144744            MOV     R0,@DSR
3889 036070  012701  000001            MOV     #1,R1
3890 036074                             RDDSRA  PCS,R2
      036074  012777  000001  144732    MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
      036102  017702  144726            MOV     @DSR,R2
3891 036106                             IFERROR DSPCE,DUADRE
      036106  020102                    CMP     R1,R2           ; OK?
      036110  001410                    BEQ     67$             ; YES.
      036112                            HRDERR  DSPCE,DUADRE,   ; NO.
```

```
                036132                                  67$:
3892            036132  012701  100000                      MOV     #100000,R1
3893            036136                                      RDDSRA  FLG,R2
                036136  012777  000002  144670              MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
                036144  017702  144664                      MOV     @DSR,R2
3894            036150                                      IFERROR DSFLE,DUADRE
                036150  020102                              CMP     R1,R2           ; OK?
                036152  001410                              BEQ     68$             ; YES.
                036154                                      HRDERR  DSFLE,DUADRE,   ; NO.
                036174                                  68$:
3895            036174  012701  000003                      MOV     #3,R1
3896            036200                                      RDDSRA  CSR,R2
                036200  012777  000003  144626              MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
                036206  017702  144622                      MOV     @DSR,R2
3897            036212                                      IFERROR DSCSE,DUADRE
                036212  020102                              CMP     R1,R2           ; OK?
                036214  001410                              BEQ     69$             ; YES.
                036216                                      HRDERR  DSCSE,DUADRE,   ; NO.
                036236                                  69$:
3898            036236  012701  000000                      MOV     #0,R1
3899            036242                                      RDDSRA  MRR,R2
                036242  012777  000004  144564              MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
                036250  017702  144560                      MOV     @DSR,R2
3900            036254  042702  170000                      BIC     #170000,R2      ; MASK OUT MPM.
3901            036260                                      IFERROR DSMRE,DUADRE
                036260  020102                              CMP     R1,R2           ; OK?
                036262  001410                              BEQ     70$             ; YES.
                036264                                      HRDERR  DSMRE,DUADRE,   ; NO.
                036304                                  70$:
3902            036304                                      RDDSRA  MPM,R2
                036304  012777  000004  144522              MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
                036312  017702  144516                      MOV     @DSR,R2
3903            036316  042702  007777                      BIC     #7777,R2        ; MASK OUT MRR.
3904            036322                                      IFERROR DSMPE,DUADRE
                036322  020102                              CMP     R1,R2           ; OK?
                036324  001410                              BEQ     71$             ; YES.
                036326                                      HRDERR  DSMPE,DUADRE,   ; NO.
                036346                                  71$:
3905            036346                                      RDDSRA  XRR,R2
                036346  012777  000006  144460              MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
                036354  017702  144454                      MOV     @DSR,R2
3906            036360  042702  170000                      BIC     #170000,R2      ; MASK OUT XPM.
3907            036364                                      IFERROR DSXRE,DUADRE
                036364  020102                              CMP     R1,R2           ; OK?
                036366  001410                              BEQ     72$             ; YES.
                036370                                      HRDERR  DSXRE,DUADRE,   ; NO.
                036410                                  72$:
3908            036410                                      RDDSRA  XPM,R2
                036410  012777  000006  144416              MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
                036416  017702  144412                      MOV     @DSR,R2
3909            036422  042702  007777                      BIC     #7777,R2        ; MASK OUT XRR.
3910            036426                                      IFERROR DSXPE,DUADRE
                036426  020102                              CMP     R1,R2           ; OK?
                036430  001410                              BEQ     73$             ; YES.
                036432                                      HRDERR  DSXPE,DUADRE,   ; NO.
                036452                                  73$:
3911            036452                                      ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
```

```
3912 036454                                             BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3913 036456                                             INITDP
     036456  004737  025402                             JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
3914 036462  012703  177760                             MOV     #177760,R3
3915 036466                                             WTDSRA  R3,CSR
     036466  012700  000013                             MOV     #SETCSR,R0      ; WRITE R3 INTO CSR .
     036472  050300                                     BIS     R3,R0
     036474  010077  144334                             MOV     R0,@DSR
3916 036500  012701  100000                             MOV     #100000,R1
3917 036504                                             RDDSRA  DSR,R2
     036504  012777  000000  144322                     MOV     #SELDSR,@DSR    ; READ DSR INTO R2 .
     036512  017702  144316                             MOV     @DSR,R2
3918 036516                                             IFERROR CSDSE,DUADRE
     036516  020102                                     CMP     R1,R2           ; OK?
     036520  001410                                     BEQ     74$             ; YES.
     036522                                             HRDERR  CSDSE,DUADRE,   ; NO.
     036542                                     74$:
3919 036542  012701  000001                             MOV     #1,R1
3920 036546                                             RDDSRA  PCS,R2
     036546  012777  000001  144260                     MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
     036554  017702  144254                             MOV     @DSR,R2
3921 036560                                             IFERROR CSPCE,DUADRE
     036560  020102                                     CMP     R1,R2           ; OK?
     036562  001410                                     BEQ     75$             ; YES.
     036564                                             HRDERR  CSPCE,DUADRE,   ; NO.
     036604                                     75$:
3922 036604  012701  100000                             MOV     #100000,R1
3923 036610                                             RDDSRA  FLG,R2
     036610  012777  000002  144216                     MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
     036616  017702  144212                             MOV     @DSR,R2
3924 036622                                             IFERROR CSFLE,DUADRE
     036622  020102                                     CMP     R1,R2           ; OK?
     036624  001410                                     BEQ     76$             ; YES.
     036626                                             HRDERR  CSFLE,DUADRE,   ; NO.
     036646                                     76$:
3925 036646  012701  000000                             MOV     #0,R1
3926 036652                                             RDDSRA  MRR,R2
     036652  012777  000004  144154                     MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
     036660  017702  144150                             MOV     @DSR,R2
3927 036664  042702  170000                             BIC     #170000,R2      ; MASK OUT MPM.
3928 036670                                             IFERROR CSMRE,DUADRE
     036670  020102                                     CMP     R1,R2           ; OK?
     036672  001410                                     BEQ     77$             ; YES.
     036674                                             HRDERR  CSMRE,DUADRE,   ; NO.
     035714                                     77$:
3929 036714                                             RDDSRA  MPM,R2
     036714  012777  000004  144112                     MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
     036722  017702  144106                             MOV     @DSR,R2
3930 036726  042702  007777                             BIC     #7777,R2        ; MASK OUT MRR.
3931 036732                                             IFERROR CSMPE,DUADRE
     036732  020102                                     CMP     R1,R2           ; OK?
     036734  001410                                     BEQ     78$             ; YES.
     036736                                             HRDERR  CSMPE,DUADRE,   ; NO.
     036756                                     78$:
3932 036756                                             RDDSRA  XRR,R2
     036756  012777  000006  144050                     MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
     036764  017702  144044                             MOV     @DSR,R2
```

```
                     037020                          79$:
3935 037020                                          RDDSRA  XPM,R2
     037020  012777  000006  144006                 MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
     037026  017702  144002                         MOV     @DSR,R2
3936 037032  042702  007777                         BIC     #7777,R2        ; MASK OUT XRR.
3937 037036                                          IFERROR CSXPE,DUADRE
     037036  020102                                  CMP     R1,R2           ; OK?
     037040  001410                                  BEQ     80$             ; YES.
     037042                                          HRDERR  CSXPE,DUADRE,   ; NO.
     037062                          80$:
3938 037062                                          ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3939 037064                                          BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3940 037066                                          INITDP
     037066  004737  025402                         JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
3941 037072  012703  007777                         MOV     #7777,R3
3942 037076                                          WTDSRA  R3,MRR
     037076  012700  000014                         MOV     #SETMRR,R0      ; WRITE R3 INTO MRR .
     037102  050300                                  BIS     R3,R0
     037104  010077  143724                         MOV     R0,@DSR
3943 037110  012701  100000                         MOV     #100000,R1
3944 037114                                          RDDSRA  DSR,R2
     037114  012777  000000  143712                 MOV     #SELDSR,@DSR    ; READ DSR INTO R2 .
     037122  017702  143706                         MOV     @DSR,R2
3945 037126                                          IFERROR MRDSE,DUADRE
     037126  020102                                  CMP     R1,R2           ; OK?
     037130  001410                                  BEQ     81$             ; YES.
     037132                                          HRDERR  MRDSE,DUADRE,   ; NO.
     037152                          81$:
3946 037152  012701  000001                         MOV     #1,R1
3947 037156                                          RDDSRA  PCS,R2
     037156  012777  000001  143650                 MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
     037164  017702  143644                         MOV     @DSR,R2
3948 037170                                          IFERROR MRPCE,DUADRE
     037170  020102                                  CMP     R1,R2           ; OK?
     037172  001410                                  BEQ     82$             ; YES.
     037174                                          HRDERR  MRPCE,DUADRE,   ; NO.
     037214                          82$:
3949 037214  012701  100000                         MOV     #100000,R1
3950 037220                                          RDDSRA  FLG,R2
     037220  012777  000002  143606                 MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
     037226  017702  143602                         MOV     @DSR,R2
3951 037232                                          IFERROR MRFLE,DUADRE
     037232  020102                                  CMP     R1,R2           ; OK?
     037234  001410                                  BEQ     83$             ; YES.
     037236                                          HRDERR  MRFLE,DUADRE,   ; NO.
     037256                          83$:
3952 037256  012701  000003                         MOV     #3,R1
3953 037262                                          RDDSRA  CSR,R2
     037262  012777  000003  143544                 MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     037270  017702  143540                         MOV     @DSR,R2
3954 037274                                          IFERROR MRCSE,DUADRE
     037274  020102                                  CMP     R1,R2           ; OK?
```

SEQ 0098

```
                037276  0014'0                          BEQ     84$             ; YES.
                037300                                  HRDERR  MRCSE,DUADRE,   ; NO.
                037320                          84$:
3955            037320  012701  000000                  MOV     #0,R1
3956            037324                                  RDDSRA  MPM,R2
                037324  012777  000004  143502          MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
                037332  017702  143476                  MOV     @DSR,R2
3957            037336  042702  007777                  BIC     #7777,R2        ; MASK OUT MRR.
3958            037342                                  IFERROR MRMPE,DUADRE
                037342  020102                          CMP     R1,R2           ; OK?
                037344  001410                          BEQ     85$             ; YES.
                037346                                  HRDERR  MRMPE,DUADRE,   ; NO.
                037366                          85$:
3959            037366                                  RDDSRA  XRP,R2
                037366  012777  000006  143440          MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
                037374  017702  143434                  MOV     @DSR,R2
3960            037400  042702  170000                  BIC     #170000,R2      ; MASK OUT XPM.
3961            037404                                  IFERROR MRXRE,DUADRE
                037404  020102                          CMP     R1,R2           ; OK?
                037406  001410                          BEQ     86$             ; YES.
                037410                                  HRDERR  MRXRE,DUADRE,   ; NO.
                037430                          86$:
3962            037430                                  RDDSRA  XPM,R2
                037430  012777  000006  143376          MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
                037436  017702  143372                  MOV     @DSR,R2
3963            037442  042702  007777                  BIC     #7777,R2        ; MASK OUT XRR.
3964            037446                                  IFERROR MRXPE,DUADRE
                037446  020102                          CMP     R1,R2           ; OK?
                037450  001410                          BEQ     87$             ; YES.
                037452                                  HRDERR  MRXPE,DUADRE,   ; NO.
                037472                          87$:
3965            037472                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3966            037474                                  BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3967            037476                                  INITDP
                037476  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
3968            037502  012703  170000                  MOV     #170000,R3
3969            037506                                  WTDSRA  R3,MPM
                037506  012700  000015                  MOV     #SETMPM,R0      ; WRITE R3 INTO MPM .
                037512  050300                          BIS     R3,R0
                037514  010077  143314                  MOV     R0,@DSR
3970            037520  012701  100000                  MOV     #100000,R1
3971            037524                                  RDDSRA  DSR,R2
                037524  012777  000000  143302          MOV     #SELDSR,@DSR    ; READ DSR INTO R2 .
                037532  017702  143276                  MOV     @DSR,R2
3972            037536                                  IFERROR MPDSE,DUADRE
                037536  020102                          CMP     R1,R2           ; OK?
                037540  001410                          BEQ     88$             ; YES.
                037542                                  HRDERR  MPDSE,DUADRE,   ; NO.
                037562                          88$:
3973            037562  012701  000001                  MOV     #1,R1
3974            037566                                  RDDSRA  PCS,R2
                037566  012777  000001  143240          MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
                037574  017702  143234                  MOV     @DSR,R2
3975            037600                                  IFERROR MPPCE,DUADRE
                037600  020102                          CMP     R1,R2           ; OK?
                037602  001410                          BEQ     89$             ; YES.
                037604                                  HRDERR  MPPCE,DUADRE,   ; NO.
```

```
                037624
                                                        89$:
                037624                                          MOV     #100000,R1
3976            037624  012701  100000                          RDDSRA  FLG,R2
3977            037630                                          MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
                037630  012777  000002  143176                  MOV     @DSR,R2
                037636  017702  143172                          IFERROR MPFLE,DUADRE
3978            037642                                          CMP     R1,R2           ; OK?
                037642  020102                                  BEQ     90$             ; YES.
                037644  001410                                  HRDERR  MPFLE,DUADRE,   ; NO.
                037646
                037666                                  90$:
3979            037666  012701  000003                          MOV     #3,R1
3980            037672                                          RDDSRA  CSR,R2
                037672  012777  000003  143134                  MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
                037700  017702  143130                          MOV     @DSR,R2
3981            037704                                          IFERROR MPCSE,DUADRE
                037704  020102                                  CMP     R1,R2           ; OK?
                037706  001410                                  BEQ     91$             ; YES.
                037710                                          HRDERR  MPCSE,DUADRE,   ; NO.
                037730                                  91$:
398C            037730  012701  000000                          MOV     #0,R1
3983            037734                                          RDDSRA  MRR,R2
                037734  012777  000004  143072                  MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
                037742  017702  143066                          MOV     @DSR,R2
3984            037746  042702  170000                          BIC     #170000,R2      ; MASK OUT MPM.
3985            037752                                          IFERROR MPMRE,DUADRE
                037752  020102                                  CMP     R1,R2           ; OK?
                037754  001410                                  BEQ     92$             ; YES.
                037756                                          HRDERR  MPMRE,DUADRE,   ; NO.
                037776                                  92$:
3986            037776                                          RDDSRA  XRR,R2
                037776  012777  000006  143030                  MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
                040004  017702  143024                          MOV     @DSR,R2
3987            040010  042702  170000                          BIC     #170000,R2      ; MASK OUT XPM.
3988            040014                                          IFERROR MPXRE,DUADRE
                040014  020102                                  CMP     R1,R2           ; OK?
                040016  001410                                  BEQ     93$             ; YES.
                040020                                          HRDERR  MPXRE,DUADRE,   ; NO.
                040040                                  93$:
3989            040040                                          RDDSRA  XPM,R2
                040040  012777  000006  142766                  MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
                040046  017702  142762                          MOV     @DSR,R2
3990            040052  042702  007777                          BIC     #7777,R2        ; MASK OUT XRR.
3991            040056                                          IFERROR MPXPE,DUADRE
                040056  020102                                  CMP     R1,R2           ; OK?
                040060  001410                                  BEQ     94$             ; YES.
                040062                                          HRDERR  MPXPE,DUADRE,   ; NO.
                040102                                  94$:
3992            040102                                          ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
3993            040104                                          BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
3994            040106                                          INITDP
                040106  004737  025402                          JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
3995            040112  012703  007777                          MOV     #7777,R3
3996            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
3997            040130  012701  100000                          MOV     #100000,R
```

```
                                        RDDSRA  DSR,R2
        040134                          MOV     #SELDSR,@DSR    ; READ DSR INTO R2 .
        040142  017702  142666         MOV     @DSR,R2
                                        IFERROR XRDSE,DUADRE
        040146  020102                  CMP     R1,R2           ; OK?
        040150  001410                  BEQ     95$             ; YES.
        040152                          HRDERR  XRDSE,DUADRE,   ; NO.
        040172                  95$:
        040172  012701  000001         MOV     #1,R1
        040176                          RDDSRA  PCS,R2
        040176  012777  000001  142630  MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
        040204  017702  142624          MOV     @DSR,R2
        040210                          IFERROR XRPCE,DUADRE
        040210  020102                  CMP     R1,R2           ; OK?
        040212  001410                  BEQ     96$             ; YES.
        040214                          HRDERR  XRPCE,DUADRE,   ; NO.
        040234                  96$:
        040234  012701  100000         MOV     #100000,R1
        040240                          RDDSRA  FLG,R2
        040240  012777  000002  142566  MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
        040246  017702  142562          MOV     @DSR,R2
        040252                          IFERROR XRFLE,DUADRE
        040252  020102                  CMP     R1,R2           ; OK?
        040254  001410                  BEQ     97$             ; YES.
        040256                          HRDERR  XRFLE,DUADRE,   ; NO.
        040276                  97$:
        040276  012701  000003         MOV     #3,R1
        040302                          RDDSRA  CSR,R2
        040302  012777  000003  142524  MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
        040310  017702  142520          MOV     @DSR,R2
        040314                          IFERROR XRCSE,DUADRE
        040314  020102                  CMP     R1,R2           ; OK?
        040316  001410                  BEQ     98$             ; YES.
        040320                          HRDERR  XRCSE,DUADRE,   ; NO.
        040340                  98$:
        040340  012701  000000         MOV     #0,R1
        040344                          RDDSRA  MRR,R2
        040344  012777  000004  142462  MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
        040352  017702  142456          MOV     @DSR,R2
        040356  042702  170000          BIC     #170000,R2      ; MASK OUT MPM.
        040362                          IFERROR XRMRE,DUADRE
        040362  020102                  CMP     R1,R2           ; OK?
        040364  001410                  BEQ     99$             ; YES.
        040366                          HRDERR  XRMRE,DUADRE,   ; NO.
        040406                  99$:
        040406                          RDDSRA  MPM,R2
        040406  012777  000004  142420  MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
        040414  017702  142414          MOV     @DSR,R2
        040420  042702  007777          BIC     #7777,R2        ; MASK OUT MPM.
        040424                          IFERROR XRMPE,DUADRE
        040424  020102                  CMP     R1,R2           ; OK?
        040426  001410                  BEQ     100$            ; YES.
        040430                          HRDERR  XRMPE,DUADRE,   ; NO.
        040450                  100$:
        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$:
   4040 041016                                  RDDSRA  MPM,R2
        041016  012777  000004  142010          MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
        041024  017702  142004                  MOV     @DSR,R2
   4041 041030  042702  007777                  BIC     #7777,R2        ; MASK OUT MRR.
   4042 041034                                  IFERROR XPMPE,DUADRE
        041034  020102                          CMP     R1,R2           ; OK?
        041036  001410                          BEQ     107$            ; YES.
        041040                                  HRDERR  XPMPE,DUADRE,   ; NO.
        041060                          107$:
   4043 041060                                  RDDSRA  XRR,R2
        041060  012777  000006  141746          MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
        041066  017702  141742                  MOV     @DSR,R2
   4044 041072  042702  170000                  BIC     #170000,R2      ; MASK OUT XPM.
   4045 041076                                  IFERROR XPXRE,DUADRE
        041076  020102                          CMP     R1,R2           ; OK?
        041100  001410                          BEQ     108$            ; YES.
        041102                                  HRDERR  XPXRE,DUADRE,   ; NO.
        041122                          108$:
   4046 041122                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
   4047 041124                                  COMEND  1$
        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.
   4048 041144                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
   4049 041150                                  END.TEST

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

```
4051
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 5    -   INCREMENTING REGISTERS
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 10.)

4052                                    ;
4053                                    ; THE READ/WRITE CAPABILITIES OF THE REGISTERS WILL BE CHECKED
4054                                    ; BY ''INCREMENTING'' EACH REGISTER FROM ITS LOWEST POSSIBLE
4055                                    ; VALUE, MINUS ONE, TO ITS HIGHEST POSSIBLE VALUE, PLUS ONE,
4056                                    ; IE. 1 TO 177777 TO 0.
4057                                    ; R1=EXPECTED, R2=RECEIVED, R0,R3=WORK REGS.
4058                                    ;
4059 041232                            INCREG: COMBEG
      041232  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      041236  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
4060  041242  012701  000023            1$:     MOV     #SELCSR+20,R1   ; INCREMENT CSR.
4061
4062 041246                                     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4063 041250                            2$:     WTDSRA  R1,CSR
      041250  012700  000013                    MOV     #SETCSR,R0      ; WRITE R1 INTO CSR .
      041254  050100                            BIS     R1,R0
      041256  010077  141552                    MOV     R0,@DSR
4064 041262                                     RDDSRA  CSR,R2
      041262  012777  000003  141544            MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      041270  017702  141540                    MOV     @DSR,R2
4065 041274                                     IFERROR ICSRE,EXPREC
      041274  020102                            CMP     R1,R2           ; OK?
      041276  001410                            BEQ     64$             ; YES.
      041300                                    HRDERR  ICSRE,EXPREC,   ; NO.
      041320                            64$:
4066 041320                                     CKLOOP          ; LOOP IF ERROR.
4067 041322  062701  000020                    ADD     #20,R1          ; INCREMENT.
4068 041326  020127  000003                    CMP     R1,#SELCSR      ; DONE?
4069 041332  001346                            BNE     2$              ; NO.
4070 041334                                     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4071
4072 041336  012701  000001                    MOV     #1,R1           ; INCREMENT MRR.
4073 041342  012703  000020                    MOV     #20,R3
4074
4075 041346                                     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4076 041350                            3$:     WTDSRA  R3,MRR
      041350  012700  000014                    MOV     #SETMRR,R0      ; WRITE R3 INTO MRR .
      041354  050300                            BIS     R3,R0
      041356  010077  141452                    MOV     R0,@DSR
4077 041362                                     RDDSRA  MRR,R2
      041362  012777  000004  141444            MOV     #SELMRR,@DSR    ; READ MRR INTO R2 .
      041370  017702  141440                    MOV     @DSR,R2
4078 041374                                     IFERROR IMRRE,EXPREC
      041374  020102                            CMP     R1,R2           ; OK?
      041376  001410                            BEQ     65$             ; YES.
      041400                                    HRDERR  IMRRE,EXPREC,   ; NO.
      041420                            65$:
4079 041420                                     CKLOOP          ; LOOP IF ERROR.
```

```
4080 041422  062703  000020                ADD     #20,R3          ; INCREMENT.
4081 041426  062701  000001                ADD     #1,R1
4082 041432  020127  000001                CMP     R1,#1           ; DONE?
4083 041436  001403                        BEQ     4$              ; YES.
4084 041440  042701  170000                BIC     #170000,R1      ; NO.
4085 041444  000741                        BR      3$              ;
4086
4087 041446                          4$:   ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4088
4089 041450  012703  000020                MOV     #20,R3          ; INCREMENT MPM.
4090
4091 041454                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4092 041456                          5$:   WTDSRA  R3,MPM
     041456  012700  000015                MOV     #SETMPM,R0      ; WRITE R3 INTO MPM .
     041462  050300                        BIS     R3,R0
     041464  010077  141344                MOV     R0,@DSR
4093 041470  010301                        MOV     R3,R1           ;GET WRITTEN DATA
4094 041472  042701  007777                BIC     #7777,R1        ;LEAVE MASK BITS IN R1
4095 041476                                RDDSRA  MPM,R2
     041476  012777  000004  141330        MOV     #SELMPM,@DSR    ; READ MPM INTO R2 .
     041504  017702  141324                MOV     @DSR,R2
4096 041510                                IFERROR IMPME,EXPREC
     041510  020102                        CMP     R1,R2           ; OK?
     041512  001410                        BEQ     66$             ; YES.
     041514                                HRDERR  IMPME,EXPREC,   ; NO.
     041534                          66$:
4097 041534  005001                        CLR     R1              ;ASSUME WRITE-PROTECT BIT IS 0
4098 041536  032703  001000                BIT     #BIT9,R3        ;DID WE WRITE IT TO 1?
4099 041542  001402                        BEQ     50$             ;BR IF NOT.
4100 041544  052701  100000                BIS     #BIT15,R1       ;SET BIT 15 FOR HBASE COMPARE IF YES.
4101 041550                          50$:  RDDSRA  HBA,R2
     041550  012777  000005  141256        MOV     #SELHBA,@DSR    ; READ HBA INTO R2 .
     041556  017702  141252                MOV     @DSR,R2
4102 041562                                IFERROR IMPWPE,EXPREC
     041562  020102                        CMP     R1,R2           ; OK?
     041564  001410                        BEQ     67$             ; YES.
     041566                                HRDERR  IMPWPE,EXPREC,  ; NO.
     041606                          67$:
4103 041606                                CKLOOP          ; LOOP IF ERROR.
4104 041610  062703  000020                ADD     #20,R3          ; INCREMENT.
4105 041614  020327  000020                CMP     R3,#20          ; DONE?
4106 041620  001316                        BNE     5$              ; NO.
4107 041622                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4108
4109 041624  012701  000001                MOV     #1,R1           ; INCREMENT XRR.
4110 041630  012703  000020                MOV     #20,R3
4111
4112 041634                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4113 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
4114 041650                                RDDSRA  XRR,R2
     041650  012777  000006  141156        MOV     #SELXRR,@DSR    ; READ XRR INTO R2 .
     041656  017702  141152                MOV     @DSR,R2
4115 041662                                IFERROR IXRRE,EXPREC
     041662  020102                        CMP     R1,R2           ; OK?
```

```
        041664  001410                          BEQ     68$             ; YES.
        041666                                  HRDERR  IXRRE,EXPREC,   ; NO.
        041706                          68$:
   4116 041706                                  CKLOOP          ; LOOP IF ERROR.
   4117 041710  062703  000020                  ADD     #20,R3          ; INCREMENT.
   4118 041714  062701  000001                  ADD     #1,R1
   4119 041720  020127  000001                  CMP     R1,#1           ; DONE?
   4120 041724  001403                          BEQ     7$              ; YES.
   4121 041726  042701  170000                  BIC     #170000,R1      ; NO.
   4122 041732  000741                          BR      6$              ;
   4123
   4124 041734                          7$:     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
   4125
   4126 041736  012703  000020                  MOV     #20,R3          ; INCREMENT XPM.
   4127
   4128 041742                                  BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
   4129 041744                          8$:     WTDSRA  R3,XPM
        041744  012700  000017                  MOV     #SETXPM,R0      ; WRITE R3 INTO XPM .
        041750  050300                          BIS     R3,R0
        041752  010077  141056                  MOV     R0,@DSR
   4130 041756  010301                          MOV     R3,R1           ;GET WRITTEN DATA
   4131 041760  042701  007777                  BIC     #7777,R1        ;LEAVE MASK BITS IN R1
   4132 041764                                  RDDSRA  XPM,R2
        041764  012777  000006  141042          MOV     #SELXPM,@DSR    ; READ XPM INTO R2 .
        041772  017702  141036                  MOV     @DSR,R2
   4133 041776                                  IFERROR IXPME,EXPREC
        041776  020102                          CMP     R1,R2           ; OK?
        042000  001410                          BEQ     69$             ; YES.
        042002                                  HRDERR  IXPME,EXPREC,   ; NO.
        042022                          69$:
   4134 042022  005001                          CLR     R1              ;ASSUME WRITE-PROTECT BIT IS 0
   4135 042024  032703  001000                  BIT     #BIT9,R3        ;DID WE WRITE IT TO 1?
   4136 042030  001402                          BEQ     80$             ;BR IF NOT.
   4137 042032  052701  040000                  BIS     #BIT14,R1       ;SET BIT 14 FOR HBASE COMPARE IF YES.
   4138 042036                          80$:    RDDSRA  HBA,R2
        042036  012777  000005  140770          MOV     #SELHBA,@DSR    ; READ HBA INTO R2 .
        042044  017702  140764                  MOV     @DSR,R2
   4139 042050                                  IFERROR IXPWPE,EXPREC
        042050  020102                          CMP     R1,R2           ; OK?
        042052  001410                          BEQ     70$             ; YES.
        042054                                  HRDERR  IXPWPE,EXPREC,  ; NO.
        042074                          70$:
   4140 042074                                  CKLOOP          ; LOOP IF ERROR.
   4141 042076  062703  000020                  ADD     #20,R3          ; INCREMENT.
   4142 042102  020327  000020                  CMP     R3,#20          ; DONE?
   4143 042106  001316                          BNE     8$              ; NO.
   4144 042110                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
   4145
   4146 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.
   4147 042132                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
```

```
4148
4149 042136                            END.TEST

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

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

                                ; (ITERATION COUNT = 20.)

                                ;
                                ; FIRST INSURE THAT START, RESUME, DNOP, DJMP, AND STOP
                                ; ALL FUNCTION CORRECTLY.
                                ; IF NOT, ABORT, NO POINT IN CONTINUING.
                                ;
042210                          SSTST:  COMBEG
042210  004737  027702                  JSR     PC,TSTGO        ; TITLE.
042214  004737  026074                  JSR     PC,DPRESET      ; DO SOFT INIT.

042220                          1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.

042222  012702  042526                  MOV     #SST2,R2        ;SET 1ST EXP'D DPC...
042226  017703  140602                  MOV     @DSR,R3
042232  042703  176000                  BIC     #^C1777,R3
042236  052703  172000                  BIS     #STOP,R3        ;...AND DSR.
042242  012777  042516  140562          MOV     #SST1,@DPC      ;XCT START, DNOPS, AND STOP.
042250  004737  027454                  JSR     PC,WAITF        ;WAIT 1/2 SEC (MORE OR LESS).
042254  027702  140552                  CMP     @DPC,R2         ;DPC SHOULD POINT TO 'RT2''.
042260  001411                          BEQ     2$              ;BR IF IT'S RIGHT.
042262                                  DFERR   SSF,PCSERR      ;START-NOP-STOP FAILS.
042302  000421                          BR      3$              ;...BYPASS RESUME TEST.
042304  012702  042550          2$:     MOV     #SST3,R2        ;OK, SET NEXT EXP'D DPC.
042310  005277  140516                  INC     @DPC            ;XCT RESUME, DNOP'S, AND DJMP.
042314  004737  027454                  JSR     PC,WAITF
042320  027702  140506                  CMP     @DPC,R2         ;NOW DPC SHOULD BE AT 'RT3''.
042324  001412                          BEQ     4$              ;BR IF SO
042326                                  DFERR   RJF,PCSERR      ;RESUME-NOP-JUMP FAILS.
042346  004737  026756          3$:     JSR     PC,RELEAS       ;UNHANG IF HUNG
042352  004737  023422          4$:     JSR     PC,IIRCHK       ;VERIFY THAT INTERNAL REGS REMAIN STATIC
042356                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.


                                ; SO FAR, SO GOOD.  NOW LET'S PUT HIM IN A JUMP-SELF
                                ; LOOP, AND SEE IF WE CAN STOP HIM FROM THE CPU.
                                ;

042360                                  BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
042362                                  DPSTART #SST2
042362  012777  042526  140442          MOV     #SST2,@DPC      ; START THE DPU.
042370  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
042374  005277  140432                  INC     @DPC            ;RESUME, XCT DJMP SELF.
042400  004737  027454                  JSR     PC,WAITF
042404  103011                          BCC     5$              ;BR IF HE'S DOING IT.
042406                                  HRDERR  JSF             ;JUMP SELF FAILS.
042426  000417                          BR      6$
042430  000240  000240          5$:     240,240
042434  005077  140374                  CLR     @DSR            ;XCT 'EXTERNAL STOP''.
042440  004737  027454                  JSR     PC,WAITF
```

```
4196 042444  103410                          BCS     6$              ;BR IF HE STOPPED.
4197 042446                                  HRDERR  ESF             ;EXT STOP FAILS.
4198 042450  004737  023422          6$:     JSR     PC,IIRCHK       ;MAKE SURE INTERNALS STILL OK.
4199 042454  004737  026756                  JSR     PC,RELEAS       ;IT DIDN'T, GO RELEASE DPU.
4200 042460                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4201
4202 042462                                  LOOPTO  1$              ; REPEAT 'TIL LOOPER EXPIRES.
     042462  004737  027646                  JSR     PC,LOOP
     042466  103645                          BCS     1$
4203 042470  004737  030072                  JSR     PC,TSTEND       ; PRINT ERROR SUMMARY IF REQ'D.
4204 042474                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
4205                                 ;
4206                                 ; DISPLAY FILE FOR THIS TEST.
4207                                 ;
4208 042500  164000  164000  164000  SST1:   DNOP,DNOP,DNOP
4209 042506  173000                          STOPN
4210 042510  164000  164000  164000  SST2:   DNOP,DNOP,DNOP
4211 042516  160000  042530                  DJMP,..+10
4212 042522  173000  173000  173000          STOPN,STOPN,STOPN       ;SHOULD SKIP OVER THESE...
4213 042530  173000                          STOPN                   ;...AND LAND HERE.
4214 042532  164000  164000          SST3:   DNOP,DNOP
4215 042536  160000  042536                  DJMP,..-2               ;JUMP SELF FOR 'EXT STOP'.
4216 042542  173000  173000                  STOPN,STOPN             ;INSURANCE.
4217
4218 042546                                  END.TEST

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

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

                                ; (ITERATION COUNT = 5.)

                                ;
                                ; FIRST, TEST ALL THOSE OPCODES WHICH XFER PIXEL DATA TO THE
                                ; DPU, OPCODES 100000 THRU 130000, (CHAR THRU REL POINT).
                                ; USE ALL 10 BITS OF THE PIXEL DATA FIELD.
                                ;
042632                          OPCOD:  COMBEG
042632  004737  027702                  JSR     PC,TSTGO        ; TITLE.
042636  004737  026074                  JSR     PC,DPRESET      ; DO SOFT INIT.
042642  005037  023054                  CLR     LOOPFL          ;CLEAR 'LOOP-FLAG'' FOR C-BASE DATA.
042646  012737  043412  022300          MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
042654  012702  043446          14$:    MOV     #OPC1+6,R2      ;SET EXPECTED DPC.
042660  012737  102000  043440          MOV     #CHAR!LO,OPC1   ;INIT OPCODE = CHAR, LEVEL 0.
042666  012703  172000          1$:     MOV     #STOP,R3        ;SET EXPECTED DSR.

042672                          2$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.

042674  012777  043440  140130          MOV     #OPC1,@DPC      ;START, XCT OPCODE+DATA.
042702  004737  027454                  JSR     PC,WAITF
042706  027703  140122                  CMP     @DSR,R3         ;DOES DSR = STOP + DATA ??
042712  001410                          BEQ     3$              ;YES
042714                                  HRDERR  PDXI,PCSERR     ;PIXEL DATA XFER WRONG.
042734  013737  043440  023056  3$:     MOV     OPC1,GDDAT      ;GET OPCODE
042742  042737  003777  023056          BIC     #3777,GDDAT     ;CLEAR OUT INTENSITY
042750  004737  022410                  JSR     PC,FLGCHK       ;SEE THAT OPCODE IS IN FLAGS.
042754  012737  000001  023056          MOV     #1,GDDAT        ;EXPECT ''EMPTY'' ON PCSAVE...
042762  004737  022360                  JSR     PC,PCSCHK       ;...AND GO CHECK FOR IT.
042766  012737  000003  023056          MOV     #SELCSR,GDDAT   ;EXPECT CLEAR CSR (EXC. SEL)...
042774  004737  022440                  JSR     PC,CSRCHK       ;...AND CHECK FOR IT.
043000  013737  023054  023056          MOV     LOOPFL,GDDAT    ;EXPECT CBASE EITHER 0 OR DNOP...
043006  004737  022624                  JSR     PC,CBCHK        ;...AND CHECK FOR IT.
043012  005037  023056                  CLR     GDDAT           ;EXPECT 0 IN REMAINING REGS...
043016  004737  022466                  JSR     PC,MMCHK        ;...MAIN MEM MGMT.
043022  004737  022566                  JSR     PC,XMCHK        ;...AUX MEM MGMT.
043026  004737  022526                  JSR     PC,HBCHK        ;...HISTOGRAM BASE.
043032                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.

043034  020327  173777                  CMP     R3,#STOP!1777   ;ALL DATA BITS DONE ??
043040  001404                          BEQ     4$              ;BR IF SO.
043042  005203                          INC     R3              ;NO BUMP DATA BITS...
043044  005237  043440                  INC     OPC1            ;...HERE TOO...
043050  000710                          BR      2$              ;...AND CONTINUE.

043052  023727  043440  133777  4$:     CMP     OPC1,#RPNT!3777 ;DONE THRU RPNT ???
043060  001407                          BEQ     5$              ;BR IF SO.
043062  062737  004000  043440          ADD     #BIT11,OPC1     ;NO, SET NEXT OPCODE...
043070  042737  001777  043440          BIC     #1777,OPC1      ;...LEVEL 0...
043076  000673                          BR      1$              ;...AND CONTINUE.
                                ;
```

```
                                                ; NOW OF THE REMAINING OPCODES (134000 - 176000), EXECUTE ALL
                                                ; EXCEPT BMO/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) ;###I
043370  004737  027646                  JSR     PC,LOOP
043374  103404                          BCS     13$
043376  004737  030072          30$:    JSR     PC,TSTEND
043402                          EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
```

```
 4322 043406  000137  042654          13$:    JMP     14$             ;USE JMP TO GET BACK FOR LOOP.
 4323                                  ;
 4324                                  ; ON ERROR, SHOW THE FAILING DPU OPCODE.
 4325                                  ;
 4326 043412                           20$:    PRINTX  #OPCFX,OPC1     ; EXTENSION SHOWS BAD OPCODE.
 4327 043436  000207                           RTS     PC
 4328                                  ;
 4329                                  ; DISPLAY CODE.
 4330                                  ;
 4331 043440  102000  164000  173000  OPC1:   CHAR:LO, D'OP, STOPN    ; DISPLAY CODE.
 4332
 4333 043446                           END.TEST
                                      ;*****************************************************************
                                      ;*
                                      ;*      END TEST 7
                                      ;*
                                      ;*****************************************************************
```

```
*** TEST 8      INCREMENTING HISTOGRAM BASE ADDRESS

4335
                                        ;*****************************************************************
                                        ;*
                                        ;*      BEGIN TEST 8    -   INCREMENTING HISTOGRAM BASE ADDRESS
                                        ;*
                                        ;*****************************************************************

                                        ; (ITERATION COUNT = 5.)

4336
4337
4338                                    ;
4339                                    ; THE READ/WRITE CAPABILITY OF THE BASE HISTOGRAM COORDINATE WILL BE
4340                                    ; CHECKED BY ''INCREMENTING'' IT FROM 2 TO 1776 TO 0.
4341                                    ;  (NOTE THAT THE WRITTEN DATA WILL APPEAR SHIFTED 1 PLACE TO THE RIGHT.)
4342                                    ; R1- EXPECTED, R2=RECEIVED, R0,R3=WORK REGS.
4343                                    ;
4344 043544                            INCHBA: COMBEG
      043544  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      043550  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
4345 043554  012703  000002            1$:     MOV     #2,R3
4346 043560                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4347 043562  010337  043712            2$:     MOV     R3,11$          ; SET VALUE FOR HBA CODE.
4348 043566  010301                            MOV     R3,R1
4349 043570  006201                            ASR     R1
4350 043572  012777  043710  137232            MOV     #10$,@DPC       ; WRITE HBA.
4351 043600  004737  027454                    JSR     PC,WAITF
4352 043604                                    RDDSRA  HBA,R2
      043604  012777  000005  137222            MOV     #SELHBA,@DSR    ; READ HBA INTO R2 .
      043612  017702  137216                    MOV     @DSR,R2
4353 043616                                    IFERROR IHBAF,EXPREC
      043616  020102                            CMP     R1,R2           ; OK?
      043620  001410                            BEQ     64$             ; YES.
      043622                                    HRDERR  IHBAE,EXPREC.   ; NO.
      043642                            64$:
4354 043642                                    CKLOOP          ; LOOP ON ERROR.
4355 043644  062703  000002                    ADD     #2,R3           ; INCREMENT.
4356 043650  042703  176001                    BIC     #176001,R3      ; MASK UNWANTED.
4357 043654  020327  000000                    CMP     R3,#0           ; DONE?
4358 043660  001340                            BNE     2$              ; NO.
4359 043662                            30$:    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4360 043664                                    COMEND  1$
      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$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
4361 043704                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
4362
4363                                    ;
4364                                    ; DISPLAY CODE.
4365                                    ;
4366 043710  150000                    10$:    SETHB
4367 043712  000000                    11$:    0
4368 043714  173000                            STOPN
```

```
4569
4570 043716                    END.TEST

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

```
4372
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 9    -   INCREMENTING CHARACTER BASE ADDRESS
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT - 2.)

4373
4374                                    ;
4375                                    ; THE READ/WRITE CAPABILITY OF THE BASE CHARACTER ADDRESS WILL BE
4376                                    ; CHECKED BY ''INCREMENTING'' IT FROM 1 TO 177777 TO 0.
4377                                    ; R1= EXPECTED, R2=RECEIVED, R0,R3=WORK REGS.
4378                                    ;
4379
4380 044014                            INCCBA: COMBEG
      044014  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      044020  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
4381  044024  012701  000001            1$:     MOV     #1,R1
4382  044030                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4383  044032  010137  044152            2$:     MOV     R1,11$          ; SET VALUE FOR CBA CODE.
4384  044036  012777  044150  136766            MOV     #10$,@DPC       ; WRITE CBA.
4385  044044  004737  027454                    JSR     PC,WAITF
4386  044050                                    RDDSRA  CBA,R2
      044050  012777  000007  136756            MOV     #SELCBA,@DSR    ; READ CBA INTO R2 .
      044056  017702  136752                    MOV     @DSR,R2
4387  044062                                    IFERROR ICBAE,EXPREC
      044062  020102                            CMP     R1,R2           ; OK?
      044064  001410                            BEQ     64$             ; YES.
      044066                                    HRDERR  ICBAE,EXPREC,   ; NO.
      044106                            64$:
4388  044106                                    CKLOOP          ; LOOP ON ERROR.
4389  044110  062701  000001                    ADD     #1,R1           ; INCREMENT.
4390  044114  020127  000000                    CMP     R1,#0           ; DONE?
4391  044120  001344                            BNE     2$              ; NO.
4392  044122                            30$:    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4393  044124                                    COMEND  1$
      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$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
4394  044144                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
4395
4396                                    ;
4397                                    ; DISPLAY CODE.
4398                                    ;
4399 044150  152000                    10$:    SETCB
4400  044152  000000                    11$:    0
4401  044154  173000                            STOPN
4402
4403 044156                                    END.TEST

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

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

```
4405
                                ;*****************************************************************************
                                ;*
                                ;*      BEGIN TEST 10   -   DJMS/DPOP
                                ;*
                                ;*****************************************************************************

                                ; (ITERATION COUNT = 10.)

4406
4407                            ;
4408                            ; CHECK THE OPERATION OF THE DJMS AND DPOP OP CODES.
4409                            ;
4410 044222                    JMSPOP: COMBEG
      044222  004737  027702            JSR     PC,TSTGO        ; TITLE.
      044226  004737  026074            JSR     PC,DPRESET      ; DO SOFT INIT.
4411
4412 044232                    1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4413
4414 044234                            DPSTART #10$            ; DO A DJMS.
      044234  012777  044630  136570    MOV     #10$,@DPC       ; START THE DPU.
      044242  004737  027454            JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4415 044246  012701  044646            MOV     #21$,R1
4416 044252  017702  136554            MOV     @DPC,R2         ; READ DPC INTO R2.
4417 044256                            IFERROR JPDPCE,EXPREC,,CKLOOP
      044256  020102                    CMP     R1,R2           ; OK?
      044260  001411                    BEQ     64$             ; YES.
      044262                            HRDERR  JPDPCE,EXPREC,CKLOOP    ; NO.
      044304                    64$:
4418 044304  012701  044634            MOV     #11$,R1
4419 044310                            RDDSRA  PCS,R2
      044310  012777  000001  136516    MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
      044316  017702  136512            MOV     @DSR,R2
4420 044322                            IFERROR JPPCSE,EXPREC,,CKLOOP
      044322  020102                    CMP     R1,R2           ; OK?
      044324  001411                    BEQ     65$             ; YES.
      044326                            HRDERR  JPPCSE,EXPREC,CKLOOP    ; NO.
      044350                    65$:
4421 044350                            DPCONT                  ; NOW DO A DPOP.
      044350  052777  000001  136454    BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
      044356  004737  027454            JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4422 044362  012701  044636            MOV     #12$,R1
4423 044366  017702  136440            MOV     @DPC,R2         ; READ DPC.
4424 044372                            IFERROR JPDPCE,EXPREC,,CKLOOP
      044372  020102                    CMP     R1,R2           ; OK?
      044374  001411                    BEQ     66$             ; YES.
      044376                            HRDERR  JPDPCE,EXPREC,CKLOOP    ; NO.
      044420                    66$:
4425 044420                            DPCONT                  ; DO A DJMS (WITH AUX SEG BIT SET).
      044420  052777  000001  136404    BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
      044426  004737  027454            JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4426 044432  012701  044647            MOV     #21$!AUXSEG,R1
4427 044436  017702  136370            MOV     @DPC,R2         ; READ DPC INTO R2.
4428 044442                            IFERROR JPDPCE,EXPREC,,CKLOOP
      044442  020102                    CMP     R1,R2           ; OK?
      044444  001411                    BEQ     67$             ; YES.
      044446                            HRDERR  JPDPCE,EXPREC,CKLOOP    ; NO.
```

```
      044470                                    67$:
4429  044470  012701  044642                          MOV     #13$,R1
4430  044474                                        RDDSRA  PCS,R2
      044474  012777  000001  136332                MOV     #SELPCS,@DSR    ; READ PCS INTO R2 .
      044502  017702  136326                        MOV     @DSR,R2
4431  044506                                        IFERROR JPPCSE,EXPREC,,CKLOOP
      044506  020102                                CMP     R1,R2           ; OK?
      044510  001411                                BEQ     68$             ; YES.
      044512                                        HRDERR  JPPCSE,EXPREC,CKLOOP    ; NO.
      044534                                    68$:
4432  044534                                        DPCONT                  ; NOW DPOP AGAIN.
      044534  052777  000001  136270                BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
      044542  004737  027454                        JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4433  044546  012701  044644                        MOV     #20$,R1
4434  044552  017702  136254                        MOV     @DPC,R2         ; READ DPC.
4435  044556                                        IFERROR JPDPCE,EXPREC
      044556  020102                                CMP     R1,R2           ; OK?
      044560  001410                                BEQ     69$             ; YES.
      044562                                        HRDERR  JPDPCE,EXPREC,  ; NO.
      044602                                    69$:
4436  044602                                        ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4437
4438  044604                                        COMEND  1$
      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$:      JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
4439  044624                                        EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
4440                                            ;
4441                                            ; DISPLAY CODE
4442                                            ;
4443  044630  160001  044644            10$:      DJMS,20$
4444  044634  173000                    11$:      STOPN
4445  044636  160001  044645            12$:      DJMS,20$!AUXSEG
4446  044642  173000                    13$:      STOPN
4447
4448  044644  173000                    20$:      STOPN
4449  044646  165000                    21$:      DPOP
4450  044650  173000                    22$:      STOPN
4451
4452  044652                                        END.TEST

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

```
4454                                    ;*****************************************************************
                                        ;*
                                        ;*      BEGIN TEST 11   -   MAIN MEM MGT ACCESS (OVER 28K)
                                        ;*
                                        ;*****************************************************************

                                        ; (ITERATION COUNT - 1.)


4455
4456
4457                                    ; CHECK THAT A DISPLAY FILE CAN BE ACCESSED VIA MAIN MEM MGT.
4458                                    ; IN MEMORY OVER 28K.
4459                                    ;
4460 044744                            MMMAC:  COMBEG
      044744  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      044750  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
4461  044754  005737  003244                    TST     KTFLG           ; GOT KT & >28K?
4462  044760  001016                            BNE     MMMACA          ; YES.
4463  044762                                    RFLAGS  R0              ; ARE WE PRINTING TEST NOS.?
4464  044764  032700  001000                    BIT     #PNT,R0         ; ?
4465  044770  001410                            BEQ     1$              ; BR IF NO -- DON'T PRINT
4466  044772                                    PRINTF  #MNOKT          ; NO.
4467  045012                            1$:     EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
4468
4469
4470  045016  004737  031322            MMMACA: JSR     PC,KTBKGD       ; SET UP BACKGROUND.
4471  045022  012703  001600            1$:     MOV     #1600,R3        ; INIT BANK PTR. (TO START OF 28K PAGE)
4472
4473  045026                            2$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4474
4475  045030  010337  172354                    MOV     R3,@#KIPAR6     ; SET KT TO TEST PAGE.
4476  045034  010301                            MOV     R3,R1           ; SET UP MRR.
4477  045036  006301                            ASL     R1
4478  045040  006301                            ASL     R1
4479  045042  006301                            ASL     R1
4480  045044  006301                            ASL     R1
4481  045046                                    WTDSRA  R1,MRR
      045046  012700  000014                    MOV     #SETMRR,R0      ; WRITE R1 INTO MRR .
      045052  050100                            BIS     R1,R0
      045054  010077  135754                    MOV     R0,@DSR
4482  045060  012705  140000                    MOV     #140000,R5      ; INIT LOC PTR.
4483  045064  004737  031270                    JSR     PC,KTON         ; ENABLE KT.
4484  045070  012715  164000                    MOV     #DNOP,(R5)      ; REPLACE ''STOPN'' WITH 'DNOP'' IN TEST LOC.
4485  045074  004737  031306                    JSR     PC,KTOFF        ; DISABLE KT.
4486  045100                                    DPSTART #0
      045100  012777  000000  135724            MOV     #0,@DPC ; START THE DPU.
      045106  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4487  045112  000413                            BR      4$
4488
4489  045114  004737  031270            3$:     JSR     PC,KTON         ; ENABLE KT.
4490  045120  012715  164000                    MOV     #DNOP,(R5)      ; REPLACE ''STOPN'' WITH 'DNOP'' IN TEST LOC.
4491  045124  004737  031306                    JSR     PC,KTOFF        ; DISABLE KT.
4492  045130                                    DPCONT
      045130  052777  000001  135674            BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
      045136  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4493  045142  010501                    4$:     MOV     R5,R1           ; R1 = EXPECTED DPC, AND -
```

```
 4494 045144  042701  170000                    BIC     #170000,R1          ; -
 4495 045150  062701  000004                    ADD     #4,R1               ; - (R3 ALREADY = EXPECTED MRR).
 4496 045154  017702  135652                    MOV     @DPC,R2             ; READ DPC INTO R2.
 4497 045160                                    RDDSRA  MRR,R4
      045160  012777  000004  135646            MOV     #SELMRR,@DSR        ; READ MRR INTO R4 .
      045166  017704  135642                    MOV     @DSR,R4
 4498 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$:      HRDERR  MMMACE,MMMAE,       ; NO.
      045222                          64$:
 4499 045222  004737  031270                    JSR     PC,KTON             ; ENABLE KT.
 4500 045226  012715  173000                    MOV     #STOPN,(R5)         ; REPLACE ''DNOP'' WITH ''STOPN''.
 4501 045232  004737  031306                    JSR     PC,KTOFF            ; DISABLE KT.
 4502
 4503 045236  062705  000002                    ADD     #2,R5               ; ADVANCE LOC PTR.
 4504 045242  020527  147774                    CMP     R5,#147774          ; DONE ENOUGH LOCS?
 4505 045246  001322                            BNE     3$                  ; NO.
 4506 045250                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 4507 045252  020337  003244                    CMP     R3,KTFLG            ; YES. ALL BANKS DONE?
 4508 045256  001403                            BEQ     9$                  ; YES.
 4509 045260  062703  000200                    ADD     #200,R3             ; NO. ADVANCE BANK PTR.
 4510 045264  000660                            BR      2$
 4511
 4512 045266                          9$:       COMFND  1$
      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.
 4513 045306  004737  031306                    JSR     PC,KTOFF            ; DISABLE KT11.
 4514 045312                                    EXIT    TST                 ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
 4515
 4516
 4517 045316      045      116      045  MNOKT:  .ASCIZ  /%N%A  !!. NO KT, OR <32K - NO TEST !../
 4518                                            .EVEN
 4519 045366                                    END.TEST

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

```
4521                                    ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 12   -   AUX MEM MGT ACCESS (OVER 28K)
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 1.)

4522
4523                                    ;
4524                                    ; CHECK THAT A DISPLAY FILE CAN BE ACCESSED VIA AUX MEM MGT.
4525                                    ; IN MEMORY OVER 28K.
4526                                    ;
4527 045456                            XMMAC:  COMBEG
     045456  004737  027702                     JSR     PC,TSTGO        ; TITLE.
     045462  004737  026074                     JSR     PC,DPRESET      ; DO SOFT INIT.
4528 045466  005737  003244                     TST     KTFLG           ; GOT KT & >28K?
4529 045472  001016                             BNE     XMMACA          ; YES.
4530 045474                                     RFLAGS  R0
4531 045476  032700  001000                     BIT     #PNT,R0         ; ARE WE PRINTING?
4532 045502  001410                             BEQ     1$              ; BR IF NO
4533 045504                                     PRINTF  #MNOKT          ; NO.
4534 045524                            1$:      EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
4535
4536
4537 045530  004737  031322            XMMACA: JSR     PC,KTBKGD       ; SET UP BACKGROUND.
4538 045534  012703  001600            1$:     MOV     #1600,R3        ; INIT BANK PTR.
4539
4540 045540                            2$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4541
4542 045542  010337  172354                     MOV     R3,@#KIPAR6     ; SET KT TO TEST PAGE.
4543 045546  010301                             MOV     R3,R1           ; SET UP XRR.
4544 045550  006301                             ASL     R1
4545 045552  006301                             ASL     R1
4546 045554  006301                             ASL     R1
4547 045556  006301                             ASL     R1
4548 045560                                     WTDSRA  R1,XRR
     045560  012700  000016                     MOV     #SETXRR,R0      ; WRITE R1 INTO XRR .
     045564  050100                             BIS     R1,R0
     045566  010077  135242                     MOV     R0,@DSR
4549 045572  012705  140000                     MOV     #140000,R5      ; INIT LOC PTR.
4550 045576  004737  031270                     JSR     PC,KTON         ; ENABLE KT.
4551 045602  012715  164000                     MOV     #DNOP,(R5)      ; REPLACE ''STOPN'' WITH 'DNOP'' IN TEST LOC.
4552 045606  004737  031306                     JSR     PC,KTOFF        ; DISABLE KT.
4553 045612                                     DPSTART #10$
     045612  012777  046030  135212             MOV     #10$,@DPC       ; START THE DPU.
     045620  004737  027454                     JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4554 045624  000413                             BR      4$
4555
4556 045626  004737  031270            3$:     JSR     PC,KTON         ; ENABLE KT.
4557 045632  012715  164000                     MOV     #DNOP,(R5)      ; REPLACE ''STOPN'' WITH 'DNOP'' IN TEST LOC.
4558 045636  004737  031306                     JSR     PC,KTOFF        ; DISABLE KT.
4559 045642                                     DPCONT
     045642  052777  000001  135162             BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
     045650  004737  027454                     JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4560 045654  010501                    4$:     MOV     R5,R1           ; R1 = EXPECTED DPC, AND -
```

```
4561 045656  042701  170000                    BIC     #170000,R1      ; -
4562 045662  062701  000005                    ADD     #5,R1           ; - (R3 ALREADY = EXPECTED XRR).
4563 045666  017702  135140                    MOV     @DPC,R2         ; READ DPC INTO R2.
4564 045672                                    RDDSRA  XRR,R4
     045672  012777  000006  135134            MOV     #SELXRR,@DSR    ; READ XRR INTO R4 .
     045700  017704  135130                    MOV     @DSR,R4
4565 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$:      HRDERR  XMMACE,XMMAE,   ; NO.
     045734                          64$:
4566 045734  004737  031270                    JSR     PC,KTON         ; ENABLE KT.
4567 045740  012715  173000                    MOV     #STOPN,(R5)     ; REPLACE "DNOP" WITH "STOPN".
4568 045744  004737  031306                    JSR     PC,KTOFF        ; DISABLE KT.
4569
4570 045750  062705  000002                    ADD     #2,R5           ; ADVANCE LOC PTR.
4571 045754  020527  147774                    CMP     R5,#147774      ; DONE ENOUGH LOCS?
4572 045760  001322                            BNE     3$              ; NO.
4573 045762                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4574 045764  020337  003244                    CMP     R3,KTFLG        ; YES. ALL BANKS DONE?
4575 045770  001403                            BEQ     9$              ; YES.
4576 045772  062703  000200                    ADD     #200,R3         ; NO. ADVANCE BANK PTR.
4577 045776  000660                            BR      2$
4578
4579 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$:      JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
4580 046020  004737  031306                    JSR     PC,KTOFF        ; DISABLE KT11.
4581 046024                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
4582
4583                                           ;
4584                                           ; DISPLAY CODE.
4585                                           ;
4586 046030  160001  000001          10$:      DJMS,AUXSEG
4587 046034  173000                            STOPN
4588
4589 046036                                    END.TEST

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

```
4594
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 13   -   STOP INTERRUPT
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 40.)

4595                                    ;
4596                                    ; CHECK THE ACTION OF THE STOP INTERRUPT.
4597                                    ; ALSO CHECKED IS THE ACTION OF THE CSR "FORCE STOP INTR" BIT.
4598                                    ; REPEAT 40 TIMES AFTER QUICK PASS.
4599                                    ;
4600 046110  004737  027702            STPTST: JSR     PC,TSTGO        ;TITLE
4601
4602 046114                            19$:    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4603
4604 046116  004737  025402                    JSR     PC,DPINIT       ;INIT THE DPU SO WE CAN CHECK INTERNAL REGS.  ;@@@I
4605 046122                                    INTSET  STP
      046122  012737  177601  026106          MOV     #^O177400!IOKCKIN.IOKSTP,INTMASK        ; PRIME FLAG, EXPECT STP
4606 046130  012777  046742  134674          MOV     #STP1,@DPC      ;XCT "STOP AND INT".
4607 046136  004737  027454            1$:     JSR     PC,WAITF
4608 046142  105737  026107                    TSTB    INTFLAG         ; DID WE GET AN INTERRUPT?
4609 046146  001412                            BEQ     2$              ;BR IF INTERRUPT REC'D.
4610 046150  012701  046136                    MOV     #1$,R1          ;PC AT WAIT TIME.
4611 046154                                    HRDERR  NSI,INTERR      ;INT EXP'D, NOT REC'D.
4612 046174  004737  023422            2$:     JSR     PC,IIRCHK       ;CHECK INTERNALS FOR INIT'ED STATE.
4613 046200                                    INTSET  STP
      046200  012737  177601  026106          MOV     #^O177400!IOKCKIN!IOKSTP,INTMASK        ; PRIME FLAG, EXPECT STP
4614 046206  012777  046746  134617          MOV     #STP2,@DPC      ;XCT "STOP", ENABLE STILL SET.
4615 046214  004737  027454            3$:     JSR     PC,WAITF
4616 046220  105737  026107                    TSTB    INTFLAG         ;DID WE GET AN INTERRUPT?
4617 046224  001412                            BEQ     4$              ;BR IF INTERRUPT REC'D.
4618 046226  012701  046214                    MOV     #3$,R1          ;SET PC AT WAIT TIME.
4619 046232                                    HRDERR  NSI,INTERR      ;INT EXP'D, NOT REC'D.
4620 046252  004737  023422            4$:     JSR     PC,IIRCHK       ;CHECK INTERNALS FOR INIT'ED STATE
4621 046256                                    INTSET
      046256  012737  177600  026106          MOV     #^O177400!IOKCKIN,INTMASK       ; PRIME FLAG, NO INTR. EXPECTED.
4622 046264  012777  046752  134540          MOV     #STP3,@DPC      ;XCT "I OFF" AND "STOP".
4623 046272  004737  027454            5$:     JSR     PC,WAITF
4624 046276  123727  026107  000377            CMPB    INTFLAG,#^O377  ;DID WE GET AN INTERRUPT?
4625 046304  001412                            BEQ     6$              ;BR IF NO INTERRUPT.
4626 046306  012701  046272                    MOV     #5$,R1
4627 046312                                    HRDERR  USI,INTERR      ;INT REC'D, NOT EXP'D.
4628 046332  004737  023422            6$:     JSR     PC,IIRCHK       ;CHECK INTERNALS FOR INIT'ED STATE
4629 046336                                    INTSET
      046336  012737  177600  026106          MOV     #^O177400!IOKCKIN,INTMASK       ; PRIME FLAG, NO INTR. EXPECTED.
4630 046344  012777  046760  134460          MOV     #STP4,@DPC      ;XCT "I ON" AND "STOP N".
4631 046352  004737  027454            7$:     JSR     PC,WAITF
4632 046356  123727  026107  000377            CMPB    INTFLAG,#^O377  ;DID WE GET AN INTERRUPT?
4633 046364  001412                            BEQ     8$              ;BR IF NO INTERRUPT.
4634 046366  012701  046352                    MOV     #7$,R1
4635 046372                                    HRDERR  USI,INTERR      ;INT REC'D, NOT EXP'D.
4636 046412  012777  000053  134414    8$:     MOV     #FORCSI!WRTCSR,@DSR     ;SET BIT TO FORCE STOP INTR.
4637 046420                                    INTSET  STP
      046420  012737  177601  026106          MOV     #^O177400!IOKCKIN!IOKSTP,INTMASK        ; PRIME FLAG, EXPECT STP
```

```
4638 046426  012777  046752  134376          MOV     #STP3,@DPC      ;XCT "I-OFF" AND STOP
4639 046434  004737  027454          9$:     JSR     PC,WAITF        ;WAIT FOR DONE
4640 046440  105737  026107                  TSTB    INTFLAG         ;DID WE GET AN INTERRUPT?
4641 046444  001412                          BEQ     10$             ;BR IF YES -- OK
4642 046446  012701  046434                  MOV     #9$,R1          ;GET CPU PC FOR ERR REPORT
4643 046452                                  HRDERR  NSI,INTERR      ;INTR EXP'D, NOT REC'D
4644 046472                          10$:    INTSET  STP
     046472  012737  177601  026106          MOV     #^O177400!IOKCKIN!IOKSTP,INTMASK        ; PRIME FLAG, EXPECT STP
4645 046500  012777  046760  134324          MOV     #STP4,@DPC      ;XCT "I-ON" AND "STOP N"
4646 046506  004737  027454          11$:    JSR     PC,WAITF        ;WAIT FOR DONE.
4647 046512  105737  026107                  TSTB    INTFLAG         ;DID WE GET AN INTERRUPT?
4648 046516  001412                          BEQ     12$             ;...BR IF YES -- OK
4649 046520  012701  046506                  MOV     #11$,R1         ;GET PC
4650 046524                                  HRDERR  NSI,INTERR      ;REPORT IF NOT
4651 046544  012777  000013  134262  12$:    MOV     #0!WRTCSR,@DSR  ;CLEAR THE CSR
4652 046552                                  SETPRI  #PRI07          ;LOCK OUT INTERRUPTS
4653 046560  105037  026106                  CLRB    INTMASK         ;DON'T EXPECT ANY INTERRUPTS.
4654 046564  012777  046742  134240          MOV     #STP1,@DPC      ;XCT "STOP AND INTR"
4655 046572  004737  027454                  JSR     PC,WAITF        ;WAIT FOR DONE
4656 046576  012737  100001  023056          MOV     #BIT15.PINTRA,GDDAT     ;EXPECT PENDING INTR
4657 046604  004737  022410                  JSR     PC,FLGCHK       ;CHECK FLAGS FOR IT.
4658 046610  012777  000012  134216          MOV     #CLFLGS,@DSR    ;CLEAR THE INTERRUPT.
4659 046616  012737  100000  023056          MOV     #100000,GDDAT
4660 046624  004737  022410                  JSR     PC,FLGCHK       ;...AND VERIFY IT.
4661 046630                                  INTSET
     046630  012737  177600  026106          MOV     #^O177400!IOKCKIN,INTMASK       ; PRIME FLAG, NO INTR. EXPECTED.
4662 046636                                  SETPRI  #PRI00          ;ALLOW INTERRUPTS
4663 046644  000240                  13$:    NOP                     ;....AND SEE IF WE GET INTR
4664 046646  000240                          NOP
4665 046650  000240                          NOP
4666 046652  000240                          NOP
4667 046654  004737  027454                  JSR     PC,WAITF
4668 046660  105737  026107                  TSTB    INTFLAG         ;DID WE GET AN INTERRUPT?
4669 046664  100412                          BMI     14$             ;OK IF NO INTR
4670 046666  012701  046644                  MOV     #13$,R1         ;SAVE CPU PC
4671 046672                                  HRDERR  USI,INTERR      ;SAY UNEXPECTED INTR
4672 046712  004737  023434          14$:    JSR     PC,IIRCNF       ;CHECK INTERNALS FOR INIT'ED STATE
4673 046716                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4674
4675 046720                                  LOOPTO  18$             ; REPEAT 'TIL LOOPER EXPIRES.
     046720  004737  027646                  JSR     PC,LOOP
     046724  103404                          BCS     18$
4676 046726  004737  030072                  JSR     PC,TSTEND
4677 046732                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
4678 046736  000137  046114          18$:    JMP     19$     ;JUMP TO START OF LOOP
4679                                 ;
4680                                 ; DISPLAY CODE
4681                                 ;
4682 046742  164002  173400          STP1:   SYNC+1,STOPI            ;STOP AND INTERRUPT.
4683 046746  164002  172000          STP2:   SYNC+1,STOP             ;AGAIN.  ENABLE SHOULD HAVE...
4684                                                                 ;...BEEN SAVED.
4685 046752  164002  171000  172000  STP3:   SYNC+1,SIOFF,STOP       ;ENABLE OFF AND STOP (NO INT).
4686 046760  164002  171400  173000  STP4:   SYNC+1,SION,STOPN       ;ENABLE ON, STOP WITHOUT INTERRUPT.
4687
4688 046766                                  END.TEST

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

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

```
                                        ;*****************************************************************
                                        ;*
                                        ;*      BEGIN TEST 14  -  DPR TIME-OUT INTERRUPT
                                        ;*
                                        ;*****************************************************************

                                        ;  ITERATION COUNT = 100.)
                                        ;
                                        ; SEND THE DISPLAY TO A NON-EXISTENT MEMORY LOCATION
                                        ; AND VERIFY THAT WE RESPONDS WITH A TIME-OUT INTERRUPT.
                                        ; REPEAT 100 TIMES AFTER QUICK PASS.
                                        ;
047050                                  TOTEST: COMREG
047050 004737 027702                            JSR     PC,TSTGO        ; TITLE.
047054 004737 026074                            JSR     PC,DPRESET      ; DO SOFT INIT.

047060                                  18:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
047062 012777 140014 133744                     MOV     #SETMAR:140000,@DSR     ;SET RELOC TO 600000, SINCE THE...
                                                                                ;...I/O PAGE IS REALLY 760000.

047070 012777 000012 133736                     MOV     #CLFLGS,@DSR    ;CLEAR THE FLAGS & PENDING INTERUPTS.
047076 012777 000013 133730                     MOV     #0,@RTCSR,@DSR  ;CLEAR THE ERROR CODE.
047104                                          INTSET  ERR
047104 012737 177604 026106                     MOV     #^O177400.IOCCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
047112 012777 160000 133712                     MOV     #160000,@DPC    ;NXT DISPLAY START...
047120 005000                           10$:    CLR     R0              ; ... SHOULD GET TIME-OUT.
047122 005300                           11$:    DEC     R0
047124 001376                                   BNE     11$
047126 105737 026107                            TSTB    INTFLAG         ; DID WE GET PROPER INTERRUPT?
047132 001412                                   BEQ     2$              ; BR IF YES
047134 012701 047120                            MOV     #10$,R1         ;IF DIDN'T, GET ERROR PC...
047140                                          HWDERR  NTO,INTERR      ;NO TIME-OUT INTERRUPT.

047160                                  2$:
047160 012737 104003 023056                     MOV     #BIT15!<1+BIT11>,SELCSR,GDDAT   ;SET UP GDDAT W/ ERROR CODE.
047166 004737 022440                            JSR     PC,CSRCHK       ;CHECK FOR THE ERROR (NUM)
047172 012737 100000 023056                     MOV     #100000,GDDAT   ;SAY FLAGS SHOULD HAVE NO PENDING INTRS
047200 004737 022410                            JSR     PC,FLGCHK       ;...CHECK FOR IT.
047204 012777 000013 133622                     MOV     #0!RTCSR,@DSR   ;CLEAR ERROR CODE
047212 005037 023056                            CLR     GDDAT           ;EXPECT 0'S FOR...
047216 004737 022526                            JSR     PC,MBCHK        ;...M-BASE
047222 004737 022624                            JSR     PC,CBCHK        ;...C-BASE
047226 004737 022566                            JSR     PC,XRCHK        ;...AND AUX REGT.
047232                                          ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.

047234                                          LOOPTO  1$              ; REPEAT...
047234 004737 027646                            JSR     PC,LOOP
047240 103707                                   BCS     1$
047242 000400                                   BR      5$              ;...TIL LOOPER EXPIRES.

047244 012777 000016 133562             5$:     MOV     #SETMAR:0,@DSR          ;INSURE MAIN-SEG RELOC IS 0...
047252 012777 000000 133554                     MOV     #SELDSR,@DSR            ;SELECT REAL DSR
                                        ;               MOV     #DTO,@TOTV      ;...AND RESTORE VECTOR.
047260 004737 030072                            JSR     PC,TSTEND
```

SEQ 0126

```
4736                                    ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 15   -   ERROR CODES
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 10.)

4737
4738                                    ;
4739                                    ; GENERATE ALL ERROR CODES NOT TESTED ELSEWHERE IN THE PROGRAM
4740                                    ; AND POSSIBLE TO GENERATE WITH SOFTWARE.  CHECK FOR THE RETURN
4741                                    ; OF THE APPROPRIATE ERROR CODE.
4742                                    ;
4743 047332                            ERCODS: COMBEG
      047332  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      047336  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
4744 047342                            1$:
4745                                    ; TEST MPE - MEMORY PROTECTION (MAIN SEGMENT).
4746
4747 047342                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4748 047344                                    INITDP
      047344  004737  025402                    JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4749 047350                                    WTDSRA  #170000,MPM     ; LIMIT MAIN SEG TO 4K
      047350  012777  170015  133456            MOV     #170000&^C17!SETMPM,@DSR
4750 047356                                    INTSET  ERR
      047356  012737  177604  026106            MOV     #^0177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4751 047364                                    DPSTART #ECMPE1         ; FORCE MPE ON MAIN SEGMENT
      047364  012777  052264  133440            MOV     #ECMPE1,@DPC    ; START THE DPU.
      047372  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4752 047376  105737  026107                    TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4753 047402  001412                            BEQ     10$             ; BR IF YES.
4754 047404  012701  047372                    MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT.
4755 047410                                    HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4756 047430                            10$:
4757 047430  012701  114003                    MOV     #MPE,R1
4758 047434                                    RDDSRA  CSR,R2
      047434  012777  000003  133372            MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      047442  017702  133366                    MOV     @DSR,R2
4759 047446                                    IFERROR ECE,CSRER
      047446  020102                            CMP     R1,R2           ; OK?
      047450  001410                            BEQ     64$             ; YES.
      047452                                    HRDERR  ECE,CSRER,      ; NO.
      047472                            64$:
4760 047472                                    ENDSEG          ; >>>>>>>>>>>>>>>>>>>>>>>>>> END SEGMENT.
4761
4762                                    ; TEST MPE - MEMORY PROTECTION (AUXILIARY SEGMENT).
4763
4764 047474                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4765 047476                                    INITDP
      047476  004737  025402                    JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4766 047502                                    WTDSRA  #170000,XPM
      047502  012777  170017  133324            MOV     #170000&^C17!SETXPM,@DSR
4767 047510                                    INTSET  ERR
      047510  012737  177604  026106            MOV     #^0177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4768 047516                                    DPSTART #ECMPE2         ; FORCE MPE ON AUXILIARY SEGMENT.
```

```
                047516  012777  052270  133306          MOV     #ECMPE2,@DPC    ; START THE DPU.
                047524  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4769            047530  105737  026107                  TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4770            047534  001412                          BEQ     11$             ; BR IF YES.
4771            047536  012701  047524                  MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4772            047542                                  HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4773            047562                          11$:
4774            047562  012701  114003                  MOV     #MPE,R1
4775            047566                                  RDDSRA  CSR,R2
                047566  012777  000003  133240          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
                047574  017702  133234                  MOV     @DSR,R2
4776            047600                                  IFERROR ECE,CSRER
                047600  020102                          CMP     R1,R2           ; OK?
                047602  001410                          BEQ     65$             ; YES.
                047604                                  HRDERR  ECE,CSRER,      ; NO.
                047624                          65$:
4777            047624                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4778            047626  005737  002502                  TST     DPUMOD          ; ARE WE IN DPU-MODE ONLY?
4779            047632  001132                          BNE     3$              ; BR IF YES -- CAN'T DO READBACK
4780                                                                            ; TO CHECK FOR WRITE-PROTECT VIOLATIONS.
4781
4782                                            ; TEST MPE - MEMCRY WRITE PROTECTION (MAIN SEGMENT).
4783
4784            047634                                  BGNSEG
4785            047636                                  INITDP
                047636  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4786            047642                                  WTDSRA  #BIT9,MPM        ; SET MAIN-SEG WRITE PROTECT
                047642  012777  001015  133164          MOV     #BIT9&^C17!SETMPM,@DSR
4787            047650                                  INTSET  ERR
                047650  012737  177604  026106          MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4788            047656                                  DPSTART #ECMPE3         ; FORCE MPE BY TRYING DMA READBACK
                047656  012777  052302  133146          MOV     #ECMPE3,@DPC    ; START THE DPU.
                047664  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4789            047670  105737  026107                  TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4790            047674  001412                          BEQ     12$             ; BR IF YES.
4791            047676  012701  047664                  MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4792            047702                                  HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4793            047722                          12$:
4794            047722  012701  114003                  MOV     #MPE,R1         ; SET EXPECTED ERROR CODE 'MPE''
4795            047726                                  RDDSRA  CSR,R2
                047726  012777  000003  133100          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
                047734  017702  133074                  MOV     @DSR,R2
4796            047740                                  IFERROR ECE,CSRER
                047740  020102                          CMP     R1,R2           ; OK?
                047742  001410                          BEQ     66$             ; YES.
                047744                                  HRDERR  ECE,CSRER,      ; NO.
                047764                          66$:
4797            047764                                  ENDSEG
4798
4799
4800                                            ; TEST MPE - MEMORY WRITE PROTECTION (AUXILIARY SEGMENT).
4801
4802            047766                                  BGNSEG
4803            047770                                  INITDP
                047770  004737  025402                  JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4804            047774                                  WTDSRA  #BIT9,XPM        ; SET AUX-SEG WRITE PROTECT
                047774  012777  001017  133032          MOV     #BIT9&^C17!SETXPM,@DSR
```

```
4805 050002
     050002  012737  177604  026106          INTSET  ERR
                                              MOV     #^0177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4806 050010
     050010  012777  052316  133014          DPSTART #ECMPE4          ; FORCE MPE BY TRYING DMA READBACK
                                              MOV     #ECMPE4,@DPC     ; START THE DPU.
     050016  004737  027454                   JSR     PC,WAITF         ; WAIT FOR DISPLAY STOP.
4807 050022  105737  026107                   TSTB    INTFLAG          ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4808 050026  001412                           BEQ     13$              ; BR IF YES.
4809 050030  012701  050016                   MOV     #.-12,R1         ; NO. SO GET PC FOR ERROR REPORT,
4810 050034                                   HRDERR  NTO,INTERR       ; ... AND REPORT THE PROBLEM.
4811 050054                              13$:
4812 050054  012701  114003                   MOV     #MPE,R1          ; SET EXPECTED ERROR CODE 'MPE''
4813 050060                                   RDDSRA  CSR,R2
     050060  012777  000003  132746           MOV     #SELCSR,@DSR     ; READ CSR INTO R2 .
     050066  017702  132742                   MOV     @DSR,R2
4814 050072                                   IFERROR ECE,CSRER
     050072  020102                           CMP     R1,R2            ; OK?
     050074  001410                           BEQ     67$              ; YES.
     050076                                   HRDERR  ECE,CSRER,       ; NO.
     050116                              67$:
4815 050116                                   ENDSEG
4816
4817
4818 050120                              3$:
4819                                     ; TEST RSVDOP - RESERVED OPCODE (ILLEGAL INSTRUCTION
4820
4821 050120                                   BGNSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4822 050122                                   INITDP
     050122  004737  025402                   JSR     PC,DPINIT        ; GO DO SOFT INIT, CHECK THE STATE.
4823 050126                                   INTSET  ERR
     050126  012737  177604  026106          MOV     #^0177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4824 050134                                   DPSTART #ECRSV1          ;START DPU ON ILLEGAL INSTRUCTION.
     050134  012777  052332  132670           MOV     #ECRSV1,@DPC     ; START THE DPU.
     050142  004737  027454                   JSR     PC,WAITF         ; WAIT FOR DISPLAY STOP.
4825 050146  105737  026107                   TSTB    INTFLAG          ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4826 050152  001412                           BEQ     14$              ; BR IF YES.
4827 050154  012701  050142                   MOV     #.-12,R1         ; NO. SO GET PC FOR ERROR REPORT,
4828 050160                                   HRDERR  NTO,INTERR       ; ... AND REPORT THE PROBLEM.
4829 050200                              14$:
4830 050200  012701  124003                   MOV     #RSVDOP,R1
4831 050204                                   RDDSRA  CSR,R2
     050204  012777  000003  132622           MOV     #SELCSR,@DSR     ; READ CSR INTO R2 .
     050212  017702  132616                   MOV     @DSR,R2
4832 050216                                   IFERROR ECE,CSRER
     050216  020102                           CMP     R1,R2            ; OK?
     050220  001410                           BEQ     68$              ; YES.
     050222                                   HRDERR  ECE,CSRER,       ; NO.
     050242                              68$:
4833 050242                                   ENDSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4834
4835                                     ; TEST RSVDOP - ODD Y COORDINATE (BIT 0 SET)
4836
4837 050244                                   BGNSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4838 050246                                   INITDP
     050246  004737  025402                   JSR     PC,DPINIT        ; GO DO SOFT INIT, CHECK THE STATE.
4839 050252  012777  000113  132554           MOV     #ENECHK.WRTCSR,@DSR      ;ENABLE RSVD-OP ERROR CHECKING.
4840 050260                                   INTSET  ERR
     050260  012737  177604  026106          MOV     #^0177400.IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
```

```
4841 050266                                    DPSTART #ECRSV2          ;START DPU ON ODD Y VALUE.
     050266  012777  052336  132536            MOV     #ECRSV2,@DPC     ; START THE DPU.
     050274  004737  027454                    JSR     PC,WAITF         ; WAIT FOR DISPLAY STOP.
4842 050300  105737  026107                    TSTB    INTFLAG          ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4843 050304  001412                            BEQ     15$              ; BR IF YES.
4844 050306  012701  050274                    MOV     #.-12,R1         ; NO. SO GET PC FOR ERROR REPORT,
4845 050312                                    HRDERR  NTO,INTERR       ; ... AND REPORT THE PROBLEM.
4846 050332                                15$:
4847 050332  012701  124103                    MOV     #RSVDOP!ENECHK,R1
4848 050336                                    RDDSRA  CSR,R2
     050336  012777  000003  132470            MOV     #SELCSR,@DSR     ; READ CSR INTO R2 .
     050344  017702  132464                    MOV     @DSR,R2
4849 050350                                    IFERROR ECE,CSRER
     050350  020102                            CMP     R1,R2            ; OK?
     050352  001410                            BEQ     69$              ; YES.
     050354                                    HRDERR  ECE,CSRER,       ; NO.
     050374                                69$:
4850 050374                                    ENDSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4851
4852                                       ; TEST RSVDOP - ODD X VALUE (BIT 0 SET)
4853
4854 050376                                    BGNSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4855 050400                                    INITDP
     050400  004737  025402                    JSR     PC,DPINIT        ; GO DO SOFT INIT, CHECK THE STATE.
4856 050404  012777  000113  132422            MOV     #ENECHK!WRTCSR,@DSR      ;ENABLE RSVD-OP ERROR CHECKING.
4857 050412                                    INTSET  ERR
     050412  012737  177604  026106            MOV     #^O177400!IOKCKIN!IOKERR,INTMASK         ; PRIME FLAG, EXPECT ERR
4858 050420                                    DPSTART #ECRSV3          ;START DPU ON ODD X VALUE.
     050420  012777  052346  132404            MOV     #ECRSV3,@DPC     ; START THE DPU.
     050426  004737  027454                    JSR     PC,WAITF         ; WAIT FOR DISPLAY STOP.
4859 050432  105737  026107                    TSTB    INTFLAG          ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4860 050436  001412                            BEQ     16$              ; BR IF YES.
4861 050440  012701  050426                    MOV     #.-12,R1         ; NO. SO GET PC FOR ERROR REPORT,
4862 050444                                    HRDERR  NTO,INTERR       ; ... AND REPORT THE PROBLEM.
4863 050464                                16$:
4864 050464  012701  124103                    MOV     #RSVDOP!ENECHK,R1
4865 050470                                    RDDSRA  CSR,R2
     050470  012777  000003  132336            MOV     #SELCSR,@DSR     ; READ CSR INTO R2 .
     050476  017702  132332                    MOV     @DSR,R2
4866 050502                                    IFERROR ECE,CSRER
     050502  020102                            CMP     R1,R2            ; OK?
     050504  001410                            BEQ     70$              ; YES.
     050506                                    HRDERR  ECE,CSRER,       ; NO.
     050526                                70$:
4867 050526                                    ENDSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4868
4869                                       ; TEST SEQERR - SEQUENCE ERROR (DJMS FROM SUBROUTINE)
4870
4871 050530                                    BGNSEG           ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4872 050532                                    INITDP
     050532  004737  025402                    JSR     PC,DPINIT        ; GO DO SOFT INIT, CHECK THE STATE.
4873 050536                                    INTSET  ERR
     050536  012737  177604  026106            MOV     #^O177400!IOKCKIN!IOKERR,INTMASK         ; PRIME FLAG, EXPECT ERR
4874 050544                                    DPSTART #ECSEQ1
     050544  012777  052356  132260            MOV     #ECSEQ1,@DPC     ; START THE DPU.
     050552  004737  027454                    JSR     PC,WAITF         ; WAIT FOR DISPLAY STOP.
4875 050556  105737  026107                    TSTB    INTFLAG          ; DID WE GET THE EXPECTED ERROR INTERRUPT?
```

```
 4876 050562  001412                        BEQ     17$             ; BR IF YES.
 4877 050564  012701  050552                MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
 4878 050570                                HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
 4879 050610                          17$:
 4880 050610  012701  120003                MOV     #SEQERR,R1
 4881 050614                                RDDSRA  CSR,R2
      050614  012777  000003  132212        MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      050622  017702  132206                MOV     @DSR,R2
 4882 050626                                IFERROR ECE,CSRER
      050626  020102                        CMP     R1,R2           ; OK?
      050630  001410                        BEQ     71$             ; YES.
      050632                                HRDERR  ECE,CSRER,      ; NO.
      050652                          71$:
 4883 050652                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 4884
 4885                                ; TEST SEQERR - SEQUENCE ERROR (CHARACTER DATA IN CHAR. SUBROUTINE)
 4886
 4887 050654                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 4888 050656                                INITDP
      050656  004737  025402                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
 4889 050662                                INTSET  ERR
      050662  012737  177604  026106        MOV     #^O177400!IOKCKIN.IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
 4890 050670                                DPSTART #ECSEQ2
      050670  012777  052374  132134        MOV     #ECSEQ2,@DPC    ; START THE DPU.
      050676  004737  027454                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 4891 050702  105737  026107                TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
 4892 050706  001412                        BEQ     18$             ; BR IF YES.
 4893 050710  012701  050676                MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
 4894 050714                                HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
 4895 050734                          18$:
 4896 050734  012701  120003                MOV     #SEQERR,R1
 4897 050740                                RDDSRA  CSR,R2
      050740  012777  000003  132066        MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      050746  017702  132062                MOV     @DSR,R2
 4898 050752                                IFERROR ECE,CSRER
      050752  020102                        CMP     R1,R2           ; OK?
      050754  001410                        BEQ     72$             ; YES.
      050756                                HRDERR  ECE,CSRER,      ; NO.
      050776                          72$:
 4899 050776                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 4900
 4901
 4902                                ; TEST SEQERR - SEQUENCE ERROR (DJMS IN CHARACTER SUBROUTINE)
 4903
 4904 051000                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 4905 051002                                INITDP
      051002  004737  025402                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
 4906 051006                                INTSET  ERR
      051006  012737  177604  026106        MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
 4907 051014                                DPSTART #ECSEQ3
      051014  012777  052442  132010        MOV     #ECSEQ3,@DPC    ; START THE DPU.
      051022  004737  027454                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 4908 051026  105737  026107                TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
 4909 051032  001412                        BEQ     19$             ; BR IF YES.
 4910 051034  012701  051022                MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
 4911 051040                                HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
 4912 051060                          19$:
```

```
4913 051060  012701  120003                    MOV     #SEQERR,R1
4914 051064                                    RDDSRA  CSR,R2
     051064  012777  000003  131742            MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     051072  017702  131736                    MOV     @DSR,R2
4915 051076                                    IFERROR ECE,CSRER
     051076  020102                            CMP     R1,R2           ; OK?
     051100  001410                            BEQ     73$             ; YES.
     051102                                    HRDERR  ECE,CSRER,      ; NO.
     051122                             73$:
4916 051122                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4917
4918
4919                                   ; TEST SEQERR - SEQUENCE ERROR (BIT-MAP-0 IN SUBROUTINE)
4920
4921 051124                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4922 051126                                    INITDP
     051126  004737  025402                    JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4923 051132                                    INTSET  ERR
     051132  012737  177604  026106            MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4924 051140                                    DPSTART #ECSEQ4
     051140  012777  052474  131664            MOV     #ECSEQ4,@DPC    ; START THE DPU.
     051146  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4925 051152  105737  026107                    TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4926 051156  001412                            BEQ     20$             ; BR IF YES.
4927 051160  012701  051146                    MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4928 051164                                    HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4929 051204                             20$:
4930 051204  012701  120003                    MOV     #SEQERR,R1
4931 051210                                    RDDSRA  CSR,R2
     051210  012777  000003  131616            MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     051216  017702  131612                    MOV     @DSR,R2
4932 051222                                    IFERROR ECE,CSRER
     051222  020102                            CMP     R1,R2           ; OK?
     051224  001410                            BEQ     74$             ; YES.
     051226                                    HRDERR  ECE,CSRER,      ; NO.
     051246                             74$:
4933 051246                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4934
4935
4936                                   ; TEST SEQERR - SEQUENCE ERROR (BIT-MAP-1 IN SUBROUTINE)
4937
4938 051250                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4939 051252                                    INITDP
     051252  004737  025402                    JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4940 051256                                    INTSET  ERR
     051256  012737  177604  026106            MOV     #^O177400.IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4941 051264                                    DPSTART #ECSEQ5
     051264  012777  052520  131540            MOV     #ECSEQ5,@DPC    ; START THE DPU.
     051272  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4942 051276  105737  026107                    TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4943 051302  001412                            BEQ     21$             ; BR IF YES.
4944 051304  012701  051272                    MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4945 051310                                    HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4946 051330                             21$:
4947 051330  012701  120003                    MOV     #SEQERR,R1
4948 051334                                    RDDSRA  CSR,R2
     051334  012777  000003  131472            MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
```

```
                                                MOV     @DSR,R2
                                                IFERROR ECE,CSRER
4949 051346                                     CMP     R1,R2           ; OK?
     051346  020102                             BEQ     75$             ; YES.
     051350  001410                             HRDERR  ECE,CSRER,      ; NO.
     051352
     051372                             75$:
4950 051372                                     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4951
4952
4953                                    ; TEST SEQERR - SEQUENCE ERROR (DMA READBACK FROM SUBROUTINE)
4954
4955 051374                                     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4956 051376                                     INITDP
     051376  004737  025402                     JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4957 051402                                     INTSET  ERR
     051402  012737  177604  026106             MOV     #^O177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4958 051410                                     DPSTART #ECSEQ6
     051410  012777  052542  131414             MOV     #ECSEQ6,@DPC    ; START THE DPU.
     051416  004737  027454                     JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4959 051422  105737  026107                     TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4960 051426  001412                             BEQ     22$             ; BR IF YES.
4961 051430  012701  051416                     MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4962 051434                                     HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4963 051454                             22$:
4964 051454  012701  120003                     MOV     #SEQERR,R1
4965 051460                                     RDDSRA  CSR,R2
     051460  012777  000003  131346             MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     051466  017702  131342                     MOV     @DSR,R2
4966 051472                                     IFERROR ECE,CSRER
     051472  020102                             CMP     R1,R2           ; OK?
     051474  001410                             BEQ     76$             ; YES.
     051476                                     HRDERR  ECE,CSRER,      ; NO.
     051516                             76$:
4967 051516                                     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
4968
4969
4970                                    ; TEST SEQERR - SEQUENCE ERROR (ILLEGAL DMA READBACK RESTART)
4971
4972 051520                                     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
4973 051522                                     INITDP
     051522  004737  025402                     JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
4974 051526                                     INTSET  ERR
     051526  012737  177604  026106             MOV     #^O177400!IOKCKIN.IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
4975 051534                                     DPSTART #ECSEQ7
     051534  012777  052572  131270             MOV     #ECSEQ7,@DPC    ; START THE DPU.
     051542  004737  027454                     JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
4976 051546  105737  026107                     TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
4977 051552  001412                             BEQ     23$             ; BR IF YES.
4978 051554  012701  051542                     MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
4979 051560                                     HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
4980 051600                             23$:
4981 051600  012701  120003                     MOV     #SEQERR,R1
4982 051604                                     RDDSRA  CSR,R2
     051604  012777  000003  131222             MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     051612  017702  131216                     MOV     @DSR,R2
4983 051616                                     IFERROR ECE,CSRER
     051616  020102                             CMP     R1,R2           ; OK?
```

```
      051620  001410                        BEQ     77$             ; YES.
      051622                                HRDERR  ECE,CSRER,      ; NO.
      051642                        77$:
 4984 051642                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 4985
 4986                                ; TEST SYNCTO - SYNC TIMEOUT FROM IMAGE MEMORY.
 4987
 4988 051644                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 4989 051646                                INITDP
      051646  004737  025402                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
 4990 051652                                INTSET  ERR
      051652  012737  177604  026106        MOV     #^0177400!IOKCKIN!IOKERR,INTMASK        ; PRIME FLAG, EXPECT ERR
 4991 051660                                DPSTART #ECSTO
      051660  012777  052620  131144        MOV     #ECSTO,@DPC     ; START THE DPU.
      051666  004737  027454                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 4992 051672  105737  026107                TSTB    INTFLAG         ; DID WE GET THE EXPECTED ERROR INTERRUPT?
 4993 051676  001412                        BEQ     24$             ; BR IF YES.
 4994 051700  012701  051666                MOV     #.-12,R1        ; NO. SO GET PC FOR ERROR REPORT,
 4995 051704                                HRDERR  NTO,INTERR      ; ... AND REPORT THE PROBLEM.
 4996 051724                        24$:
 4997 051724  012701  140003                MOV     #SYNCTO,R1
 4998 051730                                RDDSRA  CSR,R2
      051730  012777  000003  131076        MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      051736  017702  131072                MOV     @DSR,R2
 4999 051742                                IFERROR ECE,CSRER
      051742  020102                        CMP     R1,R2           ; OK?
      051744  001410                        BEQ     78$             ; YES.
      051746                                HRDERR  ECE,CSRER,      ; NO.
      051766                        78$:
 5000 051766                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 5001
 5002                                ; TEST DRDYTO - DATA READY TIMEOUT ON CURSOR READBACK
 5003
 5004 051770                                BGNSEG
 5005 051772                                INITDP
      051772  004737  025402                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
 5006 051776  105037  026106                CLRB    INTMASK
 5007 052002  012777  040000  131026        MOV     #GTJSSW,@DXR    ; SAY READ PENDING COORDS (BUT THERE ARE NONE)
 5008 052010  004737  027454                JSR     PC,WAITF        ; WAIT FOR READY
 5009 052014  012701  150003                MOV     #DRDYTO,R1      ; SET EXPECTED ERROR CODE 'DRDYTO'' INTO R1
 5010 052020                                RDDSRA  CSR,R2          ; GET THE CSR
      052020  012777  000003  131006        MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
      052026  017702  131002                MOV     @DSR,R2
 5011 052032                                IFERROR ECE,CSRER
      052032  020102                        CMP     R1,R2           ; OK?
      052034  001410                        BEQ     79$             ; YES.
      052036                                HRDERR  ECE,CSRER,      ; NO.
      052056                        79$:
 5012 052056                                ENDSEG
 5013
 5014                                ; TEST DAVTO - DATA AVAILABLE TIMEOUT PIXEL READBACK.
 5015
 5016 052060                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 5017 052062                                INITDP
      052062  004737  025402                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
 5018 052066  012777  040000  130744        MOV     #40000,@DYR     ; PROTECT ALL MEMORYS.
 5019 052074  004737  027454                JSR     PC,WAITF        ;
```

```
5020 052100  012777  040400  130732          MOV     #40400,@DYR     ;
5021 052106  004737  027454                  JSR     PC,WAITF        ;
5022 052112  012777  041000  130720          MOV     #41000,@DYR     ;
5023 052120  004737  027454                  JSR     PC,WAITF        ;
5024 052124  012777  041400  130706          MOV     #41400,@DYR     ;
5025 052132  004737  027454                  JSR     PC,WAITF        ;
5026 052136  012777  000000  130672          MOV     #0,@DXR         ; TRY PIXEL READ.
5027 052144  004737  027454                  JSR     PC,WAITF        ;
5028 052150  103410                          BCS     2$              ; BRANCH IF UNSUCCESSFUL.
5029 052152                                  DFERR   PR3H            ; PIXEL READBACK HUNG UP.
5030 052172  012701  144003          2$:     MOV     #DAVTO,R1
5031 052176                                  RDDSRA  CSR,R2
     052176  012777  000003  130630          MOV     #SELCSR,@DSR    ; READ CSR INTO R2 .
     052204  017702  130624                  MOV     @DSR,R2
5032 052210                                  IFERROR ECE,CSRER
     052210  020102                          CMP     R1,R2           ; OK?
     052212  001410                          BEQ     80$             ; YES.
     052214                                  HRDERR  ECE,CSRER,      ; NO.
     052234                          80$:
5033 052234                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5034
5035 052236                                  COMEND  1$
     052236  004737  027646                  JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
     052242  103401                          BCS     81$
     052244  000402                          BR      82$

     052246  000137  047342          81$:    JMP     1$

     052252  004737  030072          82$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
5036 052256                                  EXIT    TST     ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
5037
5038                                         ;
5039                                         ; INTERUPT SERVICE FOR THIS TEST.
5040                                         ;
5041 052262  000002                  ECISVC: RTI             ; RETURN FROM INTERUPT.
5042
5043                                         ;
5044                                         ; DISPLAY FILES.
5045                                         ;
5046 052264  164000                  ECMPE1: DNOP                    ; MPE - MEMORY PROTECTION ERROR.
5047 052266  173000                          STOPN                   ; ... ON MAIN SEGMENT
5048
5049 052270  160001  052277          ECMPE2: DJMS,1$.AUXSEG          ; CALL AUXILIARY SEGMENT
5050 052274  173000                          STOPN
5051 052276  164000                  1$:     DNOP                    ;SHOULD GET MPE ERROR
5052 052300  173000                          STOPN
5053
5054 052302  162000                  ECMPE3: IMREAD                  ;TRY DMA READBACK
5055 052304  000001  000001  000001          1,1,1,160000            ;...TO WRITE PROTECTED MAIN SEG.
5056 052314  173000                          STOPN
5057
5058 052316  162000                  ECMPE4: IMREAD                  ;TRY DMA READBACK ...
5059 052320  000001  000001  000001          1,1,1,160000.AUXSEG     ;...TO WRITE-PROTECTED AUX SEG.
5060 052330  173000                          STOPN
5061
5062 052332  161777                  ECRSV1: 161777                  ; RSVDOP - RESERVED OPCODE, OPERATION.
5063 052334  173000                          STOPN
```

```
5064
5065 052336  114000  000000  000001  ECRSV2: APNT,0,1                ; RSVDOP - ODD Y VALUE.
5066 052344  173000                          STOPN
5067
5068 052346  114000  000001  000000  ECRSV3: APNT,1,0                ; RSVDOP - ODD X VALUE.
5069 052354  173000                          STOPN
5070
5071 052356  160001  052364          ECSEQ1: DJMS,1$                 ; SEQERR - SEQUENCE ERROR.
5072 052362  173000                          STOPN
5073 052364  160001  052370          1$:     DJMS,2$                 ;DJMS FROM SUBROUTINE.
5074 052370  173000                  2$:     STOPN
5075 052372  173000                          STOPN
5076
5077 052374  114000  000000  000000  ECSEQ2: APNT,0,0                ; SEQERR.
5078 052402  152000  052414                  SETCB,2$
5079 052406  103774                          CHAR!ALL
5080 052410  000000                  1$:     0
5081 052412  173000                          STOPN
5082 052414  000000                  2$:     0                       ;DATA AS 1ST WORD -- STILL IN CHAR MODE.
5083 052416  000000  000000  000000          0,0,0,0,0,0,0,0
5084 052436  165000                          DPOP
5085 052440  173000                          STOPN
5086
5087 052442  114000  000000  000000  ECSEQ3: APNT,0,0                ; SEQERR.
5088 052450  152000  052462                  SETCB,2$
5089 052454  103774                          CHAR!ALL
5090 052456  000000                  1$:     0
5091 052460  173000                          STOPN
5092 052462  160001  052470          2$:     DJMS,3$                 ;DJMS IN CHAR SUBROUTINE.
5093 052466  173000                          STOPN
5094 052470  173000                  3$:     STOPN
5095 052472  173000                          STOPN
5096
5097 052474  170140          ECSEQ4: CLRMEM                          ; SEQERR.
5098 052476  114000  000000  000000          APNT,0,0
5099 052504  160001  052512                  DJMS,1$
5100 052510  173000                          STOPN
5101 052512  134000                  1$:     BM04
5102 052514  052514                  2$:     2$
5103 052516  173000                          STOPN
5104
5105 052520  114000  000000  000000  ECSEQ5: APNT,0,0                ; SEQERR.
5106 052526  160001  052534                  DJMS,1$
5107 052532  173000                          STOPN
5108 052534  136000                  1$:     BM14
5109 052536  052536                  2$:     2$
5110 052540  173000                          STOPN
5111
5112 052542  114000  000000  000000  ECSEQ6: APNT,0,0                ; SEQERR.
5113 052550  160001  052556                  DJMS,1$
5114 052554  173000                          STOPN
5115 052556  162000                  1$:     IMREAD                  ;DMA READBACK FROM SUBROUTINE.
5116 052560  000001                          1
5117 052562  000001                          1
5118 052564  000001                          1
5119 052566  052614                          ECSQ7B
5120 052570  173000                          STOPN
```

```
5121
5122 052572  114000  000000  000000  ECSEQ7: APNT,0,0
5123 052600  162001                          IMREAD!1                    ;ILLEGAL DMA READBACK RESTART.
5124 052602  000001                          1
5125 052604  052614                          ECSQ7B
5126 052606  173000                          STOPN
5127 052610  162101                          IMREAD!101
5128 052612  173000                          STOPN
5129 052614                          ECSQ7B: .BLKW   2
5130
5131 052620  176000                  ECSTO:  PROTEC!CH0
5132 052622  176400                          PROTEC.CH1
5133 052624  177000                          PROTEC!CH2
5134 052626  177400                          PROTEC!CH3
5135 052630  114000  040000  000000          APNT,I.0,0
5136 052636  173000                          STOPN
5137
5138 052640                                  END.TEST

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

```
5140                                    ;*****************************************************************
                                        ;*
                                        ;*      BEGIN TEST 16   -   ADDRESS RELOCATE
                                        ;*
                                        ;*****************************************************************

                                        ; (ITERATION COUNT = 20.)

5141                                    ;
5142                                    ; TEST ALL COMBINATIONS OF DPC AND RELOCATE FACTOR
5143                                    ; WHERE DPC + REL = RELT1, A LOCATION WITHIN THE PROGRAM
5144                                    ; TEST RELIES OF THE FACT THAT (DPC + REL) > 18 BITS WIL'
5145                                    ; WRAP AROUND IN THE FINAL DPC.
5146                                    ; REPEAT 20 TIMES AFTER QUICK PASS.
5147                                    ;
5148 052714                            RELTST: COMBEG
      052714  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      052720  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
5149 052724  012737  053140  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5150 052732  012777  000000  130074            MOV     #SELDSR,@DSR
5151 052740  017703  130070                    MOV     @DSR,R3
5152 052744  042703  176000                    BIC     #^C1777,R3
5153 052750  052703  172000                    BIS     #STOP,R3        ;...AND STATUS.
5154
5155 052754  012704  053166            5$:     MOV     #RELT1,R4       ;SET INITIAL DPC
5156 052760  012702  053170                    MOV     #RELT1+2,R2     ;SET INITIAL EXPECTED DPC
5157 052764  012705  000014                    MOV     #SETMRR.0,R5    ;SET INITIAL RELOCATION (0)
5158
5159 052770                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5160
5161 052772  010577  130036                    MOV     R5,@DRR         ;SET 1ST (NEXT) REL VALUE...
5162 052776  010477  130030                    MOV     R4,@DPC         ;...XCT -- RELOCATE TO 'RELT1'
5163 053002  004737  027454                    JSR     PC,WAITF
5164 053006  027702  130020                    CMP     @DPC,R2         ;CHECK DPC = R4 (INITIAL) + 2
5165 053012  001003                            BNE     2$
5166 053014  027703  130014                    CMP     @DSR,R3         ;...AND DSR = STOP+PIXEL DATA.
5167 053020  001410                            BEQ     3$
5168 053022                            2$:     HRDERR  RELF,PCSERR     ;RELOCATE FAILURE.
5169 053042  004737  026756            3$:     JSR     PC,RELEAS       ;IN CASE HE 'HUNG' SOMEWHERE.
5170 053046                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5171
5172 053050  020527  177774                    CMP     R5,#SETMRR.177760       ;MAX RELOCATION DONE?
5173 053054  001414                            BEQ     4$              ;ALL DONE IF SO.
5174
5175 053056  062705  000020                    ADD     #BIT4,R5        ;BUMP RELOCATION BY 1 (32 WORDS)
5176 053062  162702  000100                    SUB     #100,R2         ;REDUCE EXPECTED...
5177 053066  162704  000100                    SUB     #100,R4         ;... AND INITIAL DPC BY 32 WORDS)
5178 053072  100336                            BPL     1$              ;CONTINUE 'TIL WE UNDERFLOW
5179 053074  005705                            TST     R5              ;SEE IF ALREADY FORCED HIGH BITS
5180 053076  100734                            BMI     1$              ;JUST CONTINUE IF YES
5181 053100  052705  140000                    BIS     #140000,R5      ;FORCE HIGH RELOCATION BITS
5182 053104  000731                            BR      1$              ;...THEN CONTINUE
5183
5184 053106                            4$:     LOOPTO  5$              ; REPEAT 'TIL LOOPER EXPIRES.
      053106  004737  027646                    JSR     PC,LOOP
      053112  103720                            BCS     5$
```

```
5185 053114  012777  000014  127712          MOV     #SETMRR.0,@DSR  ;INSURE RR IS 0
5186 053122  012777  000000  127704          MOV     #SELDSR,@DSR    ;...AND DSR IS SELECTED
5187 053130  004737  030072                  JSR     PC,TSTEND
5188 053134                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
5189                                 ;
5190                                 ; ON ERROR, SHOW RELOCATE PARAMETERS.
5191                                 ;
5192 053140                         20$:    PRINTX  #RELFX,R4,R5    ; EXTENSION SHOWS RELOC FACTORS.
5193 053164  000207                          RTS     PC
5194
5195                                 ;DISPLAY FILE FOR RELOCATE TEST
5196 053166  173000                 RELT1:  STOPN
5197 053170  173000                         STOPN
5198 053172                                  END.TEST

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

```
5200                                    ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 17   -   CHARACTERS
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 20.)

5201                                    ;
5202                                    ; USE SETCB, CHAR, AND DPOP AS A SUBROUTINE CALL AND RETURN.
5203                                    ; USE ALL ASCII CODES (HI AND LO BYTES) ALTERING THE BASE
5204                                    ; ADDRESS SUCH THAT EACH CHAR/BASE COMBINATION YIELDS THE
5205                                    ; SAME SUBROUTINE ADDRESS.
5206                                    ; VERIFY THE DPC AT EACH SUBROUTINE ENTRY AND AGAIN AFTER
5207                                    ; FINAL DPOP.
5208                                    ; DO THE TEST FOR BOTH MAIN AND AUXILIARY SEGMENT SPECIFIED IN C-BASE.
5209                                    ; REPEAT 20 TIMES AFTER A QUICK PASS.
5210                                    ;
5211                                    ; NOTE THAT AN ERROR WHERE CHAR CODE = -1 IMPLIES THAT
5212                                    ; CHAR MODE DIDN'T TERMINATE WITH THE FINAL DPOP.
5213                                    ; IN EVERY OTHER CASE, EXP DPC - SETCB + 2(CC) + 4.
5214                                    ;
5215                                    ; ALSO, ANY ERROR IMPLIES THAT WE'VE LOST CONTROL AND
5216                                    ; CAN'T CONTINUE.
5217                                    ;
5218                                    ; CHAR ARRAY SIZE = 67. WORDS (OPCODE + 128 CHAR + DJMP,RETURN).
5219                                    ;
5220 053240                            CHRTST: COMBEG
      053240  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      053244  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
5221
5222 053250  012737  053754  053732            MOV     #CT.SUB,CHSEG   ;GET INITIAL BASE ADDRESS, MAIN SEGMENT.
5223 053256  012737  053702  022200            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5224 053264  013701  003240                    MOV     FREE,R1         ; BUFFER ADDRESS => R1.
5225 053270  010137  053750                    MOV     R1,CT.ASC       ; SET 'DJMP FREE'' CODE.
5226 053274  005002                            CLR     R2              ;1ST CHAR CODE => R2
5227 053276  012721  100000                    MOV     #CHAR,(R1)+     ;SET CHAR OPCODE...
5228 053302  110221                    6$:     MOVB    R2,(R1)+        ;...AND 128 CHARACTERS...
5229 053304  005202                             INC    R2
5230 053306  020227  000200                      CMP    R2,#128.
5231 053312  001373                               BNE   6$
5232 053314  012721  160000                    MOV     #DJMP,(R1)+     ;...AND THE RETURN LINK.
5233 053320  012721  053752                    MOV     #CT.RET,(R1)+
5234
5235 053324  017703  127504                    MOV     @DSR,R3
5236 053330  042703  176000                    BIC     #^C1777,R3
5237 053334  052703  172000                    BIS     #STOP,R3        ;SET EXP DSR FOR ERROR.
5238
5239 053340                            5$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5240 053342  013737  053732  053744            MOV     CHSEG,CT.ASC-4  ;INIT INITIAL BASE ADDRESS.
5241 053350  013737  053732  053762            MOV     CHSEG,CT.BAS    ;INIT BASE ADDRESS IN CHAR SUBTRN.
5242 053356  005001                            CLR     R1              ;R1 = CURRENT CHAR.
5243
5244 053360  013702  053732                    MOV     CHSEG,R2        ;GET SEGMENT SELECTOR
5245 053364  042702  177776                    BIC     #^CAUXSEG,R2    ;...FOR DPC READBACK
5246 053370  052702  053760                    BIS     #CT.SUB+4,R2    ;...AND SET EXPECTED DPC.
```

```
5247 053374  012777  053734  127430          MOV     #CT,@DPC        ;START, XCT 1ST CHAR.
5248 053402  000403                          SKP3                    ;SKIP OVER RESUME
5249 053404  012777  000001  127420  1$:     MOV     #1,@DPC         ;RESUME, XCT NEXT CHAR.
5250 053412  004737  027454                  JSR     PC,WAITF
5251 053416  027702  127410                  CMP     @DPC,R2         ;CHECK DPC = R2
5252 053422  001410                          BEQ     2$              ;ERROR IF IT'S WRONG.
5253 053424                                  HRDERR  CHRF,PCSERR     ;CHAR MODE FAILS.
5254 053444  013746  003240          2$:     MOV     FREE,-(SP)      ;GET START OF MAIN DISPLAY FILE,
5255 053450  005701                          TST     R1                      ;FINAL DPOP?
5256 053452  100003                          BPL     15$             ;SKIP AHEAD IF NOT.     @@@I
5257 053454  062716  000203                  ADD     #203,(SP)       ;SAY PCSAVE IS EMPTY, WITH FINAL ADDRESS.
5258 053460  000405                          BR      16$             ;...THEN GO CHECK IT.   @@@I
5259 053462  062716  000004          15$:    ADD     #4,(SP)         ;  ADD OFFSET FOR FIRST TRIP,
5260 053466  060116                          ADD     R1,(SP)         ;  AND THE CHARACTER,
5261 053470  042716  000001                  BIC     #1,(SP)         ;MAKE EVEN, 2 CHARS/WORD
5262 053474  012637  023056          16$:    MOV     (SP)+,GDDAT     ;EXPECTED DATA FOR PCSAVE
5263 053500  004737  022360                  JSR     PC,PCSCHK       ;...CHECK PCSAVE.
5264 053504  012737  100000  023056          MOV     #CHAR,GDDAT     ;NOW CHECK FOR "CHAR" OPCODE IN FLAGS
5265 053512  004737  022410                  JSR     PC,FLGCHK       ;
5266 053516  012737  000003  023056          MOV     #SELCSR,GDDAT   ;... AND CSR
5267 053524  004737  022440                  JSR     PC,CSRCHK       ;
5268 053530  013737  053762  023056          MOV     CT.BAS,GDDAT    ;CHECK THE CHARACTER BASE
5269 053536  004737  022624                  JSR     PC,CBCHK        ;
5270 053542                                  CKLOOP                  ;LOOP ON ERROR.
5271
5272 053544  005701                          TST     R1              ;HAVE WE TERMINATED ??
5273 053546  100414                          BMI     3$              ;EXIT IF SO.
5274 053550  162737  000002  053762          SUB     #2,CT.BAS       ;NO, DECR BASE ADDRESS.
5275 053556  005201                          INC     R1              ;INCR CHAR COUNT.
5276 053560  020127  000200                  CMP     R1,#128.        ;128 CHARS DONE ??
5277 053564  001307                          BNE     1$              ;LOOP IF NOT.
5278 053566  012701  177777                  MOV     #-1,R1          ;YES, SET TERMINATE FLAG...
5279 053572  012702  053754                  MOV     #CT.RET+2,R2    ;...ADJUST EXPECTED DPC...
5280 053576  000702                          BR      1$              ;...AND LOOP TO XCT FINAL DPOP.
5281
5282 053600                          3$:     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5283 053602  032737  000001  053732          BIT     #AUXSEG,CHSEG           ;DID WE DO AUX SEGMENT?
5284 053610  001004                          BNE     31$             ;BR IF YES.     @@@I
5285 053612  052737  000001  053732          BIS     #AUXSEG,CHSEG   ;NO. SO DO IT NOW
5286 053620  000647                          BR      5$              ;LOOP BACK AFTER SETTING SEG BIT.       @@@I
5287 053622  042737  000001  053732  31$:    BIC     #AUXSEG,CHSEG   ;DID AUX, SO CLEAR THE BIT
5288 053630                                  LOOPTO  5$              ; REPEAT TIL LOOPER EXPIRES.
     053630  004737  027646                  JSR     PC,LOOP
     053634  103641                          BCS     5$
5289 053636  005737  002502                  TST     DPUMOD          ; ARE WE IN DPU-MODE ONLY?
5290 053642  001013                          BNE     40$             ; BR=YES.
5291 053644  004737  026074                  JSR     PC,DPRESET      ; DO SOFT INIT. & CHECK STATE
5292 053650                                  DPSTART #CHDISP
     053650  012777  053766  127154          MOV     #CHDISP,@DPC    ; START THE DPU.
     053656  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
5293 053662  004737  027514                  JSR     PC,PAUSE1
5294 053666  004737  027514                  JSR     PC,PAUSE1
5295 053672                          40$:
5296 053672  004737  030072                  JSR     PC,TSTEND
5297 053676                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
5298                                  ;
5299                                  ; ON ERROR, SHOW BASE ADDRESS AND CURRENT CHAR CODE.
```

```
5300                                    ;
5301 053702                             20$:    PRINTX  #CHRFX,CT.BAS,R1 ; ERROR EXTENSION.
5302 053730  000207                             RTS     PC
5303                                    ;
5304 053732  053754                     CHSEG:  CT.SUB  ;CHARACTER BASE CONTROL, WITH SEGMENT SELECT.
5305                                    ;
5306                                    ; DISPLAY CODE FOR CHAR MODE TEST.
5307                                    ;
5308 053734  164000  164000  164000     CT:     DNOP,DNOP,DNOP
5309 053742  152000  053754                     SETCB,CT.SUB            ;INITIAL BASE ADDR = CT.SUB.
5310 053746  160000                             DJMP                    ;XFER PC TO THE CHAR...
5311 053750  000000                     CT.ASC: 0                       ;...DATA ARRAY.
5312 053752  173000                     CT.RET: STOPN                   ;FINAL ''DPOP'' SHOULD RETURN HERE
5313
5314 053754  053756                     CT.SUB: .+2                     ;SUBROUTINE ENTRY PONNTER.
5315 053756  173000                             STOPN                   ;STOP HERE IF CHAR ADDRESSES...
5316                                                                     ;...COME OUT RIGHT.
5317 053760  152000                             SETCB                   ;ON RESUME, SET NEW BASE ADDR...
5318 053762  053754                     CT.BAS: CT.SUB
5319 053764  165000                             DPOP                    ;...AND DPOP TO NEXT CHAR.
5320
5321                                    ; DISPLAY FILE FOR CHARACTER DISPLAY:
5322
5323 053766  176074  176474  177074     CHDISP: RDWRT!CH0,RDWRT.CH1,RDWRT!CH2,RDWRT!CH3
5324 053776  170140                             CLRMEM
5325 054000  152000  110146                     SETCB, ACAT
5326 054004  117774  001000  001500             APNT!ALL,1000,1500
5327 054012  100000                             CHAR
5328 054014     040     041     042             .ASCII  ? !''#$%&'()*+,-./?
5329                                            .EVEN
5330 054034  114000  001000  001400             APNT,1000,1400
5331 054042  100000                             CHAR
5332 054044     060     061     062             .ASCII  /0123456789/
5333 054056  114000  001000  001300             APNT,1000,1300
5334 054064  100000                             CHAR
5335 054066     072     073     074             .ASCII  /:;<=>?@ /
5336                                            .EVEN
5337 054076  114000  001000  001200             APNT,1000,1200
5338 054104  100000                             CHAR
5339 054106     101     102     103             .ASCII  /ABCDEFGHIJKLM /
5340                                            .EVEN
5341 054124  114000  001000  001100             APNT,1000,1100
5342 054132  100000                             CHAR
5343 054134     116     117     120             .ASCII  /NOPQRSTUVWXYZ /
5344                                            .EVEN
5345 054152  114000  001000  001000             APNT,1000,1000
5346 054160  100000                             CHAR
5347 054162     133     134     135             .ASCII  /[\]^_ /
5348                                            .EVEN
5349
5350 054170  117774  001000  000602             APNT!ALL,1000,602
5351 054176  100000                             CHAR
5352 054200     040     041     042             .ASCII  ? !''#$%&'()*+,-./?
5353                                            .EVEN
5354 054220  114000  001000  000502             APNT,1000,502
5355 054226  100000                             CHAR
5356 054230     060     061     062             .ASCII  /0123456789/
```

```
 5357 054242  114000  001000  000402          APNT,1000,402
 5358 054250  100000                          CHAR
 5359 054252     072     073     074          .ASCII  /:;<=>?@ /
 5360                                         .EVEN
 5361 054262  114000  001000  000302          APNT,1000,302
 5362 054270  100000                          CHAR
 5363 054272     101     102     103          .ASCII  /ABCDEFGHIJKLM /
 5364                                         .EVEN
 5365 054310  114000  001000  000202          APNT,1000,202
 5366 054316  100000                          CHAR
 5367 054320     116     117     120          .ASCII  /NOPQRSTUVWXYZ /
 5368                                         .EVEN
 5369 054336  114000  001000  000102          APNT,1000,102
 5370 054344  100000                          CHAR
 5371 054346     133     134     135          .ASCII  /[\]^_ /
 5372                                         .EVEN
 5373 054354  173000                          STOPN
 5374
 5375 054356                                  END.TEST

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

```
5377
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 18   -   ABSOLUTE POINTS
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 10.)

5378                                    ;
5379                                    ; VERIFY THAT WE CAN EXECUTE ABSOLUTE POINTS FOR THE FULL
5380                                    ; RANGE OF X/Y.
5381                                    ;
5382                                    ; PLOT ALL X POINTS, HOLDING Y AT 0.
5383                                    ; PLOT ALL Y POINTS, HOLDING X AT 0.
5384                                    ; PLOT ALL POINTS ON THE 45 (X = Y).
5385                                    ;
5386                                    ; REPEAT EACH POINT 10 TIMES AFTER QUICK PASS.
5387                                    ;
5388 054430  004737  027702            APTST:  JSR     PC,TSTGO        ;TITLE
5389 054434  012737  054740  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5390 054442  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT (IF THERE IS ONE).
5391
5392 054446  005737  002502                    TST     DPUMOD
5393 054452  001002                            BNE     .+6             ; SKIP NEXT IF DPU-ONLY.
5394 054454  004737  026650                    JSR     PC,CLRRW        ; CLEAR AND SET R/W MODE.
5395
5396 054460  012737  177777  003232            MOV     #-1,TEMP1       ;USE AS XFER SWITCH.
5397 054466  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF THE LOOPER.
5398 054474  005004                    8$:     CLR     R4              ;EXPECTED X ADDRESS.
5399 054476  005005                            CLR     R5              ;EXPECTED Y ADDRESS.
5400 054500  005037  055010                    CLR     ABPX            ;INIT X...
5401 054504  053737  003064  055010            BIS     OPTI,ABPX       ;...INTENSIFY IS OPTIONAL...
5402 054512  005037  055012                    CLR     ABPY            ;...AND Y AT 0.
5403
5404 054516                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5405
5406 054520  012777  055002  126304    9$:     MOV     #ABP,@DPC       ;START DISPLAY...
5407 054526  004737  027454                    JSR     PC,WAITF        ;WAIT FOR DONE FLAG.
5408 054532  027704  126300                    CMP     @DXR,R4         ;CHECK DXR = R4...
5409 054536  001003                            BNE     2$
5410 054540  027705  126274                    CMP     @DYR,R5         ;...AND DYR = R5.
5411 054544  001410                            BEQ     3$
5412 054546                            2$:     HRDERR  APF,XYERR       ;ABS POINT FAILS.
5413 054566                            3$:     LOOPTO  9$              ; REPEAT...TIL LOOPER EXPIRES.
     054566  004737  027646                    JSR     PC,LOOP
     054572  103752                            BCS     9$
5414 054574                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5415
5416 054576  013737  003234  027700            MOV     TEMP2,LOOPK     ;RESET LOOPER.
5417 054604  005737  003232                    TST     TEMP1           ;TEST XFER SWITCH.
5418 054610  003027                            BGT     5$              ;INCR BOTH X AND Y.
5419 054612  001413                            BEQ     4$              ;INCR Y ONLY.
5420
5421 054614  062704  000002                    ADD     #2,R4           ;INCR X ONLY.
5422 054620  062737  000002  055010            ADD     #2,ABPX
5423 054626  020427  001776                    CMP     R4,#MAXX
```

```
5424 054632  101731                        BLOS    1$              ;...AND LOOP 'TIL DONE.
5425 054634  005237  003232                INC     TEMP1           ;THEN, BUMP SWITCH...
5426 054640  000715                        BR      8$              ;...AND REINIT.
5427
5428 054642  062705  000002          4$:   ADD     #2,R5           ;INCR Y ONLY.
5429 054646  062737  000002  055012        ADD     #2,ABPY
5430 054654  020527  001776                CMP     R5,#MAXY
5431 054660  101716                        BLOS    1$              ;...AND LOOP 'TIL DONE.
5432 054662  005237  003232                INC     TEMP1           ;THEN, BUMP SWITCH...
5433 054666  000702                        BR      8$              ;...AND REINIT.
5434
5435 054670  012700  000002          5$:   MOV     #2,R0           ;INCR X AND Y.
5436 054674  060004                        ADD     R0,R4           ;INCR X'S...
5437 054676  060037  055010                ADD     R0,ABPX
5438 054702  060005                        ADD     R0,R5           ;...AND Y'S...
5439 054704  060037  055012                ADD     R0,ABPY
5440 054710  020527  001776                CMP     R5,#MAXY
5441 054714  101700                        BLOS    1$              ;...AND LOOP 'TIL DONE.
5442
5443 054716  005737  002502          10$:  TST     DPUMOD
5444 054722  001002                        BNE     11$
5445 054724  004737  027514                JSR     PC,PAUSE1       ; PAUSE IF VIDEO ON.
5446 054730  004737  030072          11$:  JSR     PC,TSTEND
5447 054734                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
5448                                    ;
5449                                    ; ON ERROR, SHOW ORIGIN AND DELTAS.
5450                                    ;
5451 054740                        20$:  PRINTX  #LONGX,ABP+2,ABP+4,ABPX,ABPY ; EXTENSION.
5452 055000  000207                        RTS     PC
5453                                    ;
5454                                    ; DISPLAY FILE FOR APNT TEST.
5455                                    ;
5456 055002  117774                ABP:    APNT.ALL
5457 055004  000000  000000                0,0
5458 055010  000000                ABPX:   0
5459 055012  000000                ABPY:   0
5460 055014  173000                        STOPN
5461
5462 055016                                END.TEST

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

```
5464
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 19   -   LONG VECTORS
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 10.)

5465                                    ;
5466                                    ; GENERATE LONG VECTORS FROM 0,0 VARYING THE LENGTH FROM
5467                                    ; MAX(X) TO HALF(XY) TO MAX(Y).
5468                                    ; THEN REPEAT FROM MAX,MAX USING NEGATIVE DELTAS.
5469                                    ;
5470                                    ;       [THIS MAY BE WORST CASE !!!]
5471                                    ;
5472                                    ; REPEAT EACH VECTOR 10 TIMES AFTER QUICK PASS.
5473                                    ;
5474 055066  004737  027702            LVTST:  JSR     PC,TSTGO        ;TITLE
5475 055072  012737  055406  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5476 055100  004737  027054                    JSR     PC,LUMIN        ; BLAST THE LUT.
5477
5478 055104  005737  002502                    TST     DPUMOD
5479 055110  001002                            BNE     .+6             ; SKIP NEXT IF DPU-ONLY.
5480 055112  004737  026650                    JSR     PC,CLRRW        ; CLEAR AND SET R/W MODE.
5481
5482 055116  005037  055452                    CLR     LV+2            ;INIT ORIGIN AT 0,0.
5483 055122  005037  055454                    CLR     LV+4
5484 055126  012704  001776                    MOV     #MAXX,R4        ;INIT EXP X AT MAX...
5485 055132  005005                            CLR     R5              ;...AND Y AT 0.
5486 055134  010437  055460                    MOV     R4,LVX          ;INIT X LENGTH...
5487 055140  053737  003064  055460            BIS     OPTI,LVX        ;...OPTIONAL INT...
5488 055146  010537  055462                    MOV     R5,LVY          ;...AND Y LENGTH.
5489 055152  012702  177776                    MOV     #-2,R2          ;INIT EXP X...
5490 055156  012703  000002                    MOV     #2,R3           ;...AND Y INCR'S.
5491 055162  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
5492
5493 055170                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5494
5495 055172  012777  055450  125632    9$:     MOV     #LV,@DPC        ;START 'EM UP.
5496 055200  004737  027454                    JSR     PC,WAITF
5497 055204  027704  125626                    CMP     @DXR,R4         ;CHECK DXR = R4...
5498 055210  001003                            BNE     2$
5499 055212  027705  125622                    CMP     @DYR,R5         ;...AND DYR = R5.
5500 055216  001410                            BEQ     3$
5501 055220                            2$:     HRDERR  LVF,XYERR       ;LONG VECTOR FAILS.
5502 055240                            3$:     LOOPTO  9$              ; REPEAT...
     055240  004737  027646                    JSR     PC,LOOP
     055244  103752                            BCS     9$
5503 055246                                    FNDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5504
5505 055250  013737  003234  027700            MOV     TEMP2,LOOPK     ;RESET LOOPER.
5506 055256  033727  055460  001777            BIT     LVX,#1777
5507 055264  001411                            BEQ     4$              ;BR IF DELTA X - 0.
5508 055266  162737  000002  055460            SUB     #2,LVX          ;OTHERWISE, DECR DX...
5509 055274  062737  000002  055462            ADD     #2,LVY          ;..AND INCR DY.
5510 055302  060204                            ADD     R2,R4           ;SET NEXT EXP X...
```

```
5511 055304  060305                          ADD     R3,R5           ;...AND EXP Y.
5512 055306  000730                          BR      1$              ;...AND CONTINUE.
5513
5514 055310  033727  055460  020000  4$:     BIT     LVX,#MXY        ;ARE WE IN THE NEG CYCLE ??
5515 055316  001022                          BNE     10$             ;WE'RE DONE IF SO.
5516 055320  012737  001776  055452          MOV     #MAXX,LV+2      ;OTHERWISE, REINIT ORIGIN...
5517 055326  012737  001776  055454          MOV     #MAXY,LV+4      ;...AT MAX.MAX.
5518 055334  052737  021776  055460          BIS     #MXY!MAXX,LVX   ;INIT X LENGTH AT -MAX.
5519 055342  012737  020000  055462          MOV     #MXY!0,LVY      ;INIT Y LENGTH AT -0.
5520 055350  005004                          CLR     R4              ;INIT EXP X = 0...
5521 055352  012705  001776                  MOV     #MAXY,R5        ;...AND Y = MAX...
5522 055356  005402                          NEG     R2              ;...REVERSE EXP INCR'S.
5523 055360  005403                          NEG     R3
5524 055362  000702                          BR      1$              ;AND GO 'ROUND'.
5525
5526 055364  005737  002502          10$:    TST     DPUMOD
5527 055370  001002                          BNE     11$
5528 055372  004737  027514                  JSR     PC,PAUSE1       ; PAUSE IF VIDEO ON.
5529 055376  004737  030072          11$:    JSR     PC,TSTEND
5530 055402                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
5531
5532                                  ;
5533                                  ; ON ERROR, SHOW ORIGIN AND DELTAS.
5534 055406                          ;
5535 055446  000207                  20$:    PRINTX  #LONGX,LV+2,LV+4,LVX,LVY ; EXTENSION.
5536                                          RTS     PC
5537                                  ;
5538                                  ; DISPLAY FILE FOR LONG VECTOR TEST.
5539 055450  114000                  ;
5540 055452  000000  000000          LV:     APNT
5541 055456  113774                          0,0
5542 055460  000000                          LVEC.ALL
5543 055462  000000                  LVX:    0
5544 055464  173000                  LVY:    0
5545                                          STOPN
5546 055466
                                             END.TEST

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

```
5548
                                        ;*****************************************************************
                                        ;*
                                        ;*      BEGIN TEST 20   -   RELATIVE POINTS
                                        ;*
                                        ;*****************************************************************

                                        ; (ITERATION COUNT = 10.)

5549                                    ;
5550                                    ; VERIFY THAT WE CAN EXECUTE RELATIVE POINTS FROM 1000,1000
5551                                    ; FOR THE FULL RANGE OF X/Y (-76 TO +76).
5552                                    ;
5553                                    ; PLOT ALL X POINTS, HOLDING Y AT -76.
5554                                    ; PLOT ALL Y POINTS, HOLDING X AT -76.
5555                                    ; PLOT ALL POINTS ON THE 45 (X = Y).
5556                                    ;
5557                                    ; REPEAT EACH POINT 10 TIMES AFTER QUICK PASS.
5558                                    ;
5559 055540  004737  027702            RPTST:  JSR     PC,TSTGO        ;TITLE
5560 055544  012737  056104  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5561 055552  004737  027054                    JSR     PC,LUMIN        ; BLAST THE LUT.
5562
5563 055556  005737  002502                    TST     DPUMOD
5564 055562  001002                            BNE     .+6             ; SKIP NEXT IF DPU-ONLY.
5565 055564  004737  026650                    JSR     PC,CLRRW        ; CLEAR AND SET R/W MODE.
5566
5567 055570  012737  177777  003232            MOV     #-1,TEMP1       ;USE AS XFER SWITCH.
5568 055576  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
5569 055604  012704  000702          8$:       MOV     #1000-76,R4     ;INIT EXPECTED X...
5570 055610  010405                            MOV     R4,R5           ;...AND Y.
5571 055612  012737  037576  056152            MOV     #37576,RPD      ;INIT DX,DY AT -MAX,-MAX.
5572 055620  053737  003064  056152            BIS     OPTI,RPD        ;OPTIONAL INTENSIFY.
5573
5574 055626                          1$:       BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5575
5576 055630  012777  056142  125174  9$:       MOV     #RPT,@DPC       ;START DISPLAY
5577 055636  004737  027454                    JSR     PC,WAITF
5578 055642  027704  125170                    CMP     @DXR,R4         ;CHECK DXR = R4...
5579 055646  001003                            BNE     2$
5580 055650  027705  125164                    CMP     @DYR,R5         ;...AND DYR = R5.
5581 055654  001410                            BEQ     3$
5582 055656                          2$:       HRDERR  RPF,XYERR       ;REL POINT FAILS.
5583 055676                          3$:       LOOPTO  9$              ; REPEAT...
     055676  004737  027646                    JSR     PC,LOOP
     055702  103752                            BCS     9$
5584 055704                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5585
5586 055706  013737  003234  027700            MOV     TEMP2,LOOPK     ;RESET LOOPER.
5587 055714  005737  003232                    TST     TEMP1           ;TEST XFER SWITCH.
5588 055720  003036                            BGT     5$              ;INCR X AND Y.
5589 055722  001420                            BEQ     4$              ;INCR Y ONLY.
5590
5591 055724  062704  000002                    ADD     #2,R4           ;INCR X ONLY.
5592 055730  010400                            MOV     R4,R0
5593 055732  004737  027312                    JSR     PC,SDX          ;CALCULATE +/- DX IN R0.
5594 055736  042737  037600  056152            BIC     #37600,RPD      ;CLEAR OLD...
```

```
5595 055744  050037  056152                BIS     R0,RPD          ;...SET NEX DX.
5596 055750  020427  001076                CMP     R4,#1076
5597 055754  101724                        BLOS    1$              ;AND LOOP 'TIL DONE.
5598 055756  005237  003232                INC     TEMP1           ;THEN, BUMP SWITCH...
5599 055762  000710                        BR      8$              ;...AND REINIT.
5600
5601 055764  062705  000002          4$:   ADD     #2,R5           ;INCR Y ONLY.
5602 055770  010500                        MOV     R5,R0
5603 055772  004737  027270                JSR     PC,SDY          ;CALCULATE +/- DY IN R0.
5604 055776  110037  056152                MOVB    R0,RPD          ;SET NEW DY.
5605 056002  020527  001076                CMP     R5,#1076
5606 056006  101707                        BLOS    1$              ;AND LOOP 'TIL DONE.
5607 056010  005237  003232                INC     TEMP1           ;THEN, BUMP SWITCH...
5608 056014  000673                        BR      8$              ;...AND REINIT.
5609
5610 056016  062704  000002          5$:   ADD     #2,R4           ;INCR X AND Y.
5611 056022  010405                        MOV     R4,R5           ; (Y = X)
5612 056024  010500                        MOV     R5,R0
5613 056026  004737  027270                JSR     PC,SDY          ;CALCULATE +/- DY IN R0.
5614 056032  010001                        MOV     R0,R1           ;SAVE FOR A MOMENT.
5615 056034  000300                        SWAB    R0
5616 056036  006200                        ASR     R0              ;REPOSITION FOR DX.
5617 056040  050100                        BIS     R1,R0           ;COMBINE DX AND DY.
5618 056042  042737  037777  056152        BIC     #37777,RPD      ;CLEAR OLD...
5619 056050  050037  056152                BIS     R0,RPD          ;...SET NEW DX,DY.
5620 056054  020527  001076                CMP     R5,#1076
5621 056060  101662                        BLOS    1$              ;AND LOOP 'TIL DONE.
5622
5623 056062  005737  002502          10$:  TST     DPUMOD
5624 056066  001002                        BNE     11$
5625 056070  004737  027514                JSR     PC,PAUSE1       ; PAUSE IF VIDEO ON.
5626 056074  004737  030072          11$:  JSR     PC,TSTEND
5627 056100                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
5628                                ;
5629                                ; ON ERROR, SHOW ORIGIN AND DELTAS.
5630                                ;
5631 056104                        20$:    PRINTX  #SHRTX,RPT+2,RPT+4,RPD ; EXTENSION.
5632 056140  000207                        RTS     PC
5633                                ;
5634                                ;DISPLAY FILE FOR REL POINT TEST.
5635                                ;
5636 056142  114000                RPT:    APNT
5637 056144  001000  001000                1000,1000               ;ORIGIN AT CENTER SCREEN.
5638 056150  133774                        RPNT!ALL
5639 056152  000000                RPD:    0                       ;DELTA X AND DELTA Y.
5640 056154  173000                        STOPN
5641
5642 056156                                END.TEST

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

```
5644
                                        ;*****************************************************************
                                        ;*
                                        ;*      BEGIN TEST 21   -   SHORT VECTORS
                                        ;*
                                        ;*****************************************************************

                                        ; (ITERATION COUNT = 10.)

5645                                    ;
5646                                    ; VERIFY THAT WE CAN EXECUTE SHORT VECTORS FROM 1000,1000
5647                                    ; FOR ALL VALUES OF DELTA X AND Y.
5648                                    ; SWEEP VECTORS FROM MAX,MAX, CCW THRU 360 DEGREES.
5649                                    ;
5650                                    ; REPEAT EACH VECTOR 10 TIMES AFTER QUICK PASS.
5651                                    ;
5652 056226  004737  027702            SVTST:  JSR     PC,TSTGO        ;TITLE
5653 056232  012737  056604  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
5654 056240  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
5655 056244  005737  002502                    TST     DPUMOD
5656 056250  001002                            BNE     .+6             ; SKIP NEXT IF DPU-ONLY.
5657 056252  004737  026650                    JSR     PC,CLRRW        ; CLEAR AND SET R/W MODE.
5658 056256  012704  001074                    MOV     #1074,R4        ;INIT EXPECTED X...
5659 056262  012705  001076                    MOV     #1076,R5        ;...AND Y TOO.
5660 056266  012737  017076  056652            MOV     #17076,SVD      ;INIT DX +74, DX +76.
5661 056274  053737  003064  056652            BIS     OPT1,SVD        ;...INT OPTIONAL.
5662 056302  005037  003232                    CLR     TEMP1           ;USE AS XFER SWITCH.
5663 056306  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPER.
5664
5665 056314                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5666
5667 056316  012777  056642  124506    9$:     MOV     #SVT,@DPC       ;START DISPLAY
5668 056324  004737  027454                    JSR     PC,WAITF
5669 056330  027704  124502                    CMP     @DXR,R4         ;CHECK DXR = R4...
5670 056334  001003                            BNE     2$
5671 056336  027705  124476                    CMP     @DYR,R5         ;...AND DYR = R5.
5672 056342  001411                            BEQ     3$
5673 056344                            2$:     HRDERR  SVF,XYERR       ;SHORT VECTOR FAILS.
5674 056364  000403                            BR      30$
5675 056366                            3$:     LOOPTO  9$              ; REPEAT...
     056366  004737  027646                    JSR     PC,LOOP
     056372  103751                            BCS     9$
5676 056374                            30$:    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5677
5678 056376  013737  003234  027700            MOV     TEMP2,LOOPK     ;RESET LOOPER.
5679 056404  005737  003232                    TST     TEMP1           ;TEST XFER SWITCH...
5680 056410  001033                            BNE     7$              ;...BR IF SET.
5681 056412  020427  000702                    CMP     R4,#1000-76
5682 056416  001413                            BEQ     5$              ;BR IF X MIN.
5683 056420  162704  000002                    SUB     #2,R4           ;OTHERWISE, DECR X.
5684 056424  010400                            MOV     R4,R0
5685 056426  004737  027312                    JSR     PC,SDX          ;TRANSLATE TO DX.
5686 056432  042737  037600  056652            BIC     #37600,SVD      ;CLEAR OLD...
5687 056440  050037  056652                    BIS     R0,SVD          ;...SET NEW DX...
5688 056444  000723                            BR      1$              ;...AND CONTINUE.
5689
5690 056446  020527  000702            5$:     CMP     R5,#1000-76
```

```
5691 056452  001410                    BEQ     6$              ;BR IF Y MIN.
5692 056454  162705  000002             SUB     #2,R5           ;OTHERWISE, DECR Y.
5693 056460  010500                    MOV     R5,R0
5694 056462  004737  027270             JSR     PC,SDY          ;TRANSLATE TO DY.
5695 056466  110037  056652             MOVB    R0,SVD          ;SET NEW DY...
5696 056472  000710                    BR      1$              ;...AND CONTINUE.
5697
5698 056474  005137  003232      6$:    COM     TEMP1           ;SET XFER SWITCH.
5699 056500  020427  001076      7$:    CMP     R4,#1000+76
5700 056504  001413                    BEQ     8$              ;BR IF X IS MAX.
5701 056506  062704  000002             ADD     #2,R4           ;OTHERWISE, INCR X.
5702 056512  010400                    MOV     R4,R0
5703 056514  004737  027312             JSR     PC,SDX          ;TRANSLATE TO DX.
5704 056520  042737  037600  056652     BIC     #37600,SVD      ;CLEAR OLD...
5705 056526  050037  056652             BIS     R0,SVD          ;...SET NEW DX...
5706 056532  000670                    BR      1$              ;...AND CONTINUE.
5707
5708 056534  020527  001076      8$:    CMP     R5,#1000+76
5709 056540  001410                    BEQ     10$             ;BR (DONE) IF Y IS MAX.
5710 056542  062705  000002             ADD     #2,R5           ;OHTERWISE, INCR Y.
5711 056546  010500                    MOV     R5,R0
5712 056550  004737  027270             JSR     PC,SDY          ;TRANSLATE TO DY.
5713 056554  110037  056652             MOVB    R0,SVD          ;SET NEW DY...
5714 056560  000655                    BR      1$              ;...AND CONTINUE.
5715
5716 056562  005737  002502     10$:    TST     DPUMOD
5717 056566  001002                    BNE     11$
5718 056570  004737  027514             JSR     PC,PAUSE1       ; PAUSE IF VIDEO ON.
5719 056574  004737  030072     11$:    JSR     PC,TSTEND
5720 056600                     EXIT    TST                     ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
5721
5722                             ;
5723                             ; ON ERROR, SHOW ORIGIN AND DELTAS.
5724 056604                     ;
5725 056640  000207             20$:    PRINTX  #SHRTX,SVT+2,SVT+4,SVD ; EXTENSION.
5726                                     RTS     PC
5727                             ;
5728                             ; DISPLAY CODE FOR SHORT VECTORS.
5729 056642  114000             ;
5730 056644  001000  001000     SVT:    APNT
5731 056650  107774                     1000,1000               ;ORIGIN AT CENTER SCREEN.
5732 056652  000000                     SVEC!ALL
5733 056654  173000             SVD:    0                       ;DELTA X AND DELTA Y.
5734                                     STOPN
5735 056656
                                        END.TES!

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

```
5737                                    ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 22   -   RUN-LENGTH
                                        ;*
                                        ;*************************************************************************
                                        ; (ITERATION COUNT = 10.)

5738
5739                                    ;
5740                                    ; ALL VARIATIONS OF RUN LENTH MODE ARE DONE AND THE END POINTS ARE CHECKED.
5741                                    ; IF TESTING IN DPU-ONLY MODE, JUST THE 'NEWLINE' COMMANDS ARE TESTED.
5742                                    ;
5743 056724                            RLMOD:  COMBEG
      056724  004737  027702                    JSR     PC,TSTGO        ; TITLE.
      056730  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
5744  056734  004737  026650                    JSR     PC,CLRRW        ; CLEAR FOR READ/WRITE.
5745  056740  005737  002502            RLMOD1: TST     DPUMOD          ; ARE WE IN DPU-ONLY MODE?
5746  056744  001121                            BNE     RLNEWL          ; IF YES, JUST DO 'NEWLINE' TESTS.
                                                        ; IF NO, DO ALL TESTS ( WRITE TO SCREEN ).
                                                        ; TEST X INCREMENT.
5747
5748
5749  056746  012705  000001            1$:     MOV     #1,R5           ; INIT COUNT FOR X TEST.
5750  056752                                    DPSTART #XINC
      056752  012777  057564  124052            MOV     #XINC,@DPC      ; START THE DPU.
      056760  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
5751  056764                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5752  056766  110537  057605            2$:     MOVB    R5,XINCB+1      ; MODIFY COUNT IN DISPLAY FILE.
5753  056772                                    DPCONT
      056772  052777  000001  124032            BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
      057000  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
5754  057004  010501                            MOV     R5,R1           ; R1 = EXPECTED DXR.
5755  057006  006301                            ASL     R1              ;
5756  057010  012703  000000                    MOV     #0,R3           ; R3 = EXPECTED DYR.
5757  057014  017702  124016                    MOV     @DXR,R2         ; READ DXR INTO R2.
5758  057020  017704  124014                    MOV     @DYR,R4         ; READ DYR INTO R4.
5759  057024                                    IFERRX2 RLXIE,RLXYE
      057024  020102                            CMP     R1,R2           ; 1ST ITEM OK?
      057026  001002                            BNE     65$             ; NO.
      057030  020304                            CMP     R3,R4           ; 2ND ITEM OK?
      057032  001410                            BEQ     64$             ; YES.
      057034                            65$:    HRDERR  RLXIE,RLXYE,    ; NO.
      057054                            64$:
5760  057054  005205                            INC     R5              ; INCREMENT X VALUE.
5761  057056  020527  000200                    CMP     R5,#200         ; DONE?
5762  057062  001341                            BNE     2$              ; NO.
5763  057064                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
5764                                                            ; TEST X2 INCREMENT.
5765  057066  012705  000001                    MOV     #1,R5           ; INIT COUNT FOR X2 TEST.
5766  057072                                    DPSTART #X2INC
      057072  012777  057614  123732            MOV     #X2INC,@DPC     ; START THE DPU.
      057100  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
5767  057104                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
5768  057106  110537  057635            3$:     MOVB    R5,X2INCB+1     ; MODIFY COUNT IN DISPLAY FILE.
5769  057112                                    DPCONT
      057112  052777  000001  123712            BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
      057120  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
```

```
                                                MOV     R5,R1           ; R1 = EXPECTED DXR.
                                                ASL     R1              ;
                                                ASL     R1              ;
                                                MOV     #0,R3           ; R3 - EXPECTED DYR.
                                                MOV     @DXR,R2         ; READ DXR INTO R2.
                                                MOV     @DYR,R4         ; READ DYR INTO R4.
                                                IFERRX2 RLX2IE,RLXYE
                                                CMP     R1,R2           ; 1ST ITEM OK?
                                                BNE     67$             ; NO.
                                                CMP     R3,R4           ; 2ND ITEM OK?
                                                BEQ     66$             ; YES.
                                        67$:    HRDERR  RLX2IE,RLXYE,   ; NO.
                                        66$:
                                                INC     R5              ; INCREMENT X VALUE.
                                                CMP     R5,#200         ; DONE?
                                                BNE     3$              ; NO.
                                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.

                                        ; TEST Y UP.
                                        RLNEWL: BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
                                                DPSTART #YUP
                                                MOV     #YUP,@DPC       ; START THE DPU.
                                                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
                                                MOV     #0,R1           ; R1 = EXPECTED DXR.
                                                MOV     #2,R3           ; R3 = EXPECTED DYR.
                                                MOV     @DXR,R2         ; READ DXR INTO R2.
                                                MOV     @DYR,R4         ; READ DYR INTO R4.
                                                IFERRX2 RLYUE,RLXYE
                                                CMP     R1,R2           ; 1ST ITEM OK?
                                                BNE     65$             ; NO.
                                                CMP     R3,R4           ; 2ND ITEM OK?
                                                BEQ     64$             ; YES.
                                        65$:    HRDERR  RLYUE,RLXYE,    ; NO.
                                        64$:
                                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.

                                        ; TEST Y DOWN.
                                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
                                                DPSTART #YDN
                                                MOV     #YDN,@DPC       ; START THE DPU.
                                                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
                                                MOV     #0,R1           ; R1 = EXPECTED DXR.
                                                MOV     #0,R3           ; R3 = EXPECTED DYR.
                                                MOV     @DXR,R2         ; READ DXR INTO R2.
                                                MOV     @DYR,R4         ; READ DYR INTO R4.
                                                IFERRX2 RLYDE,RLXYE
                                                CMP     R1,R2           ; 1ST ITEM OK?
                                                BNE     67$             ; NO.
                                                CMP     R3,R4           ; 2ND ITEM OK?
                                                BEQ     66$             ; YES.
                                        67$:    HRDERR  RLYDE,RLXYE,    ; NO.
                                        66$:
                                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.

                                        ; TEST SKIP UP.
                                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
                                                DPSTART #Y2UP
```

```
                057366  012777  057674  123436          MOV     #Y2UP,@DPC      ; START THE DPU.
                057374  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
           5805 057400  012701  000000                  MOV     #0,R1           ; R1 = EXPECTED DXR.
           5806 057404  012703  000004                  MOV     #4,R3           ; R3 = EXPECTED DYR.
           5807 057410  017702  123422                  MOV     @DXR,R2         ; READ DXR INTO R2.
           5808 057414  017704  123420                  MOV     @DYR,R4         ; READ DYR INTO R4.
           5809 057420                                  IFERRX2 RLY2UE,RLXYE
                057420  020102                          CMP     R1,R2           ; 1ST ITEM OK?
                057422  001002                          BNE     69$             ; NO.
                057424  020304                          CMP     R3,R4           ; 2ND ITEM OK?
                057426  001410                          BEQ     68$             ; YES.
                057430                          69$:    HRDERR  RLY2UE,RLXYE,   ; NO.
                057450                          68$:
           5810 057450                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
           5811
           5812                                 ; TEST SKIP DOWN.
           5813 057452                                  BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
           5814 057454                                  DPSTART #Y2DN
                057454  012777  057710  123350          MOV     #Y2DN,@DPC      ; START THE DPU.
                057462  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
           5815 057466  012701  000000                  MOV     #0,R1           ; R1 = EXPECTED DXR.
           5816 057472  012703  000000                  MOV     #0,R3           ; R3 = EXPECTED DYR.
           5817 057476  017702  123334                  MOV     @DXR,R2         ; READ DXR INTO R2.
           5818 057502  017704  123332                  MOV     @DYR,R4         ; READ DYR INTO R4.
           5819 057506                                  IFERRX2 RLY2DE,RLXYE
                057506  020102                          CMP     R1,R2           ; 1ST ITEM OK?
                057510  001002                          BNE     71$             ; NO.
                057512  020304                          CMP     R3,R4           ; 2ND ITEM OK?
                057514  001410                          BEQ     70$             ; YES.
                057516                          71$:    HRDERR  RLY2DE,RLXYE,   ; NO.
                057536                          70$:
           5820 057536                                  ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
           5821
           5822 057540                          RLMXIT: COMEND  RLMOD1
                057540  004737  027646                  JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
                057544  103401                          BCS     64$
                057546  000402                          BR      65$

                057550  000137  056740          64$:    JMP     RLMOD1

                057554  004737  030072          65$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
           5823 057560                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>
           5824
           5825
           5826
           5827                                 ;
           5828                                 ; DISPLAY CODE.
           5829                                 ;
           5830
           5831 057564  114000  000000  000000  XINC:   APNT,0,0                ; X INCREMENT.
           5832 057572  173000                          STOPN
           5833 057574  114000  040000  000000  XINCA:  APNT,I!0,0
           5834 057602  144000                          RNLN
           5835 057604  000777                  XINCB:  000777
           5836 057606  173000                          STOPN
           5837 057610  160000  057574                  DJMP,XINCA
           5838
```

```
 5839 057614  114000  000000  000000  X2INC:  APNT,0,0                ; X2 INCREMENT.
 5840 057622  173000                          STOPN
 5841 057624  114000  040000  000000  X2INCA: APNT,I!0,0
 5842 057632  144100                          RNLN.DBLPIX
 5843 057634  000777                  X2INCB: 000777
 5844 057636  173000                          STOPN
 5845 057640  160000  057624                  DJMP,X2INCA
 5846
 5847 057644  114000  000000  000000  YUP:    APNT,0,0                ; Y UP.
 5848 057652  144000  000377                  RNLN,377
 5849 057656  173000                          STOPN
 5850
 5851 057660  114000  000000  000002  YDN:    APNT,0,2                ; Y DOWN.
 5852 057666  144040  000377                  RNLN.YDOWN,377
 5853 057672  173000                          STOPN
 5854
 5855 057674  114000  000000  000000  Y2UP:   APNT,0,0                ; Y SKIP UP.
 5856 057702  144400  000377                  RNLN!RLSKIP,377
 5857 057706  173000                          STOPN
 5858
 5859 057710  114000  000000  000004  Y2DN:   APNT,0,4                ; Y SKIP DOWN.
 5860 057716  144440  000377                  RNLN!RLSKIP!YDOWN,377
 5861 057722  173000                          STOPN
 5862
 5863 057724                                  END.TEST

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

```
5865
                        ;*****************************************************************************
                        ;*
                        ;*      BEGIN TEST 23   -   CURSOR REGISTERS/SWITCH/MATCH
                        ;*
                        ;*****************************************************************************

                        ; (ITERATION COUNT = 3.)

5866                    ;
5867                    ; THIS TEST, USING THE LOADABLE CURSOR POSITION REGISTERS AND ''SOFT'' SWITCH,
5868                    ; PERFORMS THE FOLLOWING CHECKS ON THE CURSOR REGISTERS, JOYSTICK SWITCH, AND
5869                    ; CURSOR MATCH AND ASSOCIATED LOGIC FOR EACH AVAILABLE SYNC GENERATOR CHANNEL:
5870                    ;
5871                    ;  (1)  WRITE/READ TEST OF THE CURSOR POSITION REGISTERS USING DATA OF ALL 1'S
5872                    ;       AND ALL 0'S, USING PROGRAMMED I/O (VIA DXR AND DYR) TO CHECK FOR STUCK
5873                    ;       0'S, STUCK 1'S AND DUAL ADDRESSING.
5874                    ;
5875                    ;  (2)  WRITE/READ TEST OF THE CURSOR POSITION REGISTERS VIA THE DISPLAY FILE
5876                    ;       INSTRUCTION LDCP (LOAD CURSOR POSITION) AND CURD (CURSOR READ), USING
5877                    ;       AN INCREMENTING COUNT PATTERN ON THE X POSITION (0 - 3776) AND A
5878                    ;       DECREMENTING COUNT PATTERN ON THE Y POSITION (3776 - 0).
5879                    ;
5880                    ;  (3)  WRITE/READ TEST OF THE SWITCH ENABLE, MATCH ENABLE, AND CROSSHAIR
5881                    ;       INTENSITY ENABLE BITS USING PROGRAMMED I/O TO SET AND CLEAR ALL BITS.
5882                    ;
5883                    ;  (4)  WRITE/READ TEST OF THE ENABLE BITS VIA DISPLAY FILE INSTRUCTION
5884                    ;       LDJSS (LOAD JOYSTICK STATUS), USING ALL COMBINATIONS OF LOAD-ENABLES
5885                    ;       AND DATA.
5886                    ;
5887                    ;  (5)  SET THE SWITCH ENABLE AND MATCH ENABLE BITS, THEN PERFORM A WRITE TO
5888                    ;       THE JOYSTICK STATUS REGISTER OF ANOTHER CHANNEL WITH DATA CONFIGURED
5889                    ;       TO SET THE ENABLE BITS; VERIFY THAT THE ENABLES OF THE CHANNEL UNDER
5890                    ;       TEST WERE CLEARED.  FAILURE HERE INDICATES A FAULTY CHANNEL ADDRESS
5891                    ;       DECODER IN THE SYNC CHANNEL UNDER TEST.
5892                    ;
5893                    ;  (6)  TEST OPERATION OF THE JOYSTICK SWITCH INTERRUPT.
5894                    ;
5895                    ;  (7)  TEST OPERATION OF THE CURSOR MATCH INTERRUPT. FLOATING 1'S AND 0'S
5896                    ;       PATTERNS ARE USED IN THE CURSOR X AND Y POSITIONS TO VERIFY THE A
5897                    ;       MATCH OCCURS ON THE SELECTED POINT WHILE DRAWING A VECTOR THROUGH IT,
5898                    ;       AND THAT A MATCH DOES NOT OCCUR WHILE DRAWING VECTORS AROUND THE
5899                    ;       SELECTED POINT PARALLEL TO THE X AND Y AXES.
5900                    ;
5901
5902 060014  005737  002502          TST     DPUMOD          ; DPU-MODE ONLY?
5903 060020  001402                  BEQ     CSMTST          ; BR IF NOT -- OK TO EXECUTE.
5904 060022  000137  064660          JMP     CSMXIT          ; YES -- JUST GO TO EXIT.
```

```
5906
5907 060026                            CSMTST: COMBEG
     060026  004737  027702                    JSR     PC,TSTGO        ; TITLE.
     060032  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INIT.
5908 060036  004737  030414            1$:     JSR     PC,CONFIG               ; IDENTIFY THE AVAILABLE SYNC CHANNELS.
5909 060042  004737  027054                    JSR     PC,LUMIN
5910 060046  005037  065002            CSMITR: CLR     CSCHAN          ; START TESTING AT CHANNEL 0.
5911 060052  012737  065014  022300    CSMLOOP: MOV    #PRSYCH,EXTA    ; SET UP TO PRINT CHANNEL NUMBER ON ERROR.
5912 060060  004737  026074                    JSR     PC,DPRESET      ; DO SOFT INITIALIZE & CHECK REGISTERS.
5913 060064  013702  065002                    MOV     CSCHAN,R2       ; GET CHANNEL NUMBER INTO R2
5914 060070  006302                            ASL     R2              ; ... AND MAKE IT A WORD OFFSET.
5915 060072  005762  003270                    TST     SYCTAB(R2)      ; IS THERE A MODULE HERE?
5916 060076  001002                            BNE     1$              ; BR IF YES.
5917 060100  000137  064620                    JMP     CSMNEXT         ; IF NOT, GO SELECT NEXT CHANNEL.
5918 060104  006302                    1$:     ASL     R2              ; SHIFT IT L  T 5 MORE TIMES TO GET
5919 060106  006302                            ASL     R2
5920 060110  006302                            ASL     R2
5921 060112  006302                            ASL     R2
5922 060114  006302                            ASL     R2              ; ... CHANNEL NUMBER INTO BITS <7:6>
5923 060116  010237  065004                    MOV     R2,CSCH.6       ; STORE IT FOR FUTURE USE.
5924 060122  006302                            ASL     R2              ; SHIFT IT LEFT 2 MORE TIMES TO GET
5925 060124  006302                            ASL     R2              ; ... IT INTO BITS <9:8>
5926 060126  010237  065006                    MOV     R2,CSCH.8       ; STORE IT FOR FUTURE USE.
5927 060132  006302                            ASL     R2              ; SHIFT LEFT 3 MORE TIMES ...
5928 060134  006302                            ASL     R2              ; ... TO GET ...
5929 060136  006302                            ASL     R2              ; ... CHANNEL NUMBER INTO BITS <12:11>,
5930 060140  010237  065010                    MOV     R2,CSCH.11      ; AND STORE IT FOR FUTURE USE.
5931 060144  112737  000000  026106          MOVB    #0,INTMASK      ; DON'T EXPECT ANY INTERRUPT.
5932 060152  004737  026114                    JSR     PC,ENAINT       ; BUT LOOK FOR BAD ONES.
5933
5934                                    ;WRITE 0'S INTO BOTH CURSOR REGISTERS AND VERIFY THAT ALL 0'S ARE RETURNED.
5935
5936 060156                            CSMT1:  BGNSEG          ;<<<<<<<<<<<<BEGIN SEGMENT
5937 060160  012701  100000                    MOV     #BIT15,R1       ; EXPECT CURSOR X = 0 (WITH FLAG SET)
5938 060164  010103                            MOV     R1,R3           ; DITTO FOR CURSOR Y
5939 060166  013705  065012                    MOV     CSRETRY,R5      ; SET UP R5 AS RETRY COUNTER ;@@@R
5940 060172                            1$:     BISW3   CSCH.11,#WRTCPX!0,@DXR ; WRITE CURSOR X TO 0.
     060172  012746  140000                    MOV     #WRTCPX!0,-(SP)
     060176  053716  065010                    BIS     CSCH.11,(SP)
     060202  012677  122630                    MOV     (SP)+,@DXR
5941 060206  004737  027454                    JSR     PC,WAITF                ; WAIT FOR READY.
5942 060212                                    BISW3   CSCH.11,#WRTCPY!0,@DYR ; WRITE CURSOR Y TO 0.
     060212  012746  140000                    MOV     #WRTCPY!0,-(SP)
     060216  053716  065010                    BIS     CSCH.11,(SP)
     060222  012677  122612                    MOV     (SP)+,@DYR
5943 060226  004737  065336                    JSR     PC,GETCURS              ; READ CURSOR, GET DXR,DYR INTO R2,R4
5944 060232  020102                            CMP     R1,R2           ; CHECK DXR (X-CURSOR)
5945 060234  001002                            BNE     2$              ; BR IF ERROR
5946 060236  020304                            CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
5947 060240  001412                            BEQ     3$              ; BR IF OK.
5948 060242  005305                    2$:     DEC     R5              ; DECREMENT THE RETRY COUNTER ;@@@R
5949 060244  100352                            BPL     1$              ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
5950 060246                                    HRDERR  CRWRZ,RLXYE     ; REPORT THE ERROR.
5951 060266                            3$:     ENDSEG          ;>>>>>>>>>>>>END SEGMENT
```

```
                                        ;WRITE 1'S INTO BOTH CURSOR REGISTERS AND VERIFY THAT ALL 1'S ARE RETURNED.

060270                          CSMT2:  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
060272  012701  103776                  MOV     #BIT15!^O03776,R1 ; EXPECT CURSOR X = 3776 (WITH FLAG SET)
060276  010103                          MOV     R1,R3           ; DITTO FOR CURSOR Y
060300  013705  065012                  MOV     CSRETRY,R5      ; @@@R
060304                          1$:     BISW3   CSCH.11,#WRTCPX!^O03776,@DXR ; WRITE CURSOR X TO 3776.
060304  012746  143776                  MOV     #WRTCPX!^O03776,-(SP)
060310  053716  065010                  BIS     CSCH.11,(SP)
060314  012677  122516                  MOV     (SP)+,@DXR
060320  004737  027454                  JSR     PC,WAITF                ; WAIT FOR READY.
060324                                  BISW3   CSCH.11,#WRTCPY!^O03776,@DYR ; WRITE CURSOR Y TO 3776.
060324  012746  143776                  MOV     #WRTCPY!^O03776,-(SP)
060330  053716  065010                  BIS     CSCH.11,(SP)
060334  012677  122500                  MOV     (SP)+,@DYR
060340  004737  065336                  JSR     PC,GETCURS              ; READ CURSOR, GET DXR,DYR INTO R2,R4
060344  020102                          CMP     R1,R2           ; CHECK DXR (X-CURSOR)
060346  001002                          BNE     2$              ; BR IF ERROR
060350  020304                          CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
060352  001412                          BEQ     3$              ; BR IF OK.
060354  005305                  2$:     DEC     R5              ; DECREMENT THE RETRY COUNTER ;@@@R
060356  100352                          BPL     1$              ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
060360                                  HRDERR  CRWRO,RLXYE     ; REPORT THE ERROR.
060400                          3$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
```

```
5973
5974                                    ;CHECK DUAL ADDRESSING -- VERIFY THAT Y NOT CHANGED BY WRITING X.
5975
5976 060402  012701  100000      CSMT3:  MOV     #BIT15!0,R1     ; EXPECT CURSOR X TO BE 0
5977 060406  012703  103776              MOV     #BIT15!^03776,R3 ; EXPECT CURSOR Y TO BE ALL 1'S
5978 060412                             BISW3   CSCH.11,#WRTCPX,R5 ; SET UP WRITE CURSOR W/ CHANNEL
     060412  012705  140000              MOV     #WRTCPX,R5
     060416  053705  065010              BIS     CSCH.11,R5
5979
5980 060422                     1$:     BGNSEG          ;<<<<<<<<<<<<BEGIN SEGMENT
5981 060424  013700  065012              MOV     CSRETRY,R0      ; @@@R
5982 060430                     10$:    BISW3   R5,R3,@DYR      ; LOAD CURSOR Y WITH A PATTERN.
     060430  010346                      MOV     R3,-(SP)
     060432  050516                      BIS     R5,(SP)
     060434  012677  122400              MOV     (SP)+,@DYR
5983 060440  004737  027454              JSR     PC,WAITF        ; WAIT FOR STOP.
5984 060444                             BISW3   R5,R1,@DXR      ; LOAD OPPOSITE PATTERN INTO CURSOR X.
     060444  010146                      MOV     R1,-(SP)
     060446  050516                      BIS     R5,(SP)
     060450  012677  122362              MOV     (SP)+,@DXR
5985 060454  004737  065336              JSR     PC,GETCURS      ; GET THE CURSOR INTO R2 & R4.
5986 060460  020102                      CMP     R1,R2           ; CHECK DXR (X-CURSOR)
5987 060462  001002                      BNE     2$              ; BR IF ERROR
5988 060464  020304                      CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
5989 060466  001412                      BEQ     3$              ; BR IF OK.
5990 060470  005300              2$:     DEC     R0              ; DECREMENT THE RETRY COUNTER ;@@@R
5991 060472  100356                      BPL     10$             ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
5992 060474                             HRDERR  CRDAX,RLXYE     ; REPORT THE ERROR.
5993 060514                     3$:     ENDSEG          ;>>>>>>>>>>>>END SEGMENT
5994
5995 060516  032701  003776              BIT     #^03776,R1      ; DID WE WRITE 1'S INTO X YET?
5996 060522  001004                      BNE     4$              ; BR IF YES, SO WE'RE DONE.
5997 060524  010301                      MOV     R3,R1           ; NO, SO SWAP PATTERNS.
5998 060526  012703  100000              MOV     #BIT15,R3       ; ... SET UP TO WRITE Y TO 0.
5999 060532  000733                      BR      1$              ; LOOP BACK TO DO IT.
6000 060534                     4$:
```

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

060534  012701  103776      CSM14:  MOV     #BIT15!^O3776,R1 ; EXPECT CURSOR X TO BE ALL 1'S
060540  012703  100000              MOV     #BIT15!0,R3      ; EXPECT CURSOR Y TO BE 0
                              BISW3   CSCH.11,#WRTCPX,R5 ; SET UP WRITE CURSOR W/ CHANNEL
060544  012705  140000              MOV     #WRTCPX,R5
060550  053705  065010              BIS     CSCH.11,R5

060554                        1$:     BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
060556  013700  065012              MOV     CSRETRY,R0      ; @@@R
060562                        10$:    BISW3   R5,R1,@DXR      ; LOAD CURSOR X WITH A PATTERN.
060562  010146                      MOV     R1,-(SP)
060564  050516                      BIS     R5,(SP)
060566  012677  122244              MOV     (SP)+,@DXR
060572  004737  027454              JSR     PC,WAITF
060576                              BISW3   R5,R3,@DYR      ; LOAD OPPOSITE PATTERN INTO CURSOR Y.
060576  010346                      MOV     R3,-(SP)
060600  050516                      BIS     R5,(SP)
060602  012677  122232              MOV     (SP)+,@DYR
060606  004737  065336              JSR     PC,GETCURS              ; GET THE CURSOR INTO R2 & R4.
060612  020102                      CMP     R1,R2           ; CHECK DXR (X-CURSOR)
060614  001002                      BNE     2$              ; BR IF ERROR
060616  020304                      CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
060620  001412                      BEQ     3$              ; BR IF OK.
060622  005300                2$:   DEC     R0              ; DECREMENT THE RETRY COUNTER ;@@@R
060624  100356                      BPL     10$             ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
060626                              HRDERR  CRDAY,RLXYE     ; REPORT THE ERROR.
060646                        3$:   ENDSEG          ;>>>>>>>>>>>>>END SEGMENT

060650  032703  003776              BIT     #^O3776,R3      ; DID WE WRITE 1'S INTO Y YET?
060654  001004                      BNE     4$              ; BR IF YES, SO WE'RE DONE.
060656  010103                      MOV     R1,R3           ; NO, SO SWAP PATTERNS.
060660  012701  100000              MOV     #BIT15,R1       ; ... SET UP TO WRITE X TO 0.
060664  000733                      BR      1$              ; LOOP BACK TO DO IT.
060666                        4$:
```

```
                                        ; DO WRITE/READ TEST USING DISPLAY FILE.
                                        ;
060666  005001                  CSMT5:  CLR     R1              ; INIT THE X PATTERN TO 0
060670  012703  003776                  MOV     #^O03776,R3     ; INIT THE Y PATTERN TO ALL 1'S
060674                                  BISW3   CSCH.6,#LDCP,CSMF1A ; SET UP THE DISPLAY FILE
060674  012746  175000                  MOV     #LDCP,-(SP)
060700  053716  065004                  BIS     CSCH.6,(SP)
060704  012637  065042                  MOV     (SP)+,CSMF1A
060710                                  BISW3   CSCH.8,#CURD,CSMF1B ; ... BY INSERTING THE CHANNEL.
060710  012746  146100                  MOV     #CURD,-(SP)
060714  053716  065006                  BIS     CSCH.8,(SP)
060720  012637  065050                  MOV     (SP)+,CSMF1B
                                        ;
060724  010137  065044          1$:     MOV     R1,CSMF1A+2     ; INSERT X COORD. TO BE WRITTEN
060730  010337  065046                  MOV     R3,CSMF1A+4     ; ... AND THE Y COORDINATE.
060734  013705  065012                  MOV     CSRETRY,R5
060740                                  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
060742                          2$:     DPSTART #CSMF1A         ; EXECUTE LOAD THEN READ OF CURSOR.
060742  012777  065042  122062          MOV     #CSMF1A,@DPC    ; START THE DPU.
060750  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
060754  052701  100000                  BIS     #BIT15,R1       ; EXPECT THE CURSOR FLAG BIT IN DXR
060760  052703  100000                  BIS     #BIT15,R3       ; ... AND DYR.
060764  004737  065362                  JSR     PC,RDDXY        ; GET MASKED DXR AND DYR INTO R2 AND R4.
060770  020102                          CMP     R1,R2           ; CHECK DXR FOR X CURSOR.
060772  001002                          BNE     3$              ; BR IF INCORRECT.
060774  020304                          CMP     R3,R4           ; CHECK DYR FOR Y CURSOR.
060776  001412                          BEQ     4$              ; BR IF OK.
061000  005305                  3$:     DEC     R5
061002  100357                          BPL     2$
061004                                  HRDERR  CRWRE,RLXYE     ; ERROR -- DXR OR DYR WRITE/READ ERROR.
061024                          4$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
                                        ;
061026  062701  100002                  ADD     #2.BIT15,R1     ; INCREMENT THE X PATTERN.
061032  162703  100002                  SUB     #2.BIT15,R3     ; DECREMENT THE Y PATTERN.
061036  032701  003776                  BIT     #^O03776,R1     ; DID WE INCREMENT X AROUND TO 0?
061042  001330                          BNE     1$              ; LOOP BACK IF NO.
061044                          30$:
```

```
6062
6063                                    ; *** SPACE HERE FOR TESTING GENERAL DUAL ADDRESSING (ALL DBUS CODES & CHANNELS)
6064
6065                                    ; VERIFY THAT THE SWITCH, MATCH AND CROSSHAIR ENABLES CAN BE CLEARED TO 0'S.
6066
6067 061044                             CSMT6:  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
6068 061046  004737  026176                     JSR     PC,DSBINT       ; DISABLE INTERRUPTS
6069 061052  012701  100000                     MOV     #BIT15,R1       ; EXPECT DXR TO BE 0
6070 061056  010103                             MOV     R1,R3           ; DITTO FOR DYR.
6071 061060                                     BISW3   CSCH.8,#WJSS!^O77,@DXR ; FIRST SET ALL ENABLES.
     061060  012746  042077                     MOV     #WJSS!^O77,-(SP)
     061064  053716  065006                     BIS     CSCH.8,(SP)
     061070  012677  121742                     MOV     (SP)+,@DXR
6072 061074  004737  027454                     JSR     PC,WAITF                ; WAIT FOR READY.
6073 061100                                     BISW3   CSCH.8,#WJSS.BIT5!BIT3!BIT1,@DXR ; THEN CLEAR THEM.
     061100  012746  042052                     MOV     #WJSS!BIT5!BIT3.BIT1,-(SP)
     061104  053716  065006                     BIS     CSCH.8,(SP)
     061110  012677  121722                     MOV     (SP)+,@DXR
6074 061114  013705  065012                     MOV     CSRETRY,R5              ; @@@R
6075 061120  004737  065404             1$:     JSR     PC,GETJSE               ; GET THE ENABLES INTO R2 AND R4.
6076 061124  020102                             CMP     R1,R2           ; CHECK DXR (X-CURSOR)
6077 061126  001002                             BNE     2$              ; BR IF ERROR
6078 061130  020304                             CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
6079 061132  001412                             BEQ     3$              ; BR IF OK.
6080 061134  005305                     2$:     DEC     R5              ; DECREMENT THE RETRY COUNTER ;@@@R
6081 061136  100370                             BPL     1$              ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
6082 061140                                     HRDERR  JSEWRZ,RLXYE    ; REPORT THE ERROR.
6083 061160                             3$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
6084
6085                                    ; VERIFY THAT THE SWITCH, MATCH AND CROSSHAIR ENABLES CAN BE SET TO 1'S.
6086
6087 061162                             CSMT7:  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
6088 061164  012701  114000                     MOV     #BIT15!JSSIES!JSMIES,R1 ; EXPECT DXR TO HAVE SW & MAT SET.
6089 061170  012703  110000                     MOV     #BIT15!CHIE,R3  ; EXPECT DYR TO HAVE CROSSHAIR ENABLE SET.
6090 061174                                     BISW3   CSCH.8,#WJSS!BIT5!BIT3!BIT1,@DXR ; FIRST CLEAR THEM.
     061174  012746  042052                     MOV     #WJSS!BIT5!BIT3!BIT1,-(SP)
     061200  053716  065006                     BIS     CSCH.8,(SP)
     061204  012677  121626                     MOV     (SP)+,@DXR
6091 061210  004737  027454                     JSR     PC,WAITF                ; WAIT FOR READY.
6092 061214                                     BISW3   CSCH.8,#WJSS!^O77,@DXR ; THEN SET THEM.
     061214  012746  042077                     MOV     #WJSS!^O77,-(SP)
     061220  053716  065006                     BIS     CSCH.8,(SP)
     061224  012677  121606                     MOV     (SP)+,@DXR
6093 061230  013705  065012                     MOV     CSRETRY,R5              ; @@@R
6094 061234  004737  065404             1$:     JSR     PC,GETJSE               ; GET THE ENABLES INTO R2 AND R4.
6095 061240  020102                             CMP     R1,R2           ; CHECK DXR (X-CURSOR)
6096 061242  001002                             BNE     2$              ; BR IF ERROR
6097 061244  020304                             CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
6098 061246  001412                             BEQ     3$              ; BR IF OK.
6099 061250  005305                     2$:     DEC     R5              ; DECREMENT THE RETRY COUNTER ;@@@R
6100 061252  100370                             BPL     1$              ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
6101 061254                                     HRDERR  JSEWRO,RLXYE    ; REPORT THE ERROR.
6102 061274                             3$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
```

```
6104
6105                                    ; TEST ENABLES WITH GENERAL WRITE/READ FROM DISPLAY FILE.
6106
6107 061276  112737  000000  026106  CSMT8:  MOVB    #0,INTMASK
6108 061304  004737  026114                  JSR     PC,ENAINT       ; ENABLE INTERRUPTS, TO FIND BAD ONES.
6109 061310  013704  065006                  MOV     CSCH.8,R4       ; GET A COPY OF THE CHANNEL #, FOR INSERTING.
6110 061314                                  BISW3   R4,#CUIOFF!CUOFF,CSMF2A ; SET UP TO CLEAR ALL.
     061314  012746  146052                  MOV     #CUIOFF!CUOFF,-(SP)
     061320  050416                          BIS     R4,(SP)
     061322  012637  065054                  MOV     (SP)+,CSMF2A
6111 061326                                  BISW3   R4,#LDJSS,CSMF2B        ; SET UP TO GENERAL WRITE.
     061326  012746  146000                  MOV     #LDJSS,-(SP)
     061332  050416                          BIS     R4,(SP)
     061334  012637  065056                  MOV     (SP)+,CSMF2B
6112 061340                                  BISW3   R4,#CURD,CSMF2C ; SET UP FOR READ.
     061340  012746  146100                  MOV     #CURD,-(SP)
     061344  050416                          BIS     R4,(SP)
     061346  012637  065060                  MOV     (SP)+,CSMF2C
6113 061352                                  BISW3   R4,#LDJSS!^O77,CSMF2D   ; SET UP TO SET ALL.
     061352  012746  146077                  MOV     #LDJSS.^O77,-(SP)
     061356  050416                          BIS     R4,(SP)
     061360  012637  065064                  MOV     (SP)+,CSMF2D
6114
6115 061364                             1$:     BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
6116 061366  012701  100000                  MOV     #BIT15,R1               ; ASSUME JSSIE AND JSMIE WILL BE 0.
6117 061372  010103                          MOV     R1,R3                   ; ASSUME CHIE WILL BE 0.
6118 061374  013705  065056                  MOV     CSMF2B,R5               ; GET A COPY OF THE WRITTEN BITS.
6119 061400  032705  000002                  BIT     #JSWE,R5                ; ENABLED TO WRITE SWITCH ENABLE?
6120 061404  001405                          BEQ     2$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
6121 061406  032705  000001                  BIT     #JSWD,R5                ; WRITE TO 1?
6122 061412  001402                          BEQ     2$                      ; BR IF NO -- LEAVE AT 0.
6123 061414  052701  004000                  BIS     #JSSIES,R1              ; YES -- SET EXPECTED STATUS.
6124 061420                             2$:
6125 061420  032705  000010                  BIT     #JMWE,R5                ; ENABLED TO WRITE MATCH ENABLE?
6126 061424  001405                          BEQ     3$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
6127 061426  032705  000004                  BIT     #JMWD,R5                ; WRITE TO 1?
6128 061432  001402                          BEQ     3$                      ; BR IF NO -- LEAVE AT 0.
6129 061434  052701  010000                  BIS     #JSMIES,R1              ; YES -- SET EXPECTED STATUS.
6130 061440                             3$:
6131 061440  032705  000040                  BIT     #JCWE,R5                ; ENABLED TO WRITE CROSSHAIR ENABLE?
6132 061444  001405                          BEQ     4$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
6133 061446  032705  000020                  BIT     #JCWD,R5                ; WRITE TO 1?
6134 061452  001402                          BEQ     4$                      ; BR IF NO -- LEAVE AT 0.
6135 061454  052703  010000                  BIS     #CHIE,R3                ; YES -- SET EXPECTED STATUS.
6136
6137 061460  013700  065012             4$:     MOV     CSRETRY,R0              ; @@@R
6138
6139 061464                             40$:    DPSTART #CSMF2A         ; EXECUTE CLEAR ALL, THEN GENERAL WRITE.
     061464  012777  065054  121340            MOV     #CSMF2A,@DPC    ; START THE DPU.
     061472  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
6140 061476  004737  065430                    JSR     PC,RDJSE        ; GET DXR, DYR INTO R2, R4.
6141 061502  020102                            CMP     R1,R2           ; CHECK DXR (X-CURSOR)
6142 061504  001002                            BNE     41$             ; BR IF ERROR
6143 061506  020304                            CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
6144 061510  001412                            BEQ     43$             ; BR IF OK.
6145 061512  005300                     41$:   DEC     R0              ; DECREMENT THE RETRY COUNTER ;@@@R
6146 061514  100363                            BPL     40$             ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
```

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

                                                BIT     #JSWE,R5                ; ENABLED TO WRITE SWITCH ENABLE?
                                                BEQ     5$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
                                                BIT     #JSWD,R5                ; WRITE TO 0?
                                                BNE     5$                      ; BR IF NO -- LEAVE AT 1.
                                                BIC     #JSSIES,R1              ; YES -- CLEAR EXPECTED STATUS.

                                        5$:
                                                BIT     #JMWE,R5                ; ENABLED TO WRITE MATCH ENABLE?
                                                BEQ     6$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
                                                BIT     #JMWD,R5                ; WRITE TO 0?
                                                BNE     6$                      ; BR IF NO -- LEAVE AT 1.
                                                BIC     #JSMIES,R1              ; YES -- CLEAR EXPECTED STATUS.
                                        6$:
                                                BIT     #JCWE,R5                ; ENABLED TO WRITE CROSSHAIR ENABLE?
                                                BEQ     7$                      ; BR IF NO -- LEAVE EXPECTED ALONE.
                                                BIT     #JCWD,R5                ; WRITE TO 0?
                                                BNE     7$                      ; BR IF NO -- LEAVE AT 1.
                                                BIC     #CHIE,R3                ; YES -- CLEAR EXPECTED STATUS.
                                                MOV     CSRETRY,R0              ; @@@R
                                        7$:
                                                DPCONT          ; RESTART, TO SET ALL, THEN DO GENERAL WRITE.
                                                BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
                                                JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
                                                JSR     PC,WAITF                ; WAIT FOR DONE.
                                                JSR     PC,RDJSE                ; GET DXR, DYR INTO R2, R4.
                                                CMP     R1,R2           ; CHECK DXR (X-CURSOR)
                                                BNE     72$             ; BR IF ERROR
                                                CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
                                                BEQ     73$             ; BR IF OK.
                                        72$:    DEC     R0              ; DECREMENT THE RETRY COUNTER ;@@@R
                                                BPL     7$              ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
                                                HRDERR  CESWE,PLXYE     ; REPORT THE ERROR.

                                        73$:    ENDSEG          ;>>>>>>>>>>>>END SEGMENT
                                                INC     CSMF2B          ; INCREMENT THE LDJSS WRITE DATA.
                                                BIT     #^O77,CSMF2B            ; DONE ALL COMBOS?
                                                BEQ     8$                      ; BR TO EXIT IF YES.
                                                JMP     1$                      ; NO, SO LOOP BACK FOR ANOTHER ROUND.
                                        8$:
```

```
                                        ; CHECK THAT WRITING JS AND JM ENABLES OF OTHER CHANNELS CLEARS ENABLES OF
                                        ; THE CHANNEL UNDER TEST.
                                        ;
061732  004737  026074        CSMT9:  JSR     PC,DPRESET              ; DO SOFT INITIALIZE.
061736  105037  026106                CLRB    INTMASK
061742  013705  065006                MOV     CSCH.8,R5               ; GET CHANNEL # IN R5<9:8>
                                        ;
061746  062705  000400        1$:     ADD     #BIT8,R5                ; INCREMENT TARGET CHANNEL
061752  042705  176377                BIC     #^C<BIT9!BIT8>,R5       ; MASK THE CHANNEL FIELD.
061756  020537  065006                CMP     R5,CSCH.8               ; BACK TO THE CHANNEL UNDER TEST?
061762  001451                        BEQ     2$                      ; BR = YES. DONE
                                        ;
061764                                  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
061766                                  BISW3   CSCH.8,#WRTJSS!JCWE.JMWE.JMWD.JSWE.JSWD,@DXR    ; SET ENABLES
061766  012746  042057                MOV     #WRTJSS!JCWE.JMWE.JMWD!JSWE!JSWD,-(SP)
061772  053716  065006                BIS     CSCH.8,(SP)
061776  012677  121034                MOV     (SP)+,@DXR
062002  004737  027454                JSR     PC,WAITF
062006                                  BISW3   R5,#WRTJSS!JMWE!JMWD.JSWE!JSWD,@DXR
062006  012746  042017                MOV     #WRTJSS!JMWE!JMWD!JSWE.JSWD,-(SP)
062012  050516                        BIS     R5,(SP)
062014  012677  121016                MOV     (SP)+,@DXR
062020  004737  027454                JSR     PC,WAITF
062024  012701  100000                MOV     #BIT15,R1       ; EXPECT ENABLES TO BE CLEAR.
062030  012703  100000                MOV     #BIT15,R3
062034  013700  065012                MOV     CSRETRY,R0      ; @@@R
062040  004737  065404        10$:    JSR     PC,GETJSE       ; LOAD R2 & R4 W/ DXR, DYR.
062044  020102                        CMP     R1,R2           ; CHECK DXR (X-CURSOR)
062046  001002                        BNE     12$             ; BR IF ERROR
062050  020304                        CMP     R3,R4           ; CHECK DYR (Y-CURSOR)
062052  001412                        BEQ     13$             ; BR IF OK.
062054  005300                12$:    DEC     R0              ; DECREMENT THE RETRY COUNTER ;@@@R
062056  100370                        BPL     10$             ; GO TRY AGAIN IF NOT NEGATIVE ;@@@R
062060                                  HRDERR  JENOCL,RLXYE    ; REPORT THE ERROR.
                                        ;
062100                          13$:    ENDSEG          ;>>>>>>>>>>>>END SEGMENT
062102  000137  061746                JMP     1$                      ; LOOP BACK TO TRY ANOTHER TARGET.
                                        ;
062106                          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.

062106                                  CSMT10: BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
062110  004737  026176                          JSR     PC,DSBINT               ; DISABLE INTERRUPTS.
062114                                          INITDP                          ; INITIALIZE THE DPU.
062114  004737  025402                          JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
062120                                          BISW3   CSCH.8,#WJSS!JSWE.0,@DXR  ; CLEAR SWITCH ENABLE.
062120  012746  042002                          MOV     #WJSS!JSWE.0,-(SP)
062124  053716  065006                          BIS     CSCH.8,(SP)
062130  012677  120702                          MOV     (SP)+,@DXR
062134  004737  027454                          JSR     PC,WAITF
062140                                          BISW3   CSCH.8,#WECC!SWCHON,@DYR  ; SET THE SWITCH.
062140  012746  160002                          MOV     #WECC!SWCHON,-(SP)
062144  053716  065006                          BIS     CSCH.8,(SP)
062150  012677  120664                          MOV     (SP)+,@DYR
062154  004737  027454                          JSR     PC,WAITF
062160  004737  023422                          JSR     PC,IIRCHK               ; CHECK INTERNAL REGS FOR INIT STATE.
062164                                  1$:     BISW3   CSCH.8,#WECC,@DYR       ;TURN OFF THE SWITCH
062164  012746  160000                          MOV     #WECC,-(SP)
062170  053716  065006                          BIS     CSCH.8,(SP)
062174  012677  120640                          MOV     (SP)+,@DYR
062200  004737  027454                          JSR     PC,WAITF
062204  012777  040000  120624                  MOV     #GTJSSW,@DXR            ;DISMISS INTR REQ, IF ANY
062212  004737  027454                          JSR     PC,WAITF
062216                                          WTDSRA  #0,FLG                  ;CLEAR PENDING INTERRUPT
062216  012777  000012  120610                  MOV     #0&^C17.SETFLG,@DSR
062224                                          ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
```

```
6255
6256
6257
6258 062226
6259 062230  004737  026176
6260 062234
     062234  004737  025402
6261 062240
     062240  012746  160000
     062244  053716  065006
     062250  012677  120564
6262 062254  004737  027454
6263 062260
     062260  012746  042003
     062264  053716  065006
     062270  012677  120542
6264 062274  004737  027454
6265 062300  004737  023422
6266 062304
     062304  004737  025402
6267 062310

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

CSMT11: BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
        JSR     PC,DSBINT                       ; DISABLE INTERRUPTS.
        INITDP                                  ; INITIALIZE THE DPU.
        JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
        BISW3   CSCH.8,#WECC!0,@DYR  ; CLEAR THE SWITCH.
        MOV     #WECC.0,-(SP)
        BIS     CSCH.8,(SP)
        MOV     (SP)+,@DYR
        JSR     PC,WAITF
        BISW3   CSCH.8,#WJSS!JSWE!JSWD,@DXR  ; SET SWITCH ENABLE.
        MOV     #WJSS!JSWE!JSWD,-(SP)
        BIS     CSCH.8,(SP)
        MOV     (SP)+,@DXR
        JSR     PC,WAITF
        JSR     PC,IIRCHK                       ; CHECK INTERNAL REGS FOR INIT STATE.
1$:     INITDP
        JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
        ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
```

```
                                        ;(3) SET SWITCH AND ENABLE, SEE PENDING INTERRUPT

                                        CSMT12: BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
                                                INITDP
                                                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
                                                JSR     PC,DSBINT
                                                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     #CHAR!JSLCKO!PINTRB!PVEC2,R1    ; EXPECT FLAGS TO SHOW PENDING.
                                                RDDSRA  FLG,R2                  ; READ THE FLAGS REGISTER INTO R2.
                                                MOV     #SELFLG,@DSR    ; READ FLG INTO R2 .
                                                MOV     @DSR,R2
                                                IFERROR NOSWPI,EXPREC
                                                CMP     R1,R2           ; OK?
                                                BEQ     64$             ; YES.
                                                HRDERR  NOSWPI,EXPREC,  ; NO.
                                        64$:
                                        1$:
                                                BISW3   CSCH.8,#WECC,@DYR       ; TURN OFF THE SWITCH.
                                                MOV     #WECC,-(SP)
                                                BIS     CSCH.8,(SP)
                                                MOV     (SP)+,@DYR
                                                JSR     PC,WAITF
                                                BISW3   CSCH.8,#WJSS!JSWE!0,@DXR
                                                MOV     #WJSS!JSWE!0,-(SP)
                                                BIS     CSCH.8,(SP)
                                                MOV     (SP)+,@DXR
                                                JSR     PC,WAITF
                                                MOV     #GTJSSW,@DXR            ;DISMISS INTR REQ, IF ANY
                                                JSR     PC,WAITF

                                                WTDSRA  #0,FLG                  ; CLEAR THE FLAGS AND INTERRUPT.
                                                MOV     #0&^C17!SETFLG,@DSR
                                                JSR     PC,IIRCHK               ; CHECK THE INTERNAL REGISTERS
                                        2$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
                                                JSR     PC,DPRESET              ; DO SOFT INITIALIZE.
```

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

                                        CSMT13: BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
                                                INITDP
062522  004737  025402                          JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
062526  004737  026114                          JSR     PC,ENAINT
                                                INTSET  JSS
062532  012737  177610  026106                  MOV     #^O177400!IOKCKIN!IOKJSS,INTMASK        ; PRIME FLAG, EXPECT JSS
                                                BISW3   CSCH.8,#WJSS!JSWE!JSWD,@DXR     ; ENABLE SWITCH INTERRUPT.
062540  012746  042003                          MOV     #WJSS!JSWE!JSWD,-(SP)
062544  053716  065006                          BIS     CSCH.8,(SP)
062550  012677  120262                          MOV     (SP)+,@DXR
062554  004737  027454                          JSR     PC,WAITF
                                                BISW3   CSCH.8,#WECC!SWCHON,@D[illegible]        ; TURN THE SWITCH ON.
062560  012746  160002                          MOV     #WECC!SWCHON,-(SP)
062564  053716  065006                          BIS     CSCH.8,(SP)
062570  012677  120244                          MOV     (SP)+,@DYR
062574  004737  027454                          JSR     PC,WAITF
062600  012702  000100                          MOV     #100,R2
062604  105737  026107              10$:        TSTB    INTFLAG         ; DID WE INTERRUPT?
062610  100002                              BPL     11$                     ; BR IF YES.
062612  005302                              DEC     R2
062614  003373                              BGT     10$
062616  105737  026107              11$:        TSTB    INTFLAG         ; DID WE GET GOOD INTERRUPT?
062622  001414                              BEQ     2$              ; BR IF YES
062624  012701  062604                      MOV     #10$,R1         ; GET CPU PC.
062630                                      HRDERR  NOSWI,INTERR
062650  000137  062654                      JMP     3$                      ; MIGHT AS WELL QUIT.

062654                              2$:

062654                              3$:     ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
062656  004737  026074                      JSR     PC,DPRESET      ; DO SOFT INIT.
```

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

                        CSMT14: BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
                                INITDP
                                JSR     PC,DPINIT       ; GO DO SOFT INIT, CHECK THE STATE.
                                JSR     PC,ENAINT               ; ENABLE INTERRUPTS.
                                BISW3   CSCH.6,#LDCP,CSMF3A     ; SET UP DISPLAY FILE TO LOAD CURSOR,
                                MOV     #LDCP,-(SP)
                                BIS     CSCH.6,(SP)
                                MOV     (SP)+,CSMF3A
                                BISW3   CSCH.8,#CUIS,CSMF3B     ; ... AND SET SWITCH ENABLE
                                MOV     #CUIS,-(SP)
                                BIS     CSCH.8,(SP)
                                MOV     (SP)+,CSMF3B
                                BISW3   CSCH.6,#LDECC!SWCHON,CSMF3C ; ... AND TURN ON SWITCH.
                                MOV     #LDECC!SWCHON,-(SP)
                                BIS     CSCH.6,(SP)
                                MOV     (SP)+,CSMF3C
 062740  112737  000377  026107 MOVB    #^O377,INTFLAG          ; PRIME THE INTERRUPT FLAG.
                                DPSTART #CSMF3A         ; START THE DPU.
 062746  012777  065072  120056 MOV     #CSMF3A,@DPC    ; START THE DPU.
 062754  004737  027454         JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 062760  004737  027454         JSR     PC,WAITF                ; WAIT FOR STOP.
 062764  017702  120042         MOV     @DPC,R2         ; SAVE THE DPC.
 062770  105737  026107         TSTB    INTFLAG         ; TEST FOR GOOD INTERRUPT.
 062774  001412                 BEQ     1$                      ; BR IF GOOD.
 062776  012701  062760         MOV     #.-16,R1
 063002                         HRDERR  NOSWI,INTERR

                         1$:    CMP     R2,#CSMF3D              ; DID THE DPC HIT THE STOPN?
 063022  020227  065114         BLO     2$                      ; BR IF NO -- OK
 063026  103412                 MOV     #CSMF3C+4,R1            ; EXPECT A DPC PRIOR TO STOPN.
 063030  012701  065106         HRDERR  SWNOHLT,EXPREC

                         2$:    BISW3   #BIT15,CSMF3A+2,R1      ; EXPECT DXR TO HAVE X CURSOR.
 063054  013701  065074         MOV     CSMF3A+2,R1
 063060  052701  100000         BIS     #BIT15,R1
                                BISW3   #BIT15,CSMF3A+4,R3      ; EXPECT DYR TO HAVE Y CURSOR.
 063064  013703  065076         MOV     CSMF3A+4,R3
 063070  052703  100000         BIS     #BIT15,R3
 063074  005737  065012         TST     CSRETRY
 063100  001416                 BEQ     3$
 063102  004737  065362         JSR     PC,RDDXY                ; GET MASKED DXR, DYR INTO R2,R4.
                                IFERRX2 SWNOCU,RLXYE
 063106  020102                 CMP     R1,R2           ; 1ST ITEM OK?
 063110  001002                 BNE     65$             ; NO.
 063112  020304                 CMP     R3,R4           ; 2ND ITEM OK?
 063114  001410                 BEQ     64$             ; YES.
 063116                  65$:   HRDERR  SWNOCU,RLXYE,   ; NO.
 063136                  64$:

                         3$:    JSR     PC,DSBINT
                                ENDSEG          ;>>>>>>>>>>>>>END SEGMENT
 063144  004737  026074         JSR     PC,DPRESET              ; DO SOFT INIT.
```

```
                                        ; VERIFY THE CURSOR MATCH OPERATION.
                                        ;
063150  004737  026074          CSMT15: JSR     PC,DPRESET              ; DO SOFT INIT.
063154                                  BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
063156  112737  000000  026106          MOVB    #0,INTMASK              ; DON'T EXPECT INTERRUPTS
063164                                  BISW3   CSCH.8,#CUIM!CUO*,CSMF4B        ; INSERT CHANNEL # IN FILE.
063164  012746  146076                  MOV     #CUIM!CUON,-(SP)
063170  053716  065006                  BIS     CSCH.8,(SP)
063174  012637  065162                  MOV     (SP)+,CSMF4B
063200                                  BISW3   CSCH.6,#LDCP,CSMF4LC
063200  012746  175000                  MOV     #LDCP,-(SP)
063204  053716  065004                  BIS     CSCH.6,(SP)
063210  012637  065166                  MOV     (SP)+,CSMF4LC
063214                                  BISW3   CSCH.6,#LDECC!SWCHON,CSMF4C
063214  012746  175402                  MOV     #LDECC!SWCHON,-(SP)
063220  053716  065004                  BIS     CSCH.6,(SP)
063224  012637  065156                  MOV     (SP)+,CSMF4C
063230                                  BISW3   CSCH.8,#CURD!JSWE!JMWE!JCWE,CSMF4J
063230  012746  146152                  MOV     #CURD!JSWE!JMWE!JCWE,-(SP)
063234  053716  065006                  BIS     CSCH.8,(SP)
063240  012637  065210                  MOV     (SP)+,CSMF4J
063244  012737  000002  065264          MOV     #2,CSMF4F+2                     ; ASSUME NON-INTERLACED POSITION
063252  012737  020002  065302          MOV     #M!2,CSMF4G+2
063260  012737  020002  065320          MOV     #M!2,CSMF4H+2
063266  004737  026176                  JSR     PC,DSBINT
063272  112737  000202  026106          MOVB    #IOKCKIN!IOKJSM,INTMASK ; SAY MATCH INTR OK, WE WILL CHECK.
063300  032737  000001  027700          BIT     #BIT0,LOOPK     ; ARE WE IN AN ODD-NUMBERED ITERATION?
063306  001406                          BEQ     11$             ; BR IF NO.
063310                                  DPSTART #CSMF4V         ; YES --START THE SETUP FILE FOR VISUAL.
063310  012777  065116  117514          MOV     #CSMF4V,@DPC    ; START THE DPU.
063316  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
063322  000405                          BR      1$
063324                          11$:    DPSTART #CSMF4U         ; START TO PUT MEMORIES IN WRITE-ONLY MODE.
063324  012777  065132  117500          MOV     #CSMF4U,@DPC    ; START THE DPU.
063332  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
063336                          1$:     ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
063340  004737  026114                  JSR     PC,ENAINT               ; ENABLE INTERRUPTS.
063344  012737  064670  064666          MOV     #CSMPTB,CSMPY           ; CSMPY = TABLE POINTER FOR Y POSITIONS
063352  012737  064670  064664  2$:     MOV     #CSMPTB,CSMPX           ; CSMPX = TABLE POINTER FOR X POSITIONS
063360  017737  001302  065172          MOV     @CSMPY,CSMF4Y           ; SETUP Y CURSOR POSITION.
063366  017737  001272  065170  3$:     MOV     @CSMPX,CSMF4X           ; SETUP X CURSOR POSITION.
063374  005737  003230                  TST     INTLAC                  ; ARE WE IN INTERLACED MODE?
063400  001027                          BNE     4$                      ; BR IF YES -- POSITIONING IS OK.
063402  032737  000002  065172          BIT     #BIT1,CSMF4Y            ; NO -- ARE WE ON AN ODD LINE?
063410  001012                          BNE     30$                     ; BR IF YES -- CAN WRITE ABOVE, NOT BELOW
063412  012737  000004  065264          MOV     #4,CSMF4F+2             ; EVEN LINE -- SETUP FOR Y+4.
063420  012737  020004  065302          MOV     #M.4,CSMF4G+2           ; ... AND BACK DOWN BY 4
063426  012737  020002  065320          MOV     #M!2,CSMF4H+2           ; ... AND BELOW Y BY 2.
063434  000411                          BR      4$
063436  012737  000002  065264  30$:    MOV     #2,CSMF4F+2             ; ODD LINE -- SETUP FOR Y+2,
063444  012737  020002  065302          MOV     #M!2,CSMF4G+2           ; ... AND BACK DOWN TO Y,
063452  012737  020004  065320          MOV     #M.4,CSMF4H+2           ; ... BUT DOWN TO Y-4.
                                        ;
063460                          4$:     BGNSEG          ;<<<<<<<<<<<<<BEGIN SEGMENT
063462  112737  000200  026106          MOVB    #IOKCKIN,INTMASK                ; SAY WE'LL CHECK UNEXP. INTR.
```

```
 6397 063470  013705  065012                    MOV     CSRETRY,R5
 6398 063474  013701  065170                    MOV     CSMF4X,R1
 6399 063500  013703  065172                    MOV     CSMF4Y,R3
 6400 063504  052701  100000                    BIS     #BIT15,R1
 6401 063510  052703  100000                    BIS     #BIT15,R3
 6402 063514                                    ⊟ISW3   CSCH.8,#WECC,@DYR       ; CLEAR THE SWITCH
      063514  012746  160000                    MOV     #WECC,-(SP)
      063520  053716  065006                    BIS     CSCH.8,(SP)
      063524  012677  117310                    MOV     (SP)+,@DYR
 6403 063530  004737  027454                    JSR     PC,WAITF
 6404 063534  112737  000377  026107  43$:     MOVB    #^0377,INTFLAG          ; PRIME THE INTERRUPT FLAG
 6405 063542                                    DPSTART #CSMF4D
      063542  012777  065176  117262            MOV     #CSMF4D,@DPC    ; START THE DPU.
      063550  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 6406 063554  017702  117252                    MOV     @DPC,R2
 6407 063560  105737  026107                    TSTB    INTFLAG         ; ANY INTERRUPTS?
 6408 063564  002414                            BLT     403$
 6409 063566  010146                            MOV     R1,-(SP)
 6410 063570  012701  063552                    MOV     #.-14.,R1
 6411 063574                                    HRDERR  IFAULT,INTERR
 6412 063614  012601                            MOV     (SP)+,R1
 6413 063616                          403$:
 6414 063616  004737  065362                    JSR     PC,RDDXY
 6415 063622  020102                            CMP     R1,R2
 6416 063624  001002                            BNE     44$
 6417 063626  020304                            CMP     R3,R4
 6418 063630  001414                            BEQ     45$
 6419 063632  005305                  44$:      DEC     R5
 6420 063634  100337                            BPL     43$
 6421 063636                                    HRDERR  CRWRE,RLXYE
 6422 063656  000137  064552                    JMP     8$      ; GO TRY THE NEXT COORDINATES.
 6423 063662                          45$:
 6424 063662  112737  000202  026106            MOVB    #IOKCKIN!IOKJSM,INTMASK ; SAY WE'LL CHECK FOR MATCH
 6425 063670  112737  000377  026107            MOVB    #^0377,INTFLAG          ; PRIME THE INTERRUPT FLAG.
 6426 063676                                    DPCONT
      063676  052777  000001  117126            BIS     #BIT0,@DPC      ; CONTINUE THE DPU.
      063704  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
 6427 063710  012701  000100                    MOV     #100,R1                 ; SETUP A WATCHDOG COUNTER.
 6428 063714  105737  026107          40$:      TSTB    INTFLAG         ; DID WE GET AN INTERRUPT?
 6429 063720  100002                            BPL     41$                     ; BR IF YES.
 6430 063722  005301                            DEC     R1
 6431 063724  003373                            BGT     40$
 6432 063726  017702  117100          41$:      MOV     @DPC,R2         ; GET DPC FOR ERROR.
 6433 063732  017737  117100  003170            MOV     @DXR,SDXR
 6434 063740  017737  117074  003172            MOV     @DYR,SDYR
 6435 063746  105737  026107                    TSTB    INTFLAG
 6436 063752  001417                            BEQ     5$                      ; WAS IT A GOOD INTERRUPT?
 6437 063754  002434                            BLT     50$                     ; BR IF NO INTERRUPT OCCURRED.
 6438 063756  010146                            MOV     R1,-(SP)
 6439 063760  012701  063714                    MOV     #40$,R1
 6440 063764                                    HRDERR  NOCMI,INTERR
 6441 064004  012601                            MOV     (SP)+,R1
 6442 064006  000137  064046                    JMP     50$
 6443
 6444 064012                          5$:
 6445 064012  027727  117014  065234            CMP     @DPC,#CSMF4K+2  ;DID WE GET TO THE STOPN (NO MATCH)?
 6446 064020  001040                            BNE     51$                     ; 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,@DXP            ; 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$:    HRDERR  MATPOS,EXPRC2,  ; NO.
                                68$:

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

```
        064340  001410                          BEQ     7$                      ; YES.
        064342                                  HRDERR  MATFLE,FLGER,           ; NO.
        064362                          7$:
  6483  064362  112737  000377  026107  7$:     MOVB    #^0377,INTFLAG                  ; PRIME THE INTERRUPT FLAG.
  6484  064370                                  DPSTART #CSMF4E                 ; START AGAIN TO SEE IF WE GET BAD MATCH
        064370  012777  065234  116434          MOV     #CSMF4E,@DPC            ; START THE DPU.
        064376  004737  027454                  JSR     PC,WAITF                ; WAIT FOR DISPLAY STOP.
  6485  064402  004737  027454                  JSR     PC,WAITF
  6486  064406  105737  026107                  TSTB    INTFLAG                 ; DID WE GET AN INTERRUPT?
  6487  064412  002421                          BLT     17$                             ; BR IF NO
  6488  064414  001446                          BEQ     18$                             ; BR IF GOT ''GOOD'' INTERRUPT.
  6489  064416  017702  116410                  MOV     @DPC,R2
  6490  064422  010146                          MOV     R1,-(SP)
  6491  064424  012701  064404                  MOV     #.-16.,R1
  6492  064430                                  HRDERR  MATUI,INTERR
  6493  064450  012601                          MOV     (SP)+,R1
  6494  064452  000137  064552                  JMP     8$
  6495
  6496  064456                          17$:
  6497  064456  012701  065334                  MOV     #CSMF4I+2,R1                    ; GET EXPECT STOP DPC.
  6498  064462  012703  173200                  MOV     #STOPN.GOLD,R3                  ; DSR SHOULD HAVE STOP + GOLD COLOR.
  6499  064466  017702  116340                  MOV     @DPC,R2                 ; GET THE DPC.
  6500  064472  017704  116336                  MOV     @DSR,R4                 ; GET THE DSR
  6501  064476                                  IFERRX2 MENDPCS,EXPRC2
        064476  020102                          CMP     R1,R2                   ; 1ST ITEM OK?
        064500  001002                          BNE     71$                     ; NO.
        064502  020304                          CMP     R3,R4                   ; 2ND ITEM OK?
        064504  001410                          BEQ     70$                     ; YES.
        064506                          71$:    HRDERR  MENDPCS,EXPRC2,         ; NO.
        064526                          70$:
  6502  064526  000137  064552                  JMP     8$
  6503
  6504  064532                          18$:    HRDERR  MATUI,PNTCOOR
  6505
  6506  064552                          8$:     ENDSEG          ;>>>>>>>>>>>>END SEGMENT
  6507  064554  062737  000002  064664          ADD     #2,CSMPX                        ; BUMP POINTER TO X DATA.
  6508  064562  023727  064664  065000          CMP     CSMPX,#CSMPTE                   ; IS X POINTER AT END OF TABLE?
  6509  064570  103002                          BHIS    9$                              ; BR IF YES -- GO DO NEXT Y.
  6510  064572  000137  063366                  JMP     3$                              ; LOOP BACK.
  6511
  6512  064576  062737  000002  064666  9$:     ADD     #2,CSMPY
  6513  064604  023727  064666  065000          CMP     CSMPY,#CSMPTE                   ; IS Y POINTER AT END OF TABLE?
  6514  064612  103002                          BHIS    10$                             ; BR IF YES -- WE'RE DONE.
  6515  064614  000137  063352                  JMP     2$                              ; NO -- GO DO NEXT Y.
  6516  0o4620                          10$:
  6517
  6518  064620                          CSMNEXT:
  6519  064620  005237  065002                  INC     CSCHAN                          ; INCREMENT TO NEXT CHANNEL
  6520  064624  023727  065002  000004          CMP     CSCHAN,#4                       ; DID WE DO ALL AVAILABLE CHANNELS?
  6521  064632  002002                          BGE     1$                              ; BR IF YES -- GO TO EXIT.
  6522  064634  000137  060052                  JMP     CSMLOOP                         ; NO. LOOP BACK TO TEST NEXT CHANNEL.
  6523  064640                          1$:
  6524  064640                          CSMEND: COMEND  CSMITR
        064640  004737  027646                  JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
        064644  103401                          BCS     64$
        064646  000402                          BR      65$
```

```
        064650  000137  060046          64$:    JMP     CSMITR

        064654  004737  030072          65$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
  6525  064660                          CSMXIT: EXIT    TST
  6526
  6527
  6528
  6529                                  ; TABLE OF MATCH POINTS -- ALL VALUES ARE USED FOR X AND Y:
  6530
  6531  064664  000000                  CSMPX:  .WORD   0
  6532  064666  000000                  CSMPY:  .WORD   0
  6533
  6534  064670  000000  000002  000004  CSMPTB: .WORD   0,BIT1,BIT2,BIT3,BIT4          ; FIRST, FLOATING 1 ON 0'S, BITS 2
  6535  064702  000040  000100  000200          .WORD   BIT5,BIT6,BIT7,BIT8,BIT9        ; ... THRU 9.
  6536          001776                  A1=^01776                       ;ALL 1'S FOR POSITION
  6537  064714  001776  001774  001772          .WORD   A1,A1&^CBIT1,A1&^CBIT2,A1&^CBIT3,A1&^CBIT4      ; THEN, FLOATING 0 ON 1'S
  6538  064726  001736  001676  001576          .WORD   A1&^CBIT5,A1&^CBIT6,A1&^CBIT7,A1&^CBIT8,A1&^CBIT9
  6539  064740  001252  000524  001462          .WORD   1252,524,1462,636       ; ALTERNATING BITS AND PAIRS.
  6540  064750  000006  000016  000036          .WORD   6,16,36,76,176,376
  6541  064764  001770  001760  001740          .WORD   1770,1760,1740,1700,1600,1400
  6542  065000  000000                  CSMPTE: .WORD   0
  6543
  6544  065002  000000                  CSCHAN: .WORD   0
  6545  065004  000000                  CSCH.6: .WORD   0
  6546  065006  000000                  CSCH.8: .WORD   0
  6547  065010  000000                  CSCH.11:        .WORD   0
  6548
  6549  065012  000002                  CSRETRY:        .WORD   2
  6550
  6551                                  ; EXTENDED PRINTOUT FOR SYNC CHANNEL:
  6552  065014                          PRSYCH: PRINTX  #FSYCHAN,CSCHAN
  6553  065040  000207                          RTS     PC
  6554
  6555                                  ;DISPLAY FILES FOR CURSOR/SWITCH/MATCH TEST --
  6556
  6557  065042  175000  000000  000000  CSMF1A: .WORD   LDCP,0,0                ;LOAD CURSOR POSITION REGISTERS
  6558  065050  146100                  CSMF1B: .WORD   CURD                    ;READ THEM BACK
  6559  065052  173000                          .WORD   STOPN
  6560
  6561  065054  146052                  CSMF2A: .WORD   CUIOFF!CUOFF            ; CLEAR ALL ENABLES
  6562  065056  146012                  CSMF2B: .WORD   CUIOFF                  ; DO GENERAL WRITE
  6563  065060  146100  173000          CSMF2C: .WORD   CURD,STOPN              ; READ CURSOR (ENABLES COME TOO), THEN STOP
  6564  065064  146077                  CSMF2D: .WORD   CUON!^077               ; SET ALL ENABLES
  6565  065066  160000  065056                  .WORD   DJMP,CSMF2B             ; THEN GO DO GENERAL WRITE AND READ.
  6566
  6567  065072  175000  001332  002464  CSMF3A: .WORD   LDCP,^01332,^02464       ; LOAD CURSOR WITH SOME DATA
  6568  065100  146013                  CSMF3B: .WORD   CUIS                    ; ENABLE SWITCH INTERRUPT
  6569  065102  175402                  CSMF3C: .WORD   LDECC!SWCHON            ; TURN ON THE SOFT SWITCH
  6570  065104  164000  164000  164000          .WORD   DNOP,DNOP,DNOP,DNOP     ; DO SOME NO-OPS -- SHOULD STOP
  6571  065114  173000                  CSMF3D: .WORD   STOPN                   ; STOP -- SWITCH DIDN'T WORK
  6572
  6573                                  ;DISPLAY FILE FOR CURSOR MATCH TEST:
  6574
  6575  065116  176074  176474  177074  CSMF4V: .WORD   RDWRT!CH0,RDWRT.CH1,RDWRT!CH2,RDWRT.CH3  ; MEMORY SETUP, R/W.
  6576  065126  160000  065142                  .WORD   DJMP,CSMF4A
  6577
  6578  065132  176034  176434  177034  CSMF4U: .WORD   WRT!CH0,WRT.CH1,WRT.CH2,WRT!CH3 ; MEMORY SETUP, WRT-ONLY.
```

```
6579 065142  170140                    CSMF4A: .WORD   CLRMEM           ; CLEAR THE MEMORIES.
6580 065144  160001  065162                    .WORD   DJMS,CSMF4B              ; GO TURN ON CURSOR AND MATCH ENABLE
6581 065150  160001  065156  173000            .WORD   DJMS,CSMF4C,STOPN        ; TURN SWITCH ON, AND STOP
6582 065156  175402  165000            CSMF4C: .WORD   LDECC!SWCHON,DPOP        ; TURN SWITCH ON, AND RETURN.
6583
6584 065162  146076  165000            CSMF4B: .WORD   CUON.CUIM,DPOP           ; ENABLE CURSOR AND MATCH.
6585 065166  175000                    CSMF4LC: .WORD  LDCP             ; LOAD CURSOR POSITION.
6586 065170  000000                    CSMF4X:  .WORD  0                ; X
6587 065172  000000                    CSMF4Y:  .WORD  0                ; Y
6588 065174  165000                            .WORD   DPOP                     ; RETURN.
6589
6590 065176  114000  160001  065170    CSMF4D: .WORD   APNT,DJMS,CSMF4X         ; SET MATCH POSITION
6591 065204  160001  065166             CSMF4J: WORD    DJMS,CSMF4LC             ; LOAD CURSOR POSITION.
6592 065210  146152  173000            CSMF4J. .WORD   CURD.JSWE.JMWE.JCWE,STOPN                ; STOP FOR CHECK.
6593 065214  160001  065162                    .WORD   DJMS,CSMF4B              ; GO TURN ON MATCH ENABLE
6594 065220  160001  065156                    .WORD   DJMS,CSMF4C              ; GO TURN ON SWITCH
6595 065224  113774                            .WORD   LVEC!ALL                 ; SET TO WHITE.
6596 065226  040004  000000                    .WORD   I.4,0                    ; DRAW, SHOULD MATCH
6597 065232  173000                    CSMF4K: .WORD   STOPN            ; ... WITH DPC AT F4K
6598 065234  114000  160001  065170    CSMF4E: .WORD   APNT,DJMS,CSMF4X         ; RESTART. SET POINT
6599 065242  113600                            .WORD   LVEC!YELLOW              ; CHANGE COLOR
6600 065244  000002  000000                    .WORD   2,0                      ; ... THEN MOVE TO X+2, Y.
6601 065250  041776  000000  061776            .WORD   I.MAXX,0,I.M.MAXX,0      ; DRAW OUT TO RIGHT AND BACK.
6602 065260  113300                            .WORD   LVEC!VIOLET
6603 065262  020002  000002            CSMF4F:  .WORD M!2,2                     ; GO TO ABOVE MATCH POINT
6604 065266  040000  001776  040000            .WORD   I.0,MAXY,I.0,M!MAXY      ; ... AND DRAW UP AND BACK.
6605 065276  113500                            .WORD   LVEC.EGGBL
6606 065300  020002  020002            CSMF4G: .WORD   M!2,M.2
6607 065304  061776  000000  041776            .WORD   I!M!MAXX,0,I.MAXX,0      ; DRAW LEFT AND BACK.
6608 065314  113200                            .WORD   LVEC!GOLD
6609 065316  000002  020002            CSMF4H: .WORD   2,M!2
6610 065322  040000  021776  040000            .WORD   I!0,M!MAXY,I.0,MAXY
6611 065332  173000  173000            CSMF4I: .WORD   STOPN,STOPN
6612
6613                                    ;SUBROUTINES:
6614                                    ;GET CURSOR COORDINATES, AND READ MASKED DXR, DYR INTO R2, R4 --
6615 065336  004737  027454            GETCURS: JSR    PC,WAITF                 ; WAIT FOR STOP BIT.
6616 065342                                     BISW3  CSCH.8,#WRTJSS!BIT6,@DXR         ; RETRIEVE CURSOR FROM SELECTED CHAN.
     065342  012746  042100                    MOV    #WRTJSS!BIT6,-(SP)
     065346  053716  065006                    BIS    CSCH.8,(SP)
     065352  012677  115460                    MOV    (SP)+,@DXR
6617 065356  004737  027454                    JSR    PC,WAITF                 ; WAIT FOR STOP.
6618 065362                             RDDXY:  BICW3  #^C<BIT15!^03777>,@DXR,R2  ; DXR TO R2, LEAVE FLAG & COORD.
     065362  017702  115450                    MOV    @DXR,R2
     065366  042702  074000                    BIC    #^C<BIT15!^03777>,R2
6619 065372                                     BICW3  #^C<BIT15!^03777>,@DYR,R4  ; DYR TO R4, LEAVE FLAG & COORD.
     065372  017704  115442                    MOV    @DYR,R4
     065376  042704  074000                    BIC    #^C<BIT15!^03777>,R4
6620 065402  00020?                            RTS    PC
6621
6622                                    ;GET JOYSTICK ENABLES-
6623 065404  004737  027454            GETJSE: JSR     PC,WAITF                 ; WAIT FOR STOP.
6624 065410                                     BISW3  CSCH.8,#WRTJSS!BIT6,@DXR         ; READ CURSOR
     065410  012746  042100                    MOV    #WRTJSS!BIT6,-(SP)
     065414  053716  065006                    BIS    CSCH.8,(SP)
     065420  012677  115412                    MOV    (SP)+,@DXR
6625 065424  004737  027454                    JSR    PC,WAITF
```

```
6626 065430                            RDJSE:  BICW3   #^C<BIT15!JSSIES!JSMIES>,@DXR,R2 ;DXR TO R2, LEAVE ENABLES
     065430  017702  115402                    MOV     @DXR,R2
     065434  042702  063777                    BIC     #^C<BIT15!JSSIES!JSMIES>,R2
6627 065440                                     BICW3   #^C<BIT15!CHIE>,@DYR,R4 ;DYR TO R4.
     065440  017704  115374                    MOV     @DYR,R4
     065444  042704  067777                    BIC     #^C<BIT15!CHIE>,R4
6628 065450  000207                            RTS     PC
6629
6630 065452                                     END.TEST

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

```
6632
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 24   -   GRAPH-HISTOGRAM X
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 10.)

6633 065526  005737  002502                    TST     DPUMOD          ;DPU ONLY ???
6634 065532  001402                            BEQ     GHXTST          ;NO, EXECUTE.
6635 065534                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
6636                                    ;
6637                                    ; PLOT GRAPH X = 1600 TO 200 (DELTA 30), FOR Y = 76 TO 2.
6638                                    ; WHEN DONE, REPEAT IN HISTO MODE, WITH BASE = X-20.
6639                                    ; DO THE SAME FOR CONNECTED-GRAPH AND FILLED HISTOGRAM, BUT AT LOWER
6640                                    ; INTENSITY.  RUN THE MEMORIES IN 'WRITE-1'S-ONLY'' MODE TO SEE STUFF.
6641                                    ; REPEAT EACH INCREMENT 10 TIMES AFTER QUICK PASS.
6642                                    ;
6643 065540  004737  027702            GHXTST: JSR     PC,TSTGO        ;TITLE
6644 065544  012737  066142  022300            MOV     #20$,EXTA       ;SET ERROR EXTENSION ADDRESS.
6645 065552  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
6646 065556  004737  026640                    JSR     PC,CLRRW1       ;INIT IN R/W-1'S-ONLY MODE.
6647 065562  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
6648 065570  012737  123774  066212            MOV     #GHX!ALL,XA-2   ;SET G/H-X MODE WITH MAX INTENSITY
6649 065576  012737  000000  066174            MOV     #0,GHXTYP       ;START WITH GRAPH FUNCTION
6650 065604  013737  066174  066214  8$:       MOV     GHXTYP,XA       ;SETUP THE FUNCTION IN FILE
6651 065612  052737  001600  066214            BIS     #1600,XA        ;... AND SET IN THE X AMPLITUDE
6652 065620  012737  001560  066200            MOV     #1560,GHXT+2    ;INIT BASE = X-20
6653 065626  012737  174176  066202            MOV     #GYI.76,YI      ;INIT Y INCREMENT = 76.
6654 065634  012704  001600                    MOV     #1600,R4        ;INIT EXPECTED X.
6655
6656
6657 065640                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6658
6659 065642  013701  066202                    MOV     YI,R1
6660 065646  042701  177700                    BIC     #^C77,R1        ;SET CURRENT INCR IN R1.
6661 065652  005005                    2$:     CLR     R5              ;INIT EXP Y.
6662 065654  052737  010000  066214            BIS     #GHIINH,XA
6663 065662  012777  066176  115142            MOV     #GHXT,@DPC      ;START NEW INCREMENT.
6664 065670  000402                            SKP2
6665 065672  005277  115134            3$:     INC     @DPC            ;RESUME, PLOT NEXT POINT.
6666 065676  004737  027454                    JSR     PC,WAITF
6667 065702  032737  010000  066214            BIT     #GHIINH,XA
6668 065710  001001                            BNE     13$
6669 065712  060105                            ADD     R1,R5           ;EXP Y - Y + INCR.
6670 065714  027704  115116            13$:    CMP     @DXR,R4         ;CHECK RETURNED X = X AMP...
6671 065720  001003                            BNE     4$
6672 065722  027705  115112                    CMP     @DYR,R5         ;...AND Y = Y + INCR.
6673 065726  001410                            BEQ     5$
6674 065730                            4$:     HRDERR  GHXF,XYERR      ;GHX FAILS.
6675 065750  042737  010000  066214  5$:       BIC     #GHIINH,XA
6676 065756  020527  001776                    CMP     R5,#MAXY        ;DONE THIS INCREMENT ??
6677 065762  103743                            BLO     3$              ;NOT YET, CONTINUE.
6678 065764                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6679 065766                                    LOOPTO  1$              ; REPEAT...
     065766  004737  027646                    JSR     PC,LOOP
```

```
       065772  103722                          BCS     1$

6680
6681
6682 065774  013737  003234  027700            MOV     TEMP2,LOOPK     ;RESET LOOPER.
6683 066002  023727  066202  174102            CMP     YI,#GYI.2
6684 066010  001414                            BEQ     6$              ;BR IF LAST INCR DONE.
6685 066012  162737  000002  066202            SUB     #2,YI           ;OTHERWISE, SET NEXT INCR...
6686 066020  162737  000030  066214            SUB     #30,XA          ;...AMPLITUDE...
6687 066026  162704  000030                    SUB     #30,R4
6688 066032  162737  000030  066200            SUB     #30,GHXT+2      ;...AND BASE LINE.
6689 066040  000677                            BR      1$              ;DO NEXT SET.
6690
6691 066042  005737  066174          6$:       TST     GHXTYP          ;JUST DID GRAPH?
6692 066046  001004                            BNE     30$             ;...BR AHEAD IF NOT.
6693 066050  012737  040000  066174            MOV     #HST,GHXTYP     ;YES. SET HISTO MODE
6694 066056  000652                            BR      8$              ;...AND GO DO IT.
6695 066060  023727  066174  060000  30$:      CMP     GHXTYP,#FHST    ;JUST DID FILLED-HISTO?
6696 066066  001417                            BEQ     10$             ;DONE IF YES.
6697 066070  162737  000400  066212            SUB     #BIT8,XA-2      ;IF NOT, DECREMENT INTENSITY.
6698 066076  023727  066174  040000            CMP     GHXTYP,#HST     ;JUST DID HISTO.?
6699 066104  001004                            BNE     31$             ;...BR AHEAD IF NOT.
6700 066106  012737  020000  066174            MOV     #CGRPH,GHXTYP   ;IF YES, NOW DO CONN-GRAPH
6701 066114  000633                            BR      8$              ;...
6702 066116  012737  060000  066174  31$:      MOV     #FHST,GHXTYP    ;DID C-GRPH, NOW DO F-HIST
6703 066124  000627                            BR      8$              ;...
6704
6705 066126  004737  027514          10$:      JSR     PC,PAUSE1
6706 066132  004737  030072                    JSR     PC,TSTEND
6707 066136                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
6708
6709                                   ;
6710                                   ; ON ERROR, SHOW X AMPLITUDE AND Y INCREMENT.
6711 066142                           ;
6712 066172  000207                   20$:    PRINTX  #GHXYX,XA,YI    ;ERROR EXTENSION.
6713                                           RTS     PC
6714                                  ;
6715                                  ;DISPLAY CODE FOR GRAPH/HISTO X TEST.
6716 066174  000000                  ;
6717 066176  150000  000000          GHXTYP: 0               ;FUNCTION TYPE
6718 066202  174100                  GHXT:   SETHB,0         ;SET BASE LINE.
6719 066204  114000                  YI:     GYI.0           ;SET Y INCR.
6720 066206  000000  000000                  APNT
6721 066212  123774                          0,0
6722 066214  000000                          GHX.ALL
6723 066216  173000                  XA:     0               ;X AMPLITUDE (BIT 14 = HST MODE).
6724 066220  160000  066212                  STOPN
6725                                         DJMP,XA-2
6726 066224
                                      ;
                                      ;       END TEST 24
                                      ;
                                      ;
```

```
6728
                                        ;*****************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 25   -   GRAPH-HISTOGRAM Y
                                        ;*
                                        ;*****************************************************************************

                                        ; (ITERATION COUNT = 10.)

6729 066300  005737  002502                    TST     DPUMOD          ;DPU ONLY ??
6730 066304  001402                            BEQ     GHYTST          ;NO, EXECUTE.
6731 066306                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
6732                                    ;
6733                                    ; PLOT GRAPH Y = 1600 TO 200 (DELTA 30), FOR X = 76 TO 2.
6734                                    ; WHEN DONE, REPEAT IN HISTO MODE, WITH BASE = Y-20.
6735                                    ; DO THE SAME FOR CONNECTED-GRAPH AND FILLED HISTOGRAM, BUT AT LOWER
6736                                    ; INTENSITY.  RUN THE MEMORIES IN 'WRITE-1'S-ONLY'' MODE TO SEE STUFF.
6737                                    ; REPEAT EACH INCREMENT 10 TIMES AFTER QUICK PASS.
6738                                    ;
6739 066312  004737  027702            GHYTST: JSR     PC,TSTGO        ;TITLE
6740 066316  012737  066714  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
6741 066324  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
6742 066330  004737  026640                    JSR     PC,CLRRW1       ;INIT IN R/W-1'S-ONLY MODE.
6743 066334  013737  027700  003234            MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
6744 066342  012737  127774  066764            MOV     #GHY!ALL,YA-2   ;SET G/H-Y MODE WITH MAX INTENSITY
6745 066350  012737  000000  066746            MOV     #0,GHYTYP       ;START WITH GRAPH FUNCTION
6746 066356  013737  066746  066766  8$:       MOV     GHYTYP,YA       ;SETUP THE FUNCTION IN FILE
6747 066364  052737  001600  066766            BIS     #1600,YA        ;... AND SET IN THE Y AMPLITUDE
6748 066372  012737  001560  066752            MOV     #1560,GHYT+2    ;INIT BASE LINE = Y-20.
6749 066400  012737  174176  066754            MOV     #GXI!76,XI      ;INIT X INCREMENT = 76.
6750 066406  012705  001600                    MOV     #1600,R5        ;INIT EXP Y.
6751
6752
6753 066412                            1$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6754
6755 066414  013701  066754                    MOV     XI,R1
6756 066420  042701  177700                    BIC     #^C77,R1        ;SET CURRENT INCR IN R1.
6757 066424  005004                    2$:     CLR     R4              ;INIT EXP X...
6758 066426  052737  010000  066766            BIS     #GHIINH,YA
6759 066434  012777  066750  114370            MOV     #GHYT,@DPC      ;START NEW INCREMENT.
6760 066442  000402                            SKP2
6761 066444  005277  114362            3$:     INC     @DPC            ;RESUME, PLOT NEXT POINT.
6762 066450  004737  027454                    JSR     PC,WAITF
6763 066454  032737  010000  066766            BIT     #GHIINH,YA
6764 066462  001001                            BNE     13$
6765 066464  060104                            ADD     R1,R4           ;EXP X = X + INCR.
6766 066466  027704  114344            13$:    CMP     @DXR,R4         ;CHECK RETURNED X = X + INCR...
6767 066472  001003                            BNE     4$
6768 066474  027705  114340                    CMP     @DYR,R5         ;...AND Y = Y AMPL.
6769 066500  001410                            BEQ     5$
6770 066502                            4$:     HRDERR  GHYF,XYERR      ;GHY FAILS.
6771 066522  042737  010000  066766  5$:       BIC     #GHIINH,YA
6772 066530  020427  001776                    CMP     R4,#MAXX        ;DONE THIS INCREMENT ??
6773 066534  103743                            BLO     3$              ;NOT YET, CONTINUE.
6774 066536                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6775 066540                                    LOOPTO  1$              ; REPEAT...
     066540  004737  027646                    JSR     PC,LOOP
```

```
      066544  103722                          BCS     1$
 6776
 6777
 6778 066546  013737  003234  027700          MOV     TEMP2,LOOPK     ;RESET LOOPER.
 6779 066554  023727  066754  174102          CMP     XI,#GXI!2
 6780 066562  001414                          BEQ     6$              ;BR IF LAST INCR DONE.
 6781 066564  162737  000002  066754          SUB     #2,XI           ;OTHERWISE, SET NEXT INCR...
 6782 066572  162737  000030  066766          SUB     #30,YA          ;...AMPLITUDE...
 6783 066600  162705  000030                  SUB     #30,R5
 6784 066604  162737  000030  066752          SUB     #30,GHYT+2      ;...AND BASE LINE.
 6785 066612  000677                          BR      1$              ;DO NEXT SET.
 6786
 6787 066614  005737  066746          6$:     TST     GHYTYP          ;JUST DID GRAPH?
 6788 066620  001004                          BNE     30$             ;...BR AHEAD IF NOT.
 6789 066622  012737  040000  066746          MOV     #HST,GHYTYP     ;YES. SET HISTO MODE
 6790 066630  000652                          BR      8$              ;...AND GO DO IT.
 6791 066632  023727  066746  060000  30$:    CMP     GHYTYP,#FHST    ;JUST DID FILLED-HISTO?
 6792 066640  001417                          BEQ     10$             ;DONE IF YES.
 6793 066642  162737  000400  066764          SUB     #BIT8,YA-2      ;IF NOT, DECREMENT INTENSITY.
 6794 066650  023727  066746  040000          CMP     GHYTYP,#HST     ;JUST DID HISTO.?
 6795 066656  001004                          BNE     31$             ;...BR AHEAD IF NOT.
 6796 066660  012737  020000  066746          MOV     #CGRPH,GHYTYP   ;IF YES, NOW DO CONN-GRAPH
 6797 066666  000633                          BR      8$              ;...
 6798 066670  012737  060000  066746  31$:    MOV     #FHST,GHYTYP    ;DID C-GRPH, NOW DO F-HIST
 6799 066676  000627                          BR      8$              ;...
 6800
 6801 066700  004737  027514          10$:    JSR     PC,PAUSE1
 6802 066704  004737  030072                  JSR     PC,TSTEND
 6803 066710                                  EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
 6804                                 ;
 6805                                 ; ON ERROR, SHOW Y AMPLITUDE AND X INCREMENT.
 6806                                 ;
 6807 066714                          20$:    PRINTX  #GHXYX,YA,XI    ; ERROR EXTENSION.
 6808 066744  000207                          RTS     PC
 6809                                 ;
 6810                                 ;DISPLAY CODE FOR GRAPH/HISTO Y TEST.
 6811                                 ;
 6812 066746  000000                  GHYTYP: 0               ;FUNCTION TYPE
 6813 066750  150000  000000          GHYT:   SETHB,0         ;SET BASE LINE.
 6814 066754  174100                  XI:     GXI!0           ;SET X INCR.
 6815 066756  114000                          APNT
 6816 066760  000000  000000                  0,0
 6817 066764  127774                          GHY!ALL
 6818 066766  000000                  YA:     0               ;Y AMPLITUDE (BIT 14   HST MODE).
 6819 066770  173000                          STOPN
 6820 066772  160000  066764                  DJMP,YA-2
 6821
 6822 066776                                  END.TEST

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

```
6824
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 26   -   BIT MAP (1)
                                        ;*
                                        ;*************************************************************************

                                        ; (ITERATION COUNT = 10.)

6825 067044  005737  002502                    TST     DPJMOD          ;DPU ONLY ??
6826 067050  001402                            BEQ     BM1TST          ;NO, EXECUTE.
6827 067052                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
6828                                    ;
6829                                    ; EXECUTE BIT MAP MODE 1 WITH 4 AND 8 BIT PIXELS FOR ALL
6830                                    ; POSSIBLE RUN LENGTHS (0 THRU 777).  TEST FINAL X/Y FOR EACH.
6831                                    ; IF VISUAL, DISPLAYS A WEDGE STARTING AT 0,0 WITH EVERY
6832                                    ; OTHER PIXEL (IN X) TURNED OFF.
6833                                    ; REPEAT AFTER QUICK AS BEFORE.
6834                                    ; PIXEL BUFFER SIZE = 256. WORDS (512. 8 BIT PIXELS MAX).
6835                                    ;
6836                                    ;       IN THE NEW DPU, A COUNT OF 0 DOES 512. PIXELS, SO IT'S LAST.
6837 067056  004737  027702            BM1TST: JSR     PC,TSTGO        ; TITLE.
6838 067062  012737  067350  022300            MOV     #20$,EXTA       ; SET ERROR EXTENSION ADDRESS.
6839 067070  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
6840 067074  013737  027700  003234            MOV     LOOPK,TEMP2     ; SAVE A COPY OF THE LOOPER.
6841 067102  012700  000377                    MOV     #377,R0         ; SET PIXEL DATA PATTERN...
6842 067106  004737  027014                    JSR     PC,FILL1        ; ..AND FILL THE BUFFER.
6843                                                                    ; 2 ON, 2 OFF IN 4 BIT MODE.
6844                                                                    ; 1 ON, 1 OFF IN 8 BIT MODE.
6845
6846 067112  012777  000001  113714            MOV     #SELPCS,@DSR    ;SAVE PCSAVE...
6847 067120  017737  113710  003146            MOV     @DSR,IPCSAV     ;
6848 067126  012701  136001                    MOV     #BM14+1,R1      ; 1ST OPCODE W/ COUNT=1.
6849 067132  013737  003240  067412            MOV     FREE,BM1PIX     ; BUFFER ADDRESS.
6850 067140  004737  026650          1$:       JSR     PC,CLRRW        ; INIT IN R/W MODE.
6851 067144  012704  000002                    MOV     #2,R4           ; INIT EXPECTED X...
6852 067150  005005                            CLR     R5              ;...AND Y.
6853 067152  010137  067410          2$:       MOV     R1,BM1PIX-2     ; SET 1ST/NEXT OPCODE...
6854 067156  010537  067406                    MOV     R5,BM1Y         ;...AND Y CO-ORD.
6855
6856 067162                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6857
6858 067164  012777  067374  113640            MOV     #BM1,@DPC       ;START, XCT BIT MAP MODE.
6859 067172  004737  027454                    JSR     PC,WAITF
6860 067176  027704  113634                    CMP     @DXR,R4         ;CHECK DXR - R4...
6861 067202  001003                            BNE     3$
6862 067204  027705  113630                    CMP     @DYR,R5
6863 067210  001410                            BEQ     4$
6864 067212                          3$:       HRDERR  BMF,XYERR       ;BM1 FAILS.
6865 067232                          4$:       ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6866
6867 067234                                    LOOPTO  2$              ; REPEAT...
     067234  004737  027646                    JSR     PC,LOOP
     067240  103744                            BCS     2$
6868
6869 067242  013737  003234  027700            MOV     TEMP2,LOOPK     ;RESET LOOPER.
6870 067250  032701  000777                    BIT     #777,R1         ;DONE 512. PIXELS?
```

```
6871 067254  001416                        BEQ     5$              ;...IF YES, GO SEE IF ALL DONE.
6872 067256  005201                        INC     R1              ; NO, INCREMENT RUN LENGTH.
6873 067260  062704  000002                ADD     #2,R4           ; NO, INCR EXPECTED X...
6874 067264  062705  000002                ADD     #2,R5           ;...AND Y...
6875 067270  032701  000777                BIT     #777,R1         ;COUNT OVERFLOW?
6876 067274  001326                        BNE     2$              ;CONTINUE IF NOT.
6877 067276  020127  137000                CMP     R1,#BM18        ;INTO BM18?
6878 067302  001011                        BNE     6$              ;BR IF NOT.
6879 067304  012701  136000                MOV     #BM14,R1        ;IF YES, DO BM14 W/ COUNT=0
6880 067310  000720                        BR      2$              ;...AND CONTINUE.
6881 067312  020127  137000        5$:     CMP     R1,#BM18        ;HERE IF ACTUAL COUNT-0.
6882 067316  001406                        BEQ     10$             ;DONE IF ON BM18.
6883 067320  012701  137001                MOV     #BM18+1,R1      ;GO TO BM18
6884 067324  000705                        BR      1$              ;...AND CONTUNUE.
6885 067326  012701  137000        6$:     MOV     #BM18+0,R1      ;DO BM1.8 W/ COUNT=0
6886 067332  000707                        BR      2$              ;...AND CONTUNUE.
6887
6888 067334  004737  027514        10$:    JSR     PC,PAUSE1
6889 067340  004737  030072                JSR     PC,TSTEND
6890 067344                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
6891
6892                                ;
6893                                ; ON ERROR, SHOW THE FAILING BIT MAP OPCODE.
6894 067350                        ;
6895 067372  000207           20$:     PRINTX  #BMX,R1         ; EXTENSION SHOWS BAD OPCODE.
6896                                        RTS     PC
6897                                ;
6898                                ; DISPLAY CODE FOR BIT MAP MODE 1.
6899 067374  164000  164000  164000 ;
6900 067402  114000           BM1:     DNOP, DNOP, DNOP        ; *** WRITE MODE (176024) ***
6901 067404  000000                        APNT
6902 067406  000000           BM1X:    0
6903 067410  136000           BM1Y:    0
6904 067412  000000                        BM14!0                  ; BIT MAP MODE...
6905 067414  164000  164000  164000 BM1PIX: 0                      ;...PIXEL BUFFER ADDRESS.
6906 067422  173000                        DNOP, DNOP, DNOP        ; *** READ MODE ( 176040) ***
6907                                       STOPN
6908 067424
                                           END.TEST

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

```
6910
                                        ;*****************************************************************
                                        ;*
                                        ;*      BEGIN TEST 27   -   BIT MAP (0)
                                        ;*
                                        ;*****************************************************************

                                        ; (ITERATION COUNT = 10.)

6911 067472  005737  002502            1$:     TST     DFJMOD          ; DPU ONLY ??
6912 067476  001402                            BEQ     BMOTST          ; NO, EXECUTE.
6913 067500                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
6914                                    ;
6915                                    ; INITIALIZE AND FILL THE BUFFER.
6916                                    ;
6917 067504  004737  027702            BMOTST: JSR     PC,TSTGO        ;TITLE
6918 067510  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
6919 067514  004737  026650                    JSR     PC,CLRRW        ;INIT IN R/W MODE.
6920 067520  012700  000377                    MOV     #377,R0         ; DATA FOR 2 OR 4 PIXELS.
6921 067524  004737  027014                    JSR     PC,FILL1        ; FILL THE BUFFER.
6922 067530  013737  003240  070426            MOV     FREE,BMOP1+12   ; 1ST FRAME(S) USE BUFFER.
6923 067536  005037  070452                    CLR     BMOP2+12        ; 2ND USES ABS 0.
6924                                    ;
6925                                    ; FIRST TEST THE RESOLUTION BIT ON A 32 SQUARE ARRAY.
6926                                    ;
6927 067542  013737  027700  003234    1$:     MOV     LOOPK,TEMP2     ; SAVE A COPY OF THE LOOPER.
6928 067550  012701  134000                    MOV     #BM04,R1        ; MODE 0, 512 RESOLUTION.
6929 067554  010137  070424                    MOV     R1,BMOP1+10
6930 067560  005037  070420                    CLR     BMOP1+4
6931 067564  005037  070422                    CLR     BMOP1+6         ; ORIGIN 0,0.
6932 067570  012704  000100                    MOV     #100,R4
6933 067574  010405                            MOV     R4,R5
6934
6935 067576                            2$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6936 067600  012777  070414  113224            MOV     #BMOP1,@DPC
6937 067606  004737  070320                    JSR     PC,BMOCHK       ; ERROR CHECK.
6938 067612                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6939 067614                                    LOOPTO  2$              ; REPEAT...
     067614  004737  027646                    JSR     PC,LOOP
     067620  103766                            BCS     2$
6940
6941 067622  013737  003234  027700            MOV     TEMP2,LOOPK     ;...THEN RESTORE LOOPER.
6942 067630  012701  134100                    MOV     #BM04!R256,R1   ; SHIFT TO 256 RESOLUTION.
6943 067634  010137  070424                    MOV     R1,BMOP1+10
6944 067640  062737  000040  070420          ADD     #40,BMOP1+4
6945 067646  062737  000040  070422          ADD     #40,BMOP1+6
6946 067654  062704  000140                    ADD     #140,R4
6947 067660  010405                            MOV     R4,R5
6948
6949 067662                            3$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
6950 067664  012777  070414  113140            MOV     #BMOP1,@DPC
6951 067672  004737  070320                    JSR     PC,BMOCHK       ; ERROR CHECK.
6952 067676                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
6953 067700                                    LOOPTO  3$              ; REPEAT...
     067700  004737  027646                    JSR     PC,LOOP
     067704  103766                            BCS     3$
6954
```

```
 6955 067706  013737  003234  027700          MOV     TEMP2,LOOPK     ;...AND RESTORE.
 6956                                ;
 6957                                ; NOW CHECK THE EXPANSION OF A 32 SQUARE ARRAY.
 6958                                ;
 6959 067714  012701  134000        4$:      MOV     #BM04,R1        ; 4 BIT, 32 SQUARE, NO EXP.
 6960 067720  000402                         SKP2
 6961 067722  012701  135000        5$:      MOV     #BM08,R1        ; 8 BIT, 32 SQUARE, NO EXP.
 6962 067726  005037  070420                 CLR     BM0P1+4         ; 1ST BLOCK AT 0,0
 6963 067732  005037  070422                 CLR     BM0P1+6
 6964 067736  012704  000100                 MOV     #100,R4         ; SET EXP X AND Y.
 6965 067742  010405                         MOV     R4,R5
 6966 067744  010137  070424                 MOV     R1,BM0P1+10     ; SET OPCODE.
 6967 067750  004737  026650                 JSR     PC,CLRRW
 6968
 6969 067754                        6$:      BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 6970 067756  012777  070414  113046         MOV     #BM0P1,@DPC
 6971 067764  004737  070320                 JSR     PC,BM0CHK       ; ERROR CHECK.
 6972 067770                                 ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 6973 067772                                 LOOPTO  6$              ; REPEAT...
      067772  004737  027646                 JSR     PC,LOOP
      067776  103766                         BCS     6$
 6974
 6975 070000  013737  003234  027700         MOV     TEMP2,LOOPK
 6976 070006  062737  000040  070420         ADD     #40,BM0P1+4
 6977 070014  062737  000040  070422         ADD     #40,BM0P1+6
 6978 070022  062704  000140                 ADD     #140,R4
 6979 070026  010405                         MOV     R4,R5
 6980 070030  042701  000777                 BIC     #777,R1
 6981 070034  052701  000010                 BIS     #EX2,R1         ; EXPAND X2.
 6982 070040  010137  070424                 MOV     R1,BM0P1+10
 6983
 6984 070044                        7$:      BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 6985 070046  005277  112760                 INC     @DPC
 6986 070052  004737  070320                 JSR     PC,BM0CHK       ; ERROR CHECK.
 6987 070056                                 ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 6988 070060                                 LOOPTO  7$
      070062  004737  027646                 JSR     PC,LOOP
      070064  103767                         BCS     7$
 6989
 6990 070066  013737  003234  027700         MOV     TEMP2,LOOPK
 6991 070074  062737  000100  070420         ADD     #100,BM0P1+4
 6992 070102  062737  000100  070422         ADD     #100,BM0P1+6
 6993 070110  062704  000300                 ADD     #300,R4
 6994 070114  010405                         MOV     R4,R5
 6995 070116  042701  000777                 BIC     #777,R1
 6996 070122  052701  000020                 BIS     #EX4,R1         ; EXPAND X4.
 6997 070126  010137  070424                 MOV     R1,BM0P1+10
 6998
 6999 070132                        8$:      BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
 7000 070134  005277  112672                 INC     @DPC
 7001 070140  004737  070320                 JSR     PC,BM0CHK       ; ERROR CHECK.
 7002 070144                                 ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
 7003 070146                                 LOOPTO  8$              ; REPEAT
      070146  004737  027646                 JSR     PC,LOOP
      070152  103767                         BCS     8$
 7004
 7005 070154  013737  003234  027700         MOV     TEMP2,LOOPK
```

```
7006 070162  020127  135020                    CMP     R1,#BM08.EX4    ; 8 BIT MODE DONE ??
7007 070166  001255                            BNE     5$              ; NO, GO 'ROUND.
7008                                    ;
7009                                    ; NOW TEST THE 64, 128, AND 256 SQUARE SIZES BY DISPLAYING
7010                                    ; THE CONTENTS OF CPU MEMORY STARTING AT LOC 0.
7011                                    ;
7012 070170  012701  134001            MAPMEM: MOV     #BM04!M64,R1    ; 4 BIT, 64 SQ.
7013 070174  012704  000200                    MOV     #200,R4
7014 070200  010405                            MOV     R4,R5           ; EXP X,Y.
7015 070202  010137  070450            1$:     MOV     R1,BMOP2+10     ; SET OPCODE.
7016
7017 070206                                    BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7018 070210  012777  070440  112614            MOV     #BMOP2,@DPC
7019 070216  004737  027454                    JSR     PC,WAITF
7020 070222  103375                            BCC     .-4             ; BIG ONES WILL OUTLAST WAITF TIMER.
7021 070224  004737  070324                    JSR     PC,BMOCHK+4     ; ERROR CHECK 200,200.
7022                                                                   ;             400,400 (128 SQ).
7023                                                                   ;            1000,1000 (256 SQ).
7024 070230                                    ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7025
7026 070232  004737  027522                    JSR     PC,PAUS.5
7027 070236  032701  001000                    BIT     #1000,R1        ; 8 BIT MODE ??
7028 070242  001010                            BNE     2$              ; YES.
7029 070244  052701  001000                    BIS     #1000,R1        ; NO, SHIFT TO 8 BIT MODE.
7030 070250  000754                            BR      1$              ; *** OR NOP ***
7031 070252  023727  070450  135003            CMP     BMOP2+10,#BM08!M256
7032 070260  001401                            BEQ     2$              ; QUIT AT 8 BIT, 256 SQ. ...
7033 070262  000747                            BR      1$              ;... OR WE RUN OUT OF MEMORY.
7034 070264  042701  001000            2$:     BIC     #1000,R1        ; RESET 4 BIT MODE...
7035 070270  005201                            INC     R1              ;...SET NEXT SIZE...
7036 070272  006304                            ASL     R4
7037 070274  006305                            ASL     R5              ;...AND EXP X,Y.
7038 070276  032701  000004                    BIT     #4,R1           ; DONE THRU 256 SQ. ??
7039 070302  001737                            BEQ     1$              ; NO, GO 'ROUND.
7040
7041 070304  004737  027514            10$:    JSR     PC,PAUSE1       ; ALL DONE.
7042 070310  004737  030072                    JSR     PC,TSTEND
7043 070314                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
7044
7045 070320  004737  027454            BMOCHK: JSR     PC,WAITF        ; WAIT FOR DONE...
7046 070324  027704  112506                    CMP     @DXR,R4         ;...AND TEST X...
7047 070330  001003                            BNE     1$
7048 070332  027705  112502                    CMP     @DYR,R5         ;...AND Y.
7049 070336  001413                            BEQ     2$
7050 070340                            1$:
7051 070340  012737  070370  022300            MOV     #3$,EXTA        ; SET ERROR EXTENSION ADDRESS.
7052 070346                                    HRDERR  BMF,XYERR
7053 070366                            2$:
7054 070366  000207                            RTS     PC
7055 070370                            3$:     PRINTX  #BMX,R1         ; EXTENSION SHOWS BAD OPCODE.
7056 070412  000207                            RTS     PC
7057                                    ;
7058                                    ; DISPLAY CODE FOR BMO TESTS.
7059                                    ;
7060 070414  164000            BMOP1:  DNOP
7061 070416  114000  000000  000000            APNT, 0, 0
7062 070424  134000                            BM04!0                  ; 32 SQUARE, NO EXP.
```

```
7063 070426  000000                          0                          ; BUFFER ADDRESS.
7064 070430  164000  173000                  DNOP, STOPN
7065 070434  160000  070416                  DJMP, BMOP1+2
7066
7067 070440  170140                  BMOP2:  CLRMEM
7068 070442  114000  000000  000000          APNT, 0, 0
7069 070450  134000                          BM04!0
7070 070452  000000                          0
7071 070454  164000  173000                  DNOP, STOPN
7072 070460  160000  070440                  DJMP, BMOP2
7073
7074 070464                                  END.TEST

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

```
7076
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 28   -   IMAGE MEMORY CLEAR-SET
                                        ;*
                                        ;*************************************************************************
                                        ; (ITERATION COUNT = 10.)

7077 070546  005737  002502                    TST     DPJMOD          ;DPU ONLY ??
7078 070552  001402                            BEQ     CLRTST          ;NO, EXECUTE.
7079 070554                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
7080                                    ;
7081                                    ; ''CLRMEM'' TO ZERO, EXECUTE PIXEL-READ-BACK, AND TEST DATA.
7082                                    ; ''SETMEM'' TO 1'S, AND REPEAT.
7083                                    ; DISPLAY MODE (INTERLACE/NON-INTERLACE) IS TRANSPARENT.
7084                                    ;
7085                                    ; IF SHORT MODE, CYCLE X FOR Y = 0 AND 2 (ODD VS EVEN),
7086                                    ; THEN CYCLE Y HOLDING X AT 0.
7087                                    ; REPEAT EACH PIXEL READ 10 TIMES AFTER QUICK PASS.
7088                                    ; IF LONG MODE, CYCLE X FOR EVERY Y (512 X 512 ADDRESSES),
7089                                    ; READING EACH PIXEL ONLY ONCE.
7090                                    ;
7091 070560  004737  027702            CLRTST: JSR     PC,TSTGO        ;TITLE
7092 070564  004737  030414                    JSR     PC,CONFIG       ; CONFIGURE VSV SYSTEM.
7093 070570  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
7094 070574                                    BGNSUB
7095 070576  005037  072414                    CLR     IMC.CH
7096 070602  013701  072414            CLRT1:  MOV     IMC.CH,R1
7097 070606  00630.                            ASL     R1
7098 070610  016101  003256                    MOV     MEMTAB(R1),R1
7099 070614  001002                            BNE     .+6             ; BR IF MEMORY ON THIS CHANNEL
7100 070616  000137  071116                    JMP     11$
7101 070622  010137  003072                    MOV     R1,PDF
7102 070626  005101                            COM     R1
7103 070630  010137  003074                    MOV     R1,PDM
7104 070634  013701  072414                    MOV     IMC.CH,R1
7105 070640  000301                            SWAB    R1
7106 070642                                    BISW3   #RDWRT,R1,CLR.ZA        ; INSERT CHANNEL OF MEMORY TO TEST
     070642  010146                            MOV     R1,-(SP)
     070644  052716  176074                    BIS     #RDWRT,(SP)
     070650  012637  071434                    MOV     (SP)+,CLR.ZA
7107 070654  005737  002504                    TST     IMTMOD
7108 070660  001403                            BEQ     .+10            ;SKIP NEXT IF SHORT MODE...
7109 070662  012737  000001  027700           MOV     #1,LOOPK        ;...OTHERWISE, SET ONCE ONLY.
7110 070670  013737  027700  003234           MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
7111 070676  005003                            CLR     R3              ;R3 = PATTERN (0'S FIRST).
7112 070700  012777  071424  112124           MOV     #CLR.Z,@DPC     ;XCT ''CLEAR MEM TO 0''.
7113 070706  000402                            SKP2
7114 070710  005277  112116            1$:     INC     @DPC            ;RESUME, XCT ''SETMEM TO 1''S.
7115 070714  004737  027454                    JSR     PC,WAITF        ;WAIT 'TIL DONE
7116 070720  005001                            CLR     R1              ;INIT X ADDRESS...
7117 070722  005002                            CLR     R2              ;...AND Y ADDRESS.
7118
7119 070724                            2$:
7120 070724  010277  112110                    MOV     R2,@DYR         ;SET Y ADDRESS.
7121 070730  004737  027454                    JSR     PC,WAITF
```

```
7122 070734                                3$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7123 070736  010177  112074                       MOV     R1,@DXR         ;NOW READ-BACK PIXEL DATA...
7124 070742  004737  027454                       JSR     PC,WAITF
7125 070746  004737  023524                       JSR     PC,ERRCHK
7126 070752  017704  112056                       MOV     @DSR,R4         ;...TO R4.
7127 070756  100015                              BPL     10$             ;BR IF DPU HUNG (PRB FAILURE).
7128 070760  043704  003074                       BIC     PDM,R4          ;OK, JUST KEEP THE DATA BITS.
7129 070764  020304                              CMP     R3,R4           ;IS DATA RIGHT ??
7130 070766  001423                              BEQ     4$              ;BR IF WRONG.
7131 070770                                      HRDERR  IMDI,IMDERR     ;DATA INCORRECT.
7132 071010  000412                              BR      4$
7133 071012  004737  026756                10$:  JSR     PC,RELEAS       ;RELEASE ''HUNG'' DPU...
7134 071016                                      HRDERR  PRRHNG          ;...REPORT ERROR AND QUIT.
7135 071036                                4$:   ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7136
7137 071040  013737  003234  027700        5$:   MOV     TEMP2,LOOPK     ;RESET LOOPER.
7138 071046  020227  000004                6$:   CMP     R2,#4           ;IF ''SHORT'' MODE, HOLD X...
7139 071052  103006                              BHIS    8$              ;...AT 0 FOR Y - 4 TO MAX.
7140
7141 071054  020127  001776                7$:   CMP     R1,#MAXX
7142 071060  001403                              BEQ     8$              ;BR IF AT MAX X.
7143 071062  062701  000002                       ADD     #2,R1           ;OTHERWISE, BUMP X...
7144 071066  000722                              BR      3$              ;...AND CONTINUE.
7145 071070  020237  003250                8$:   CMP     R2,YMAX
7146 071074  001404                              BEQ     9$              ;DONE IF AT MAX Y.
7147 071076  005001                              CLR     R1              ;OTHERWISE, RESET X...
7148 071100  062702  000002                       ADD     #2,R2           ;...BUMP Y...
7149 071104  000707                              BR      2$              ;...AND CONTINUE.
7150 071106  005103                        9$:   COM     R3              ;COMPLIMENT DATA PATTERN.
7151 071110  043703  003074                       BIC     PDM,R3
7152 071114  001275                              BNE     1$              ;RESTART 1'S PATTERN.
7153
7154 071116                                11$:
7155 071116  005237  072414                       INC     IMC.CH          ; DO NEXT CHANNEL
7156 071122  023727  072414  000004              CMP     IMC.CH,#4
7157 071130  103624                              BLO     CLRT1           ; LOOP THRU ALL 4 CHANNELS
7158 071132                                      ENDSUB
7159
7160 071134                                CLTNEW: BGNSUB
7161 071136  052737  000006  072440              BIS     #6,IMC.IR               ;LET READ SEE ALL BITS
7162 071144  012737  071454  071452              MOV     #CLMTAB,CLMPTR          ;START AT BEGINNING OF PATTERN TABLE
7163 071152  017700  000274                1$:   MOV     @CLMPTR,R0              ;GET PATTERN TO CLEAR TO
7164 071156  100515                              BMI     8$                      ;NEGATIVE IF DONE ALL
7165 071160  042737  001777  071562              BIC     #1777,CLR.D2            ;CLEAR OUT OLD PIXEL PATTERN
7166 071166  050037  071562                       BIS     R0,CLR.D2               ;INSERT THE NEW ONE
7167 071172  005700                              TST     R0                      ;IS NEW ONE 0?
7168                                       ;      BNE     2$                      ;IF NO, GO DO ''SETMEM'' FUNCTION.
7169 071174  000404                              BR      2$                      ;DO ''SETMEM'' TO 0 UNTIL NEW MICROCODE READY
7170 071176  012777  071546  111626              MOV     #CLR.Z2,@DPC            ;IF YES, DO THE ''CLRMEM''
7171 071204  000403                              BR      3$
7172 071206  012777  071562  111616        2$:   MOV     #CLR.D2,@DPC            ;''CLEAR'' TO THE PATTERN
7173 071214  004737  027454                3$:   JSR     PC,WAITF                ;WAIT FOR DONE
7174
7175 071220  005037  072414                       CLR     IMC.CH                  ;START CHECKING AT CHANNEL 0.
7176 071224  013700  072414                4$:   MOV     IMC.CH,R0               ;GET CHANNEL TO CHECK
7177 071230  006300                              ASL     R0                      ;MAKE A WORD INDEX
7178 071232  016001  003256                       MOV     MEMTAB(R0),R1           ;GET MASK FOR VALID PIXEL BITS
```

```
7179 071236  001445                        BEQ     7$                      ;IF 0, NO CHANNEL HERE. GO TO NEXT.
7180 071240  010137  003072                MOV     R1,PDF                  ;SAVE VALID MASK
7181 071244  005101                        COM     R1                      ;COMPLEMENT IT.
7182 071246  057701  000200                BIS     @CLMPTR,R1              ;INSERT THE PATTERN
7183 071252  006201                        ASR     R1                      ;SHIFT IT DOWN TO BITS <7:0>
7184 071254  006201                        ASR     R1
7185 071256  010146                        MOV     R1,-(SP)                ;PUT PATTERN IN LOW BYTE,
7186 071260  110166  000001                MOVB    R1,1(SP)                ;AND HIGH BYTE.
7187 071264  012601                        MOV     (SP)+,R1                ;PUT IT BACK INTO R1.
7188 071266  012700  000400                MOV     #256.,R0                ;GET COUNT OF NUMBER OF WORDS TO FILL (256.)
7189 071272  013702  003240                MOV     FREE,R2                 ;GET ADDRESS TO PUT TEST DATA.
7190 071276  010122              5$:       MOV     R1,(R2)+                ;FILL THE TEST BUFFER
7191 071300  005300                        DEC     R0
7192 071302  001375                        BNE     5$
7193 071304  013737  003240  072416        MOV     FREE,IMC.TA             ;TELL CHECK ROUTINE WHERE IT IS.
7194 071312  010237  072450                MOV     R2,IMC.RA               ;... AND WHERE TO PUT THE MEM DATA READ.
7195 071316  005037  072436                CLR     IMC.Y                   ;START AT Y = 0.
7196
7197 071322                        6$:     BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7198 071324  004737  071600                JSR     PC,IMCHK                ;READ AND CHECK A LINE OF MEMORY DATA.
7199 071330                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7200
7201 071332  023737  072436  003250        CMP     IMC.Y,YMAX              ;DONE LAST LINE?
7202 071340  001404                        BEQ     7$                      ;MOVE ON TO NEXT CHANNEL IF YES.
7203 071342  062737  000002  072436        ADD     #2,IMC.Y                ;IF NO, BUMP Y & GO FOR MORE.
7204 071350  000764                        BR      6$
7205 071352  005237  072414        7$:     INC     IMC.CH                  ;INCREMENT THE CHANNEL NUMBER.
7206 071356  023727  072414  000004        CMP     IMC.CH,#4               ;DONE LAST CHANNEL?
7207 071364  002717                        BLT     4$                              ;IF NOT, GO FOR MORE.
7208 071366  062737  000002  071452        ADD     #2,CLMPTR               ;IF YES, MOVE ON TO NEXT PATTERN
7209 071374  005737  002504                TST     IMTMOD          ; SHOPT TEST?
7210 071400  001264                        BNE     1$              ; NO.
7211 071402  023727  071452  071470        CMP     CLMPTR,#CLMSHT  ; YES. DONE?
7212 071410  001260                        BNE     1$              ; NO.
7213                                                                ; YES.
7214
7215 071412                        8$:
7216 071412                                ENDSUB
7217 071414  004737  030072                JSR     PC,TSTEND
7218 071420                                EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
7219
7220                               ;CLR.Z: DNOP,CLRMEM,STOPN       ;CLEAR IMAGE MEMORY.
7221 071424  176000  176400  177000  CLR.Z:  PROTEC!CH0,PROTEC!CH1,PROTEC!CH2,PROTEC!CH3
7222 071434  176074                CLR.ZA: RDWRT
7223 071436  116000  170100  173000        APNT!LO,SETMEM,STOPN    ;CLEAR MEMORY THIS WAY UNTIL NEW MICROCODE READY
7224 071444  117774  170100  173000  CLR.D:  APNT!ALL,SETMEM,STOPN   ;SET IMAGE MEMORY TO 1'S.
7225
7226
7227                                       .MACRO  CLP.D   CPAT
7228                                       <CPAT*BIT6>!<-CPAT-1&17*BIT2>
7229                                       .ENDM
7230 071452  071454                CLMPTR: .WORD   CLMTAB
7231
7232 071454  000000                CLMTAB: 0
7233 071456                                CLP.D   17
7234 071460                                CLP.D   10
7235 071462                                CLP.D   4
```

```
 7236 071464                                        CLP.D   2
 7237 071466                                        CLP.D   1
 7238 071470                                CLMSHT: CLP.D   14
 7239 071472                                        CLP.D   11
 7240 071474                                        CLP.D   12
 7241 071476                                        CLP.D   3
 7242 071500                                        CLP.D   5
 7243 071502                                        CLP.D   6
 7244 071504                                        CLP.D   7
 7245 071506  001774                                1774
 7246 071510  177777                                -1
 7247 071512  000000  000000  000000                0,0,0,0,0,0
 7248 071526  000000  000000  000000                0,0,0,0,0,0,0,0
 7249
 7250                                       ;DISPLAY FILES FOR CLEARING MEMORY:
 7251 071546  176074  176474  177074      CLR.Z2: RDWRT!CH0,RDWRT.CH1,RDWRT.CH2,RDWRT.CH3
 7252 071556  170140  173000                      CLRMEM,STOPN
 7253
 7254 071562  117774                      CLR.D2: APNT.ALL
 7255 071564  176074  176474  177074              RDWRT.CH0,RDWRT.CH1,RDWRT.CH2,RDWRT.CH3
 7256 071574  170100  173000                      SETMEM,STOPN
 7257
 7258
 7259                                       ;
 7260                                       ; SUBROUTINE - READ & VERIFY 1 LINE OF IMAGE MEMORY.
 7261                                       ;
 7262                                       ;ENTER WITH:
 7263                                       ;       IMC.CH = MEMORY CHANNEL TO READ (0, 1, 2, OR 3)
 7264                                       ;       IMC.Y  = Y POSITION OF LINE TO BE READ
 7265                                       ;       IMC.RA = ADDRESS OF 512.-BYTE BUFFER TO READ DATA INTO
 7266                                       ;       IMC.TA = ADDRESS OF 512.-BYTE BUFFER TO TEST AGAINST
 7267                                       ;                (DATA MUST INCLUDE 1'S IN NON-EXISTENT BIT POSITIONS TO
 7268                                       ;                 CATCH BAD DRIVERS & DBUS LINES)
 7269                                       ;
 7270 071600  013700  072414          IMCHK:  MOV     IMC.CH,R0               ;GET CHANNEL #
 7271 071604  000300                          SWAB    R0                      ;... INTO HIGH BYTE OF R0.
 7272 071606  042737  001400  072430          BIC     #BIT9!BIT8,IMC.ME       ;CLEAR OLD CHANNEL # FROM ENABLING INSTR.
 7273 071614  050037  072430                  BIS     R0,IMC.ME               ;SET IN THE NEW CHANNEL
 7274 071620  013737  072352  072354          MOV     IMCTRY,IMCRCT
 7275 071626  012777  072420  111176  IMCHKR: MOV     #IMC.SR,@DPC            ;START THE READ.
 7276 071634  004737  027454                  JSR     PC,WAITF                ;... WAIT FOR IT TO FINISH.
 7277 071640  013702  072450                  MOV     IMC.RA,R2
 7278 071644  062702  001000                  ADD     #512.,R2                ;R2 GETS FINAL ADDRESS
 7279 071650  013703  072440                  MOV     IMC.IR,R3               ;R3 GETS WHAT DSR SHOULD BE.
 7280 071654  027702  111152                  CMP     @DPC,R2                 ;IS DPC CORRECT?
 7281 071660  001003                          BNE     1$
 7282 071662  027703  111146                  CMP     @DSR,R3                 ;IS DSR CORRECT?
 7283 071666  001410                          BEQ     2$
 7284 071670                          1$:
 7285 071670                                  HRDERR  IMRF,PCSERR             ;PRINT IF NOT.
 7286 071710  005004                  2$:     CLR     R4                              ;DXR SHOULD BE 0.
 7287 071712  013705  072436                  MOV     IMC.Y,R5
 7288 071716  062705  000002                  ADD     #2,R5                   ;DYR SHOULD BE Y+2
 7289 071722  027704  111110                  CMP     @DXR,R4                 ;TEST DXR
 7290 071726  001003                          BNE     3$
 7291 071730  027705  111104                  CMP     @DYR,R5                 ;TEST DYR
 7292 071734  001410                          BEQ     4$
```

```
7293 071736                                3$:
7294 071736                                        HRDERR  IMRF,XYERR
7295 071756  012701  000000                4$:     MOV     #0,R1           ;R1 = X ADDRESS UNDER TEST.
7296 071762  013704  072416                        MOV     IMC.TA,R4       ;R4 POINTS TO EXPECTED DATA
7297 071766  013705  072450                        MOV     IMC.RA,R5       ;R5 POINTS TO RECEIVED DATA
7298 071772  022425                        5$:     CMP     (R4)+,(R5)+     ;CHECK THE DATA
7299 071774  001006                                BNE     10$
7300 071776  062701  000004                6$:     ADD     #4,R1           ;BUMP THE X ADDRESS
7301 072002  020127  002000                        CMP     R1,#MAXX+2
7302 072006  002771                                BLT     5$
7303 072010  000471                                BR      9$                      ; EXIT WHEN DONE WITH LINE.
7304
7305 072012  005337  072354                10$:    DEC     IMCRCT
7306 072016  100402                                BMI     .+6
7307 072020  000137  071626                        JMP     IMCHKR
7308 072024  010137  072346                        MOV     R1,IMCBDA       ;GET POSITION OF BAD DATA
7309 072030  12 465  177776  177776                CMPB    -2(R4),-2(R5)   ;LOW BYTE BAD?
7310 072036  001440                                BEQ     12$             ;BR IF NOT.
7311 072040  116437  177776  023056                MOVB    -2(R4),GDDAT    ;IF YES, GET THE DATA
7312 072046  116537  177776  023060                MOVB    -2(R5),BADDAT
7313 072054  006337  023056                11$:    ASL     GDDAT           ;SHIFT IT TO LOOK LIKE DBUS DATA
7314 072060  006337  023056                        ASL     GDDAT
7315 072064  006337  023060                        ASL     BADDAT
7316 072070  006337  023060                        ASL     BADDAT
7317 072074  042737  176003  023056                BIC     #^C1774,GDDAT
7318 072102  042737  176003  023060                BIC     #^C1774,BADDAT
7319 072110                                        HRDERR  IMDI,IMCERR
7320 072130  032737  000002  072346                BIT     #2,IMCBDA       ;DONE HIGH BYTE?
7321 072136  001317                                BNE     6$                      ;IF YES, GO BACK TO MAIN LOOP.
7322 072140  062737  000002  072346  12$:    ADD     #2,IMCBDA
7323 072146  126465  177777  177777                CMPB    -1(R4),-1(R5)   ;COMPARE HIGH BYTES
7324 072154  001710                                BEQ     6$
7325 072156  116437  177777  023056                MOVB    -1(R4),GDDAT
7326 072164  116537  177777  023060                MOVB    -1(R5),BADDAT
7327 072172  000730                                BR      11$
7328
7329 072174  000207                        9$:     RTS     PC
7330
7331 072176                                BGNMSG  IMCERR
7332 072176                                        PRINTX  TNAM
7333 072216  013737  072346  072350                MOV     IMCBDA,IMCRAM
7334 072224  006237  072350                        ASR     IMCRAM
7335 072230                                        PRINTX  #IMDIXC,IMCBDA,IMC.Y,PDF,GDDAT,BADDAT
7336 072274                                        PRINTX  #IMDIX2,IMCRAM,IMC.CH
7337 072324                                        PRINTX  #NULCR                  ;PRINT BLANK LINE
7338 072344                                ENDMSG
7339
7340 072346  000000                        IMCBDA: .WORD   0                       ;X POSITION FOR PRINTOUT
7341 072350  000000                        IMCRAM: .WORD   0                       ;SUSPECT RAM
7342 072352  000000                        IMCTRY: .WORD   0
7343 072354  000000                        IMCRCT: .WORD   0
7344
7345 072356     040      040      104      IMRF:   .ASCIZ  /  DMA PIXEL READBACK FAILURE/
7346                                                .EVEN
7347
7348         163000                                 IMR8=163000
7349 072414  000000                        IMC.CH: 0
```

```
7350 072416  000000                        IMC.TA: 0                ;ADDRESS OF TEST BUFFER
7351 072420  176000  176400  177000        IMC.SR: PROTEC!CH0,PROTEC.CH1,PROTEC.CH2,PROTEC!CH3
7352 072430  176074                        IMC.ME: RDWRT            ;ENABLE SELECTED MEMORY
7353 072432  114000  000000                        APNT,0           ;START AT X=0
7354 072436  000000                        IMC.Y:  0                ;Y
7355 072440  163006                        IMC.IR: IMR8!6           ;READ, 2 8-BIT PIXELS/WORD, ALL BITS
7356 072442  001000  000001                        512.,1           ;1 FULL LINE
7357 072446  000400                                256.             ;WORD COUNT = 256.
7358 072450  160000                        IMC.RA: .WORD   160000   ;READ BUFFER
7359 072452  173000                                STOPN            ;(SHOULD STOP BY ITSELF)
7360 072454                                        END.TEST

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

```
7362
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 29   -   IMAGE MEMORY INTERLACE
                                ;*
                                ;*************************************************************************
                                ; (ITERATION COUNT = 10.)

7363 072536  005737  002502            TST     DPJMOD          ; DPU ONLY?
7364 072542  001402                    BEQ     INLAC           ; NO. EXECUTE.
7365 072544                            EXIT    TST     ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>
7366
7367                            ;
7368                            ; CHECK MEMORY INTERLACE / NON- INTERLACE BY WRITING A POINT AT 0,0.
7369                            ; IF MEM IS INTERLACED, THE PIXEL DATA AT 0,2 SHOULD BE (NON-ZERO).
7370                            ; .F MEM IS NON-INTERLACED, THE PIXEL DATA AT 0,2 SHOULD BE (ZERO).
7371                            ;
7372 072550                    INLAC:  COMBEG
       072550  004737  027702            JSR     PC,TSTGO        ; TITLE.
       072554  004737  026074            JSR     PC,DPRESET      ; DO SOFT INIT.
7373 072560  004737  030414            JSR     PC,CONFIG       ; CONFIGURE VSV SYSTEM.
7374 072564  005005                    1$:     CLR     R5              ; INIT CHAN PTR.
7375 072566  004737  026650            JSR     PC,CLRRW        ; CLEAR ALL MEMORIES.
7376
7377 072572                            BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7378 072574  010504                    2$:     MOV     R5,R4           ; GET MEMORY TABLE ENTRY FOR THIS CHAN.
7379 072576  006304                    ASL     R4              ;
7380 072600  016401  003256            MOV     MEMTAB(R4),R1   ; (SAVE IT IN R1 & R3). IS THERE A MEM?
7381 072604  010103                    MOV     R1,R3           ;
7382 072606  001450                    BEQ     4$              ; NO.
7383 072610  010504                    MOV     R5,R4           ; GET CHAN IN BITS 8,9.
7384 072612  000304                    SWAB    R4              ;
7385 072614  052704  176034            BIS     #WRT,R4         ; SET WRITE INST FOR CURRENT CHAN -
7386 072620  010437  072776            MOV     R4,11$          ; - IN DISPLAY FILE.
7387 072624                            DPSTART #10$            ; YES.
       072624  012777  072766  110200    MOV     #10$,@DPC       ; START THE DPU.
       072632  004737  027454            JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
7388 072636  032737  020000  003270    BIT     #BIT13,SYCTAB   ; IS MEM NON-INTERLACED?
7389 072644  001401                    BEQ     3$              ; YES. (R1 = EXPECTED FOR NON-INTERLACED).
7390 072646  005001                    CLR     R1              ; NO. ADJUST R1 TO EXPECT FOR INTERLACED.
7391 072650  012777  000002  110162    3$:     MOV     #2,@DYR         ; PIXEL READ (POINT 0,2) INTO R2.
7392 072656  004737  027454            JSR     PC,WAITF        ;
7393 072662  012777  000000  110144    MOV     #0,@DSR         ;
7394 072670  017702  110140            MOV     @DSR,R2         ;
7395 072674  042702  176003            BIC     #176003,R2      ;
7396 072700  074203                    XOR     R2,R3           ; AND PIXEL READ WITH MEM TAB VAL.
7397 072702  040302                    BIC     R3,R2           ;
7398 072704                            IFERROR INLE,INLEX
       072704  020102                    CMP     R1,R2           ; OK?
       072706  001410                    BEQ     64$             ; YES.
       072710                            HRDERR  INLE,INLEX,     ; NO.
       072730                    64$:
7399 072730  005205                    4$:     INC     R5              ; NEXT CHAN.
7400 072732  020527  000004            CMP     R5,#4           ; DONE?
7401 072736  001316                    BNE     2$              ; NO.
7402                                                           ; YES.
```

```
                                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
                                                COMEND  1$
 072742  004737  027646                         JSR     PC,LOOP         ; REPEAT 'TIL LOOPER EXPIRES.
 072746  103401                                 BCS     65$
 072750  000402                                 BR      66$

 072752  000137  072564          65$:    JMP     1$

 072756  004737  030072          66$:    JSR     PC,TSTEND       ; PRINT ERROR SUMMARY, IF REQ'D.
                                         EXIT    TST     ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>

                                ;
                                ; DISPLAY FILE.
                                ;
 072766  176000  176400  177000 10$:    PROTEC!CH0,PROTEC!CH1,PROTEC!CH2,PROTEC!CH3
 072776  176034                 11$:    WRT!CH0
 073000  117774  040000  000000         APNT!ALL,I!0,0
 073006  173000                         STOPN

 073010                                  END.TEST

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

```
7417
                                        ;*****************************************************************
                                        ;*
                                        ;*      BEGIN TEST 30   -   IMAGE MEMORY PATTERNS
                                        ;*
                                        ;*****************************************************************

                                        ; (ITERATION COUNT = 1.)

7418 073070  005737  002502                    TST     DPUMOD          ;DPU ONLY ??
7419 073074  001402                            BEQ     PATTST          ;NO, EXECUTE.
7420 073076                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>
7421                                    ;
7422                                    ; WRITE, THEN READ-BACK AND TEST PIXEL DATA PATTERNS.
7423                                    ; USE FOUR DATA PATTERNS PER PIXEL AS FOLLOWS:
7424                                    ;       1.  00000000    FOUR 2 BIT 0'S.
7425                                    ;       2.  01010101    FOUR 2 BIT 1'S
7426                                    ;       3.  10101010    FOUR 2 BIT 2'S
7427                                    ;       4.  11111111    FOUR 2 BIT 3'S.
7428                                    ;
7429                                    ; DISPLAY MODE (INTERLACE/NON-INTERLACE) IS TRANSPARENT.
7430                                    ; IF SHORT MODE, CYCLE X FOR Y = 0 AND 2 (ODD VS EVEN),
7431                                    ; THEN CYCLE Y HOLDING X AT 0.
7432                                    ; READ EACH PATTERN 10 TIMES AFTER QUICK PASS.
7433                                    ; IF LONG MODE, CYCLE X FOR EACH Y (512 X 512 ADDRESSES),
7434                                    ; READING EACH PATTERN ONLY ONCE.
7435                                    ;
7436 073102  004737  027702            PATTST: JSR     PC,TSTGO        ;TITLE
7437 073106  004737  030414                    JSR     PC,CONFIG       ; CONFIGURE VSV SYSTEM.
7438 073112  004737  027054                    JSR     PC,LUMIN        ; BLAST LUT.
7439
7440 073116                                    BGNSUB
7441 073120  005037  072414                    CLR     IMC.CH          ;START AT CHANNEL 0
7442 073124  013701  072414            PATT1:  MOV     IMC.CH,R1
7443 073130  006301                            ASL     R1
7444 073132  016101  003256                    MOV     MEMTAB(R1),R1
7445 073136  001002                            BNE     .+6             ; BR IF MEMORY ON THIS CHANNEL.
7446 073140  000137  073506                    JMP     11$
7447 073144  010137  003072                    MOV     R1,PDF
7448 073150  005101                            COM     R1
7449 073152  010137  003074                    MOV     R1,PDM
7450 073156  013737  027700  003234          MOV     LOOPK,TEMP2     ;SAVE A COPY OF LOOPK.
7451 073164  013701  072414                    MOV     IMC.CH,R1
7452 073170  000301                            SWAB    R1
7453 073172                                    BISW3   #RDWRT,R1,CLR.ZA
     073172  010146                            MOV     R1,-(SP)
     073174  052716  176074                    BIS     #RDWRT,(SP)
     073200  012637  071434                    MOV     (SP)+,CLR.ZA
7454 073204                                    DPSTART #CLR.Z
     073204  012777  071424  107620            MOV     #CLR.Z,@DPC     ; START THE DPU.
     073212  004737  027454                    JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
7455 073216  005001                            CLR     R1              ;R1 = X POINTER.
7456 073220  005002                            CLR     R2              ;R2 = Y POINTER.
7457 073222  012737  040000  074126            MOV     #I,WRT.X        ;INIT DISPLAY CODE X...
7458 073230  005037  074130                    CLR     WRT.Y           ;...AND Y.
7459 073234  012705  074112            1$:     MOV     #PATS,R5        ;PATTERN POINTER => R5.
7460 073240  012503                            MOV     (R5)+,R3        ;1ST PATTERN...
```

```
7461 073242  043703  003074          2$:   BIC     PDM,R3          ;...NEXT PATTERN => R3.
7462 073246  042737  001777  074124        BIC     #1777,WRT.D     ;SET NEXT PIXEL PATTERN...
7463 073254  050337  074124                BIS     R3,WRT.D        ;...IN DISPLAY CODE.
7464
7465 073260                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7466
7467 073262  012777  074124  107542        MOV     #WRT.D,@DPC     ;WRITE NEXT PATTERN, THIS PIXEL.
7468 073270  004737  027454                JSR     PC,WAITF
7469 073274  010177  107536                MOV     R1,@DXR         ;SET THE X ADDRESS.
7470 073300  004737  027454                JSR     PC,WAITF
7471 073304  010277  107530          3$:   MOV     R2,@DYR         ;NOW READ-BACK PIXEL DATA...
7472 073310  004737  027454                JSR     PC,WAITF
7473 073314  017704  107514                MOV     @DSR,R4         ;... TO R4.
7474 073320  043704  003074                BIC     PDM,R4          ;OK, STRIP ANY CRAP.
7475 073324  020304                       CMP     R3,R4           ;AND TEST THE DATA.
7476 073326  001410                       BEQ     9$              ;BR IF OK
7477 073330                                HRDERR  IMDI,IMDERR     ;DATA INCORRECT ERROR.
7478 073350                          9$:   LOOPTO  3$              ; REPEAT...
     073350  004737  027646                JSR     PC,LOOP
     073354  103753                       BCS     3$
7479 073356  013737  003234  027700        MOV     TEMP2,LOOPK     ;RESET LOOPER.
7480 073364                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7481
7482 073366  013737  003234  027700  4$:   MOV     TEMP2,LOOPK     ;RESET LOOPER.
7483 073374  012503                       MOV     (R5)+,R3        ;GET NEXT PATTERN...
7484 073376  100321                       BPL     2$              ;...BR IF NOT TERMINATOR.
7485 073400  020227  000004          5$:   CMP     R2,#4           ;HOLD X...
7486 073404  103011                       BHIS    7$              ;...AT 0, FOR Y 4 TO MAX.
7487
7488 073406  020127  001776          6$:   CMP     R1,#MAXX
7489 073412  001406                       BEQ     7$              ;BR IF AT MAX X.
7490 073414  062701  000002                ADD     #2,R1           ;OTHERWISE, BUMP X...
7491 073420  062737  000002  074126        ADD     #2,WRT.X
7492 073426  000702                       BR      1$              ;...AND CONTINUE.
7493 073430  020237  003250          7$:   CMP     R2,YMAX
7494 073434  001424                       BEQ     11$             ;ALL DONE IF MAX Y.
7495 073436  005001                       CLR     R1              ;OTHERWISE, RESET X...
7496 073440  012737  040000  074126        MOV     #1,WRT.X
7497 073446  062702  000002                ADD     #2,R2           ;...BUMP Y...
7498 073452  062737  000002  074130        ADD     #2,WRT.Y
7499 073460  000665                       BR      1$              ;...AND CONTINUE.
7500
7501 073462  004737  026756          10$:  JSR     PC,RELEAS       ;RELEASE 'HUNG' DPU...
7502 073466                                HRDERR  PRBHNG          ;...REPORT ERROR AND QUIT.
7503 073506                          11$:
7504 073506  005237  072414                INC     IMC.CH
7505 073512  023727  072414  000004        CMP     IMC.CH,#4
7506 073520  103601                       BLO     PATT1
7507 073522                                ENDSUB
7508
7509 073524                          MPTNEW: BGNSUB
7510 073526  013700  003240                MOV     FREE,R0         ;START THE 'WRITE BUFFER' AT 1ST FREE LOCATION
7511 073532  012701  000045                MOV     #256./MPLEN+MPLEN,R1  ;SETUP # OF TIMES TO REPEAT PATTERN TABLE
7512 073536  012702  000011          1$:   MOV     #MPLEN,R2       ;GET LENGTH OF PATTERN TABLE
7513 073542  012703  074150                MOV     #MPAT2,R3       ;...AND ITS ADDRESS
7514 073546  012320                 2$:   MOV     (R3)+,(R0)+     ;COPY PATTERN TABLE INTO WRITE BUFFER
7515 073550  005302                       DEC     R2
```

```
7516 073552  001375                        BNE     2$
7517 073554  005301                        DEC     R1                      ;DO IT ENOUGH TIMES FOR 1 SCAN LINE, ALL PATTERNS
7518 073556  001367                        BNE     1$
7519 073560  010037  074174                MOV     R0,MPCBAS       ;NEXT FREE LOCATION IS BASE OF ''COMPARE BUFFER''
7520 073564  005037  072414                CLR     IMC.CH          ;START TESTING AT CHANNEL 0.
7521
7522 073570  004737  026650          3$:   JSR     PC,CLRRW        ;CLEAR ALL MEMORIES
7523 073574  013700  072414                MOV     IMC.CH,R0       ;GET CHANNEL NUMBER
7524 073600  000300                        SWAB    R0                      ;...INTO BITS <9:8>
7525 073602  042737  001400  074206        BIC     #C^3,MPT.WE     ;CLEAR OLD CHANNEL FROM ''WRITE-ENABLE'' INSTRUCTION
7526 073610  050037  074206                BIS     R0,MPT.WE       ;...AND INSERT THE CURRENT CHANNEL
7527 073614  013700  072414                MOV     IMC.CH,R0       ;GET THE CHANNEL AGAIN
7528 073620  006300                        ASL     R0                      ;...AND MAKE IT A WORD INDEX
7529 073622  016001  003256                MOV     MEMTAB(R0),R1   ;GET THE MEMORY PARAMETER (PIXEL MASK)
7530 073626  001516                        BEQ     8$                      ;IF ZERO, NO MEMORY ON THIS CHANNEL.
7531 073630  010137  003072                MOV     R1,PDF          ;SAVE PIXEL MASK (FOR ERROR REPORTS)
7532 073634  006201                        ASR     R1                      ;GET PIX MASK INTO LOW BYTE
7533 073636  006201                        ASR     R1
7534 073640  005101                        COM     R1                      ;COMPLEMENT IT, TO ALLOW SETTING OF UNUSED
7535 073642  010146                        MOV     R1,-(SP)        ;...BITS IN COMPARE WORDS
7536 073644  110166  000001                MOVB    R1,1(SP)        ;BOTH BYTES.
7537 073650  012601                        MOV     (SP)+,R1
7538 073652  013700  074174                MOV     MPCBAS,R0       ;START AT BEGINNING OF COMPARE BUFFER
7539 073656  012702  000412                MOV     #256.+MPLEN+1,R2
7540 073662  013703  C03240                MOV     FREE,R3         ;AND AT WRITE BUFFER
7541 073666  012310                  4$:   MOV     (R3)+,(R0)      ;COPY WRITE BUFFER INTO COMPARE BUFFER,
7542 073670  050120                        BIS     R1,(R0)+        ;...SETTING UNUSED BITS.
7543 073672  005302                        DEC     R2
7544 073674  001374                        BNE     4$
7545 073676  010037  072450                MOV     R0,IMC.RA       ;MAKE NEXT FREE ADDRESS THE START OF READ BUFFER.
7546
7547 073702  005037  072436                CLR     IMC.Y           ;START TESTING AT FIRST LINE (Y 0)
7548 073706  013737  072436  074214  5$:   MOV     IMC.Y,MPT.WY    ;TELL WRITER WHERE TO WRITE
7549 073714  005037  074172                CLR     MPOFF           ;CLEAR THE PATTERN OFFSET
7550 073720  013737  003240  074220  6$:   MOV     FREE,MPT.WB     ;GET BASE OF WRITE BUFFER
7551 073726  063737  074172  074220        ADD     MPOFF,MPT.WB    ;...AND ADD THE OFFSET
7552
7553 073734                                BGNSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< BEGIN SEGMENT.
7554
7555 073736  012777  074176  107066        MOV     #MPT.ST,@DPC    ;THEN WRITE DATA FROM THERE.
7556 073744  004737  027454                JSR     PC,WAITF
7557 073750  013737  074174  072416        MOV     MPCBAS,IMC.TA   ;GET BASE OF COMPARE BUFFER
7558 073756  063737  074172  072416        ADD     MPOFF,IMC.TA    ;...AND ADD THE OFFSET TO IT.
7559 073764  004737  071600                JSR     PC,IMCHK        ;READ THE LINE AND COMPARE IT TO COMPARE BUFFER
7560 073770                                ENDSEG          ; <<<<<<<<<<<<<<<<<<<<<<<<<< END SEGMENT.
7561
7562 073772  005737  003230                TST     INTLAC          ;INTERLACED MODE?
7563 073776  001014                        BNE     7$                      ;IF YES, SKIP AHEAD.
7564 074000  032737  000002  072436        BIT     #2,IMC.Y
7565 074006  001010                        BNE     7$
7566 074010  062737  000002  072436        ADD     #2,IMC.Y        ;IF NONINTERLACED, NEXT LINE SHOULD READ THE SAME
7567 074016  004737  071600                JSR     PC,IMCHK        ;CHECK FOR IT.
7568 074022  162737  000002  072436        SUB     #2,IMC.Y
7569 074030  062737  000002  074172  7$:   ADD     #2,MPOFF        ;BUMP THE OFFSET
7570 074036  023727  074172  000022        CMP     MPOFF,#MPLEN*2  ;DONE ALL PATTERNS?
7571 074044  002725                        BLT     6$                      ;IF NOT, LOOP FOR MORE ON THIS SAME LINE.
7572 074046  062737  000002  072436        ADD     #2,IMC.Y        ;IF DONE ALL, BUMP Y
```

```
7573 074054  023737  072436  003250          CMP     IMC.Y,YMAX      ;DONE LAST LINE?
7574 074062  101711                          BLOS    5$                      ;IF NOT, GO FOR MORE
7575 074064  005237  072414              8$: INC     IMC.CH          ;BUMP THE CHANNEL NUMBER.
7576 074070  023727  072414  000004        CMP     IMC.CH,#4       ;DONE ALL CHANNELS?
7577 074076  002634                        BLT     3$                      ;IF NOT, DO THEM.
7578 074100                                 ENDSUB
7579
7580 074102  004737  030072                JSR     PC,TSTEND
7581 074106                                 EXIT    TST             ;EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>>>
7582 074112  000000                  PATS: 0                       ;PATRN 1 (00000000)
7583 074114  000524                        0524                    ;PATRN 2 (01010101)
7584 074116  001250                        1250                    ;PATRN 3 (10101010)
7585 074120  001774                        1774                    ;PATRN 4 (11111111)
7586 074122  177777                        -1                      ;LIST TERMINATOR.
7587
7588 074124  116000                  WRT.D: APNT.LO                ;PIXEL DATA SET ON THE FLY.
7589 074126  040000                  WRT.X: I.0                    ;LIKEWISE ADDRESSES.
7590 074130  000000                  WRT.Y: 0
7591 074132  164000  164000  164000        DNOP,DNOP,DNOP          ;IN CASE WE WANT TO CHANGE...
7592 074140  164000  164000  164000        DNOP,DNOP,DNOP          ;...MODE FOR P-R-B DEBUG.
7593 074146  173000                        STOPN                   ;STOP AND TEST THIS PIXEL.
7594
7595                                ;++++++++++++++++++++++++++++++++++
7596                                        .MACRO  PAT.D MDX
7597                                        <MDX*BIT8>!<MDX*BIT4>.<-MDX-1&17*BIT12>.<-MDX-1&17>
7598                                        .ENDM
7599
7600 074150                        MPAT2:  PAT.D   0               ;PUT 4-BIT PATTERN INTO BITS <7:4> AND <11:8>,
7601 074152                                PAT.D   1               ;...AND ITS COMPLEMENT INTO <15:12> AND <3:0>
7602 074154                                PAT.D   2               ;USE FLOATING 1'S AND 0'S
7603 074156                                PAT.D   4
7604 074160                                PAT.D   10
7605 074162                                PAT.D   7
7606 074164                                PAT.D   13
7607 074166                                PAT.D   15
7608 074170                                PAT.D   16
7609         000011                MPLEN=.-MPAT2/2         ;USE ODD # OF WORDS TO MAXIMIZE CHANCES OF FINDING
7610                                ;               DUAL-ADDRESSING PROBLEMS.
7611
7612 074172  000000                  MPOFF:  0       ;OFFSET FOR WRITING AND COMPARING
7613 074174  000000                  MPCBAS: 0               ;BASE ADDRESS OF COMPARE BUFFER
7614
7615                                ;COMMANDS FOR WRITING THE PATTERN INTO THE LINE
7616 074176  176000  176400  177000  MPT.ST: PROTEC.CH0,PROTEC.CH1,PROTEC!CH2,PROTEC.CH3
7617 074206  176034                  MPT.WE: WRT!CH0         ;ENABLE A CHANNEL FOR WRITING
7618 074210  114000  000000                  APNT,0          ;START AT X=0
7619 074214  000000                  MPT.WY: 0
7620 074216  137000                          BM18            ;USE BIT-MAP-1, WITH A COUNT OF 512. PIXELS
7621 074220  000000                  MPT.WB: 0
7622 074222  173000                          STOPN
7623
7624                                ;++++++++++++++++++++++++++++++++++++++++++++++++++
7625 074224                                 END.TEST

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

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

```
7627
                                ;*************************************************************************
                                ;*
                                ;*      BEGIN TEST 31   -   SPARE
                                ;*
                                ;*************************************************************************

                                ; (ITERATION COUNT = 1.)

7628 074264  005737  002502            TST     DPJMOD          ;DPU ONLY ??
7629 074270  001402                    BEQ     SPARTS          ;NO, EXECUTE.
7630 074272                            EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
7631
7632 074276  004737  027702    SPARTS: JSR     PC,TSTGO
7633 074302                            EXIT    TST     ; ***NOT YET CODED***
7634
7635                            ;*********************
7636 074306                            END.TEST

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

```
7639
                                        ;*************************************************************************
                                        ;*
                                        ;*      BEGIN TEST 32   -   SYSTEM VERIFICATION DISPLAY
                                        ;*
                                        ;*************************************************************************
                                        ; (ITERATION COUNT = 1.)

7640 074374  005737  002502                    TST     DPJMOD          ;DPU ONLY ??
7641 074400  001402                            BEQ     VIS1            ;NO, EXECUTE.
7642 074402                                    EXIT    TST             ; EXIT THIS TEST >>>>>>>>>>>>>>>>>>>>>>>>>>
7643                                    ;
7644                                    ; SMORGASBORD, ONE FRAME THAT HAS A BIT OF EVERYTHING.
7645                                    ; FIRST, A DEC LOGO (SORT OF), THEN VIOLATE THE EDGES
7646                                    ; AT LOWER LEFT AND UPPER RIGHT.
7647                                    ;
7648 074406  004737  027702            VIS1:   JSR     PC,TSTGO
7649 074412  004737  026650                    JSR     PC,CLRRW        ; CLEAR.
7650 074416  004737  105222                    JSR     PC,DBLINK       ; DISABLE BLINKING.
7651 074422  004737  027054                    JSR     PC,LUMIN        ; BLAST THE LUT.
7652
7653 074426  012777  074442  106376    1$:     MOV     #LOGO,@DPC      ; START WITH THE LOGO.
7654 074434  004737  027454                    JSR     PC,WAITF
7655                                    ;       JSR     PC,PAUS.5
7656 074440  000446                            BR      VIS2            ;...AND CONTINUE.
7657
7658 074442  114000  000540  000100    LOGO:   APNT, 540, 100
7659 074450  152000  110146                    SETCB,ACAT
7660 074454  103774                            CHAR!ALL
7661 074456     104     040     111            .ASCIZ  /D I G I T A L/
7662
7663 074474  114000  000140  000140            APNT,140,140            ;EXCEED EDGES AT LOWER LEFT...
7664 074502  110000                            LVEC
7665 074504                                    LXY     I,-200,0
7666 074510                                    LXY     I,0,-200
7667 074514                                    LXY     I,200,0
7668 074520                                    LXY     I,0,200
7669 074524  114000  001640  001540            APNT,1640,1540          ;...AND AT UPPER RIGHT.
7670 074532  110000                            LVEC
7671 074534                                    LXY     I,200,0
7672 074540                                    LXY     I,0,300
7673 074544                                    LXY     I,-200,0
7674 074550                                    LXY     I,0,-300
7675 074554  173000                            STOPN
```

```
7677                                        ;
7678                                        ; NOW 1 LARGE AND 2 SMALL OCTAGONS.
7679                                        ;
7680 074556  012777  074616  106246  VIS2:          MOV     #LOCT,@DPC      ;1 LARGE ONE...
7681 074564  004737  027454                 JSR     PC,WAITF
7682 074570  012777  074670  106234          MOV     #SOCT1,@DPC     ;...AND 2 SMALL ONES.
7683 074576  004737  027454                 JSR     PC,WAITF
7684 074602  012777  074702  106222          MOV     #SOCT2,@DPC
7685 074610  004737  027454                 JSR     PC,WAITF
7686                                  ;      JSR     PC,PAUS.5
7687 074614  000456                        BR      VIS3            ;...AND CONTINUE.
7688
7689 074616  114000  000500  000002  LOCT:   APNT,500,2
7690 074624  113774                        LVEC!ALL                ; 1 LARGE OCTAGON...
7691 074626                                 LXY     I,576,0
7692 074632                                 LXY     I,436,436
7693 074636                                 LXY     I,0,576
7694 074642                                 LXY     I,-436,436
7695 074646                                 LXY     I,-576,0
7696 074652                                 LXY     I,-436,-436
7697 074656                                 LXY     I,0,-576
7698 074662                                 LXY     I,436,-436
7699 074666  173000                        STOPN
7700
7701 074670  114000  000220  000620  SOCT1:  APNT,220,620            ;...AND 2 SMALL ONES.
7702 074676  160000  074710                 DJMP, .+10
7703 074702  114000  001460  000620  SOCT2:  APNT,1460,620
7704 074710  107774                        SVEC.ALL
7705 074712                                 SXY     I,76,0
7706 074714                                 SXY     I,56,56
7707 074716                                 SXY     I,0,76
7708 074720                                 SXY     I,-56,56
7709 074722                                 SXY     I,-76,0
7710 074724                                 SXY     I,-56,-56
7711 074726                                 SXY     I,0,-76
7712 074730                                 SXY     I,56,-56
7713 074732  130000                        RPNT                    ;5 REL POINTS IN CENTER.
7714 074734                                 SXY     U,20,50
7715 074736                                 SXY     I,20,50         ;CENTER.
7716 074740                                 SXY     I,20,0
7717 074742                                 SXY     I,-20,20
7718 074744                                 SXY     I,-20,-20
7719 074746                                 SXY     I,20,-20
7720 074750  173000                        STOPN
```

```
7722                                        ;
7723                                        ; NEXT, SHOW GRAPH/HISTO Y IN UPPER LEFT.
7724                                        ;
7725 074752  012737  011240  075124  VIS3:          MOV     #1240!GHIINH,GY+2 ;PLOT GRAPH Y STARTING AT 1240.GHIINH.
7726 074760  012777  075100  106044               MOV     #GRY,@DPC         ;START GRAPH Y.
7727 074766  000402                               SKP2
7728 074770  005277  106036          1$:          INC     @DPC
7729 074774  004737  027454                       JSR     PC,WAITF
7730 075000  042737  010000  075124               BIC     #GHIINH,GY+2
7731 075006  062737  000004  075124               ADD     #4,GY+2           ;INCR Y AMPLITUDE...
7732 075014  023727  075124  001604               CMP     GY+2,#1604
7733 075022  001362                               BNE     1$                ;LOOP UNTIL DONE.
7734
7735 075024  012737  051240  075152               MOV     #HST!1240!GHIINH,HY+2 ;NOW, HISTO Y STARTING AT 1240.GHIINH.
7736 075032  012777  075134  105772               MOV     #HSY,@DPC
7737 075040  000402                               SKP2
7738 075042  005277  105764          2$:          INC     @DPC
7739 075046  004737  027454                       JSR     PC,WAITF
7740 075052  042737  010000  075152               BIC     #GHIINH,HY+2
7741 075060  062737  000010  075152               ADD     #10,HY+2          ;INCR Y AMPLITUDE...
7742 075066  023727  075152  041550               CMP     HY+2,#HST!1550
7743 075074  001362                               BNE     2$                ;...AND LOOP TIL DONE.
7744
7745                                    ;          JSR     PC,PAUS.5
7746 075076  000431                               BR      VIS4              ;...AND CONTINUE.
7747
7748 075100  114000  000100  001240  GRY:         APNT,100,1240
7749 075106  113774                               LVEC:ALL                  ;DRAW BASE LINE.
7750 075110                                        LXY     1,340,0
7751 075114                                        LXY     0,-340,0
7752 075120  174104                               GXI!4                     ;X INCR - 4 FOR GRAPH.
7753 075122  127774  001240          GY:          GHY!ALL,0!1240
7754 075126  173000  160000  075122               STOPN,DJMP,GY
7755
7756 075134  150000  001240          HSY:         SETHB,1240                ;HISTO BASE LINE.
7757 075140  114000  000140  001240               APNT,140,1240
7758 075146  174110                               GXI!10                    ;X INCR = 10 FOR HISTO.
7759 075150  127774  041240          HY:          GHY!ALL,HST!1240
7760 075154  173000  160000  075150               STOPN,DJMP,HY
```

```
7762                                        ;
7763                                        ; NEXT, GRAPH/HISTO X IN LOWER RIGHT.
7764                                        ;
7765 075162  012737  011340  075334  VIS4:          MOV     #1340!GHIINH,GX+2 ;PLOT GRAPH X STARTING AT 1340!GHIINH.
7766 075170  012777  075310  105634                MOV     #GRX,@DPC         ;START GRAPH X.
7767 075176  000402                                 SKP2
7768 075200  005277  105626              1$:      INC     @DPC
7769 075204  004737  027454                        JSR     PC,WAITF
7770 075210  042737  010000  075334                BIC     #GHIINH,GX+2
7771 075216  062737  000004  075334                ADD     #4,GX+2           ;INCR X AMPLITUDE...
7772 075224  023727  075334  001704                CMP     GX+2,#1704
7773 075232  001362                                BNE     1$                ;LOOP UNTIL DONE.
7774
7775 075234  012737  051340  075362                MOV     #HST!1340!GHIINH,HX+2 ;NOW, HISTO X STARTING AT 1340!GHIINH.
7776 075242  012777  075344  105562                MOV     #HSX,@DPC
7777 075250  000402                                 SKP2
7778 075252  005277  105554              2$:      INC     @DPC
7779 075256  004737  027454                        JSR     PC,WAITF
7780 075262  042737  010000  075362                BIC     #GHIINH,HX+2
7781 075270  062737  000010  075362                ADD     #10,HX+2          ;INCR X AMPLITUDE...
7782 075276  023727  075362  041650                CMP     HX+2,#HST.1650
7783 075304  001362                                BNE     2$                ;...AND LOOP TIL DONE.
7784
7785                                        ;       JSR     PC,PAUS.5
7786 075306  000431                                BR      VIS5              ;...AND CONTINUE.
7787
7788 075310  114000  001340  000100  GRX:      APNT,1340,100
7789 075316  113774                             LVEC!ALL                      ;DRAW BASE LINE.
7790 075320                                      LXY     I,0,340
7791 075324                                      LXY     U,0,-340
7792 075330  174104                             GYI!4                         ;Y INCR = 4 FOR GRAPH.
7793 075332  123774  001340              GX:      GHX!ALL,0!1340
7794 075336  173000  160000  075332              STOPN,DJMP,GX
7795
7796 075344  150000  001340              HSX:     SETHB,1340                    ;HISTO BASE LINE.
7797 075350  114000  001340  000140              APNT,1340,140
7798 075356  174110                             GYI,10                        ;Y INCR = 10 FOR HISTO.
7799 075360  123774  041340              HX:      GHX!ALL,HST!1340
7800 075364  173000  160000  075360              STOPN,DJMP,HX
```

```
                                        ;
                                        ; NEXT SHOW BIT MAP MODE 1 AT BOTTOM CENTER,
                                        ; AND BIT MAP MODE 0 AT TOP CENTER.
                                        ;
075372  012700  177777        VISS:     MOV     #-1,R0          ; SET PIXEL DATA PATTERN...
075376  004737  027020                  JSR     PC,FILL2        ;...AND FILL BUFFER WITH...
                                                                ;...ALTERNATE WORDS.

075402  013737  003240  075552          MOV     FREE,B148       ; SET BUFFER ADDRESS.
075410  012737  136340  075550          MOV     #BM14!340,B148-2 ; SET OPCODE = BM14 X 340
075416  012777  075530  105406          MOV     #B14,@DPC       ;XCT BM14, DRAWS A DASHED...
075424  004737  027454                  JSR     PC,WAITF        ;...LINE (4 ON, 4 OFF)...
                                                                ;...FOR 340 PIXELS.
075430  012737  137400  075550          MOV     #BM18!400,B148-2 ;SET OPCODE = BM18 X 400
075436  012777  075542  105366          MOV     #B18,@DPC       ;XCT BM18, DRAWS A DASHED...
075444  004737  027454                  JSR     PC,WAITF        ;...LINE (2 ON, 2 OFF)...
                                                                ;...FOR 400 PIXELS.

075450  012700  000377                  MOV     #377,R0
075454  004737  027014                  JSR     PC,FILL1

075460  013737  003240  075600          MOV     FREE,B048       ; SET BUFFER ADDRESS.
                                ;       MOV     #BM04!M64,B048-2 ; 4 BIT X 64 SQUARE.
075466  012737  134010  075576          MOV     #BM04!M32!EX2,B048-2 ; 4 BIT X 64 SQUARE.
075474  012777  075556  105330          MOV     #B04,@DPC       ; DO IT.
075502  004737  027454                  JSR     PC,WAITF
                                ;       MOV     #BM08!M64,B048-2 ; 8 BIT X 64 SQUARE.
075506  012737  135010  075576          MOV     #BM08!M32!EX2,B048-2 ; 8 BIT X 64 SQUARE.
075514  012777  075570  105310          MOV     #B08,@DPC       ; AND DO THAT TOO.
075522  004737  027454                  JSR     PC,WAITF

075526                          1$:     ;JSR    PC,PAUS.5
075526  000430                          BR      VISSA           ;...AND CONTINUE.

075530  114000  000400  000240  B14:    APNT,400,240            ; 4 ON 4 OFF FOR 340 PIXELS.
075536  160000  075550                  DJMP, .+10
075542  114000  000300  000340  B18:    APNT,300,340            ; 2 ON 2 OFF FOR 400 PIXELS.
075550  136340                          BM14!340
075552  000000                  B148:   0
075554  173000                          STOPN

075556  114000  000540  001340  B04:    APNT, 540, 1340
075564  160000  075576                  DJMP, .+10
075570  114000  001040  001340  B08:    APNT, 1040, 1340
075576  134001                          BM04!M64
075600  000000                  B048:   0
075602  164000  164000                  DNOP, DNOP
075606  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).
                                        ;
075610  112737  000004  075671  VIS5A:        MOVB    #4,VRLF2+1      ;SET INITIAL COUNT FOR 0-INTENS AT 4
075616  012777  075660  105206          MOV     #VRLF,@DPC      ;START THE FILE
075624  000403                          BR      2$              ;SKIP OVER THE RESUME
075626  012777  000001  105176  1$:     MOV     #1,@DPC         ;RESUME THE DISPLAY
075634  004737  027454          2$:     JSR     PC,WAITF        ;WAIT FOR DONE
075640  123727  075671  000024          CMPB    VRLF2+1,#24     ;DONE 20(8) LINES?
075646  103003                          BHIS    3$                      ;FINISH IF YES.
075650  105237  075671                  INCB    VRLF2+1         ;BUMP INITIAL COUNT IF NO.
075654  000764                          BR      1$                      ;...THEN XCT AGAIN.
075656                          3$:     ;JSR    PC,PAUS.5       ;PAUSE
075656  000430                          BR      VIS6            ;...THEN CONTINUE.
                                        ;
                                        ;DISPLAY FILE FOR VISUAL RUN-LENGTH MODE
075660  114000  000300  000320  VRLF:   APNT,300,320            ;START BETWEEN BM1 & BM0
075666  144040                  VRLF1:  RNLN!YDOWN              ;ENTER RUN-LENGTH MODE, GO DOWN THE SCREEN
075670     000     004          VRLF2:  .BYTE   0,4     ;0 INTENSITY, START AT LENGTH=4
075672  006020  006040  006060          .WORD   6020,6040,6060  ;INCREASING INTENSITY, LENGTH-14
075700  006100  006120  006140          .WORD   6100,6120,6140,6160     ;....AND MORE
075710  006200  006220  006240          .WORD   6200,6220,6240,6260     ;....AND MORE
075720  006300  006320  006340          .WORD   6300,6320,6340,6360     ;....AND MORE
075730  173000                          STOPN           ;STOP
075732  000000                          .WORD   0               ;COUNT = 0 FOR NEWLINE
075734  160000  075670                  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).
                                        ;
075740  012737  112000  076400  VIS6:         MOV     #LVEC!LO,RBL    ;INIT DISPLAY FILE.
075746  012737  040000  076402          MOV     #I!0,RBX
075754  012737  000300  076404          MOV     #300,RBY
075762  012701  000003                  MOV     #3,R1           ; SET FOR 3 VECTORS PER LEVEL...
075766  012703  000004                  MOV     #4,R3           ;...AND INCR LEVEL BY '1'.
075772  010102                  1$:     MOV     R1,R2           ; SCRATCH COPY LEVEL COUNTER.
075774  012777  076372  105030          MOV     #RBO,@DPC       ;START, QUAD 1.
076002  004737  027454                  JSR     PC,WAITF
076006  023727  076404  000000  2$:     CMP     RBY,#0          ;DELTA Y MIN ??
076014  001403                          BEQ     3$              ;DONE QUAD 1 IF SO.
076016  004737  076230                  JSR     PC,Q1
076022  000771                          BR      2$
076024  052737  020000  076404  3$:     BIS     #MXY,RBY        ,QUAD 2, DX POS, DY NEG.
076032  023727  076402  040000  4$:     CMP     RBX,#I!0        ;DELTA X MIN ??
076040  001403                          BEQ     5$              ;DONE QUAD 2 IF SO.
076042  004737  076266                  JSR     PC,Q2
076046  000771                          BR      4$
076050  052737  020000  076402  5$:     BIS     #MXY,RBX        ;QUAD 3, DX NEG, DY NEG.
076056  023727  076404  020000  6$:     CMP     RBY,#MXY.0      ;DELTA Y -MIN ??
076064  001403                          BEQ     7$              ;DONE QUAD 3 IF SO.
076066  004737  076230                  JSR     PC,Q3
076072  000771                          BR      6$
076074  042737  020000  076404  7$:     BIC     #MXY,RBY        ;QUAD 4, DX NEG, DY POS.
076102  023727  076402  060000  8$:     CMP     RBX,#I.MXY!0    ;DELTA X -MIN ??
076110  001403                          BEQ     9$              ;DONE QUAD 4 IF SO.
076112  004737  076266                  JSR     PC,Q4
076116  000771                          BR      8$
076120  012737  004000  003220  9$:     MOV     #4000,SHADLY    ; FINALLY, SET SHADING DELAY...
076126  012705  110070                  MOV     #NTSC8+2,R5     ;...AND CYCLE THRU 16 SHADES...
076132  012501                  10$:    MOV     (R5)+,R1        ;...OF EACH BASIC (NTSC) COLOR.
076134  001403                          BEQ     11$             ; BR WHEN ALL DONE.
076136  004737  027172                  JSR     PC,SHAD16
076142  000773                          BR      10$
076144  004737  104626          11$:    JSR     PC,OUTLIN       ; OUTLINE THE SCREEN.
076150  004737  027514                  JSR     PC,PAUSE1
076154  004737  027514                  JSR     PC,PAUSE1
076160  004737  027514                  JSR     PC,PAUSE1       ; PAUSE 3 SECONDS...
076164  004737  105174                  JSR     PC,EBLINK       ; ENABLE BLINKING.
076170  004737  027514                  JSR     PC,PAUSE1       ;
076174  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  RB0:     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,RB0
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$             ;GC 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
```

```
                                        ;
                                        ; 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/
```

```
100560  114000  001366  000716              APNT,1526-140,HAFY60-20
100566  103774                              CHAR!ALL
100570     061     065     062              .ASCII  /1526/
100574  114000  001636  000716              APNT,1776-140,HAFY60-20
100602  103774                              CHAR!ALL
100604     061     067     067              .ASCII  /1776/
100610  114000  000716  000010              APNT,HAFX-60,10
100616  103774                              CHAR!ALL
100620     060     060     060              .ASCII  /0000/
100624  114000  000716  000250              APNT,HAFX-60,250
100632  103774                              CHAR!ALL
100634     060     062     064              .ASCII  /0240/
100640  114000  000716  000510              APNT,HAFX-60,510
100646  103774                              CHAR!ALL
100650     060     065     060              .ASCII  /0500/
100654  114000  000716  001136              APNT,HAFX-60,1176-40
100662  103774                              CHAR!ALL
100664     061     061     067              .ASCII  /1176/
100670  114000  000716  001376              APNT,HAFX-60,1436-40
100676  103774                              CHAR!ALL
100700     061     064     063              .ASCII  /1436/
100704  114000  000716  001636              APNT,HAFX-60,1676-40
100712  103774                              CHAR!ALL
100714     061     066     067              .ASCII  /1676/

100720  114000  001706  000010              APNT,1776-70,10           ;FINALLY...
100726  103774                              CHAR!ALL
100730     107     120                      .ASCII  /GP/              ;...MY INITIALS.

100732  164000  164000  164000    BOXES:    DNOP,DNOP,DNOP            ;*** FOR DEBUG ***
100740  146016                    JCXB:     CUIM                      ; MATCH ON, SWITCH OFF.
100742  114000  000000  000000              APNT,0,0
100750  113774                              LVEC!ALL                  ;AND DRAW NESTED BOXES.
100752                                      LXY     I,1776,0
100756                                      LXY     I,0,1676
100762                                      LXY     I,-1776,0
100766                                      LXY     I,0,-1676
100772  114000  000250  000240              APNT,250,240
101000  110000                              LVEC
101002                                      LXY     I,1256,0
101006                                      LXY     I,0,1176
101012                                      LXY     I,-1256,0
101016                                      LXY     I,0,-1176
101022  114000  000520  000500              APNT,520,500
101030  110000                              LVEC
101032                                      LXY     I,536,0
101036                                      LXY     I,0,476
101042                                      LXY     I,-536,0
101046                                      LXY     I,0,-476
101052  114000  000676  000636              APNT,HAFX-64.,HAFY50-64.
101060  134001                              BM04!M64                  ; A 64 SQUARE BIT MAP AT CENTER.
101062  000000                    BMXX.     0                         ; POINTS TO FREE CORE BUFFER.
101064  114000  000756  000736              APNT,HAFX-20,HAFY60
101072  106000                              SVEC!LO                   ; CROSS AT CENTER (BLK ON WHITE)
101074                                      SXY     I,40,0
101076                                      SXY     U,-20,20
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.
                                        ;
102356  004737  026650          BARS:   JSR     PC,CLRRW        ;INIT DISPLAY.
102362  004737  027066                  JSR     PC,NTSC         ; NTSC COLORS (IF LUT AVAILABLE)
102366  012737  112000  102466          MOV     #LVEC!LO,BV     ;INIT DISPLAY CODE AT LVL 0...
102374  012701  000004                  MOV     #4,R1           ;..AND INCREMENT BY LVL 1.
102400  012702  000400          1$:     MOV     #256.,R2        ; 256 PAIRS = 512 TOTAL.
102404  012777  102460  100420          MOV     #BGEN,@DPC
102412  000402                          SKP2
102414  005277  100412          2$:     INC     @DPC
102420  004737  027454                  JSR     PC,WAITF
102424  005302                          DEC     R2
102426  001411                          BEQ     3$              ;BR WHEN FINISHED.
102430  060137  102466                  ADD     R1,BV           ; OTHERWISE, INCREMENT...
102434  042737  176000  102466          BIC     #^C1777,BV      ;...AND STRIP THE LEVEL...
102442  052737  112000  102466          BIS     #LVEC!LO,BV     ;...AND REPLACE OPCODE.
102450  000761                          BR      2$              ;...AND LOOP.

102452  004737  104762          3$:     JSR     PC,OUTLIN       ; OUTLINE THE SCREEN.
102456  000207                          RTS     PC              ; RE URN TO IDLE LOOP.

102460  114000  000000  000000  BGEN:   APNT, 0, 0
102466  112000                  BV:     LVEC!LO
102470  040000  001776                  I!0, MAXY               ; UP...
102474  000002  000000                  2, 0                    ;...AND OVER...
102500  040000  021776                  I!0, MXY!MAXY           ;...DOWN...
102504  000002  000000                  2, 0                    ;...AND OVER AGAIN.
102510  173000                          STOPN
102512  160000  102466                  DJMP, BV
```

```
                                        ;
                                        ; 7 X 7 POINT TABLE FOR DOTS AND CROSS-HATCH.
                                        ;
102516                                  XTBL:                   ;X POINTS...
102516  000104                          Y50TBL: 34.*2           ;...AND 50 HZ Y POINTS.
102520  000330                                  108.*2          ;DELTA = 74.
102522  000554                                  182.*2
102524  001000                                  256.*2
102526  001224                                  330.*2
102530  001450                                  404.*2
102532  001674                                  478.*2
102534  177777                                  -1
102536  000110                          Y60TBL: 36.*2           ; 60 HZ Y POINTS.
102540  000320                                  104.*2          ;DELTA = 68.
102542  000530                                  172.*2
102544  000740                                  240.*2
102546  001150                                  308.*2
102550  001360                                  376.*2
102552  001570                                  444.*2
102554  177777                                  -1
```

```
8495                                    ;
8496                                    ; DOT GENERATOR.
8497                                    ;
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$,@DPC       ; START THE DPU.
103142  004737  027454                  JSR     PC,WAITF        ; WAIT FOR DISPLAY STOP.
103146  000405                          BR      3$
                                        ;
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.
```

```
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
```

```
8715                                    ;
8716                                    ; DISPLAYS FOR FILLING SCREEN WITH SOLID COLOR
8717                                    ;
8718 104526  012746  003774            SCRWHT: MOV #ALL,-(SP)
8719 104532  000410                            BR      SCRCOM          ; GO TO COMMON SECTION
8720
8721 104534  012746  002200            SCRRED: MOV     #L0!200,-(SP)   ; GET RED ONTO STACK,
8722 104540  000405                            BR      SCRCOM          ; GO TO COMMON
8723
8724 104542  012746  002100            SCRBLU: MOV     #L0!100,-(SP)   ; GET BLUE ONTO STACK,
8725 104546  000402                            BR      SCRCOM          ; GO TO COMMON
8726
8727 104550  012746  003400            SCRGRN: MOV     #L0!1400,-(SP)  ; GET FULL GREEN ONTO STACK,
8728
8729 104554  004737  026650            SCRCOM: JSR     PC,CLRRW        ;CLEAR FOR READ/WRITE
8730 104560  004737  027066                    JSR     PC,NTSC         ;SETUP COLOR TABLE (IF AVAILABLE)
8731 104564  012737  114000  104612            MOV     #APNT,10$       ;INIT PIXEL DATA SELECTION
8732 104572  052637  104612                    BIS     (SP)+,10$       ;INSERT THE COLOR
8733 104576  012777  104612  076226            MOV     #10$,@DPC       ;START THE DPU
8734 104604  004737  027454                    JSR     PC,WAITF        ;WAIT FOR DONE
8735 104610  000207                            RTS     PC              ;RETURN TO IDLE
8736
8737 104612  117774                    10$:    APNT!ALL
8738 104614  170100                            SETMEM          ;SET THE MEMORY
8739 104616  173000                            STOPN
```

```
8741                                    ;
8742                                    ; 'BLOOM'' TEST DISPLAY
8743                                    ; THIS DISPLAY ALTERNATELY DISPLAYS A FULL WHITE SCREEN AND A PERIMETER OUTLINE.
8744                                    ; THE DISPLAY CHANGES ONCE EVERY 2 SECONDS.
8745                                    ;
8746 104620  004737  025402            BLOOM:  JSR     PC,DPINIT       ;INIT THE DPU
8747 104624  004737  102320                    JSR     PC,TBOFF        ;TURN BLINKING OFF
8748 104630  004737  027066                    JSR     PC,NTSC         ;SET UP THE COLOR TABLE (IF AVAILABLE)
8749 104634  013737  003250  104730            MOV     YMAX,12$        ;SET UP THE MAX Y VALUES, BASED ON FREQUENCY.
8750 104642  013737  003250  104740            MOV     YMAX,13$
8751 104650  052737  020000  104740            BIS     #MXY,13$
8752 104656  012777  104666  076146            MOV     #10$,@DPC       ;START THE DISPLAY & LET IT RUN.
8753 104664  000207                            RTS     PC              ;RETURN TO THE IDLE LOOP
8754
8755                                    ;
8756                                    ; DISPLAY FILE:
8757                                    ;
8758 104666  176034                    10$:    WRT!CH0         ;SET ALL CHANNELS TO WRITE-ONLY, SWITCH-ENABLED.
8759 104670  176434                            WRT!CH1
8760 104672  177034                            WRT!CH2
8761 104674  177434                            WRT!CH3
8762                                    ;LOOP TO HERE:
8763 104676  117774  000000  000000    11$:    APNT!ALL,0,0    ;SET WHITE PIXEL DATA
8764 104704  170100                            SETMEM          ;SET MEMORIES TO ALL WHITE
8765 104706  170200                            SWTCH           ;SWITCH TO READ-ONLY TO DISPLAY WHITE SCREEN
8766 104710  164170                            DNOP+120.       ;SHOW IT FOR 2 SECONDS.
8767 104712  170200                            SWTCH           ;SWITCH TO WRITE-ONLY MODE.
8768 104714  170140                            CLRMEM          ;CLEAR MEMORIES.
8769 104716  117774  000000  000000            APNT!ALL,0,0    ;START PERIMETER OUTLINE AT <0,0>
8770 104724  110000                            LVEC            ;ENTER LONG-VECTOR MODE.
8771 104726  040000                            I!0             ;DRAW LEFT EDGE ...
8772 104730  001776                    12$:    MAXY            ;... TO TOP OF VISIBLE AREA.
8773 104732  041776                            I!MAXX          ;DRAW TOP EDGE ...
8774 104734  000000                            0               ;... TO RIGHT SIDE.
8775 104736  040000                            I!0             ;DRAW RIGHT EDGE ...
8776 104740  021776                    13$:    MXY!MAXY        ;... TO BOTTOM.
8777 104742  061776                            I!MXY!MAXX      ;DRAW BOTTOM EDGE ...
8778 104744  000000                            0               ;... BACK TO <0,0>
8779 104746  170200                            SWTCH           ;SWITCH TO READ-ONLY TO DISPLAY PERIMETER
8780 104750  164170                            DNOP+120.       ;SHOW IT FOR 2 SECONDS.
8781 104752  170200                            SWTCH           ;SWITCH BACK TO WRITE-ONLY.
8782 104754  160000  104676                    DJMP,11$        ;LOOP BACK TO DO IT ALL OVER AGAIN.
8783
8784

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

```
                                        ;
                                        ;       SOME MISCELLANEOUS DISPLAY SUBROUTINES.
                                        ;
                                        ;
                                        ;
                                        ; SUBROUTINE TO OUTLINE THE PERIMETER OF THE SCREEN.
                                        ;
104762  013737  003250  105050  OUTLIN: MOV     YMAX,11$        ; SET Y COORDINATES AS PER HZ.
104770  013737  003250  105060          MOV     YMAX,12$
104776  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.LO,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!LO,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!LO
 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
```

```
                                        ;
                                        ; DISPLAY CODE.
                                        ;
8897 105372  175400                     10$:    NBLINK
8898 105374  175500                             NBLINK+100
8899 105376  175600                             NBLINK+200
8900 105400  175700                             NBLINK+300
8901 105402  173000                             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)
                                        ;
9168 110302  110702                     ACAT:   ..NULL          ;0-37 ARE NULL (NLISTED).
9174
9175         110402                             $SVPC=.         ;SAVE PC, BACK UP AND...
9176         110326                             .=ACAT+<12*2>
9177 110326  110704                     ..LF                    ;...INSERT LINE FEED...
9178         110334                             .=ACAT+<15*2>
9179 110334  110714                     ..CR                    ;...AND CARRIAGE RETURN ADDRESSES.
9180         110402                             .=$SVPC         ;RESTORE PC
9181
9182 110402  110730                     ..SPC                   ;SPACE
9183 110404  110736                     ..EXC                   ;EXCLAM
9184 110406  110754                     ..QUO                   ;DOUBLE QUOTE.
9185 110410  110772                     ..NUM                   ;NUMBER SIGN
9186 110412  111020                     ..DOL                   ;DOLLAR
9187 110414  111052                     ..PCT                   ;PER CENT
9188 110416  111112                     ..AND                   ;AMPERSAND
9189 110420  111152                     ..QUO1                  ;SINGLE QUOTE
9190 110422  111164                     ..LPAR                  ;LFT PAREN
9191 110424  111202                     ..RPAR                  ;RT PAREN
9192 110426  111220                     ..AST                   ;ASTERISK
9193 110430  111242                     ..PLS                   ;PLUS
9194 110432  111260                     ..CMA                   ;COMMA
9195 110434  111272                     ..MNS                   ;MINUS
9196 110436  111304                     ..DOT                   ;PERIOD
9197 110440  111314                     ..SLSH                  ;SLASH
9198 110442  111330                     ..00
9199 110444  111362                     ..11
9200 110446  111402                     ..22
9201 110450  111434                     ..33
9202 110452  111474                     ..44
9203 110454  111514                     ..55
9204 110456  111544                     ..66
9205 110460  111576                     ..77
9206 110462  111616                     ..88
9207 110464  111666                     ..99
9208 110466  111716                     ..COLN                  ;COLON
9209 110470  111734                     ..SEMI                  ;SEMI COLON
9210 110472  111752                     ..LANG                  ;LFT ANGLE
9211 110474  111766                     ..EQ                    ;EQUALS
9212 110476  112004                     ..RANG                  ;RT ANGLE
9213 110500  112016                     ..QUES                  ;QUESTION
9214 110502  112044                     ..ATS                   ;AT SIGN
9215 110504  112100                     ..AA
9216 110506  112124                     ..BB
9217 110510  112160                     ..CC
9218 110512  112206                     ..DD
9219 110514  112230                     ..EE
9220 110516  112236                     ..FF
9221 110520  112254                     ..GG
9222 110522  112272                     ..HH
9223 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
```

```
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
```

SYMBOL TABLE

```
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$YFS = 000010          IMCBDA  072504
DUADR   035704          ERTABE  003502          F$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          F$END = 002100          GHYF    005344          HSX     075502          IMPWPE  007234
ECMPE3  052436          E$LOAD= 000035          GHYT    067106          HSY     075272          IMREAD= 162000 G
```

SYMBOL TABLE

```
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          SCMMM0  106704
MAPMEM  070326          MRMPE   011111          O$APTS= 000000          RBL     076536          SCMMS0  106772
MARK    100442          MRPCE   010751          O$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
MFGM0   003022          NOMAN   017735          PDI     003070 G        RLMOD   057062          SELCBA= 000007 G
MFGS0   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        PRI00 = 000000 G        RPF     005175          SETDSR= 000010 G
MPE   = 114003 G        NXR     003504          PRI01 = 000040 G        RPNT  = 130000 G        SETFLG= 000012 G
MPFLE   011351          NXRERR  020032 G        PRI02 = 000100 G        RPT     056300          SETHB = 150000 G
MPLEN = 000011          NXRX    003543          PRI03 = 000140 G        RPTST   055676          SETMEM= 170100 G
MPMER   021252 G        NXTU    032206          PRI04 = 000200 G        RSTPOS= 100000 G        SETMPM= 000015 G
MPMERA  013273          ODDSKP= 000200 G        PRI05 = 000240 G        RSVDOP= 124003 G        SETMRR= 000014 G
MPMRE   011451          OPCF    004676          PRI06 = 000300 G        R256  = 000100 G        SETU    032272
MPOFF   074330          OPCFX   004724          PRI07 = 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
```

SYMBOL TABLE

```
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
```

H 4

SEQ 0254

```
..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
```