

IDENTIFICATION

PRODUCT CODE:	MAINDEC-08-DKVTA-A-D
PRODUCT NAME:	VT78 MOS MEMORY DIAGNOSTIC
DATE: RELEASED:	JUNE 1977
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	B. J. LECOQ

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.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

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

COPYRIGHT (C) 1977 BY DIGITAL EQUIPMENT CORPORATION

TABLE OF CONTENTS

1.	ABSTRACT
2.	REQUIREMENTS
2.1	EQUIPMENT
2.2	STORAGE
2.3	PRELIMINARY PROGRAMS
3.	LOADING PROCEDURE
4.	ERRORS
4.1	MARCH TEST ERROR TYPEOUTS
4.2	RELOCATION ERROR TYPEOUTS
5.	RESTRICTIONS
5.1	STARTING RESTRICTIONS
5.2	OPERATING RESTRICTIONS
6.	EXECUTION TIME
7.	PROGRAM DESCRIPTION
7.1	GENERAL
7.2	FUNCTIONAL TEST
7.3	DYNAMIC TESTS
7.4	PROGRAM RELOCATION
8.	CONSOLE PACKAGE ADDENDUM
8.1	DESCRIPTION
8.2	PROGRAM START
8.3	CONTROL CHARACTERS
8.3.1	CONTROL G
8.3.2	CONTROL S
8.3.3	CONTROL Q
8.3.4	CONTROL C
8.3.5	ILLEGAL CHARACTERS
8.4	END OF PASS REPORTING
9.	APT NOTES

1. ABSTRACT

THE VT78 MOS MEMORY TEST IS INTENDED FOR USE ON THE VT78 SYSTEM, ALTHOUGH IT WILL RUN ON MOST PDP-8 SYSTEMS. ALTHOUGH DESIGNED TO TEST 16K OF MOS MEMORY, THE PROGRAM WILL TEST SYSTEMS EQUIPPED WITH FROM 4K TO 32K WORDS OF MOS MEMORY BY CHANGING BITS 6-11 OF HCW1 (LOC 21). AUTOMATIC PROGRAM RELOCATION IS PROVIDED IN ORDER TO TEST ALL MEMORY FIELDS FROM EACH MEMORY FIELD.

SINCE THE VT78 HAS NO HARDWARE SWITCH REGISTER, THE STANDARD CONSOLE PACKAGE HAS BEEN INCLUDED IN THE PROGRAM. THIS PROVIDES AN INTERFACE BETWEEN THE OPERATOR AND THE DIAGNOSTIC VIA THE VT78 VIDEO DISPLAY-KEYBOARD TERMINAL.

THIS DIAGNOSTIC WILL RUN UNDER APT CONTROL.

2. REQUIREMENTS

2.1 EQUIPMENT

A VT78 VIDEO-KEYBOARD TERMINAL

2.2 STORAGE

THE PROGRAM OCCUPIES CORE LOCATIONS 0000 TO 3177, WITH LOCATIONS 5600 TO 7777 USED AS A BUFFER AREA.

2.3 PRELIMINARY PROGRAMS

THE VT78 RESIDENT CONTROL PANEL PROGRAM AND THE VT78 CPU DIAGNOSTIC MUST HAVE BEEN SUCCESSFULLY RUN.

3. LOADING PROCEDURE

THE PROGRAM IS NORMALLY LOADED FROM THE FLOPPY DISKETTE WHICH IS PROVIDED WITH THE RELEASE PACKAGE. THE PROGRAM MAY BE LOADED INTO ANY DESIRED CORE STACK BY HAVING BIN IN THAT CORE STACK.

4. ERRORS

THE CONTENTS OF A GIVEN MEMORY TEST LOCATION SHOULD ALWAYS BE EQUAL TO ITS ADDRESS OR THE COMPLEMENT OF ITS ADDRESS. IF IT IS NOT, A TEST ERROR WILL RESULT. A RELOCATION ERROR WILL OCCUR IF THE RELOCATION COMPARISON CHECK FAILS. FOR THE DYNAMIC TESTS THE MESSAGE "A DYNAMIC ERROR HAS OCCURED IN FIELD X" WILL BE DISPLAYED, WHERE X IS THE FIELD UNDER TEST WHEN THE ERROR OCCURED.

NOTE: THE PURPOSE OF THE DYNAMIC TESTS IS JUST TO DETECT DYNAMIC PROBLEMS IN MOS MEMORY. DUE TO THE NATURE OF THESE TESTS, PROGRAM DIAGNOSIS DOWN TO A CHIP LEVEL IS IMPOSSIBLE. THE ONLY INFORMATION WHICH MAY BE OF USE IN DEBUGGING IS LOCATION 100 OF THE FAILING FIELD. IT MAY CONTAIN THE ADDRESS +1 FROM WHICH THE PROGRAM ENTERED THE ERROR ROUTINE.

4.1 MARCH TEST ERROR TYPEOUTS

FOR THE FIRST ERROR ENCOUNTERED A HEADER WILL BE TYPED OUT FOLLOWED BY THE PERTINENT DATA. FOR ALL SUBSEQUENT ERRORS, ONLY THE PERTINENT DATA WILL BE TYPED. THE FORMAT IS AS FOLLOWS:

PR LOC ADDR GOOD BAD

PR LOC = THE PROGRAM ADDRESS WHERE THE ERROR JMS OCCURRED.
(INCLUDES FIELD)

ADDR = THE ADDRESS OF THE LOCATION IN ERROR. (INCLUDES FIELD)

GOOD = WHAT THE DATA SHOULD BE.

BAD = WHAT THE DATA IS.

4.2 RELOCATION ERROR TYPEOUTS

ALL RELOCATION ERRORS ARE IN THE FOLLOWING FORMAT:

XXXXX RELOCATION ERROR AT LOCATION YYYYY

XXXXX = THE PROGRAM ADDRESS WHERE THE ERROR JMS OCCURRED.
(INCLUDES FIELD)

YYYYY = THE ADDRESS OF THE LOCATION IN ERROR (INCLUDES FIELD)

NOTE: AFTER EACH ERROR PRINT-OUT THE PROGRAM CONTINUES ON WITH THE NEXT SEQUENTIAL MEMORY LOCATION.

5. RESTRICTIONS

5.1 STARTING RESTRICTIONS

THE PROGRAM MAY BE RESTARTED AT ANY TIME FROM LOCATION 0200 OF THE STACK THE PROGRAM IS PRESENTLY IN.

5.2 OPERATING RESTRICTIONS

NONE

6. EXECUTION TIME

THE TIME FOR ONE COMPLETE PASS IS APPROXIMATELY 1 MIN 15 SEC.

7. PROGRAM DESCRIPTION

7.1 GENERAL

THE VT78 MOS MEMORY TEST IS DESIGNED TO DETECT ANY LOCATION THAT CANNOT BE UNIQUELY ADDRESSED. FUNCTIONAL AND DYNAMIC TEST ARE PERFORMED BY THE DIAGNOSTIC (SEE 7.2 AND 7.3). ALL OPERATOR INTERACTION IS DONE USING THE CONSOLE PACKAGE (SEE SECTION 8).

THE PROGRAM AUTOMATICALLY RELOCATES ITSELF TO EACH MEMORY FIELD UNDER TEST TO ENSURE THAT ALL FIELDS MAY BE CORRECTLY REFERENCED FROM ANY FIELD. THE PROGRAM WILL NOT RELOCATE INTO A KNOWN BAD FIELD.

CONTROL OF THE PROGRAM IS GIVEN TO THE OPERATOR BY MEANS OF THE PSEUDO SWITCH REGISTER (PSR). THE OPERATOR MAY HALT AFTER ERROR, INHIBIT ERROR PRINTOUTS, LOOP ON ERROR OR TESTS, HALT AFTER PASS, SELECT ALL OR ANY OF THE TESTS, AND AT ANY TIME RESTART THE PROGRAM USING THE CONSOLE PACKAGE.

7.2 FUNCTIONAL TEST

THE TEST USED TO TEST THE FUNCTIONALITY OF THE MEMORY IS THE MARCH TEST. THE TEST RUNS ALL FIELDS IN THE SYSTEM REGARDLESS OF OPERATOR SPECIFICATIONS. THIS TEST ENSURES THAT A ONE AND A ZERO CAN BE WRITTEN INTO EACH MEMORY CELL.

7.3

DYNAMIC TEST

THE PROGRAM CONTAINS THREE DYNAMIC TESTS WHICH CHECK FOR MULTIPLE SELECTION OF ADDRESS DECODERS, SLOW RECOVERY OF SENSE AMPLIFIERS, AND READ ACCESS TIME.

7.4

PROGRAM RELOCATION

PROGRAM RELOCATION IS CONTROLLED BY THE PROGRAM. THE PROGRAM INITIALLY TESTS ALL STACKS EXCEPT FOR THE ONE IT WAS LOADED IN. IT THEN RELOCATES TO THE NEXT HIGHEST ERROR FREE STACK, AND TEST ALL OTHER STACKS AGAIN. THIS IS REPEATED UNTIL THE PROGRAM HAS BEEN RELOCATED INTO EVERY ERROR FREE STACK. ONCE RESIDING IN THE HIGHEST STACK IN THE SYSTEM THE PROGRAM RELOCATES TO THE LOWEST ERROR FREE STACK.

THE PROGRAM PROVIDES A DEGREE OF PROTECTION FOR ITSELF BY REMEMBERING ALL STACKS WHERE ERRORS OCCUR. WHEN A FAULTY STACK IS NEXT IN SEQUENCE TO CONTAIN THE PROGRAM, THE PROGRAM WILL SKIP THE FAULTY STACK AND RELOCATE TO THE NEXT HIGHEST ORDER STACK WHICH IS ERROR FREE. IF ALL OTHER SELECTED STACKS ARE FAULTY, PROGRAM RELOCATION WILL NOT TAKE PLACE.

DURING RELOCATION A COMPARISON CHECK IS MADE TO INSURE NO PROGRAM LOSS

FOR FURTHER UNDERSTANDING OF HOW THE TESTS ARE PERFORMED, REFER TO THE LISTING.

8.0 CONSOLE PACKAGE ADDENDUM

8.1 DESCRIPTION

A CONSOLE PACKAGE HAS BEEN ADDED TO THIS DIAGNOSTIC TO ALLOW THE PROGRAM TO RUN WITH NO HARDWARE SWITCH REGISTER AND TO HAVE COMMUNICATIONS WITH THE DIAGNOSTIC VIA A TERMINAL. THE CONSOLE PACKAGE ALLOWS THE OPERATOR CONTROL OF THE DIAGNOSTIC THROUGH THE TERMINAL. THE DIAGNOSTIC WILL ASK FOR THE VALUE OF THE PSEUDO SWITCH REGISTER BEFORE CONTINUING WITH THE EXECUTION OF THE DIAGNOSTIC. ALL ERRORS AND PASS COMPLETES WILL BE PRINTED AT THE TERMINAL. NO HALTS WILL BE EXECUTED.

8.2 PROGRAM START

THE PROGRAM WILL TYPEOUT A HEADING AND THEN PRINT SR=XXXX. XXXX IS THE VALUE OF THE SWITCH REGISTER USED. SELECT A VALUE TO USE FOR THE SWITCH REGISTER FROM THE SWITCH REGISTER TABLE BELOW AND INPUT THE NUMBER USING THE CONSOLE KEYBOARD. TYPING IN FOUR DIGITS WILL CAUSE THE PROGRAM TO ECHO THE NEW VALUE OF THE SWITCH REGISTER AND START THE PROGRAM. TYPING A CARRIAGE RETURN AFTER INPUTTING A DIGIT WILL ALSO CAUSE THE PROGRAM TO ECHO THE NEW VALUE OF THE SWITCH REGISTER AND START THE PROGRAM. IF IT IS NOT DESIRED TO CHANGE THE SWITCH REGISTER VALUE, A CARRIAGE RETURN CAN BE TYPED AND THE PROGRAM WILL BE STARTED WITHOUT ECHOING THE SWITCH REGISTER VALUE.

PSEUDO SWITCH REGISTER

4000	BIT 0=1	INHIBIT ERROR HALT
2000	BIT 1=1	LOOP ON ERROR
1000	BIT 2=1	LOOP ON TEST
0400	BIT 3=1	HALT ON COMPLETION OF PROGRAM PASS
0200	BIT 4=1	INHIBIT ERROR TYPEOUT
0100	BIT 5=1	DON'T RUN MARCH TEST
0040	BIT 6=1	DON'T RUN DYNAMIC TESTS
0020	BIT 7=1	RING BELL ON ERROR

8.3 CONTROL CHARACTERS

8.3.1 CONTROL G

TO GAIN CONTROL OF THE CONSOLE SWITCH REGISTER PACKAGE WHILE RUNNING THE PROGRAM, A CONTROL G MUST BE TYPED IN ON THE CONSOLE TERMINAL. WHEN CONTROL G IS TYPED THE PROGRAM WILL RESPOND BY TYPING AN UP ARROW THEN G FOLLOWED BY SR=XXXX. XXXX IS THE PRESENT CONTENTS OF THE PSEUDO SWITCH REGISTER. THE OPERATOR CAN NOW CHANGE THE SETTING OF THE SWITCH REGISTER BY TYPING IN A NEW NUMBER OR NOT CHANGE IT BY TYPING IN A CARRIAGE RETURN. TYPING IN FOUR DIGITS WILL CAUSE THE PROGRAM TO ECHO THE SWITCH REGISTER VALUE TYPED AND CONTINUE THE PROGRAM USING THE VALUE TYPED AS THE NEW SWITCH REGISTER. TYPING IN ONE TO THREE DIGITS FOLLOWED BY A CARRIAGE RETURN WILL ALSO CAUSE THE PROGRAM TO ECHO THE NEW VALUE OF THE SWITCH REGISTER TYPED AND CONTINUE THE PROGRAM. TYPING ONLY A CARRIAGE RETURN WILL CAUSE THE PROGRAM TO CONTINUE WITHOUT CHANGING THE SWITCH REGISTER, AND THE PROGRAM WILL NOT ECHO THE SWITCH REGISTER VALUE. BY TYPING A LINE FEED, THE PROGRAM WILL BE RELOCATED BACK TO THE ORIGINAL PROGRAM FIELD AND RESTARTED.

TYPING A CONTROL C WILL CAUSE THE PROGRAM TO RETURN TO THE OPERATING SYSTEM (OS/8) AT 7600 IN FIELD 0. ILLEGAL CHARACTERS TYPED WILL RESULT IN THE CHARACTER BEING ECHOED FOLLOWED BY A QUESTION MARK AND THE SWITCH REGISTER MESSAGE BEING RETYPED.

8.3.2 CONTROL S

THIS IS A CONTROL CHARACTER TO STOP SENDING DATA TO A TERMINAL. IT IS USUALLY A FUNCTION OF THE TERMINAL AND IS AUTOMATICALLY SENT WHEN THE TERMINAL BUFFER IS FULL. THE BUFFER IS EMPTIED AS THE DATA IS PRINTED. AFTER ALL THE DATA IN THE TERMINAL BUFFER IS PRINTED THE TERMINAL SENDS A CONTROL Q (START SENDING DATA TO TERMINAL). THE CONTROL S IS NOT ECHOED.

BY TYPING A CONTROL S WHILE RUNNING THE DIAGNOSTIC, THE PROGRAM, WHEN THE CONTROL S IS RECOGNIZED, WILL WAIT FOR A CONTROL Q TO CONTINUE THE PROGRAM, A LINE FEED TO RESTART THE PROGRAM, OR A CONTROL C TO RETURN TO THE OPERATING SYSTEM.

8.3.3 CONTROL Q

THIS CONTROL CHARACTER IS TO RESUME SENDING DATA TO THE TERMINAL. IT IS USUALLY SENT AUTOMATICALLY BY A TERMINAL WHEN IT IS READY TO ACCEPT MORE DATA. THIS CONTROL CHARACTER IS NOT ECHOED.

BY TYPING A CONTROL Q AFTER A CONTROL S HAS BEEN TYPED, THE PROGRAM WILL CONTINUE FROM WHERE IT WAS INTERRUPTED FROM.

8.3.4 CONTROL C

THIS CONTROL CHARACTER IS USED TO RETURN CONTROL BACK TO AN OPERATING SYSTEM KEYBOARD MONITOR. THE OPERATING SYSTEM SELECTED FOR THE PDP-8 IS THE OS/8 SYSTEM WITH ITS BOOTSTRAP LOCATED IN THE LAST PAGES OF FIELD 0 AND 1. WHEN THIS CONTROL CHARACTER IS RECOGNIZED, THE PROGRAM WILL ECHO THE CONTROL CHARACTER AS AN UP ARROW THEN C, RESTORE THE LAST PAGE OF FIELD 0 AND 1, AND JUMP TO LOCATION 7600 IN FIELD 0.

8.3.5 ILLLEGAL CHARACTERS

A CHARACTER TYPED ON THE KEYBOARD, OTHER THAN A CONTROL G, CONTROL S, OR A CONTROL C, WILL RESULT IN THE CHARACTER BEING ECHOED FOLLOWED BY A QUESTION MARK AND THE PROGRAM WILL BE CONTINUED.

8.4

END OF PASS REPORTING

THERE WILL BE AN END OF PASS MESSAGE CONTAINING THE DIAGNOSTIC NAME, THE END OF PASS MESSAGE AND AN OCTAL NUMBER OF PASSES.

9.

APT NOTES

ALL OF THE FOLLOWING NOTES APPLY ONLY WHEN THE PROGRAM IS BEING RUN ON AN APT SYSTEM.

1. FOR MORE INFORMATION SEE THE FOLLOWING DOCUMENTS.
 - A. STANDARD APT SYSTEM TO PDP8 DIAGNOSTIC INTERFACE.
 - B. APT SYSTEM MANAGERS GUIDE.

2. IF BIT 0 OF HCW1(LOC 21) IS A '1' THEN THE PROGRAM WILL BE RUN UNDER APT CONTROL. ALL OUTPUT FROM THE PROGRAM WILL BE SUPPRESSED.

3. APT IS NOTIFIED THAT THE PROGRAM IS RUNNING WITHOUT ERRORS BETWEEN EVERY .2 SECONDS AND 4.0 SECONDS.

4. ERRORS:

ALL ERRORS CALL APT.
ONLY THE ERROR PC IS REPORTED TO APT-8.
THE TYPE OF ERROR CAN BE DETERMINED FROM THE CORRESPONDING ADDRESS IN THE PROGRAM LISTING.

LISTING

```

1 /DKVTAA MOS MEMORY DIAGNOSTIC MAINDEC-08-DKVTA-A-D
2 /16K MOS MEMORY TEST FOR VT78 (VER A)
3 /COPYRIGHT 1977 DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASS 01754
4 /PROGRAMMER: ROBERT J LFCOQ
5 /
6 /
7 /
8 /
9 /
10 /
11 /
12 /
13 /IOT COMMANDS
14 6201 CDF0=6201 /CHANGE TO DATA FIELD 0
15 6211 CDF1=6211 /CHANGE TO DATA FIELD 1
16 6221 CDF2=6221 /CHANGE TO DATA FIELD 2
17 6231 CDF3=6231 /CHANGE TO DATA FIELD 3
18 6203 CBF0=6203 /CHANGE TO DATA AND INSTRUCTION FIELD 0
19 6213 CBF1=6213 /CHANGE TO DATA AND INSTRUCTION FIELD 1
20 6223 CBF2=6223 /CHANGE TO DATA AND INSTRUCTION FIELD 2
21 6233 CBF3=6233 /CHANGE TO DATA AND INSTRUCTION FIELD 3
22 6214 RDF=6214 /READ DATA FIELD INTO AC BITS 6-8
23 6224 RIF=6224 /READ INSTRUCTION FIELD INTO AC BITS 6-8
24
25
26
27
28
29 0000 0000 *0
30 0000 0301 301
31 0001 5001 JMP
32
33
34 0020 *20
35
36 0020 0000 PSR, 0 /PSEUDO SWITCH REGISTER
37 /4000 BIT 0=1 INHIBIT ERROR HALT
38 /2000 BIT 1=1 LOOP ON ERROR
39 /1000 BIT 2=1 LOOP ON TEST
40 /0400 BIT 3=1 HALT ON COMPLETION OF PROGRAM PASSES
41 /0200 BIT 4=1 INHIBIT ERROR TYPEOUT
42 /0100 BIT 5=1 DON'T RUN MARCH TEST
43 /0040 BIT 6=1 DON'T RUN DYNAMIC TEST
44 /0020 BIT 7=1 BELL ON ERROR
45 0021 4017 HCW1, 4017 /APT/ LAST TWO DIGITS INDICATE MEMORY SIZE
46 0022 0000 HCW2, 0 /APT/
47 0023 7000 NOP
48
49 4424 LISN=JMS I.
50 0024 2675 XLISN
51 4425 PRNTMS=JMS I.
52 0025 3000 MESAGX
53 4426 ONEOCT=JMS I.
54 0026 2601 ONEOCK
55 4427 TWOUCT=JMS I.

```

```

56 0027 2611 TWOOCK
57 4430 PRNT1=JMS I.
58 0030 3066 XPRNT1
59 4431 FOROCT=JMS I.
60 0031 3037 FOROCK
61 4432 PRNT2=JMS I.
62 0032 2624 XPRNT2
63 4433 PRNT4=JMS I.
64 0033 3053 XPRNT4
65 4434 SPACE2=JMS I.
66 0034 2667 SPACX2
67 4435 TYPE=JMS I.
68 0035 2635 XTYPE
69 4436 CRLF=JMS I.
70 0036 2652 XCRLF
71
72
73
74 /
75 /CONSTANTS AND POINTERS
76 0037 7777 HEAD1, 7777 /ERROR HEADING CONTROL
77 0040 0000 TESTAD, 0 /TEST ADDRESS COUNTER
78 0041 0000 TOPSTK, 0 /HIGHEST ACTUAL STACK IN SYSTEM
79 0042 0000 STKPIN, 0 /STACK PROG IS IN 00X0
80 0043 0000 STKTST, 0 /STACK SEL FOR TEST 00X0
81 0044 0000 BDATA, 0 /BAD DATA
82 0045 0000 GDATA, 0 /GOOD DATA
83 0046 0000 MOVE, 0 /RELOCATION ADDRESS
84 0047 0000 COUNT, 0 /ERROR COUNTER
85 0050 0000 COLCNT, 0 /COLUMN COUNTER - DYNAMIC TEST
86 0051 0200 CONST, 200 /EITHER + OR = 200 DYNAMIC TESTS
87
88
89 0052 7741 M37, -37
90 0053 7703 M75, -75
91 0054 0002 K2, 2
92 0055 0003 K3, 3
93 0056 0007 K7, 7
94 0057 0010 K10, 10
95 0060 0020 K20, 20
96 0061 0030 K30, 30
97 0062 0077 K77, 77
98 0063 0100 K100, 100
99 0064 0200 K200, 200
100 0065 0207 K207, 207
101 0066 0212 K212, 212
102 0067 0215 K215, 215
103 0070 0240 K240, 240
104 0071 0260 K260, 260
105 0072 0261 K261, 261
106 0073 1000 K1000, 1000
107 0074 4000 K4000, 4000
108 0075 4060 K4060, 4060
109 0076 6201 K6201, 6201 /CDF 0
110 0077 6203 K6203, 6203

```

```

111 0100 7577 K7577, 7577
112 0101 7600 K7600, 7600 /DYNAMIC TEST
113 0102 7700 K7700, 7700 /DYNAMIC TEST
114 0103 1713 XSTOP, STOP /STOP ROUTINE POINTER
115
116
117 0104 2523 KSFCHK, CHKKSF
118 0105 2255 IAPTEP, APTER /APT/
119 0106 2234 IAPTOK, APTOK /APT/
120
121 0200 *200
122 /
123 /VT78 MOS MEMORY TEST
124 /
125
126 0200 7300 START, CLA CLL
127 0401 6002 IDF /TURN INTERRUPT OFF
128 0202 1022 TAD HCW2
129 0203 7710 SPA CLA /RUN UNDER APT CONTROL?
130 0204 5777 JMP XAPT /YES - SET UP FOR APT
131 0205 4776 JMS SAVOS8 /NO - SAVE OS8 HANDLER
132 0206 6224 XSTRT, RIF
133 0207 1076 TAD K6201
134 0210 3216 DCA PRGSTK /MAKE INSTR FIELD = DATA FIELD
135 0211 4425 PRNTMS
136 0212 0314 TITLE /TYPEOUT PROGRAM TITLE
137 /APT/ IF UNDER APT CONTROL NEXT LOC WILL = NOP.
138
139 APTN00, /APT/
140
141 0213 4775 JMS PSEUDO /TYPEOUT PSEUDO SWITCH REGISTER
142 0214 7300 BEGIN, CLA CLL
143 0215 4504 JMS I KSFCHK /CHECK IF CONSOLE ACTIVE
144 0216 6201 PRGSTK, CDF0
145 0217 1021 TAD HCW1
146 0220 0257 AND K34 /SELECT TOP STACK-USUALLY 16K
147 0221 7012 RTR
148 0222 3041 DCA TOPSTK
149 /APT/ IF UNDER APT CONTROL NEXT LOC WILL = NOP.
150 APTN01,
151 0223 4774 JMS TSTSYS /TYPE NUMBER OF STACKS IN SYSTEM
152 0224 4425 PRNTMS
153 0225 0331 PREL /TYPE PROGRAM WILL RELOCATE
154 0226 3276 DCA PASCNT
155 0227 4277 JMS INIT /CLEAR FIELD STATUS LIST
156 0230 1041 NXTPAS, TAD TOPSTK
157 0231 7040 CMA
158 0232 3256 DCA KOUNT
159 0233 7240 STA
160 0234 3037 DCA HEAD1 /RESET ERROR HEADING
161 0235 6224 TEST, RIF
162 0236 3042 DCA STKPIN /SET UP CURRENT PROGRAM FIELD
163 0237 4425 PRNTMS
164 0240 0546 PRFLD /TYPE PROGRAM FIELD MESSAGE
165 0241 6224 RIF

```

```

166 0242 7112 RTR CLL
167 0243 7010 RAR
168 0244 1071 TAD K260 /FIELD PROGRAM IN
169 0245 4435 TYPE
170 0246 4436 CRLF
171 0247 4773 JMS MOSTST /GO EXECUTE MARCH TEST
172 0250 4772 JMS DYNST /GO EXECUTE DYNAMIC TESTS
173 0251 4771 JMS RELO /GO RELOCATE PROGRAM
174 0252 2256 ISZ KOUNT /HAVE WE RELOCATED TO ALL FIELDS YET?
175 0253 5235 JMP TEST /NO
176 0254 4260 JMS PCOUNT /YES - DONE ONE PASS
177 0255 5230 JMP NXTPAS
178 0256 0000 KOUNT, 0
179 0257 0034 K34, 34
180
181
182 /
183 0260 0000 PCOUNT, 0
184 /APT/ IF UNDER APT CONROL NEXT LOC WILL = JMS I IAPTOK.
185
186 0261 4504 APTOK, JMS I KSFCHK /THIS LOC OVERLAID IF APT
187 0262 7000 NOP /
188 0263 4425 PRNTMS /PRINT END OF PASS MESSAGE
189 0264 0345 PASHES /POINTER TO MESSAGE
190 0265 2276 ISZ PASCNT /ADD 1 TO THE PASS COUNTER
191 0266 1276 TAD PASCNT /GET THE COUNTER
192 0267 4433 PRNT4 /PRINT THE 4 OCTAL DIGITS
193 0270 4436 CRLF
194 0271 1020 TAD PSR
195 0272 0371 AND (400 /HALT ON PASS COMPLETION?
196 0273 7640 SZA CLA
197 0274 4775 JMS PSEUDO /YES - GO ASK SWITCH REGISTER
198 0275 5660 JMP I PCOUNT /NO - RETURN
199
200 0276 0000 PASCNT, 0
201
202 /ROUTINE TO CLEAR FIELD STATUS LIST (FLDLST)
203 /
204 0277 0000 INIT, 0
205 0300 1313 TAD M7
206 0301 3312 DCA CLRCNT /SETUP COUNTER TO MINUS 7
207 0302 1370 TAD (FLDLST /GET ADDRESS OF LIST
208 0303 3311 DCA CLRLST /SAVE IT
209 0304 3711 DCA I CLRLST /CLEAR CONTENTS OF ADDRESS
210 0305 2311 ISZ CLRLST /NEXT ADDRESS
211 0306 2312 ISZ CLRCNT /ARE WE DONE?
212 0307 5304 JMP ,-3 /NO - DO NEXT ADDRESS
213 0310 5677 JMP I INIT /YES - RETURN
214
215 0311 0000 CLRLST, 0
216 0312 0000 CLRCNT, 0 /MINUS NUMBER OF POSSIBLE STACKS
217 0313 7771 M7, -7
218
219 0314 4343 TITLE, TEXT "##DKVTAA MOS MEMORY TEST#"
0315 0413

```

```

0316 2624
0317 0101
0320 4015
0321 1723
0322 4015
0323 0515
0324 1722
0325 3140
0326 2405
0327 2324
220 0330 4300 PREL, TEXT "PROGRAM WILL RELOCATE##"
0331 2022
0332 1707
0333 2201
0334 1540
0335 2711
0336 1414
0337 4022
0340 0514
0341 1703
0342 0124
0343 0543
221 0344 4300 PASMES, TEXT "#DKVTAA - END OF PASS "
0345 4304
0346 1326
0347 2401
0350 0140
0351 5540
0352 0516
0353 0440
0354 1706
0355 4020
0356 0123
0357 2340
0360 0000

222
223 0370 1725
224 0371 0400
225 0372 1010
226 0373 0600
227 0374 0523
228 0375 2440
229 0376 2267
230 0377 2277
PAGE
231
232 /
233 /
234 /
235 /RELOCATE THE PROGRAM
236 /
237 0400 0000 RELO, 0
238 0401 7300 CLA CLL
239 0402 3047 DCA COUNT /CLEAR ERROR COUNTER
240 0403 4243 JMS BADSTK /GO CHECK IF STACK IS GOOD

```

```

241 0404 1300 TAD NXSTK /SETUP STACK TO MOVE INTO
242 0405 7106 RTL CLL
243 0406 7004 RAL
244 0407 1076 TAD K6201
245 0410 3224 DCA RELO2 /MODIFY CDF INSTRUCTION AT LOC RELO2
246 0411 1042 TAD STKPIN /SETUP STACK MOVING FROM
247 0412 1076 TAD K6201
248 0413 3222 DCA RELO1
249 0414 1222 TAD RELO1
250 0415 3227 DCA RELO3
251 0416 1224 TAD RELO2
252 0417 1054 TAD K2
253 0420 3241 DCA RELO4
254 0421 3046 DCA MOVE
255 0422 6201 RELO1, CDF0 /MOVE FROM DATA FIELD
256 0423 1446 TAD I MOVE /GET WORD
257 0424 6201 RELO2, CDF0 /MOVE TO DATA FIELD
258 0425 3446 DCA I MOVE /SAVE IT
259 0426 1446 TAD I MOVE
260 0427 6201 RELO3, CDF0 /MOVE FROM DATA FIELD
261 0430 7041 CIA
262 0431 1446 TAD I MOVE /WAS TRANSFER OK?
263 0432 7640 SZA CLA /YES - SKIP
264 0433 4777 JMS ERRM /NO - MOVE ERROR
265 0434 2046 ISZ MOVE /DONE 4096 TIMES?
266 0435 5222 JMP RELO1 /NO - GET NEXT WORD
267 0436 1047 TAD COUNT
268 0437 7640 SZA CLA /SKIP IF NO MOVE ERROR
269 0440 5600 JMP I RELO /RETURN -WITHOUT CHANGING TO NEW DATA FIELD
270 0441 6203 RELO4, CBFO /CHANGE TO NEW PROG FIELD
271 0442 5600 JMP I RELO
272 /
273 /
274 /ROUTINE TO INHIBIT RELOCATION INTO BAD STACK
275 /
276 /
277
278 0443 0000 BADSTK, 0
279 0444 1042 TAD STKPIN /STACK PROGRAM IS IN
280 0445 7012 RTR
281 0446 7010 RAR
282 0447 3300 DCA NXSTK
283 0450 1041 TAD TOPSTK /GET HIGHEST STACK IN SYSTEM
284 0451 7041 CIA
285 0452 3301 DCA BADCNT /SET UP BAD FIELD COUNTER
286 0453 1041 X, TAD TOPSTK /TOP STACK
287 0454 7041 CIA
288 0455 1300 TAD NXSTK
289 0456 7650 SNA CLA /ARE WE AT HIGHEST STACK
290 0457 5275 JMP ATTOP /YES
291 0460 2300 ISZ NXSTK /NO
292 0461 1300 TAD NXSTK
293 0462 1376 CHECK, TAD (FLDLST /LIST CONTAINING STATUS OF FIELDS TESTED
294 0463 3277 DCA BADPTR /SAVE POINTER
295 0464 1677 TAD I BADPTR

```

```

296 0465 7650 SNA CLA /IS FIELD SELECTED GOOD?
297 0466 5643 JMP I BADSTK /YES - RETURN
298
299 0467 2301 ISZ BADCNT /NO- SKIP IF ALL FIELDS TEST BAD
300 0470 5253 JMP X /- TRY NEXT STACK
301 0471 4425 PRNTMS
302 0472 0502 ALLBAD
303 0473 4775 JMS PSEUDD /GO ASK SW REG QUESTION,
304 0474 5201 JMP RELO+1 /TRY AGAIN
305
306 0475 3300 ATTOP, DCA NXSTK /SET FOR FIELD 0
307 0476 5262 JMP CHECK
308
309 0477 0000 BADPTR, 0
310 0400 0000 NXSTK, 0
311 0501 0000 BADCNT, 0
312
313 0402 0301 ALLBAD, TEXT "CANNOT RELOCATE INTO ANY FIELD!!"
0503 1616
0504 1724
0405 4022
0406 0514
0507 1703
0510 0124
0411 0540
0412 1116
0513 2417
0514 4001
0515 1631
0416 4006
0517 1105
0420 1404
0521 4141
0522 4300
314 /
315 /
316 /TYPEOUT NUMBER OF STACKS IN SYSTEM
317 /
318 0523 0000 TSTSYS, 0
319 0524 1041 TAD TOPSTK
320 0525 1072 TAD K261
321 0526 4435 TYPE /TYPEOUT NUMBER
322 0527 4425 PRNTMS
323 0430 0532 STACKS
324 0431 5723 JMP I TSTSYS
325
326 0432 4023 STACKS, TEXT " STACKS IN THIS SYSTEM"
0433 2401
0534 0313
0535 2340
0436 1116
0437 4024
0440 1011
0541 2340
0442 2331
    
```

```

0543 2324
0444 0515
0545 4300
327
328 0546 2022 PRFLD, TEXT "PROG FIELD = "
0547 1707
0450 4006
0551 1105
0452 1404
0553 4075
0554 4000
329
330 0575 2440
331 0576 1725
332 0577 2000
0600
333 PAGE
334 /
335 /THE FOLLOWING TEST IS A MARCH PATTERN DEVELOPED FOR TESTING
336 /THE MS8-C MOS MEMORY, THE TEST RUN ALL FIELDS IN THE SYSTEM.
337 /
338 / THE TEST SELECTED FOR THE MOS MEMORY TESTING IS A TYPICAL MARCH
339 / PATTERN, THE TEST BEGINS BY LOADING THE ENTIRE MEMORY WITH
340 / A 2525 PATTERN, THEN STARTING AT ADDRESS ZERO OF LOWEST POSSIBLE
341 / FIELD THE TEST READS THE CONTENTS, COMPARES IT, AND THEN WRITES BACK
342 / THE COMPLIMENT VALUE, THE PROCESS IS REPEATED THROUGHOUT THE ENTIRE
343 / MEMORY.
344 /
345 / NEXT THE PROCESS REPEATS FROM MAXIMUM TO MINIMUM, COMPLIMENTING
346 / AS IT IS BEING DONE.
347 /
348 / THE ENTIRE SEQUENCE IS THEN REPEATED USING A BACKGROUND OF
349 / 5252. THIS INSURES THAT A ONE AND A ZERO CAN BE WRITTEN INTO
350 / EACH MEMORY CELL.
351 /
352 0600 0000 MOSTST, 0
353 0601 1020 TAD PSR /SEE IF MOSTST TO BE RUN
354 0602 0063 AND K100
355 0603 7640 SZA CLA /MOSTST TO BE RUN
356 0604 5600 JMP I MOSTST /NO, EXIT
357 0605 4425 PRNTMS
358 0606 1600 MARCH /PRINT MARCH TEST MESSAGE
359 0607 7344 CLL CLA CMA RAL /-2
360 0610 3372 DCA PATCNT /SET UP LOOP COUNTER FOR BACK GROUND PATTERN
361 0611 4777 JMS SETPAT /GO SETUP INITIAL VALUE OF PAT1 & PAT2
362 0612 3040 DCA TESTAD /START WITH ADDRESS ZERO
363 0613 1041 TAD TOPSTK /GET MAX VALUE IN SYSTEM,
364 0614 7040 CMA
365 0615 3365 DCA FLDCNT /SAVE FOR LOOP COUNTER,
366 0616 7344 CLL CLA CMA RAL /-2
367 0617 3370 DCA TSTCNT
368 0620 3043 DCA STKTST /START WITH FIELD ZERO
369 0621 1042 TAD STKPIN
370 0622 1076 TAD K6201 /SET UP RETURN DATA FIELD
371 0623 3234 DCA NOSFLD
    
```


372	0624	4776*	MOSLUP, JMS	SAME	/SEE IF STACK SELECTED IS SAME AS
373					/PROGRAM FIELD.
374	0625	5243	JMP	STKUPD	/SAME STACK AS PROGRAM, JUST UPDATE THE
375					/FIELD TO NEXT ONE SELECTED.
376	0626	1043	TAD	STKTST	
377	0627	1076	JAD	K6201	/SET UP STACK TO LOAD.
378	0630	3232	DCA	*+2	
379	0631	1373	MEMLOD, TAD	PAT1	/LOAD BACKGROUND PATTERN INTO
380					/STACK SELECTED.
381	0632	6201	DCF		/THIS IS CHANGED TO LOAD STACK.
382	0633	3440	DCA I	TESTAD	
383	0634	6201	MOSFLD, CDF		/CHANGED BACK TO STACK OF PROGRAM.
384	0635	1040	TAD	TESTAD	
385	0636	7001	IAC		/UPDATE TEST ADDRESS
386	0637	3040	DCA	TESTAD	
387	0640	1040	TAD	TESTAD	
388	0641	7640	SZA CLA		/HAS IT GONE BACK TO ZERO YET?
389	0642	5231	JMP MEMLOD		/NO, GO BACK AND DO NEXT ADDRESS
390					
391					/NOW UPDATE THE FIELD TO TEST.
392	0643	1043	STKUPD, TAD	STKTST	
393	0644	1057	TAD	K10	
394	0645	3043	DCA	STKTST	/SAVE NEW OFFSET
395	0646	2365	ISZ	FLDCNT	/ALL FIELDS AVAILABLE DONE?
396	0647	5224	JMP	MOSLUP	/NO, GO DO NEXT.
397					/AT THIS POINT ALL AVAILABLE MEMORY IS FILLED
398					/WITH THE BACKGROUND PATTERN.
399					
400					/START READING MEMORY.
401	0650	1057	TAD	K10	
402	0651	3367	DCA	FLDINC	/SET UP INITIAL VALUE
403	0652	7301	CLL CLA IAC		
404	0653	3366	DCA	ADDINC	/INITIAL ADDRESS VALUE
405	0654	3043	DCA	STKTST	/START WITH FIELD ZERO
406	0655	1042	TAD	STKPIN	
407	0656	1076	TAD	K6201	/SET UP PROGRAM RETURN VALUE
408	0657	3300	DCA	REDFLD	
409	0660	1041	TAD	TOPSTK	/SET UP LOOP COUNTER
410	0661	7040	CMA		
411	0662	3365	DCA	FLDCNT	
412	0663	4776*	REDLUP, JMS	SAME	/PROGRAM FIELD = TEST FIELD?
413	0664	5313	JMP	REDUPD	/YES UPDATE TO NEW FIELD
414	0665	4504	JMS I	KSFCHK	/CHECK IF CONSOLE ACTIVE
415	0666	1373	TAD	PAT1	/SET UP COMPARE VALUE
416	0667	3045	DCA	GDATA	
417	0670	1043	TAD	STKTST	
418	0671	1076	TAD	K6201	/SET UP FIELD TO TEST
419	0672	3273	DCA	*+1	
420	0673	6201	DCF		/CHANGED TO TEST FIELD
421	0674	1440	TAD I	TESTAD	/GET VALUE STORED
422	0675	3044	DCA	BDATA	
423	0676	1374	TAD	PAT2	/CHANGE VALUE IN SELECTED ADDRESS
424	0677	3440	DCA I	TESTAD	
425	0700	6201	REDFLD, CDF		/RESTORE TO PROGRAM FIELD
426	0701	1045	TAD	GDATA	

427	0702	7041	CIA		/DO COMPARE
428	0703	1044	TAD	BDATA	/ACTUAL = EXPECTED
429	0704	7640	SZA CLA		/NO REPORT AN ERROR
430	0705	4775*	JMS	MOSERR	
431	0706	1040	TAD	TESTAD	
432	0707	1366	TAD	ADDINC	/UPDATE ADDRESS TO DO
433	0710	3040	DCA	TESTAD	
434	0711	2371	ISZ	ADDNT	/4096 LOOPS DONE
435	0712	5265	JMP	REDLUP+2	/NO GO BACK AND REPEAT FOR ALL FIELDS
436	0713	1043	REDUPD, TAD	STKTST	
437	0714	1367	TAD	FLDINC	/UPDATE TEST FIELD
438	0715	3043	DCA	STKTST	/SAVE THE NEW VALUE
439	0716	2365	ISZ	FLDCNT	/ALL FIELDS DONE YET
440	0717	5263	JMP	REDLUP	/NO, THEN CONTINUE
441	0720	1367	TAD	FLDINC	
442	0721	7041	CIA		/UPDATE OFFSET
443	0722	3367	DCA	FLDINC	
444	0723	1366	TAD	ADDINC	
445	0724	7041	CIA		
446	0725	3366	DCA	ADDINC	/UPDATE ADDRESS OFFSET
447	0726	1366	TAD	ADDINC	
448	0727	7700	SMA CLA		/READING BACKWARDS
449	0730	5334	JMP	*+4	/NO.
450	0731	1040	TAD	TESTAD	
451	0732	1366	TAD	ADDINC	/START AT ADDRESS 7777
452	0733	3040	DCA	TESTAD	
453	0734	1367	TAD	FLDINC	
454	0735	1043	TAD	STKTST	/GET TO STARTING FIELD
455	0736	3043	DCA	STKTST	
456	0737	1373	TAD	PAT1	/UPDATE PATTERN
457	0740	7040	CMA		
458	0741	3373	DCA	PAT1	
459	0742	1374	TAD	PAT2	
460	0743	7040	CMA		
461	0744	3374	DCA	PAT2	
462	0745	7000	APTK1, NOP		/APT/
463	0746	2370	ISZ	TSTCNT	/ALL DONE THIS TEST
464	0747	5260	JMP	REDLUP=3	
465	0750	1373	TAD	PAT1	
466	0751	7040	CMA		
467	0752	3373	DCA	PAT1	
468	0753	1374	TAD	PAT2	
469	0754	7040	CMA		
470	0755	3374	DCA	PAT2	/SET UP NEXT PATTERN
471	0756	2372	ISZ	PATCNT	/SEE IF ALL DONE?
472	0757	5212	JMP	MOSLOD	/NO, GO DO NEXT PATTERN
473	0760	1020	TAD	PSR	/YES
474	0761	0073	AND	K1000	/TEST BIT 2
475	0762	7640	SZA CLA		/LOOP ON TEST?
476	0763	5207	JMP	MOSLOD=3	/YES - GO BACK TO START OF TEST
477	0764	5600	JMP I	MOSTST	/NO - RETURN TO PROGRAM.
478					
479	0765	0000	FLDCNT,	0	
480	0766	0000	ADDINC,	0	
481	0767	0000	FLDINC,	0	

```

482 0770 0000 TSCNT, 0
483 0771 0000 ADDCNT, 0
484 0772 0000 PATCNT, 0
485 0773 2525 PAT1, 2525
486 0774 5252 PAT2, 5252
487
488 0775 1636
489 0776 1144
490 0777 1000
491 PAGE
492 /SETUP INITIAL VALUE OF PAT1 AND PAT2
493 /
494 1000 0000 SETPAT, 0
495 1001 1377 TAD (2525 /CHECKERBOARD PATTERN
496 1002 3776 DCA PAT1 /SAVE IT
497 1003 1776 TAD PAT1
498 1004 7040 CMA /COMPLEMENT IT
499 1005 3775 DCA PAT2 /SAVE INVERTED CHECKERBOARD PATTERN
500 1006 3774 DCA ADDCNT
501 1007 5600 JMP I SETPAT /RETURN TO PROGRAM
502 /
503 /THE FOLLOWING TESTS ARE DYNAMIC TEST WHICH CHECK FOR
504 /MULTIPLE SELECTION OF ADDRESS DECODERS, SLOW
505 /RECOVERY OF SENSE AMPLIFIERS, AND READ ACCESS TIME.
506 /
507 /
508 /DYNST 1
509 /
510 / 1) LOAD BACKGROUND PATTERN OF 4100 (JMS 100) IN ENTIRE MEMORY
511 / 2) LOAD 5700 (JMP I 100) IN FIRST ADDRESS OF EACH PAGE,
512 / EXCEPT PAGE ZERO.
513 / 3) LOAD INDIRECT ADDRESSING DATA IN LOC 100 OF EACH PAGE,
514 / 4) LOAD RETURN AND ERROR RETURN CODE INTO EACH FIELD UNDER TEST
515 / 5) AFTER CHECKING IF STACK SELECTED TO BE TESTED IS GOOD,
516 / NO HARD ERRORS, THE CODE IS EXECUTED IN EACH TEST FIELD,
517 / 6) THIS PROCESS IS REPEATED 63 TIMES, INCREMENTING THE COLUMN
518 / BITS EACH TIME.
519 / 7) THIS TEST IS ESPECIALLY GOOD FOR CHECKING SLOW RECOVERY
520 / TIME IN SENSE AMPS OF BITS 6-11.
521 /
522 /DYNST 2
523 /
524 / 1) THIS TEST DOES THE SAME THING AS DYNST 1 EXCEPT INSTEAD
525 / OF JUMPING IN A FORWARD DIRECTION, IT JUMPS BACK AND
526 / FORTH FROM BOTTOM TO TOP OF MEMORY TOWARDS THE MIDDLE.
527 / 2) THIS TEST IS ESPECIALLY GOOD FOR CHECKING ADDRESS
528 / DECODER OF BITS 0-5.
529 /
530 /DYNST 3
531 /
532 / 1) LOADS ALL MEMORY UNDER TEST WITH 4100 (JMS 100),
533 / 2) LOADS LOC'S 0-76 WITH JMP INSTRUCTIONS,
534 / 3) LOADS RETURN AND ERROR RETURN CODE, THEN EXECUTES IT,
535 / 4) DOES ONLY 1 PASS.
536 / 5) THIS TEST IS ESPECIALLY GOOD FOR CHECKING ADDRESS
537 / DECODER OF BITS 6-11.

```

```

536 /
537 1010 0000 DYNST, 0
538 1011 7300 CLA CLL
539 1012 6224 RIF
540 1013 1076 TAD K6201
541 1014 3215 DCA .+1
542 1015 6201 CDF /MAKE IF FLD= DF FLD
543 1016 1020 TAD PSR /SEE IF DYNST TO BE RUN
544 1017 0373 AND (40
545 1020 7640 SZA CLA /DYNST TO BE RUN
546 1021 5610 JMP I DYNST /NO. EXIT
547 1022 4425 PRNTMS
548 1023 1606 PRDYN /PRINT DYNAMIC TEST MESSAGE
549 1024 1072 TAD K261 /#1
550 1025 4435 TYPE
551 1026 4436 CRLF
552 1027 1372 TAD (NXTTST /SETUP TEST POINTER
553 1030 3357 DCA TSTPTR
554 1031 1101 TAD K7600
555 1032 3353 DCA STRTAD /SETUP START ADDRESS FOR RETURN
556 1033 1064 TAD K200
557 1034 3051 DCA CONST /SETUP CONSTANT
558 1035 1064 TAD K200
559 1036 3360 DCA TSTORG /START POINT OF CODE IN TEST FIELD
560 1037 3040 XDYN1, DCA TESTAD /START WITH ADDRESS ZERO
561 1040 7000 APTOK2, NOP /OVERRITTEN WHEN RUNNING UNDER APT
562 1041 3050 DCA COLCNT /CLEAR COLUMN COUNTER
563 1042 1371 TAD (5700 /DYNST 1 AND 2
564 1043 3354 DCA INST /SETUP INITIAL JMP I INSTRUCTION
565 1044 1042 TAD STKPIN
566 1045 1076 TAD K6201 /SET UP RETURN DATA FIELD
567 1046 3770 DCA PSTK1
568 1047 1041 XDYN2, TAD TOPSTK /GET MAX VALUE IN SYSTEM,
569 1050 7040 CMA
570 1051 3355 DCA STKKNT /SAVE FOR LOOP COUNTER,
571 1052 1355 TAD STKKNT
572 1053 3356 DCA STKKT1
573 1054 3043 DCA STKTST /START WITH FIELD ZERO
574 1055 4344 XDYN3, JMS SAME /SEE IF STACK SELECTED IS SAME AS
575 /PROGRAM FIELD,
576 1056 5267 JMP UPSTK /SAME STACK AS PROGRAM, JUST UPDATE THE
577 /FIELD TO NEXT ONE SELECTED,
578 1057 1043 TAD STKTST
579 1060 1076 TAD K6201 /SET UP STACK TO LOAD,
580 1061 3767 DCA TSTK1
581 1062 1360 TAD TSTORG
582 1063 7650 SNA CLA /DONE TESTS 1 & 2
583 1064 4766 JMS LODFLD /YES - LOAD PATTERN FOR DYN TEST 3
584 1065 4765 JMS LODSTK /GO LOAD PATTERN FOR DYN TEST 1 OR 2
585 1066 4764 JMS RETCOD /GO SET UP CODE FOR RETURN
586 /FROM TEST FIELD.
587
588 /NOW UPDATE THE FIELD TO TEST.
589 /
590 1067 4504 UPSTK, JMS I KSFCHK /CHECK IF CONSOLE ACTIVE

```

```

591 1670 1643 TAD STKTST
592 1671 1657 TAD K10
593 1672 3043 DCA STKTST /SAVE NEW OFFSET
594 1673 2355 ISZ STKNT /ALL FIELDS AVAILABLE DONE?
595 1674 5255 JMP XDYN3 /NO, GO DO NEXT.
596
597 /AT THIS POINT ALL AVAILABLE MEMORY IS FILLED
598 /WITH THE TEST PATTERN.
599 1675 5763* JMP PART2
600
601
602 /INCREMENT COLUMN COUNTER AND INST
603 1676 2354 NXCOL, ISZ INST /DYN TEST 1 AND 2
604 1677 2050 ISZ COLCNT /INCREMENT COLUMN COUNTER
605 1100 1050 TAD COLCNT
606 1101 0062 AND K77 /DONE ALL COLUMNS?
607 1102 7640 SZA CLA
608 1103 5247 JMP XDYN2 /NO
609 1104 1020 DONE3, TAD PSR /GET SWITCH REGISTER
610 1105 0073 AND K1000 /TEST BIT 2
611 1106 7640 SZA CLA /LOOP ON TEST?
612 1107 5237 JMP XDYN1 /YES
613 1110 2357 ISZ TSTPTR /SETUP TO DO NEXT TEST
614 1111 5757 NXITST, JMP I TSTPTR /JUMP TO NEXT TEST
615 1112 5315 JMP DYN2S2 /* DO DYNAMIC TEST #2
616 1113 5331 JMP DYN2S3 /* DO DYNAMIC TEST #3
617 1114 5610 JMP I DYN2ST /*RETURN TO PROGRAM
618
619 /DONE DYNAMIC TEST #1
620 1115 1051 /CHANGE CONST FOR DYN2ST 2
621 1116 7041 DYN2S2, TAD CONST /200
622 1117 3051 CIA /-200
623 1120 1074 DCA CONST
624 1121 3353 TAD K4000
625 1122 4425 DCA STRTAD /START OF RETURN CODE
626 1123 1606 PRNTMS
627 1124 1072 PRDYN /PRINT DYN TEST MESSAGE
628 1125 7001 TAD K261
629 1126 4435 IAC /#2
630 1127 4436 TYPE
631 1130 5237 CRLF
632 1131 7300 JMP XDYN1 /
633 1132 3360 DYN2S3, CLA CLL
634 1133 1362 DCA TSTORG /SETUP START POINT
635 1134 3353 TAD (110
636 1135 4425 DCA STRTAD /START OF RETURN CODE
637 1136 1606 PRNTMS
638 1137 1071 PRDYN
639 1140 1055 TAD K260
640 1141 4435 TAD K3 /#3
641 1142 4436 TYPE
642 1143 5237 CRLF
643 JMP XDYN1
644
645 /RETURN IF PROGRAM IS IN SELECTED STACK

```

```

646 /RETURN +1 IF PROGRAM IS NOT IN SELECTED FIELD
647 /
648 /
649 1144 0000 SAME, 0
650 1145 1042 TAD STKPIN
651 1146 7041 CIA
652 1147 1043 TAD STKTST
653 1150 7640 SZA CLA
654 1151 2344 ISZ SAME /PROG NOT IN SELECTED STACK
655 1152 5744 JMP I SAME
656
657 1153 0000 STRTAD, 0
658 1154 0000 INST, 0
659 1155 0000 STKNT, 0
660 1156 0000 STKNT1, 0
661 1157 0000 TSTPTR, 0
662 1160 0000 TSTORG, 0
663
664
665
666 1162 0110
667 1163 1455
668 1164 1400
669 1165 1200
670 1166 1274
671 1167 1341
672 1170 1343
673 1171 5700
674 1172 1111
675 1173 0040
676 1174 0771
677 1175 0774
678 1176 0773
679 1177 2525
680 1200 PAGE
681 /LOAD BACKGROUND PATTERN AND JMP INSTRUCTIONS
682 /FOR DYN2ST 1 OR 2.
683 1200 0000 LODSTK, 0
684 1201 1050 TAD COLCNT /*-HAVE ALL TEST FIELDS BEEN
685 1202 7740 SZA CLA CLL /* LOADED WITH BACKGROUND PATTERN?
686 1203 5212 JMP LODJMS /YES
687 1204 3040 /LOAD ALL LOCATIONS OF TEST FIELD WITH JMS TO LOC. 100
688 1205 1377 JMSLOD, TAD (4100 /LOAD BACKGROUND PATTERN INTO
689 /STACK SELECTED.
690 1206 4340 JMS CHGFLD /CHANGE DATA FIELD AND DEPOSIT AC
691 1207 2040 ISZ TESTAD /DONE 4096 TIMES?
692 1210 5205 JMP JMSLOD /NO, GO BACK AND DO NEXT ADDRESS
693 1211 5227 JMP LDJMP
694
695 /REWRITE JMS'S IN ALL LOC THAT WERE CHANGED IN TEST FIELDS
696 1212 7340 LODJMS, CLA CLL CMA
697 1213 1050 TAD COLCNT
698 1214 1064 TAD K200
699 1215 3040 DCA TESTAD /START INITIALLY WITH ADDRESS 200

```

```

700 1216 1053 IAD M75
701 1217 3346 DCA XM75 /SETUP COUNTER
702 1220 1377 LDJMS1, IAD (4100 /GET JMS 100 INST
703 1221 4340 JMS CHGFLD /DEPOSIT IT IN TEST FIELD
704 1222 1040 TAD TESTAD
705 1223 1063 TAD K100
706 1224 3040 DCA TESTAD /ADDED 100 TO ADDRESS
707 1225 2346 ISZ XM75 /ARE WE DONE?
708 1226 5220 JMP LDJMS1 /NO
709 /LOAD FIRST LOC OF EACH PAGE, EXCEPT PAGE 0, WITH 5700 = JMP I (XX00
710 /LOC INCREMENTED BY ONE AFTER EACH COMPLETE PASS BY COLCNT
711 1227 7300 LDJMP, CLA CLL
712 1230 1052 TAD M37
713 1231 3350 DCA XM37 /SETUP COUNTER
714 1232 1050 TAD COLCNT
715 1233 1064 TAD K200
716 1234 3040 DCA TESTAD /START INITIALLY AT ADDRESS 200
717 1235 1776 LDJMP1, TAD INST /INITIALLY 5700
718 1236 4340 JMS CHGFLD /DEPOSIT IT IN TEST FIELD
719 1237 1040 TAD TESTAD
720 1240 1064 TAD K200
721 1241 3040 DCA TESTAD /ADDED 200 TO ADDRESS
722 1242 2350 ISZ XM37 /DONE?
723 1243 5235 JMP LDJMP1 /NO
724 /LOAD LOC 100 OF EACH PAGE, EXCEPT 0, WITH DATA FOR INDIRECT ADDRESSING
725 /LOC INCREMENTED AFTER EACH COMPLETE PASS BY COLCNT
726 1244 1063 TAD K100
727 1245 1050 TAD COLCNT
728 1246 3040 DCA TESTAD /START INITIALLY AT ADDRESS 200
729 1247 1052 TAD M37
730 1250 3347 DCA XM37 /SETUP COUNTER
731 1251 1051 TAD CONST /200 FOR DYN TST 1 - 200 FOR DYN TST 2
732 1252 3345 DCA DATA /SETUP DATA
733 1253 1050 TAD COLCNT /GET COLUMN #
734 1254 1051 LDDATA, TAD CONST /200
735 1255 1345 TAD DATA
736 1256 3345 DCA DATA /UPDATE DATA
737 1257 1040 TAD TESTAD /300 TO 7500 IN INC'S OF 200
738 1260 1064 TAD K200
739 1261 3040 DCA TESTAD /UPDATED ADDRESS
740 1262 1345 TAD DATA /DATA IN INCREMENTS OF 200
741 1263 4340 JMS CHGFLD /DEPOSIT IN TEST FIELD
742 1264 1345 TAD DATA
743 1265 0102 AND K7700
744 1266 1074 TAD K4000 /CHECK IF HALF DONE LOADING FIELD
745 1267 7650 SNA CLA /HALF DONE?
746 1270 4330 JMS HAFDON /YES - GO CHECK WHICH TEST IS RUNNING

747
748 1271 2347 ISZ XM37 /DONE WHOLE FIELD YET?
749 1272 5254 JMP LDDATA /NO
750 1273 5600 JMP I LODSTK /RETURN
751
752 /LOAD BACKGROUND PATTERN AND JUMP INSTRUCTIONS
753 /FOR DYN TST #3
754 1274 0000 LODFLD, 0
    
```

```

755 1275 7300 CLA CLL
756 1276 3040 DCA TESTAD /CLEAR TESTAD
757 1277 7001 IAC
758 1300 3345 DCA DATA /SETUP DATA
759 1301 1377 LOAD, TAD (4100
760 1302 4340 JMS CHGFLD /STORE IT
761 1303 2040 ISZ TESTAD /NEXT ADDRESS - DONE 4096 TIMES?
762 1304 5301 JMP LOAD /NO - DO NEXT ADDRESS
763 /LOAD PAGE ZERO LOC'S 0 TO 76 WITH JUMP INSTRUCTIONS
764 1305 1345 AGAIN, TAD DATA /GET JUMP-TO LOCATION DATA
765 1306 1351 TAD M40
766 1307 7640 SZA CLA /ARE WE AT HALFWAY POINT?
767 1310 5314 JMP +4 /NO - CONTINUE
768 1311 1375 TAD (5110 /MAKE A JUMP TO 110 INST
769 1312 4340 JMS CHGFLD /STORE IT
770 1313 2040 ISZ TESTAD /UP THE ADDRESS
771 1314 1345 TAD DATA /GET DATA
772 1315 7040 CMA /COMPLEMENT IT
773 1316 0374 AND (5077 /MAKE JUMP INST
774 1317 4340 JMS CHGFLD /STORE IT
775 1320 2345 ISZ DATA
776 1321 2040 ISZ TESTAD
777 1322 1040 TAD TESTAD /CHECK IF DONE
778 1323 0062 AND K77 /ISOLATE BITS 6 TO 11
779 1324 7640 SZA CLA /DONE?
780 1325 5305 JMP AGAIN /NO - STORE NEXT INST
781 1326 2274 ISZ LODFLD /INCREMENT RETURN
782 1327 5674 JMP I LODFLD /YES - RETURN +1
783
784 /CHECKS TO SEE IF DYN TEST #2 IS RUNNING
785 /IF IT IS 200 IS ADDED TO TESTAD TO SET UP PATTERN PROPERLY
786 1330 0000 HAFDON, 0
787 1331 1051 TAD CONST
788 1332 7700 SNA CLA /RUNNING DYN TEST #2?
789 1333 5730 JMP I HAFDON /NO - RETURN
790 1334 1064 TAD K200 /YES - ADD 200 TO TESTAD
791 1335 1040 TAD TESTAD
792 1336 3040 DCA TESTAD
793 1337 5730 JMP I HAFDON /RETURN
794
795
796 1340 0000 CHGFLD, 0
797 1341 6201 TSTK1, CDF /TEST FIELD
798 1342 3440 DCA I TESTAD
799 1343 6201 PSTK1, CDF
800 1344 5740 JMP I CHGFLD
801
802 1345 0000 DATA, 0
803 1346 0000 XM75, 0
804 1347 0000 XM37, 0
805 1350 0000 XM37, 0
806 1351 77 M40, -40
807
808
809 1374 5077
    
```

```

810 1375 5110
811 1376 1154
812 1377 4100
      PAGE
813 /ROUTINE TO SETUP RETURN FROM TEST FIELD
814 /
815 1400 0000 RETCOD, 0
816 1401 1777* TAD STRTAD /7600 OR 4000 OR 110
817 1402 1050 TAD COLCNT
818 1403 3040 DCA TESTAD
819 1404 4232 JMS XAND /SETUP AND INSTRUCTION
820 1405 2040 ISZ TESTAD /NEXT ADDRESS
821 1406 4242 JMS XCDF /SETUP CDF INSTRUCTION
822 1407 2040 ISZ TESTAD /NEXT ADDRESS
823 1410 4247 JMS XJMPI /SETUP JMP I ,+1 INSTRUCTION
824 1411 2040 ISZ TESTAD /NEXT ADDRESS
825 1412 1376 TAD (RETURN /GET RETURN ADDRESS
826 1413 4775* JMS CHGFLD
827
828 1414 1355 TAD ADRCHK /LOCATION EQUAL TO COMPLEMENT OF BITS 6-11
829 1415 3040 DCA TESTAD /SETUP ADDRESS
830 1416 1102 TAD K7700 /MASKING DATA
831 1417 4775* JMS CHGFLD /DEPOSIT IN TEST FIELD
832
833 /SETUP ERROR RETURN FROM TEST FIELD
834 1420 1063 TAD K100
835 1421 7001 IAC
836 1422 3040 DCA TESTAD /START AT ADDRESS 101
837 1423 4242 JMS XCDF /SETUP CDF INSTRUCTION
838 1424 2040 ISZ TESTAD /NEXT ADDRESS
839 1425 4247 JMS XJMPI /SETUP JMP I ,+1 INSTRUCTION
840 1426 2040 ISZ TESTAD /NEXT ADDRESS
841 1427 1374 TAD (REPERR /GET ADDRESS OF CALL TO ERROR ROUTINE
842 1430 4775* JMS CHGFLD
843 1431 5600 JMP I RETCOD
844
845
846 /ROUTINE TO MAKE AN AND INST, WITH ADDRESS EQUAL
847 /TO THE COMPLEMENT OF COLUMN COUNTER.
848 1432 0000 XAND, 0
849 1433 1050 TAD COLCNT /GET COLUMN COUNTER
850 1434 7040 CMA /COMPLEMENT IT
851 1435 0062 AND K77 /
852 1436 3355 DCA ADRCHK /SAVE FOR LATER USE
853 1437 1355 TAD ADRCHK
854 1440 4775* JMS CHGFLD
855 1441 5632 JMP I XAND
856
857 /MAKE A CDF INST OF CURRENT PROGRAM FIELD
858 1442 0000 XCDF, 0
859 1443 1042 TAD STKPIN
860 1444 1077 TAD K6203
861 1445 4775* JMS CHGFLD
862 1446 5642 JMP I XCDF
863

```

```

864 /MAKE A JMP I ,+1 INST
865 1447 0000 XJMPI, 0
866 1450 1040 TAD TESTAD
867 1451 0373 AND (177
868 1452 1372 TAD (5601
869 1453 4775* JMS CHGFLD
870 1454 5647 JMP I XJMPI
871
872 /CHECK IF STACK SELECTED IS OK TO EXECUTE IN,
873 /THEN EXECUTE CODE STORED THERE.
874
875 1455 7300 PART2, CLA CLL
876 1456 1050 TAD COLCNT
877 1457 1771* TAD TSTORG /START POINT OF CODE IN TEST FIELD
878 1460 3356 DCA XSTART
879 1461 3043 DCA STKTST /START WITH FIELD ZERO
880 1462 4770* TOP, JMS SAME /CHECK IF PROG FLD EQUAL DATA FLD
881 1463 1057 TAD K10 /SAME
882 1464 1043 TAD STKTST /NOT SAME
883 1465 3043 DCA STKTST
884 1466 1043 TAD STKTST
885 1467 7112 RTR CLL
886 1470 7010 RAR
887 1471 3357 DCA NWFLD /SAVE TEST FIELD
888 1472 2767* ISZ STKKT1 /ARE WE DONE?
889 1473 5275 JMP ,+2 /NO - CONT
890 1474 5323 JMP CHKST /DONE PASS -GO CHECK WHICH TEST IS RUNNING
891 1475 1337 TAD NWFLD
892 1476 1366 TAD (FIDLST /LIST OF STATUS OF FIELDS
893 1477 3360 DCA PTR
894 1400 1760 TAD I PTR
895 1401 7640 SZA CLA /IS FIELD SELECTED GOOD?
896 1402 5317 JMP ADD1 /NO - TRY NEXT ONE
897 1403 1357 TAD NWFLD /YES
898 1404 7106 RTL CLL
899 1405 7004 RAL
900 1406 1077 TAD K6203 /MAKE CDF INST
901 1407 3310 DCA ,+1
902 1410 6201 CDF
903 1411 1101 TAD K7600 /STORE IN AC
904 1412 5756 JMP I XSTART /GO EXECUTE CODE IN TEST FIELD
905 /IF NO ERRORS OCCURRED PROGRAM CONTROL WILL RETURN HERE
906
907 1413 7041 RETURN, CIA /COMPLEMENT DATA PREVIOUSLY STORED IN AC
908 1414 1101 TAD K7600
909 1415 7740 SZA CLA CLL /WAS DYN TEST GOOD?
910 1416 4327 REPERR, JMS DYNERR /NO - REPORT AN ERROR
911
912 /SETUP TO EXECUTE IN NEXT FIELD
913
914 1417 1043 ADD1, TAD STKTST
915 1420 1057 TAD K10
916 1421 3043 DCA STKTST
917 1422 5262 JMP TOP
918 /IF DYNTEST 3 IS RUNNING JUMP OUT (DOES ONLY ONE PASS)

```

```

919                /IF DYNTST 1 OR 2 IS RUNNING GO INCREMENT TO NEXT COLUMN
920
921 1523 1771*  CHKTST, TAD  TSTORG  /O IF DYNTST 3
922 1524 7640   SZA CLA  /DYNTST 3 RUNNING?
923 1525 5765* JMP  NXCOL  /NO - DO NEXT COLUMN
924 1526 5764* JMP  DONE3  /YES - DONE DYNTST 3
925
926
927
928 1527 0000  DYNERR, 0
929 1530 7000  APTE03, NOP  /OVERWRITTEN IF UNDER APT
930 1531 4763* JMS  BELL  /CHECK FOR BELL- RETURN +1 IF NOBELL
931 1532 5727  JMP I  DYNERR /BELL - RETURN TO PROGRAM
932 1533 1020  TAD  PSR  /CHECK FOR INHIBIT TYPEOUT
933 1534 0064  AND  K200 /TEST BIT 4
934 1535 7640  SZA CLA  /INHIBIT TYPEOUT?
935 1536 5345  JMP  OVER  /YES -BYPASS TYPEOUT ROUTINE
936 1537 4425  PRNTMS
937 1540 1613  DYNERR1 /PRINT ERROR MESSAGE
938 1541 1357  TAD  NWFLD /FIELD UNDER TEST
939 1542 1071  TAD  K260  / AT TIME OF ERROR
940
941 1544 4436  CRLF
942 1545 1020  OVER,  TAD  PSR  /INHIBIT ERROR HALT?
943 1546 7700  SMA CLA /TEST BIT 0
944 1547 4762* JMS  PSEUDO /NO
945 1550 1020  TAD  PSR  /GET SWITCH REGISTER
946 1551 7004  RAL  /TEST BIT 1
947 1552 7710  SPA CLA /LOOP ON ERROR?
948 1553 5761* JMP  XDYN1 /YES
949 1554 5727  JMP I  DYNERR /RETURN TO PROGRAM
950
951
952 1555 0000  ADRCHK, 0
953 1556 0000  XSTART, 0  /STARTING ADDRESS=1 OF PROGRAM IN TEST FIELD
954 1557 0000  NWFLD, 0
955 1560 0000  PTR, 0
956
957
958
959
960 1561 1037
961 1562 2440
962 1563 2056
963 1564 1104
964 1565 1076
965 1566 1725
966
967 1567 1156
968 1570 1144
969 1571 1160
970 1572 5601
971 1573 0177
972 1574 1516
973 1575 1340
974 1576 1513

```

```

974 1577 1153
975 1600 1501  PAGE
      1601 2203  MARCH, TEXT  "MARCH TEST*"
      1602 1040
      1603 2405
      1604 2324
      1605 4300
976
977 1606 0431  PRDYN, TEXT  "DYN TEST "
      1607 1640
      1610 2405
      1611 2324
      1612 4000
978 1613 0140  DYNERR1, TEXT  "A DYNAMIC ERROR HAS OCCURED IN FIELD "
      1614 0431
      1615 1601
      1616 1511
      1617 0340
      1620 0522
      1621 2217
      1622 2240
      1623 1001
      1624 2340
      1625 1703
      1626 0325
      1627 2205
      1630 0440
      1631 1116
      1632 4006
      1633 1105
      1634 1404
      1635 4000
979
980
981  /MOSTST ERROR ROUTINE
982
983 1636 0000  MOSERR, 0
984 1637 7300  CLL CLA
985 1640 7000  APTE01, NOP
986 1641 2047  ISZ  COUNT
987 1642 7410  SKP
988 1643 5241  JMP  ,=2
989 1644 4777* JMS BELL  /CHECK FOR BELL - RETURN +1 IF NOBELL
990 1645 5636  JMP I MOSERR /BELL - RETURN TO PROGRAM
991 1646 1020  TAD PSR  /NO BELL - CHECK FOR INHIBIT TYPEOUT
992 1647 0064  AND K200
993 1650 7640  SZA CLA
994 1651 5313  JMP STOP  /BYPASS ERROR TYPEOUT ROUTINE
995 1652 2037  ISZ HEAD1
996 1653 5256  JMP PERR1
997 1654 4425  PRNTMS
998 1655 1735  HEAD  /TYPEOUT ERROR HEADING
999 1656 1042  PERR1, TAD STKPIN /PROGRAM FIELD
1000 1657 7012  RTR

```

```

1001 1660 7010 RAR
1002 1661 1071 TAD K260
1003 1662 4435 TYPE
1004 1663 7040 CMA
1005 1664 1236 TAD MOSERR /PROGRAM LOCATION OF CALL TO SUB
1006 1665 4433 PRNT4
1007 1666 4434 SPACE2
1008 1667 1043 TAD STKTST /FIELD OF ERROR
1009 1670 7012 RTR
1010 1671 7010 RAR
1011 1672 3323 DCA XFLD /SAVE # OF FIELD OF ERROR
1012 1673 1323 TAD XFLD
1013 1674 1071 TAD K260
1014 1675 4435 TYPE
1015 1676 1040 TAD TESTAD /FAILING ADDRESS
1016 1677 4433 PRNT4
1017 1700 1045 TAD GOATA /GOOD DATA
1018 1701 4433 PRNT4
1019 1702 1044 TAD BDATA /BAD DATA
1020 1703 4433 PRNT4
1021 1704 4425 PRNTMS
1022 1705 1600 MARCH /PRINT MARCH TEST
1023 1706 1323 TAD XFLD /FIELD OF ERROR
1024 1707 1376 TAD (FLDLST
1025 1710 3324 DCA BADFLD
1026 1711 7240 STA
1027 1712 3724 DCA I BADFLD /LOAD ADDRESS OF BAD FIELD WITH ONES
1028 1713 1020 STOP, TAD PSR /INHIBIT ERROR HALT
1029 1714 7700 SMA CLA /TEST BIT 0
1030 1715 4775 JMS PSEUDO /NO
1031 1716 1020 TAD PSR /LOOP ON ERROR?
1032 1717 7004 RAL /TEST BIT 1
1033 1720 7710 SPA CLA
1034 1721 5774 JMP MOSLOD-3 /YES
1035 1722 5636 JMP I MOSERR /NO - RETURN TO PROGRAM
1036
1037 1723 0000 XFLD, 0
1038 1724 0000 BADFLD, 0
1039
1040 1725 0000 FLDLST, 0 /F0 - ZEROS FOR GOOD STACK, ONES FOR BAD
1041 1726 0000 0 /F1
1042 1727 0000 0 /F2
1043 1730 0000 0 /F3
1044 1731 0000 0 /F4
1045 1732 0000 0 /F5
1046 1733 0000 0 /F6
1047 1734 0000 0 /F7
1048 1735 2022 HEAD, TEXT "PR LOC ADDR GOOD BAD#"
1736 4014
1737 1703
1740 4040
1741 4001
1742 0404
1743 2240
1744 4040
    
```

```

1745 0717
1746 1704
1747 4040
1750 0201
1751 0443
1752 0000
1049
1050
1051 1774 0607
1052 1775 2440
1053 1776 1725
1054 1777 2056
1055
1056
1057
1058 2000 0000 /RELOCATION ERROR ROUTINE
1059 2001 7200 ERRM, 0
1060 2002 7000 APTE02, NOP
1061 2003 2047 ISZ COUNT /RELO ERROR OCCURED
1062 2004 7410 SKP
1063 2005 5203 JMP ,-2
1064 2006 4256 JMS BELL /CHECK FOR BELL - RETURN +1 IF NO BELL
1065 2007 4600 JMS I ERRM /BELL - RETURN TO PROGRAM
1066 2010 1020 TAD PSR /NO BELL - CHECK FOR INHIBIT TYPEOUT
1067 2011 0064 AND K200
1068 2012 7640 SZA CLA
1069 2013 5230 JMP STOP1 /BYPASS TYPEDOUT ROUTINE
1070 2014 4425 PERRM, PRNTMS
1071 2015 2040 RELERR
1072 2016 1043 TAD STKTST
1073 2017 7112 CIL RTR
1074 2020 7010 RAR
1075 2021 1071 TAD K260 /FIELD OF ERROR
1076 2022 4435 TYPE
1077 2023 1046 TAD MOVE /ADDRESS OF ERROR
1078 2024 4433 PRNT4
1079 2025 4436 CRLF
1080 2026 7240 STA
1081 2027 3037 DCA HEAD1
1082 2030 1020 STOP1, TAD PSR /INHIBIT ERROR HALT?
1083 2031 7700 SMA CLA /TEST BIT 0
1084 2032 4777 JMS PSEUDO /NO
1085 2033 1020 TAD PSR /LOOP ON ERROR?
1086 2034 7004 RAL
1087 2035 7710 SPA CLA
1088 2036 5776 JMP RELO1-1 /YES
1089 2037 5600 JMP I ERRM /RETURN TO PROGRAM
1090
1091 2040 5252 RELERR, TEXT "****WARNING*** RELO ERR AT "
2041 5227
2042 0122
2043 1611
2044 1607
2045 5252
    
```

```

2046 5240
2047 2205
2050 1417
2051 4005
2052 2222
2053 4001
2054 2440
2055 0000
1092
1093 2056 0000 BELL, 0
1094 2057 7300 CLA CLL
1095 2060 1020 TAD PSR /CHECK FOR BELL ON ERROR
1096 2061 0080 AND K20 /BIT 7
1097 2062 7650 SWA CLA
1098 2063 5270 JMP NOBELL
1099 2064 1065 RBELL, TAD K207 /BELL CODE
1100 2065 4435 TYPE
1101 2066 4504 JMS I KSFCHK /CHECK FOR CONSOLE RECEIVE FLAG
1102 2067 5656 JMP I BELL
1103 2070 2256 NOBELL, ISZ BELL /RETURN +1
1104 2071 5656 JMP I BELL
1105 2176 0421
1106 2177 2440
PAGE /APT/
APTMOD, 0
1107 2200 0000 TAD (JMS I IAPTOK /SETUP FOR APT CONTROL
1108 2201 1377 DCA APTOK0 /
1109 2202 3776 TAD (JMS I IAPTOK /SETUP FOR APT CONTROL
1110 2203 1377 DCA APTOK1 /
1111 2204 3775 TAD (JMS I IAPTOK /SETUP FOR APT CONTROL
1112 2205 1377 DCA APTOK2 /
1113 2206 3774 TAD (7000 /MODIFY SOME LOCS TO: NOP.
1114 2207 1373 DCA APTN00 /
1115 2210 3772 TAD (7000 /
1116 2211 1373 DCA APTN01 /
1117 2212 3771 TAD (7000 /
1118 2213 1373 DCA APTN02 /
1119 2214 3770 TAD (7000 /
1120 2215 1373 DCA 6505 /OVERWRITES APT CODE
1121 2216 3767 TAD (7000 /
1122 2217 1373 DCA 6523 /OVERWRITES APT CODE
1123 2220 3766 TAD (JMS I IAPTER /MODIFY SOME LOCS TO: JMS I IAPTER.
1124 2221 1365 DCA APTE01 /
1125 2222 3764 TAD (JMS I IAPTER /
1126 2223 1365 DCA APTE02 /
1127 2224 3763 TAD (JMS I IAPTER /
1128 2225 1365 DCA APTE03 /
1129 2226 3762
1130 2227 1361 TAD (INHMES /INHIBIT USE OF MESAGX ROUTINE
1131 2230 3760 DCA APTMES /
1132 2231 1357 TAD (INH TYP /INHIBIT USE OF TYPE ROUTINE
1133 2232 3756 DCA APTNTP /
1134 2233 5600 JMP I APTMOD
1135
1136
1137

```

```

1138
1139 /APT/ ROUTINE TO 'NOTIFY' APT THAT THE PROGRAM IS RUNNING OK.
1140 /PROGRAM JUMPS TO APT CODE THAT WAS SAVED IN PROGRAM FIELD
1141
1142 2234 0000 APTOK, 0 /APT/
1143 2235 6002 IOF /APT/
1144 2236 7200 CLA /APT/
1145 2237 1251 TAD APTIMX /APT/DELAY 100MS.
1146 2240 3253 DCA APTCTX /APT/
1147 2241 1252 TAD APTINY /APT/
1148 2242 3254 DCA APTCTY /APT/
1149 2243 2254 ISZ APTCTY /APT/
1150 2244 5243 JMP ,=1 /APT/
1151 2245 2253 ISZ APTCTX /APT/
1152 2246 5241 JMP ,=5 /APT/
1153 2247 4755 JMS 6500 /APT/CALL APT = 'PRG OK'.
1154 2250 5634 JMP I APTOK /APT/RTN FROM APT = RTN TO CALL+1.
1155
1156 2251 7776 APTIMX, -2 /APT/
1157 2252 0000 APTINY, 0 /APT/
1158 2253 0000 APTCTX, 0 /APT/
1159 2254 0000 APTCTY, 0 /APT/
1160
1161
1162
1163 /APT/ ROUTINE TO HANDLE ERRORS UNDER APT CONTROL.
1164 /PROGRAM JUMPS TO APT CODE THAT WAS SAVED IN PROGRAM FIELD
1165
1166 2255 0000 APTER, 0 /APT/
1167 2256 6002 IOF /APT/
1168 2257 7200 CLA /APT/
1169 2260 6224 RIF /APT/AC=IF.
1170 2261 1076 TAD K6201 /APT/CREATE A CDF INST.
1171 2262 3265 DCA ,+3 /APT/MODIFY NEXT CDF INST.
1172 2263 7240 CLA CMA /APT/
1173 2264 1255 TAD APTER /APT/AC=ERROR PC.
1174 2265 6201 CDF /APT/(MODIFIED CDF) DF=IF.
1175 2266 5754 JMP 6520 /APT/CALL APT = 'ERROR'.
1176
1177
1178 /CB/ ROUTINE TO SAVE PAGE 37 OF FIELD 0 & 1 OR SAVE APT HANDLER
1179
1180 /SAVE EITHER APT OR OS8 CODE
SAVOS8, 0
1181 2267 0000 TAD K7577 /SETUP OS8 HANDLER POINTER -1
1182 2270 1100 DCA 10 /SAVE IN AUTO INDEX 10
1183 2271 3010 TAD K6201 /FIELD 1
1184 2272 1076 TAD K10 /SAVE OS8 FIELD 1
1185 2273 1057 DCA FLD /DO IT
1186 2274 3314 JMS SAVFLD /RETURN TO PROGRAM
1187 2275 4307 JMP I SAVOS8 /SETUP APT HANDLER POINTER -1
1188 2276 5667 XAPT, TAD (5577 /SETUP APT HANDLER POINTER -1
1189 2277 1353 DCA 10 /SAVE IN AUTO INDEX 10
1190
1191 2300 3010 TAD K6201
1192 2301 1076

```



```

1193 2302 1061 TAD K30 //FIELD 1
1194 2303 3314 DCA FLD /MODIFY CDF INSTRUCTION IN LOC FLD
1195 2304 4307 JMS SAVFLD /GO SAVE APT HANDLER
1196 2305 4200 JMS APTMOD /GO MODIFY LOCATIONS FOR APT
1197 2306 5752* JMP BEGIN
1198
1199 2307 0000 SAVFLD, 0
1200 2310 1751* TAD PRGSTK /INSTRUCTION FIELD
1201 2311 3316 DCA PRGFLO /MODIFY INSTRUCTION AT LOC PRGFLO
1202 2312 1325 TAD STOADD /GET ADDRESS OF STORAGE ADDRESS
1203 2313 3011 DCA 11 /SAVE IN AUTO INDEX 11
1204 2314 6701 FLD, CDFD /CHANGE DATA FIELD
1205 2315 1410 TAD I 10 /GET WORD
1206 2316 6201 PRGFLO, CDFD /CHANGE DATA FIELD TO PROGRAM FIELD
1207 2317 3411 DCA I 11 /SAVE IN STORE AREA
1208 2320 1010 TAD 10
1209 2321 7040 CMA
1210 2322 7640 SZA CLA /CHECK TO SEE IF DONE
1211 2323 5314 JMP FLD /NO = DO NEXT WORD
1212 2324 5707 JMP I SAVFLD /YES = RETURN
1213
1214 2325 5577 STOADD, 5577
1215 2326 4323 SRMMSG, TEXT "#SR#"
1216 2327 2275
1217 2330 0000
1218 2331 7743 GESTMK, TEXT "T#"
1219 2332 0000
1220 2333 3603 UPARRC, TEXT "C#"
1221 2334 4300
1222 2335 3607 UPARRG, TEXT "G#"
1223 2336 4300
1224
1225 2351 0216
1226 2352 0214
1227 2353 5577
1228 2354 6520
1229 2355 6500
1230 2356 2637
1231 2357 5635
1232 2360 3002
1233 2361 5600
1234 2362 1530
1235 2363 2002
1236 2364 1640
1237 2365 4505
1238 2366 6523
1239 2367 6505
1240
1241 2370 2524
1242 2371 0223
1243 2372 0213
1244 2373 7000
1245 2374 1040
1246 2375 0745
1247 2376 0261
1248 2377 4506

```

```

1243 2400 PAGE /APT/
1244
1245 /CB/ ROUTINE TO RESTORE PAGES 37 OF FIELD 0 AND 1
1246
1247 2400 7200 CBRM, CLA
1248 2401 4425 PRNTMS /GO PRINT UPARROW C
1249 2402 2333 UPARRC /POINTER TO MESSAGE
1250 2403 6224 RIF /GET THE PRESENT DATA FIELD
1251 2404 1076 TAD K6201 /GET THE CDF INSTRUCTION
1252 2405 3216 DCA CBRMO /SAVE THE NEW CDF INSTRUCTION
1253 2406 1100 TAD K7577 /SET UP AUTO INDEX FOR RESTORE OF 0
1254 2407 3010 DCA 10 /SAVE IN AUTO INDEX 10
1255 2410 1377 TAD (5577 /SETUP STORAGE POINTER
1256 2411 3011 DCA 11 /SAVE IN AUTO INDEX 11
1257 2412 1100 TAD K7577 /SEUP AUTO INDEX OF RESTORE OF FILD 1
1258 2413 3012 DCA 12 /SAVE IN AUTO INDEX 12
1259 2414 1100 TAD K7577 /SETUP NEXT POINTER
1260 2415 3013 DCA 13 /SAVE IN AUTO INDEX 13
1261 2416 6201 CBRMO, CDF /MODIFIED CDF INSTRUCTION TO PRG FIELD
1262 2417 1010 TAD 10 /RESTORATION DONE
1263 2420 7040 CMA
1264 2421 7450 SNA
1265 2422 5235 JMP CBRM1 /SKIP IF NO
1266 2423 7621 7621 /DONE-GO TO MONITOR AT 7600
1267 2424 1410 TAD I 10 /CLEAR AC AND MQ
1268 2425 7421 7421 /GET DATA FROM PROGRAM FIELD
1269 2426 1411 TAD I 11 /PUT IT IN THE MQ
1270 2427 6211 CDF 10 /GET DATA TO BE PUT IN FIELD 1
1271 2430 3413 DCA I 13 /CHANGE DATA FIELD TO 1
1272 2431 7521 7521 /PUT IT IN FIELD 1
1273 2432 6201 CDF 00 /SWAP AC AND MQ
1274 2433 3412 DCA I 12 /CHANGE DATA FIELD TO 0
1275 2434 5216 JMP CBRMO /RESTORE FIELD 0 PAGE 37
1276 2435 6203 CBRM1, CDF CIF /GO DO NEXT WORD
1277 2436 5637 JMP I ,+1 /CHANGE DATA AND INSTR FIELD TO 0
1278 2437 7600 7600 /GO TO 7600 OF THAT FIELD
1279 /MONITOR STARTING ADDRESS
1280
1281 /ROUTINE USED FOR CONSOLE SWITCH REGISTER CHANGES
1282
1283 2440 0000 PSEUDO, 0
1284 2441 7200 CLA
1285 2442 4425 SRGEST, PRNTMS /PRINT SR QUESTION
1286 2443 2326 SRMMSG /POINTER TO MESSAGE
1287 2444 1020 TAD PSR /GET THE VALUE OF THE SWITCH REGISTER
1288 2445 4433 PRNT4 /PRINT THE 4 DIGITS
1289 2446 7346 CLA CLL CMA RTL /SETUP A COUNTER TO ACCEPT 4 DIGITS
1290 2447 3322 DCA TTYCNT /SAVE THE COUNTER
1291 2450 1376 TAD (CHARRO /GET POINTER FOR FIRST CHARACTER
1292 2451 3254 DCA CHGCHR /SAVE THE POINTER FOR DIGITS
1293 2452 4424 LISN /WAIT FOR KEYBOARD INPUT
1294 2453 0001 I /CHECK FOR A OCTAL DIGIT
1295 2454 2472 CHGCHR, CHARRO /THIS LOCATION WILL GET MODIFIED
1296 2455 7566 -212 /CHECK FOR LINE FEED

```

```

1297 2456 3124 RESTRT /LINE FEED = RETURN TO START VIA RESTRT
1298 2457 7563 -215 /CHECK FOR CARRIAGE RETURN
1299 2460 2506 RETYPE /RETYPE SR AND CONT IF DIGITS TYPED
1300 2461 7575 -203 /CHECK FOR A CONTROL C
1301 2462 2400 CRRM /CONTROL C TYPED =RETURN TO MONITOR
1302 2463 7555 -223 /CHECK FOR A CONTROL S
1303 2464 2553 CNTRS /WAS CONTROL S WAIT FOR "Q OR "C
1304 2465 0000 0 /NONE OF ABOVE CHARACTERS=ILLEGAL CHAR
1305 2466 2467 +1 /GO TO NEXT ADDRESS TO PRINT ?
1306 2467 4425 PRNTMS /GO PRINT ?
1307 2470 2331 GESTMK /POINTER TO ? MESSAGE
1308 2471 5242 JMP SRQEST /RETURN AND ASK QUESTION AGAIN
1309 2472 3020 CHARRO, DCA PSR /SAVE THE LEAST SIGNIFICANT BIT
1310 2473 1375 TAD (CHARR1 /UPDATA POINTER FOR CHARACTERS 2 3 4
1311 2474 3254 DCA CHGCHR /SAVE THE POINTER ADDRESS
1312 2475 5252 JMP CHGCHR-2 /RETURN FOR NEXT CHARACTER INPUT
1313 2476 3321 CHARR1, DCA SAVCHR /SAVE THE CHARACTER TYPED
1314 2477 1020 TAD PSR /GET THE VALUE OF SR
1315 2400 7106 CLL RTL /MOVE IT INTO NEXT POSITION
1316 2401 7004 RAL
1317 2402 1321 TAD SAVCHR /ADD NEW CHARACTER TO IT
1318 2403 3020 DCA PSR /SAVE THE NEW VALUE
1319 2404 2322 ISZ TTYCNT /DONE ALL 4 CHARACTERS
1320 2405 5252 JMP CHGCHR-2 /NO GET NEXT INPUT FROM KEYBOARD
1321 2406 1376 RETYPE, TAD (CHARRO /GET POINTER TO SEE IF SR ECHOED
1322 2407 7041 CIA /NEGATE THE POINTER
1323 2410 1254 TAD CHGCHR /GET THE POINTER STORED
1324 2411 7650 SNA CLA /ECHO VALUE OF SR?
1325 2412 5640 JMP I PSEUDO /NO-ONLY CR WAS TYPED=USE ORIGINAL VALUE
1326 2413 4425 PRNTMS /RE-ECHO VALUE TYPED
1327 2414 2326 SRMSG /POINTER TO SR MESSAGE
1328 2415 1020 TAD PSR /GET VALUE OF SR
1329 2416 4433 PRNT4 /PRINT THE 4 OCTAL DIGITS
1330 2417 4436 CRLF /ISSUE A CR AND LF
1331 2420 5640 JMP I PSEUDO /RETURN TO PROGRAM
1332
1333 2421 0000 SAVCHR, 0
1334 2422 0000 TTYCNT, 0
1335
1336
1337
1338
1339
1340 2423 0000 /ROUTINE TO CHECK FOR CONSOLE RECEIVE FLAG
1341 2424 6031 CHKKSF, 0
1342 2425 5723 APTNO2, KSF /SKIP ON CONSOLE RECEIVE FLAG
1343 2426 1022 JMP I CHKKSF /RECEIVE FLAG NOT SET RETURN TO PROGRAM
TAD HCW2 /CHECK TO SEE IF RUNNING UNDER APT?
1344 2427 7700 SMA CLA /YES = CLEAR FLAG AND RETURN
1345 2430 5333 JMP +3 /NO =CHECK FOR "C OR "G
1346 2431 6032 KCC /CLEAR CONSOLE RECEIVE FLAG
1347 2432 5723 JMP I CHKKSF /RETURN TO PROGRAM
1348 2433 4424 LISN /CHECK THE KEYBOARD CHARACTER
1349 2434 7575 -203 /CODE FOR "C
1350 2435 2400 CRRM /WAS A CONTROL C=EXIT TO MONITOR
1351 2436 7571 -207 /CODE FOR "G

```

```

1352 2437 2547 CNTRLG /WAS "G ECHO CHAR=ENTER SR QUESTION
1353 2440 7555 -223 /CHECK FOR A CONTROL S
1354 2441 2555 CNTRS1 /WAS A CONTROL S WAIT FOR "Q OR "C
1355 2442 0000 0 /CHAR WAS NOT "C OR "G
1356 2443 2544 +1 /ECHO CHAR AND QUESTION MARK
1357 2444 4425 PRNTMS /PRINT ? AND CR LF
1358 2445 2331 GESTMK /POINTER TO MESSAGE
1359 2446 5723 JMP I CHKKSF /RETURN TO PROGRAM
1360
1361 2447 4425 CNTRLG, PRNTMS /PRINT "G AND CR LF
1362 2450 2335 UPARRG /POINTER TO MESSAGE
1363 2451 4240 JMS PSEUDO /GO ASK THE SR QUESTION
1364 2452 5723 JMP I CHKKSF /RETURN TO THE PROGRAM
1365
1366 2453 4774 CNTRS, JMS WAITQC /GO WAIT FOR A CONTROL Q OR C
1367 2454 5252 JMP CHGCHR-2 /GO WAIT FOR NEXT CHAR
1368
1369 2455 4774 CNTRS1, JMS WAITQC /WAIT FOR A CONTROL Q OR C
1370 2456 5723 JMP I CHKKSF /RETURN TO PROGRAM
1371
1372
1373 2474 3106
1374 2475 2476
1375 2476 2472
1376 2477 5577
2600 PAGE
1377
1378
1379
1380 2600 0000 FILLER, 0 /SET TO NUMBER OF FILLERS REQUIRED
1381
1382 /INPUT ONE OCTAL NUMBER TO AC 9 THRU 11
1383 /GOOD RETURN IS JMS+2
1384
1385 2601 0000 ONEOCK, 0 /CALL BY "ONEOCK"
1386 2602 4424 LISN
1387 2603 0001 1
1388 2604 2607 +3
1389 2605 0000 0
1390 2606 2610 +2
1391 2607 2201 ISZ ONEOCK
1392 2610 5601 JMP I ONEOCK
1393
1394 /INPUT TWO OCTAL NUMBERS TO AC 6 THRU 11
1395 /GOOD RETURN IS JMS+2
1396
1397 2611 0000 TWOOCK, 0 /CALL BY "TWOOCK"
1398 2612 4201 JMS ONEOCK
1399 2613 5611 JMP I TWOOCK
1400 2614 7104 CLL RAL
1401 2615 7006 RTL
1402 2616 3224 DCA XPRNT2
1403 2617 4201 JMS ONEOCK
1404 2620 5611 JMP I TWOOCK
1405 2621 1224 TAD XPRNT2

```

```

1406 2622 2211      ISZ   TWOOCK
1407 2623 5611      JMP I  TWOOCK
1408
1409                /PRINT THE TWO OCTAL NUMBERS IN THE AC 6 THRU 11
1410
1411 2624 0000      XPRNT2, 0          /CALL BY "PRNT2"
1412 2625 3211      DCA    TWOOCK
1413 2626 1211      TAD    TWOOCK
1414 2627 7012      RTR
1415 2630 7010      RAR
1416 2631 4430      PRNT1
1417 2632 1211      TAD    TWOOCK
1418 2633 4430      PRNT1
1419 2634 5624      JMP I  XPRNT2
1420
1421                /TYPE THE ASCII CHARACTER IN THE AC
1422
1423 2635 0000      XTYPE, 0          /CALL BY "TYPE"
1424 2636 3251      DCA    CHAR          /SAVE THE CHARACTER
1425 2637 7000      APTNTP, NOP        /APT
1426 2640 7610      SKP    CLA          /CONSOLE INACTIVE-TYPE THE CHARACTER
1427 2641 4777      JMS    CNTRLS        /CONSOLE ACTIVE-CHECK FOR CONTROL S
1428 2642 1251      IAD    CHAR          /GET THE CHARACTER SAVED AND PRINT
1429 2643 6046      TLS
1430 2644 7200      CLA
1431 2645 6041      TSP
1432 2646 5245      JMP    *-1
1433 2647 6042      TCF
1434 2650 5635      JMP I  XTYPE
1435
1436 2651 0000      CHAR, 0
1437 2652 5635      INHTYP=JMP I XTYPE
1438
1439
1440
1441                /TYPE A CR AND LF WITH NUMBER OF FILLERS
1442                /AS DETERMINED BY LOCATION "FILLER"
1443
1444 2652 0000      XCRLF, 0          /CALL BY "CRLF"
1445 2653 7200      CLA
1446 2654 1067      TAD    K215
1447 2655 4435      TYPE
1448 2656 1200      TAD    FILLER
1449 2657 7040      CMA
1450 2660 3266      DCA    XURS
1451 2661 1066      TAD    K212
1452 2662 4435      TYPE
1453 2663 2266      ISZ   XURS
1454 2664 5262      JMP    *-2
1455 2665 5652      JMP I  XCRLF
1456
1457 2666 0000      XORS, 0
1458
1459                /PRINT 2 SPACES
1460

```

```

1461 2667 0000      SPACX2, 0          /CALL BY "SPACE2"
1462 2670 4425      PRNTMS
1463 2671 2673      *-2
1464 2672 5667      JMP I  SPACX2
1465 2673 4040      4040
1466 2674 0010      0010 /USED BY LISN
1467
1468
1469                /COMPARE INPUT TO LIST FOLLOWING CALL
1470                /INPUT ONE CHARACTER IF AC=0
1471                /USE TAD PSRT INPUT IF AC NON ZERO
1472
1473 2675 0000      XLISN, 0          /CALL BY "LISN"
1474 2676 7640      SZA  CLA
1475 2677 5325      JMP   LISN1          /USE TAD PSRT INPUT SINCE AC NOT ZERO
1476 2700 6031      KSF
1477 2701 5300      JMP    *-1
1478 2702 6036      KPB
1479 2703 0357      AND    K177
1480 2704 1064      TAD    K200
1481 2705 3267      DCA    SPACX2
1482 2706 1267      TAD    SPACX2
1483 2707 1361      TAD    M212
1484 2710 7450      SNA
1485 2711 5315      JMP    *-4          /IS IT A LF?
1486 2712 1360      TAD    M3          /YES
1487 2713 7640      SZA  CLA          /IS IT A CR?
1488 2714 5317      JMP    *-3          /NO
1489 2715 4436      CRLF
1490 2716 5325      JMP   LISN1
1491 2717 1267      TAD    SPACX2
1492 2720 1376      TAD    (-223
1493 2721 7650      SNA  CLA          /CHECK FOR A CONTROL S
1494 2722 5325      JMP   LISN1          /WAS IT A CONTROL S
1495 2723 1267      TAD    SPACX2          /YES-DO NOT ECHO CHARACTER
1496 2724 4435      TYPE
1497 2725 1675      LISN1, TAD I  XLISN        /PRINT THE CHARACTER
1498 2726 2275      ISZ   XLISN          /GET COMPARE VALUE
1499 2727 7450      SNA
1500 2730 5336      JMP   LISN3          /EXIT?
1501 2731 7500      SNA
1502 2732 5346      JMP   LISNUM        /YES
1503 2733 1267      TAD    SPACX2          /LOOK FOR OCTAL NUMBER
1504 2734 7640      SZA  CLA          /COMPARE
1505 2735 5343      JMP   LISN2        /EQUAL?
1506 2736 3266      LISN3, DCA    XURS          /NO
1507 2737 1675      TAD I  XLISN
1508 2740 3275      DCA    XLISN
1509 2741 1266      TAD    XURS
1510 2742 5675      JMP I  XLISN          /AC IS ZERO UNLESS OCTAL NUMBER
1511 2743 7200      LISN2, CLA
1512 2744 2275      ISZ   XLISN
1513 2745 5325      JMP   LISN1
1514 2746 7200      LISNUM, CLA          /LOOK FOR OCTAL NUMBER
1515 2747 1267      TAD    SPACX2

```

```

1516 2750 1354 TAD M270
1517 2751 7500 SWA
1518 2752 5343 JMP LISN2 /IS IT LESS THAN 8?
1519 2753 1057 TAD K10 /NO, SO NOT AN OCTAL NUMBER
1520 2754 7510 M270, SPA /IS IT GREATER THAN ZERO?
1521 2755 5343 JMP LISN2 /NO, SO NOT A NUMBER
1522 2756 5336 JMP LISN3
1523 2757 0177 K177, U177
1524 2760 7775 M3, 7775
1525 2761 7566 M212, 7566
1526
1527
1528
1529
1530 2776 7555
1531 2777 3073
1532 3000 PAGE
1533
1534
1535 /PRINT PACKED ASCII TEXT TERMINATED BY
1536 /SIX-BIT 00
1537
1538 3A00 0000 MESAGX, 0 /CALL BY "MESSAGE"
1539 3A01 7200 CLA
1540 3A02 7000 APTMES, NOP /APT
1541 3A03 1600 TAD I MESAGX
1542 3A04 3237 DCA FOROCK
1543 3A05 2200 ISZ MESAGX /SET UP RETURN
1544 3A06 1637 TAD I FOROCK
1545 3A07 7012 RTR
1546 3A10 7012 RTR
1547 3A11 7012 RTR
1548 3A12 4217 JMS MESAGF
1549 3A13 1637 TAD I FOROCK
1550 3A14 4217 JMS MESAGF
1551 3A15 2237 ISZ FOROCK
1552 3A16 5206 JMP *-10
1553 3A17 0000 MESAGF, 0
1554 3A20 0062 AND K77
1555 3A21 7450 SWA /TERMINATOR (00)?
1556 3A22 5600 JMP I MESAGX /YES
1557 3A23 1236 TAD M43
1558 3A24 7450 SWA /CRLF?
1559 3A25 5234 JMP *,+7 /YES
1560 3A26 1055 TAD K3
1561 3A27 7510 SPA /200 OR 300
1562 3A30 1063 TAD K100 /300
1563 3A31 1070 TAD K240 /200
1564 3A32 4435 TYPE
1565 3A33 5617 JMP I MESAGF
1566 3A34 4436 CRLF
1567 3A35 5617 JMP I MESAGF
1568 3A36 7735 M43, 7735
1569 5600 INHMES=JMP I MESAGX
    
```

```

1570
1571
1572 /INPUT 4 OCTAL NUMBERS TO AC
1573 /GOOD RETURN IS CALL+2
1574
1575 3A37 0000 FOROCK, 0 /CALL BY "FOROCK"
1576 3A40 4427 TWOOCT
1577 3A41 5637 JMP I FOROCK
1578 3A42 7106 CLL RTL
1579 3A43 7006 RTL
1580 3A44 7006 RTL
1581 3A45 3253 DCA XPRNT4
1582 3A46 4427 TWOOCT
1583 3A47 5637 JMP I FOROCK
1584 3A50 1253 TAD XPRNT4
1585 3A51 2237 ISZ FOROCK
1586 3A52 5637 JMP I FOROCK
1587
1588
1589 /PRINT FOUR OCTAL NUMBERS IN AC 11 THRU 0 FOLLOWED
1590 /BY TWO SPACES
1591 3A53 0000 XPRNT4, 0 /CALL BY "PRNT4"
1592 3A54 3237 DCA FOROCK
1593 3A55 1237 TAD FOROCK
1594 3A56 7012 RTR
1595 3A57 7012 RTR
1596 3A60 7012 RTR
1597 3A61 4432 PRNT2
1598 3A62 1237 TAD FOROCK
1599 3A63 4432 PRNT2
1600 3A64 4434 SPACE2
1601 3A65 5653 JMP I XPRNT4
1602
1603 /PRINT THE OCTAL NUMBER IN AC 9 THRU 11
1604 XPRNT1, 0 /CALL BY "PRNT1"
1605 3A67 0056 AND K7
1606 3A70 1071 TAD K260
1607 3A71 4435 TYPE
1608 3A72 5666 JMP I XPRNT1
1609
1610
1611 /ROUTINE TO CHECK FOR A CONTROL S WHILE TYPING OUT MESSAGES
1612 /TO EXIT ROUTINE IF-A CONTROL S WAS TYPED-A CONTROL Q OR C MUST BE
1613 /INPUTTED ON THE KEYBOARD
1614
1615 3A73 0000 CNTRLS, 0
1616 3A74 6031 KSF /SKIP ON CONSOLE KEYBOARD FLAG
1617 3A75 5673 JMP I CNTRLS /RETURN TO TYPE ROUTINE-FLAG NOT SET
1618 3A76 6034 KRS /READ THE CHARACTER STATICALLY
1619 3A77 0377 AND (177) /MASK TO 7 BIT ASCII
1620 3A00 1376 TAD (=23) /CHECK FOR A CONTROL S
1621 3A01 7640 SZA CLA /WAS IT A CONTROL S
1622 3A02 5673 JMP I CNTRLS /NO-RETURN WITH KEYBOARD FLAG STILL SET
1623 3A03 6032 KCC /CLEAR KEYBOARD FLAG FROM "S
1624 3A04 4306 JMS WAITQC /WAIT FOR CONTROL Q OR C
    
```

1625 3105 5673
1626

JMP I CNIRLS

/RETURN TO PRINT MESSAGE BEING TYPED

```

1627 3106 0000 WAITQC, 0 /ROUTINE TO WAIT FOR CONTROL Q OR C
1628 3107 5306 JMP *-1 /READ THE CHARACTER TYPED
1629 3110 6036 KRB /MASK TO 7 BIT ASCII
1630 3111 0377 AND (-177 /CHECK FOR A CONTROL C
1631 3112 1375 TAD (-3 /WAS IT A CONTROL C?
1632 3113 7450 SNA /YES-RESTORE MONITOR AND RETURN
1633 3114 5774* JMP CBRM /CHECK FOR A LINE FEED CHARACTER
1634 3115 1373 TAD (-7 /WAS IT A LINE FEED
1635 3116 7450 SNA /YES GO RESTART THE PROGRAM
1636 3117 5324 JMP RESTRT /CHECK FOR A CONTROL Q "Q
1637 3120 1373 TAD (-7 /WAS IT A CONTROL Q
1638 3121 7640 SZA CLA /NO-WAIT FOR APPROPRIATE KEY
1639 3122 5307 JMP WAITQC+1 /RETURN TO WHENCE IT CAME
1640 3123 5706 JMP I WAITQC
1641
1642 /ROUTINE TO RELOCATE PROGRAM TO ORIGINAL PROGRAM FIELD
1643 /AND RESTART PROGRAM.
1644 3124 6224 RESTRT, RIF
1645 3125 1076 TAD K6201
1646 3126 3337 DCA RST1 /MAKE CDF INSTRUCTIONS
1647 3127 1337 TAD RST1
1648 3130 3344 DCA RST3
1649 3131 1772* TAD PRGSTK /ORIGINAL PROGRAM FIELD
1650 3132 3341 DCA RST2
1651 3133 1341 TAD RST2
1652 3134 1054 TAD K2
1653 3135 3353 DCA RST4 /MAKE CBF INST
1654 3136 3046 DCA MOVE /CLEAR IT
1655 3137 6201 RST1, CDF0 /MOVE FROM CURRENT PROGRAM FIELD
1656 3140 1446 TAD I MOVE
1657 3141 6201 RST2, CDF0 /MOVE TO ORIGINAL PROGRAM FIELD
1658 3142 3446 DCA I MOVE
1659 3143 1446 TAD I MOVE
1660 3144 6201 RST3, CDF0 /MOVE FROM CURRENT PROGRAM FIELD
1661 3145 7041 CIA
1662 3146 1446 TAD I MOVE /WAS TRANSFER OK?
1663 3147 7640 SZA CLA /YES - SKIP
1664 3150 4771* JMS ERRM /NO - REPORT AN ERROR
1665 3151 2046 ISZ MOVE /DONE 4096 TIMES?
1666 3152 5337 JMP RST1 /NO
1667 3153 6203 RST4, CBF0 /CHANGE TO NEW PROG FIELD
1668 3154 5770* JMP XSTRT /RESTART PROGRAM
1669
1670
1671 3170 0206
1672 3171 2000
1673 3172 0216
1674 3173 7771
1675 3174 2400
1676 3175 7775
1677 3176 7755
1678 3177 0177
1679 5600
1680

```

/LOC'S 5600 TO 7777 USED TO SAVE EITHER APT OR O88 HANDLERS

4000
4100

4200
4300

4400
4500

4600
4700

5000
5100

5200
5300

5400
5500

5600
5700

6000
6100

6200
6300

6400
6500

6600
6700

7000
7100

7200
7300

7400
7500

7600
7700

ADD1	1517	CLRLST	0311	K34	0257	PART2	1455
ADDCNT	0771	CNTRLG	2547	K4000	0074	PASCNT	0276
ADDINC	0766	CNTRLS	3073	K4060	0075	PASMES	0345
ADRCHK	1555	CNTRS	2553	K6201	0076	PAT1	0773
AGATN	1305	CNTRS1	2555	K6203	0077	PAT2	0774
ALLRAD	0502	COLCNT	0050	K7	0056	PATCNT	0772
APTCTX	2253	CONST	0051	K7577	0100	PCOUNT	0260
APTCTY	2254	COUNT	0047	K7600	0101	PERR1	1656
APTR01	1640	CRLF	4436	K77	0062	PERRM	2014
APTR02	2002	DATA	1345	K7700	0102	PRDYN	1606
APTR03	1530	DONE3	1104	KOUNT	0256	PREL	0331
APTRR	2255	DYNER1	1613	KSPCHK	0104	PRFLD	0546
APTRMX	2251	DYNERR	1527	LDDATA	1254	PRGFLD	2316
APTRMY	2252	DYNTS2	1115	LDJMP	1227	PRGSTK	0216
APTRMS	3002	DYNTS3	1131	LDJMP1	1235	PRNT1	4430
APTRWD	2200	DYNTST	1010	LDJMB1	1220	PRNT2	4432
APTRW0	0213	ERRM	2000	LISN	4424	PRNT4	4433
APTRW1	0223	FILLER	2600	LISN1	2725	PRNTMS	4425
APTRW2	2524	FLD	2314	LISN2	2743	PSEUDO	2440
APTRTP	2637	FLODNT	0765	LISN3	2736	PSR	0020
APTRK	2234	FLODNC	0767	LISNUM	2746	PSTK1	1343
APTRKO	0261	FLOLST	1725	LOAD	1301	PTR	1560
APTRK1	0745	FOROCK	3037	LODFLD	1274	QESTMK	2331
APTRK2	1040	FOROCT	4431	LODUMS	1212	RBELL	2064
ATTP	0475	GDATA	0045	LODSTK	1200	RDF	6214
BADCNT	0501	HAPDON	1330	M212	2761	REDFLD	0700
BADFLD	1724	HCW1	0021	M270	2754	REDLUP	0663
BADPTR	0477	HCW2	0022	M3	2760	REDUPD	0713
BADATK	0443	HEAD	1735	M37	0052	RELERR	2040
BDATA	0044	HEAD1	0037	M40	1351	RELO	0400
BEGIN	0214	IAPTR	0105	M43	3036	RELO1	0422
BELL	2056	IAPTRK	0106	M7	0313	RELO2	0424
BRM	2400	INHMS	5600	M75	0053	RELO3	0427
BRM0	2416	INHYP	5635	MARCH	1600	RELO4	0441
BRM1	2435	INIT	0277	MEMLOD	0631	REPERR	1516
BRF0	6203	INST	1154	MESAGF	3017	RFSTRT	3124
BRF1	6213	JMSLOD	1205	MESAGY	3000	RETCOD	1400
BRF2	6223	K10	0057	MOSERR	1636	RETURN	1513
BRF3	6233	K100	0063	MOSFLD	0634	RETYPE	2506
BRF0	6201	K1000	0073	MOSLOD	0612	RIF	6224
BRF1	6211	K177	2757	MOSLUP	0824	RST1	3137
BRF2	6221	K2	0054	MOSTST	0600	RST2	3141
BRF3	6231	K20	0060	MOVE	0046	RST3	3144
CHAR	2651	K200	0064	NOBELL	2070	RST4	3153
CHARRO	2472	K207	0065	NHFLD	1557	SAME	1144
CHARR1	2476	K212	0066	NACOL	1076	SAVCHR	2521
CHECK	0462	K215	0067	NASTK	0500	SAVFLD	2307
CHGCHR	2454	K240	0070	NXTPAS	0230	SAVOS8	2267
CHGFLD	1340	K260	0071	NXTTST	1111	SETPAT	1000
CHKKSF	2523	K261	0072	ONEOCK	2601	SPACE2	4434
CHKST	1523	K3	0055	ONEOCT	4426	SPACX2	2667
CLRCNT	0312	K30	0061	OVER	1545	SRMESG	2326

SQFST 2442
STACKS 0532
STACT 0200
STKNT 1155
STKNT1 1156
STKIN 0042
STKST 0043
STKHPD 0643
STOADD 2325
STOP 1713
STOP1 2030
STPTAD 1153
TEST 0235
TESTAD 0040
TITLE 0314
TOP 1462
TOPSTK 0041
TSTCNT 0770
TSTK1 1341
TSTORG 1160
TSTPTR 1157
TSTYS 0523
TTYCNT 2522
TWOCK 2611
TWOCKT 4427
TYPE 4435
UPAPRC 2333
UPARRG 2335
UPSTK 1067
WAITGC 3106
X 0453
XAND 1432
XAPT 2277
XCDF 1442
XCRLF 2652
XDY1 1037
XDY2 1047
XDY3 1055
XFLD 1723
XJMI 1447
XLIN 2675
XM37 1347
XM75 1346
XN37 1350
XORS 2666
XPRNT1 3066
XPRNT2 2624

XPRNT4 3053
XSTART 1556
XSTNP 0103
XSTRT 0206
XTYPE 2635

ERRORS DETECTED: 0
LINKS GENERATED: 70
RUN-TIME: 4 SECONDS
3K CORE USED

.L1777	989	1054#							
.L2176	1088	1105#							
.L2177	1084	1106#							
.L2351	1200	1220#							
.L2352	1197	1221#							
.L2353	1189	1222#							
.L2354	1175	1223#							
.L2355	1153	1224#							
.L2356	1133	1225#							
.L2357	1132	1226#							
.L2360	1131	1227#							
.L2361	1130	1228#							
.L2362	1129	1229#							
.L2363	1127	1230#							
.L2364	1125	1231#							
.L2365	1124	1126	1128	1232#					
.L2366	1123	1233#							
.L2367	1121	1234#							
.L2370	1119	1235#							
.L2371	1117	1236#							
.L2372	1115	1237#							
.L2373	1114	1116	1118	1120	1122	1238#			
.L2374	1113	1239#							
.L2375	1111	1240#							
.L2376	1109	1241#							
.L2377	1108	1110	1112	1242#					
.L2574	1366	1369	1373#						
.L2575	1310	1374#							
.L2576	1291	1321	1375#						
.L2577	1255	1376#							
.L2776	1492	1530#							
.L2777	1427	1531#							
.L3170	1668	1671#							
.L3171	1664	1672#							
.L3172	1649	1673#							
.L3173	1634	1637	1674#						
.L3174	1633	1675#							
.L3175	1631	1676#							
.L3176	1620	1677#							
.L3177	1619	1630	1678#						
.V0040	444	675#							
.V0110	434	666#							
.V0177	867	970#	1619	1630	1678#				
.V0206	1668	1671#							
.V0213	1115	1237#							
.V0214	1197	1221#							
.V0216	1200	1220#	1649	1673#					
.V0223	1117	1236#							
.V0261	1109	1241#							
.V0400	173	195	224#						
.V0421	1088	1105#							
.V0523	151	227#							
.V0600	171	226#							

.V0607	1034	1051#							
.V0745	1111	1240#							
.V0771	499	676#							
.V0773	495	496	678#						
.V0774	498	677#							
.V1000	361	490#							
.V1010	172	225#							
.V1037	948	960#							
.V1040	1113	1239#							
.V1076	423	964#							
.V1104	424	963#							
.V1111	552	674#							
.V1144	372	412	489#	880	967#				
.V1153	816	974#							
.V1154	717	811#							
.V1156	888	966#							
.V1160	877	921	968#						
.V1200	584	669#							
.V1274	583	670#							
.V1340	826	831	842	854	861	869	972#		
.V1341	880	671#							
.V1343	667	672#							
.V1400	685	668#							
.V1455	599	667#							
.V1513	825	973#							
.V1516	841	971#							
.V1530	1129	1229#							
.V1636	430	488#							
.V1640	1125	1231#							
.V1725	207	223#	293	331#	892	965#	1024	1053#	
.V2000	264	332#	1664	1672#					
.V2002	1127	1230#							
.V2056	930	962#	989	1054#					
.V2267	131	229#							
.V2277	130	230#							
.V2400	1633	1675#							
.V2440	141	197	228#	303	330#	944	961#	1030	1052#
.V2472	1291	1321	1375#						
.V2476	1310	1374#							
.V2524	1119	1235#							
.V2525	494	679#							
.V2637	1133	1225#							
.V3002	1131	1227#							
.V3073	1427	1531#							
.V3106	1366	1369	1373#						
.V4100	688	702	759	812#					
.V4505	1124	1126	1128	1232#					
.V4506	1108	1110	1112	1242#					
.V5077	773	809#							
.V5110	768	810#							
.V5577	1189	1222#	1255	1376#					
.V5600	1130	1228#							
.V5601	868	969#							

.V5635	1132	1226#				
.V5700	563	673#				
.V6500	1153	1224#				
.V6505	1121	1234#				
.V6520	1175	1223#				
.V6573	1123	1233#				
.V7000	1114	1116	1118	1120	1122	1238#
.V7555	1492	1530#				
.V7755	1620	1677#				
.V7771	1634	1637	1674#			
.V7775	1631	1676#				

