

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DQKDR-A-D
PRODUCT NAME: 11/6X TRAPS TEST
PROGRAM DATE: MARCH 1977
MAINTAINER: DIAGNOSTIC GROUP
AUTHOR: BRUCE BURGESS

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

40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69

TABLE OF CONTENTS

- 1. ABSTRACT
- 2. REQUIREMENTS
 - 2.1 EQUIPMENT
 - 2.2 STORAGE
 - 2.3 PRE-REQUISITE PROGRAMS
 - 2.4 EXECUTION TIME
- 3. LOADING PROCEDURE
 - 3.1 METHOD
- 4. STARTING PROCEDURE
 - 4.1 STARTING ADDRESS
 - 4.2 PROGRAM AND/OR OPERATOR ACTION
- 5. OPERATION PROCEDURE
 - 5.1 OPERATIONAL SWITCHES
 - 5.2 FUNCTION ABSTRACTS
- 6. ERRORS
 - 6.1 ERROR RECOVERY
- 7. RESTRICTIONS
- 8. PROGRAM DESCRIPTION
- 9. ACT/APT COMPATABILITY

70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125

1. ABSTRACT
THIS IS A TEST OF ALL OPERATIONS AND INSTRUCTIONS THAT CAUSE TRAPS. ALSO TESTED ARE TRAP OVERFLOW CONDITIONS, ODDITIES OF REGISTER 6, INTERRUPTS AND THE RESET INSTRUCTIONS. ALL THE RESERVED INSTRUCTION CODES ARE TESTED OUT.
2. REQUIREMENTS
 - 2.1 EQUIPMENT
PDP-11/6X STANDARD COMPUTER
 - 2.2 STORAGE
 - 2.2.1 PROGRAM STORAGE - THE ROUTINE USES MEMORY FROM 0000 TO 22000.
 - 2.3 PRE-REQUISITE PROGRAMS
ALL APPLICABLE BASIC CPU PROGRAMS SHOULD BE RUN, TO VERIFY CORRECT OPERATION OF THE BASIC INSTRUCTIONS.
 - 2.4 EXECUTION TIME
FIRST PASS (NO ITERATIONS), QUICK VERIFY=15 SECONDS
3. LOADING PROCEDURE
 - 3.1 METHOD
PROCEDURE FOR NORMAL ABSOLUTE TAPES SHOULD BE FOLLOWED.
 4. STARTING PROCEDURE
 - 4.1 STARTING ADDRESS
THE PROGRAM STARTS AT 200.
 - 4.2 PROGRAM AND/OR OPERATOR ACTION
LOAD PROGRAM INTO MEMORY.
SET SWITCH REGISTER TO STARTING ADDRESS.
LOAD ADDRESS.
PRESS START.
THE PROGRAM WILL IDENTIFY ITSELF AND TEST EXECUTION WILL BEGIN IT WILL PRINT AN "END OF PASS" MESSAGE AT THE END OF EACH PASS. DURING THE FIRST PASS THERE ARE NO ITERATIONS. SUBSEQUENT PASSES HAVE 15 ITERATIONS.
IF AN ERROR IS DETECTED, THERE WILL BE A HALT.

126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181

WHEN A HALT OCCURS AND IT IS NECESSARY TO SCOPE ON IT, INSERT A "JMP#R0" INSTRUCTION IN THE "LOOP" LOCATION FOLLOWING THE HALT. THE "JMP" INSTRUCTION SHOULD BRANCH YOU TO THE PREVIOUS SCOPE LOCATION.

- 5. OPERATION PROCEDURE
- 5.1 OPERATIONAL SWITCHES
- NO SWITCHES ARE USED
- 5.2 FUNCTION ABSTRACTS

5.2.1 BEGIN SA 200

5.2.2 HLT

INDICATES THE UNIQUE ADDRESS THAT TAGS THE FAILING SUBTEST. THE INCORRECT DATA AT THE TIME OF THE FAILURE MAY OR MAY NOT BE DISPLAYED IN REGISTER ZERO, WHICH IS THE DATA REGISTER ON A HALT.

5.2.3 SCOPE

IS A "MOV R7,R0" INSTRUCTION THAT IS PLACED BETWEEN EACH SUBTEST IN THE INSTRUCTION SECTION. IT ESTABLISHES A POINT TO WHICH THE TEST WILL LOOP BACK TO, IN CASE A SCOPE LOOP IS ENTERED.

5.2.4 LOOP

IS A "NOP" INSTRUCTION PLACED AFTER EACH "HLT". IF A SCOPE LOOP IS DESIRED REPLACE THE "LOOP" BY A "JMP #R0" INSTRUCTION (110) AND PRESS CONTINUE. COMMENTS IN THE LISTINGS ADJACENT TO "LOOP" EXPLAIN HOW TO USE THE SCOPE LOOP. THE TEST WILL LOOP BACK TO THE PREVIOUS SCOPE INSTRUCTION.

5.2.5 TRAPCATCHER

THIS IS A SERIES OF INSTRUCTIONS DESIGNED TO DETECT AND ISOLATE UNEXPECTED TRAPS AND INTERRUPTS, THAT OCCUR IN THE TRAP AND INTERRUPT VECTOR AREA OF MEMORY.

THE PRINCIPAL OF THIS ROUTINE IS: THE VECTOR ENTRANCE ADDRESS POINTS TO THE NEXT SEQUENTIAL WORD WHICH WILL CONTAIN A HALT (000000) (THIS LOCATION IS ALSO THE STATUS WORD FOR THAT VECTOR ENTRANCE, BUT THIS WILL HAVE NO EFFECT ON IT, ALSO BEING THE NEXT INSTRUCTION).

IF A HALT OCCURS IN THE TRAP OR INTERRUPT VECTOR AREA, REGISTER SIX SHOULD BE EXAMINED TO DETERMINE ITS CONTENTS. THEN USE REGISTER SIX CONTENTS AS AN ADDRESS TO DETERMINE THE LOCATION THE PROGRAM WAS AT, WHEN THE INTERRUPT OR

182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237

TRAP OCCURRED. (MEMORY AS SPECIFIED BY R6 CONTAINS THE PC OF THE INSTRUCTION FOLLOWING THE INSTRUCTION WHERE TRAP OCCURRED).

THE CONTENTS OF LOCATION "\$TESTN" CONTAINS THE TEST NUMBER THAT WAS BEING EXECUTED AT THE TIME OF TRAP.

6. ERRORS

6.1 ALL ERRORS WILL CAUSE A HALT.

6.2 ERROR RECOVERY

ON TRAP ERRORS - RESTART AT STARTING ADDRESS
DEPRESS CONTINUE TO CONTINUE TEST

IF THE MACHINE GETS "HUNG" OR FAILS IN A FATAL UNPREDICTABLE MANNER, THE USER CAN FIND OUT THE TEST WHICH WAS BEING EXECUTED AT THE TIME OF FAILURE. THE CONTENTS OF LOCATION "\$TESTN" CONTAINS THE TEST NUMBER THAT WAS BEING EXECUTED AT THE TIME OF FAILURE.

USEFUL INFORMATION CAN ALSO BE GAINED BY EXAMINING THE "PC" AT WHICH THE PROGRAM HALTED AND CROSS-REFERENCING THAT "PC" IN THE PROGRAM LISTINGS.

7. RESTRICTIONS

7.1 STARTING RESTRICTION

THE PROGRAM MUST BE STARTED AT 200.

7.2 OPERATIONAL RESTRICTION

NONE

8. PROGRAM DESCRIPTION

THIS PROGRAM CHECKS THAT ON ALL TRAP OPERATIONS REGISTER 6 IS DECREMENTED THE CORRECT AMOUNT, THAT THE CORRECT PC IS SAVED ON THE STACK, THAT THE OLD CONDITION CODES AND PRIORITY ARE PLACED ON THE STACK AND THAT THE NEW STATUS AND CONDITION CODES ARE CORRECT. BOTH THE "TRAP" AND "EMT" TRAP INSTRUCTIONS ARE TESTED TO SEE THAT ALL COMBINATIONS WILL TRAP. CHECKED ALSO IS THAT ALL RESTRICTED INSTRUCTIONS WILL TRAP. VERIFICATION OF THE "TRT" INSTRUCTION (00003) WHICH IS USED FOR SOFTWARE DEBUG ROUTINES; ODT,DDT, IS DONE. ALSO, THE TRACE BIT IS CHECKED TO SEE IF IT CAUSES A TRAP. STACK OVERFLOW IS CHECKED FOR ALL TRAP INSTRUCTIONS. YELLOW AND RED ZONE VIOLATIONS ARE CHECKED. THE RTI AND RTT INSTRUCTIONS ARE CHECKED FOR CORRECT STACK

230
 239
 240
 241
 242
 243
 244
 245
 246
 247
 248
 249
 250
 251
 252
 253
 254
 255
 256
 257
 258
 259
 260
 261
 262
 263
 264
 265
 266
 267
 268
 269
 270
 271
 272
 273
 274
 275
 276
 277
 278
 279
 280
 281
 282
 283
 284
 285
 286
 287
 288
 289
 290
 291
 292
 293

000000
 000001
 000001
 000002
 010700
 000000
 000003
 000003
 000004
 000004
 000014
 000030
 000014
 000014
 000014
 000020
 000034
 177564
 177560
 177564
 177566
 000240
 000240
 000240
 177776
 000007
 000010
 004700
 000100

OPERATIONS.
 9. ACT/APT COMPATABILITY
 THE PROGRAM IS COMPATIBLE WITH ACT AND APT. UNDER APT,
 BEFORE HALTING ON ERROR, THE PROGRAM PASSES THE TEST
 NUMBER AND ERROR NUMBER TO LOCATIONS "\$TESTN" AND "\$FATAL"
 RESPECTIVELY.
 \$
 ;PDP-11/60 TRAP INSTRUCTION TEST AND ODD ON UNIQUE R6 OPERATIONS
 ;REVISED: SEPTEMBER 1976, JIM KAPADIA
 .LIST ME ;LIST MACRO EXPANSIONS
 .NLIST MC,MD,CMD ;OR MACRO CALLS AND DEFINITIONS
 .ENABL ABS,AMA ;ABSOLUTE BINARY OUTPUT
 TAB=\$0
 LAST=\$1
 \$ENO=1
 \$SN=1
 FIRST=\$2
 SCOPE=01#700 ;MOV REGISTER 7 TO REGISTER ZERO TO TAG LAST TEST
 HLT=HALT
 TRT=3
 BPT=3
 ITRAP5=4
 ERRVEC=4 ;RESEPVED INST AND ILLEGAL ADDRESSES
 TRITVEC=14
 EMTVEC=30
 TRTVEC=14
 BPTVEC=14
 IOTVEC=20
 TPAPVEC=34
 TTCSR=177564
 TRCSR=177560
 TPS=177564
 TPB=177566
 BELL=240
 LOOP=NOP
 NOP=240
 STATUS=177776
 RESINST=000007 ;A RESERVED INSTRUCTION
 RESVEC=10 ;RESERVED INSTRUCTION TRAP VECTOR
 ILLA=004700
 ILLR=100

```

294          177776          CC=177776
295          000006          R6=06
296          000005          P5=05
297
298          076000          MED=076000          ;MAINT. EXAM & DEPOSIT
299
300
301
302          000200          ;=200
303 000200 070137 001102    JWP BEGIN
304          000300          ;=300
305          .SHTTL ACT11 HOOKS
306
307          ;*****
308          ;HOOKS REQUIRED BY ACT11
309          $SVPC=,          ;SAVE PC
310          ;=46
311 000046 021520          $ENDAD          ;:1)SET LOC.46 TO ADDRESS OF $ENDAD IN ;EOP
312          000052          ;=52
313 000052 000000          ;=52          ;:2)SET LOC.52 TO ZERO
314          000300          ;=300          ;: RESTORE PC
315          .SHTTL APT MAILBOX-ETABLE
316
317          ;*****
318          ;FVEN
319 000300          $MAIL:          ;;APT MAILBOX
320 000300 000000          $MSGTY: .WORD AMSGTY ;;MESSAGE TYPE CODE
321 000302 000000          $FATAL: .WORD AFATAL ;;FATAL ERROR NUMBER
322 000304 000000          $TESTN: .WORD ATESTN ;;TEST NUMBER
323 000306 000000          $PASS: .WORD APASS ;;PASS COUNT
324 000310 000000          $DEVCT: .WORD ADEVCT ;;DEVICE COUNT
325 000312 000000          $UNIT: .WORD AUNIT ;;I/O UNIT NUMBER
326 000314 000000          $MSGAD: .WORD AMSGAD ;;MESSAGE ADDRESS
327 000316 000000          $MSGLG: .WORD AMSGLG ;;MESSAGE LENGTH
328 000320          $ETABLE:          ;;APT ENVIRONMENT TABLE
329 000320 000          $ENV: .BYTE AENV ;;ENVIRONMENT BYTE
330 000321 000          $ENVM: .BYTE AENVM ;;ENVIRONMENT MODE BITS
331 000322 000000          $SWREG: .WORD ASWREG ;;APT SWITCH REGISTER
332 000324 000000          $USWR: .WORD AUSWR ;;USER SWITCHES
333 000326 000000          $CPUOP: .WORD ACPUOP ;;CPU TYPE,OPTIONS
334          ;*
335          ;*          BITS 15-11=CPU TYPE
336          ;*          11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
337          ;*          11/70=06,PDO=07,Q=10
338          ;*          BIT 10=REAL TIME CLOCK
339          ;*          BIT 9=FLOATING POINT PROCESSOR
340          ;*          BIT 8=MEMORY MANAGEMENT
341          $ETEND:
342          .MEXIT
343          .SBTTL APT PARAMETER BLOCK
344
345          ;*****
346          ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
347          ;*****
348          ;$X=,          ;;SAVE CURRENT LOCATION
349 000024 000200          ;=24          ;;SET POWER FAIL TO POINT TO START OF PROGRAM
350          200          ;;FOR APT START UP

```

```

350          000044          ;=44          ;;POINT TO APT INDIRECT ADDRESS PNTR.
351 000044 000330          $APTHDR ;;POINT TO APT HEADER BLOCK
352          000330          ;=$X          ;;RESET LOCATION COUNTER
353          ;*****
354          ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
355          ;INTERFACE SPEC.
356
357 000330          $APTHD:
358 000330 000000          $HIRTS: .WORD 0          ;;TWO HIGH BITS OF 10 BIT MAILBOX ADDR.
359 000332 000300          $MBADR: .WORD $MAIL          ;;ADDRESS OF APT MAILBOX (BITS 0-15)
360 000334 000002          $STMT: .WORD 2          ;;RUN TIME OF LONGEST TEST
361 000336 000002          $PASTM: .WORD 2          ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
362 000340 000000          $UNITM: .WORD 0          ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
363 000342 000014          .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
364
365          000304          $STNM=$TFSTN
366          000302          $ERROR=$FATAL
367
368

```

```
369          001000          .,1000
370
371
372 001000 000000          K1:  0
373 001002 000000          K2:  0
374 001004 000000          K3:  0
375 001006 000000          K4:  0
376 001010 000000          K5:  0
377 001012 000000          K6:  0
378 001014 052525          K7:  52525
379 001016 052400          K10: 52400
380 001020 000000          K11:  0
381 001022 000000          K12:  0
382
383 001024 000000          NAMFLG: 0
384
385
386 001026 005215 040515 047111 MSGNAM: .ASCIZ <15><12>.MAINDEC=11-DOKDB-A PDP 11/6X TRAPS TEST.
387 001034 042504 026503 030461
388 001042 042055 045521 041104
389 001050 040455 020040 042120
390 001056 020120 030461 033057
391 001064 020130 051124 050101
392 001072 020123 042524 052123
393 001100 000
394
395          001102          .,EVEN
396
397 001102 012737 177777 021546 BFGIN: MOV      #=1,0#PASSPT      ;CLEAR THE ITERATION COUNTER
398
399
400
401 001110 005037 000300          BEGIN1: CLR      $MSGTY
402 001114 012737 021570 000024      MOV      $PWRDWN,24      ;SET UP THE POWER DOWN VECTOR
403 001122 012737 000340 000026      MOV      $340,26      ;SET UP POWER DOWN PRIORITY
404 001130 005037 000304          CLR      $STSNM
405 001134 005037 000302          CLR      $ERROR
406 001140 012702 000300          MOV      $$MSGTY,R2
407 001144 012703 000302          MOV      $$FATAL,R3
408 001150 012706 021774          MOV      $RUFF,%6      ;SET UP STACK POINTER
409
410          ;PRINT MAINDEC NUMBER AND NAME ONLY THE FIRST TIME
411
412 001154 005737 001024          TST      NAMFLG          ;NAME PRINTED NCE?
413 001160 001021          BNE      BGN3          ;IF YES DONT PRINT AGAIN
414 001162 005237 001024          INC      NAMFLG          ;SET FLAG
415 001166 132737 000040 000321      HITB    $40,$ENVM      ;WILL APT ALLOW PRINTING?
416 001174 001013          BNE      BGN3          ;NO
417 001176 012700 001026          MOV      $MSGNAM,R0
418 001202 105737 177564          BGN2:  TSTB    $#TPS          ;TTY READY?
419 001206 100375          BPL      BGN2
420 001210 112037 177566          MOVB    (R0)+,$#TPB      ;PRINT CHARACTER
421 001214 001372          BNE      BGN2          ;PRINT NEXT ONE IF NOT DONE
422
423 001216 105737 177564          BGN2A: TSTR    $#TPS          ;WAIT FOR DONE
424 001222 100375          BPL      BGN2A
```

```
425 001224          BGN3:
426
427 001224 010700          SCOPE
428          ;*****
429          ;*TEST 1 TEST THAT A TRAP OCCURS ON A RESERVED INSTRUCTION (76000)
430          ;*****
431 001226 005237 000304          TST1:  INC      $#TESTN
432 001232 012706 021774          MOV      $BUFF,%6          ;STACK POINTER SETUP
433 001236 012737 001264 000010      MOV      $RETA,$RESVEC      ;LOAD RESERVED INST. TRAP VECTOR
434 001244 005037 000012          CLR      $RESVEC+2          ;AND STATUS
435 001250 000007          RESINST          ;RESEVED INSTRUCTION (SHOULT TRAP)
436
437          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
438 001254 012713 000001          MOV      $1,(R3)          ;MOVE TO MAIL BOX ($FATAL) THE ERROR
439 001260 000000          HLT      ;ERROR! RESERVED INSTRUCTION FAILED TO TRAP
440 001262 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
441          ;"JMP $R0" (000110) TO GET A
442          ;SCOPE LOOP AND HIT CONTINUE
443 001264 010700          RFTA:  SCOPE
444
445          ;*****
446          ;*TEST 2 TEST DECREMENT OF STACK POINTER ON A RESERVED INSTRUCTION TRAP
447          ;*****
448 001266 005237 000304          TST2:  INC      $#TESTN
449 001272 012706 021774          MOV      $BUFF,%6          ;STACK POINTER SETUP
450 001276 012737 001306 000010      MOV      $RETB,$RESVEC      ;RETURN POINTER
451 001304 000007          RESINST          ;DO A RESERVED INSTRUCTION
452 001306 020627 021770          CMPB    %6,$BUFF-4          ;TEST DECREMENT OF %6
453 001312 001405          BEQ     15
454
455          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
456 001316 012713 000002          MOV      $2,(R3)          ;MOVE TO MAIL BOX ($FATAL) THE ERROR
457 001322 000000          HIT      ;STACK POINTER (%6) WAS NOT PUSHED
458          ;DOWN BY TWO WORDS WHEN A RESERVED
459          ;INSTRUCTION TRAPPED
460 001324 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
461          ;"JMP $R0" (000110) TO GET A
462          ;SCOPE LOOP AND HIT CONTINUE
463 001326 010700          16:  SCOPE
464
465          ;*****
466          ;*TEST 3 TEST THAT PROPER P.C. IS SAVED ON A RESERVED INSTRUCTION TRAP
467          ;*****
468 001330 005237 000304          TST3:  INC      $#TESTN
469 001334 012706 021774          MOV      $BUFF,%6          ;STACK POINTER SETUP
470 001340 012737 001350 000010      MOV      $RETC,$RESVEC      ;RETURN POINTER
471 001346 000007          INSTC: RESINST          ;TRAP ON THIS INSTRUCTION
472 001350 022737 001350 021770 RFTC:  CMPB    $,$BUFF-4          ;CHECK FOR INCREMENTED P.C.
473 001356 001405          BFD     15
474
475          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
476 001362 012713 000003          MOV      $3,(R3)          ;MOVE TO MAIL BOX ($FATAL) THE ERROR
477 001366 000000          HLT      ;WRONG "OLD" PC WAS SAVED ON THE STACK
478          ;WHEN A RESERVED INSTR. TRAPPED
479 001370 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
480          ;"JMP $R0" (000110) TO GET A
```



```

481
482 001372 010700          18:  SCOPE                ;SCOPE LOOP AND HIT CONTINUE
483
484
485
486 001374 005237 000304    TST4:  INC  #0$TESTN          ;*****
487 001400 012706 021774    MOV  #BUFF,%6              ;TEST 4 TEST THAT "OLD" CC AND PRIORITY ARE SAVED ON A RESERVED INSTR. TRAP
488 001404 012737 001420 000010  MOV  #RETF,RESVEC          ;*****
489 001412 005037 177776    CLR  CC                    ;SET UP
490 001416 000007              RESINST                   ;CLEAR CC AND PRIORITY
491 001420 013700 021772    RETD:  MOV  BUFF-2,%0        ;TRAP ON RESERVED INSTRUCTION
492 001424 001405              HEQ  1$                  ;GET SAVED STATUS & TEST FOR ALL 0'S
493
494 001426 005212              INC  (R2)                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
495 001430 012713 000004    MOV  #4,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROP
496 001434 000000              HLT                      ;WRONG "OLD" PSW SAVED ON STACK
497
498
499 001436 000240              LOOP                      ;WHEN A RESERVED INSTR. TRAPPED
500
501
502 001440 010700          18:  SCOPE                ;EXPECT OLD PSW=0
503
504 001442 012706 021774    MOV  #BUFF,%6              ;REPLACE THIS INSTRUCTION BY A
505 001446 012737 001464 000010  MOV  #RETE,RESVEC          ;JMP #00 (000110) TO GET A
506 001454 012737 000357 177776    MOV  #357,CC              ;SCOPE LOOP AND HIT CONTINUE
507 001462 000007              RESINST                   ;PRE SET THE STATUS WORD
508 001464 013700 021772    RETE:  MOV  BUFF-2,%0        ;RESERVED INSTRUCTION TRAP
509
510 001470 022700 000357    CMP  #357,%0              ;GET SAVED STATUS
511 001474 001405              BEQ  1$                  ;SAVED STATUS CORRECT?
512
513 001476 005212              INC  (R2)                ;BRANCH IF CORRECT
514 001500 012713 000005    MOV  #5,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROP
515 001504 000000              HLT                      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
516
517 001506 000240              LOOP                      ;WRONG "OLD" PSW SAVED ON STACK
518
519
520 001510 010700          18:  SCOPE                ;WHEN A RESERVED INSTR. TRAPPED
521
522
523
524
525 001512 005237 000304    TST5:  INC  #0$TESTN          ;*****
526 001516 012706 021774    MOV  #BUFF,%6              ;TEST 5 TEST THAT "NEW" STATUS IS CORRECT ON A RESERVED INSTRUCTION TRAP
527 001522 012737 001544 000010  MOV  #RETF,RESVEC          ;*****
528 001530 005037 000012 177776    CLR  RESVEC+2              ;INITIALIZE THE STACK POINTER
529 001534 012737 000357 177776    MOV  #357,CC              ;SET UP VECTOR
530 001542 000007              RESINST                   ;CLEAR "NEW" STATUS
531 001544 013700 177776    RETF:  MOV  CC,%0          ;PRE SET THE STATUS WORD
532 001550 001405              HEQ  1$                  ;DO A RESERVED INSTRUCTION
533
534 001552 005212              INC  (R2)                ;GET & TEST THE "NEW" STATUS WORD
535 001554 012713 000006    MOV  #6,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROP
536 001560 000000              HLT                      ;BRANCH IF ALL 0'S
537
538
539 001562 000240              LOOP                      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
540
541
542 001564 005037 177776    18:  CLR  CC                ;MOVE TO MAIL BOX ($FATAL) THE ERROP
543 001570 010700          SCOPE                      ;"NEW" PSW WAS INCORRECT WHEN A
544
545 001572 012706 021774    MOV  #BUFF,%6              ;RESERVED INSTR. TRAPPED
546 001576 012737 001620 000010  MOV  #RETF,RESVEC          ;EXPECT NEW PSW=0
547 001604 012737 000357 000012  MOV  #357,RESVEC+2        ;REPLACE THIS INSTRUCTION BY A
548 001612 005037 177776    CLR  CC                    ;JMP #00 (000110) TO GET A
549 001616 000007              RESINST                   ;SCOPE LOOP AND HIT CONTINUE
550 001620 013700 177776    RETG:  MOV  CC,%0          ;LOAD THE NEW STATUS WORD
551 001624 022700 000357    CMP  #357,%0              ;PRE SET THE STATUS WORD
552 001630 001405              HEQ  1$                  ;DO A RESERVED INSTRUCTION
553
554 001632 005212              INC  (R2)                ;GET THE "NEW" STATUS WORD
555 001634 012713 000007    MOV  #7,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROP
556 001640 000000              HLT                      ;WAS "NEW" STATUS CORRECTLY LOADED
557
558
559 001642 000240              LOOP                      ;BRANCH IF CORRECT
560
561
562 001644              18:
563 001644 012737 000012 000010  MOV  #RESVEC+2,RESVEC      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
564 001652 005037 000012 000010  CLR  RESVEC+2             ;MOVE TO MAIL BOX ($FATAL) THE ERROP
565 001656 010700          SCOPE                      ;"NEW" PSW WAS INCORRECT WHEN A
566
567
568
569
570 001660 005237 000304    TST6:  INC  #0$TESTN          ;*****
571 001664 012706 021774    MOV  #BUFF,%6              ;TEST 6 TEST THAT A TRAP OCCURES FOR A "TRAP" INSTRUCTION
572 001670 012737 001716 000034  MOV  #RETA1,TRAPVEC        ;*****
573 001676 005037 000036 177776    CLR  TRAPVEC+2            ;STACK POINTER SETUP
574 001702 104400          TRAP                      ;LOAD TRAP VECTOR
575
576 001704 005212              INC  (R2)                ;DO A TRAP INSTRUCTION
577 001706 012713 000010    MOV  #10,(R3)             ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
578 001712 000000              HLT                      ;MOVE TO MAIL BOX ($FATAL) THE ERROP
579 001714 000240              LOOP                      ;"TRAP" INSTRUCTION DID NOT TRAP
580
581
582 001716 010700          RETA1: SCOPE              ;REPLACE THIS INSTRUCTION BY A
583
584
585
586
587 001720 005237 000304    TST7:  INC  #0$TESTN          ;*****
588 001724 012706 021774    MOV  #BUFF,%6              ;TEST 7 TEST DECREMENT OF STACK POINTER ON A "TRAP" INSTRUCTION
589 001730 012737 001740 000034  MOV  #RETB1,TRAPVEC        ;*****
590 001736 104400          TRAP                      ;STACK POINTER SETUP
591 001740 020627 021770    RETB1: CMP  #6,#BUFF-4        ;RETURN POINTER
592 001744 001405          BFO  1$                  ;DO A TRAP INSTRUCTION
593
594
595
596
597
598
599
600

```

```

537
538
539 001562 000240          LOOP                      ;RESERVED INSTR. TRAPPED
540
541
542 001564 005037 177776    18:  CLR  CC                ;EXPECT NEW PSW=0
543 001570 010700          SCOPE                      ;REPLACE THIS INSTRUCTION BY A
544
545 001572 012706 021774    MOV  #BUFF,%6              ;JMP #00 (000110) TO GET A
546 001576 012737 001620 000010  MOV  #RETF,RESVEC          ;SCOPE LOOP AND HIT CONTINUE
547 001604 012737 000357 000012  MOV  #357,RESVEC+2        ;LOAD THE NEW STATUS WORD
548 001612 005037 177776    CLR  CC                    ;PRE SET THE STATUS WORD
549 001616 000007              RESINST                   ;DO A RESERVED INSTRUCTION
550 001620 013700 177776    RETG:  MOV  CC,%0          ;GET THE "NEW" STATUS WORD
551 001624 022700 000357    CMP  #357,%0              ;WAS "NEW" STATUS CORRECTLY LOADED
552 001630 001405              HEQ  1$                  ;BRANCH IF CORRECT
553
554 001632 005212              INC  (R2)                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
555 001634 012713 000007    MOV  #7,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROP
556 001640 000000              HLT                      ;"NEW" PSW WAS INCORRECT WHEN A
557
558
559 001642 000240              LOOP                      ;RESERVED INSTR. TRAPPED
560
561
562 001644              18:
563 001644 012737 000012 000010  MOV  #RESVEC+2,RESVEC      ;EXPECT NEW PSW=357
564 001652 005037 000012 000010  CLR  RESVEC+2             ;REPLACE THIS INSTRUCTION BY A
565 001656 010700          SCOPE                      ;JMP #00 (000110) TO GET A
566
567
568
569
570 001660 005237 000304    TST6:  INC  #0$TESTN          ;*****
571 001664 012706 021774    MOV  #BUFF,%6              ;TEST 6 TEST THAT A TRAP OCCURES FOR A "TRAP" INSTRUCTION
572 001670 012737 001716 000034  MOV  #RETA1,TRAPVEC        ;*****
573 001676 005037 000036 177776    CLR  TRAPVEC+2            ;STACK POINTER SETUP
574 001702 104400          TRAP                      ;LOAD TRAP VECTOR
575
576 001704 005212              INC  (R2)                ;DO A TRAP INSTRUCTION
577 001706 012713 000010    MOV  #10,(R3)             ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
578 001712 000000              HLT                      ;MOVE TO MAIL BOX ($FATAL) THE ERROP
579 001714 000240              LOOP                      ;"TRAP" INSTRUCTION DID NOT TRAP
580
581
582 001716 010700          RETA1: SCOPE              ;REPLACE THIS INSTRUCTION BY A
583
584
585
586
587 001720 005237 000304    TST7:  INC  #0$TESTN          ;*****
588 001724 012706 021774    MOV  #BUFF,%6              ;TEST 7 TEST DECREMENT OF STACK POINTER ON A "TRAP" INSTRUCTION
589 001730 012737 001740 000034  MOV  #RETB1,TRAPVEC        ;*****
590 001736 104400          TRAP                      ;STACK POINTER SETUP
591 001740 020627 021770    RETB1: CMP  #6,#BUFF-4        ;RETURN POINTER
592 001744 001405          BFO  1$                  ;DO A TRAP INSTRUCTION
593
594
595
596
597
598
599
600

```

```

593
594 001746 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
595 001750 012713 000011  MOV      #11,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
596 001754 000000      HLT                    ;STACK POINTER (R6) DID NOT DECREMENT
597                                     ;TWO WORDS WHEN "TRAP" TRAPPED
598 001756 000244      LOOP                   ;REPLACE THIS INSTRUCTION BY A
599                                     ;"JMP #R0" (000110) TO GET A
600                                     ;SCOPE LOOP AND HIT CONTINUE
601 001760 010700      1S:   SCOPE
602                                     ;*****
603                                     ;*TEST 10 TEST THAT PROPER P.C. IS SAVED ON A "TRAP" INSTRUCTION
604                                     ;*****
605
606 001762 005237 000304  TST10: INC      #0TESTN
607 001766 012706 021774      MOV      #BUFF,#6     ;STACK POINTER SETUP
608 001772 012737 002052 000034  MOV      #RETCL,TRAPVEC ;RETURN FROM TRAP POINTER
609 002000 104400      TRAP                   ;TRAP ON THIS INSTRUCTION
610 002002 022737 002170 RETC1: CMP      #,,BUFF-4   ;CHECK INCREMENTED P.C.
611 002010 001405      HFQ      1$
612
613 002012 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
614 002014 012713 000012  MOV      #12,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
615 002020 000000      HLT                    ;WRONG "OLD" PC SAVED ON THE STACK
616                                     ;WHEN "TRAP" TRAPPED
617 002022 000240      LOOP                   ;REPLACE THIS INSTRUCTION BY A
618                                     ;"JMP #R0" (000110) TO GET A
619                                     ;SCOPE LOOP AND HIT CONTINUE
620 002024 010700      1S:   SCOPE
621                                     ;*****
622                                     ;*TEST 11 TEST THAT "OLD" CC & PRIORITY ARE PLACED ON STACK, ON A "TRAP" TRAP
623                                     ;*****
624 002026 005237 000304  TST11: INC      #0TESTN
625 002032 012706 021774      MOV      #BUFF,#6     ;SET UP
626 002036 012737 002052 000034  MOV      #RETCL,TRAPVEC ;SET UP
627 002044 005037 177776      CLR      CC           ;CLEAR CC AND PRIORITY
628 002050 104400      TRAP                   ;DO A TRAP INSTRUCTION
629 002052 013700 021772  RETD1: MOV      #BUFF-2,#0 ;GET & TEST THE SAVED STATUS
630 002056 001405      BEQ      1$           ;BRANCH IF ALL 0'S
631
632 002060 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
633 002062 012713 000013  MOV      #13,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
634 002066 000000      HLT                    ;WRONG "OLD" PC SAVED ON THE STACK
635                                     ;WHEN "TRAP" TRAPPED, EXPECT "0"
636 002070 000240      LOOP                   ;REPLACE THIS INSTRUCTION BY A
637                                     ;"JMP #R0" (000110) TO GET A
638                                     ;SCOPE LOOP AND HIT CONTINUE
639 002072 010700      1S:   SCOPE
640
641 002074 012706 021774      MOV      #BUFF,#6     ;INITIALIZE THE STACK POINTER
642 002100 012737 002116 000034  MOV      #RETCL,TRAPVEC ;SET UP
643 002106 012737 000357 177776      MOV      #357,CC      ;PRE SET THE STATUS WORD
644 002114 104400      TRAP                   ;DO A TRAP INSTRUCTION
645 002116 013700 021772  PETE1: MOV      #BUFF-2,#0 ;GET THE SAVED STATUS WORD
646 002122 022700 000357      CMP      #357,#0      ;WAS CORRECT STATUS SAVED ON THE STACK
647 002126 001405      BEQ      1$           ;BRANCH IF CORRECT
648

```

```

649 002130 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
650 002132 012713 000014  MOV      #14,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
651 002136 000000      HLT                    ;WRONG "OLD" PC SAVED ON THE STACK
652                                     ;WHEN "TRAP" TRAPPED, EXPECT "357"
653 002140 000240      LOOP                   ;REPLACE THIS INSTRUCTION BY A
654                                     ;"JMP #R0" (000110) TO GET A
655                                     ;SCOPE LOOP AND HIT CONTINUE
656 002142 010700      1S:   SCOPE
657
658                                     ;*****
659                                     ;*TEST 12 TEST THAT "NEW" STATUS IS CORRECT ON A "TRAP" TRAP
660                                     ;*****
661 002144 005237 000304  TST12: INC      #0TESTN
662 002150 012706 021774      MOV      #BUFF,#6     ;CLEAR FUTURE PRIORITY AND CC
663 002154 012737 002176 000034  MOV      #RETF1,TRAPVEC ;PRE SET THE STATUS WORD
664 002162 005037 000036      CLR      TRAPVEC+2
665 002166 012737 000357 177776      MOV      #357,CC      ;PRE SET THE STATUS WORD
666 002174 104400      TRAP                   ;DO A TRAP INSTRUCTION
667 002176 013700 177776  RETF1: MOV      CC,#0   ;GET THE "NEW" STATUS WORD
668 002202 001405      BEQ      1$           ;BRANCH IF ALL 0'S
669
670 002204 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
671 002206 012713 000015  MOV      #15,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
672 002212 000000      HLT                    ;FPPOR! INCORRECT "NEW" PSW LOADED
673 002214 000240      LOOP                   ;REPLACE THIS INSTRUCTION BY A
674                                     ;"JMP #R0" (000110) TO GET A
675                                     ;SCOPE LOOP AND HIT CONTINUE
676 002216 005037 177776      1S:   CLR      CC
677 002222 010700      SCOPE                   ;WHEN "TRAP" TRAPPED, EXPECT 0
678
679 002224 012706 021774      MOV      #BUFF,#6     ;LOAD "NEW" STATUS WORD
680 002230 012737 002252 000034  MOV      #RETF1,TRAPVEC ;PRE SET THE STATUS WORD
681 002236 012737 000357 000036  MOV      #357,TRAPVEC+2 ;DO A TRAP INSTRUCTION
682 002244 005037 177776      CLR      CC           ;GET THE "NEW" STATUS
683 002250 104400      TRAP                   ;IS IT CORRECT?
684 002252 013700 177776  RETG1: MOV      CC,#0   ;BRANCH IF CORRECT
685 002256 022700 000357      CMP      #357,#0
686 002262 001405      BEQ      1$
687
688 002264 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
689 002266 012713 000016  MOV      #16,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
690 002272 000000      HLT                    ;INCORRECT "NEW" PSW LOADED WHEN
691                                     ;"TRAP" TRAPPED, EXPECT 357
692 002274 000240      LOOP                   ;REPLACE THIS INSTRUCTION BY A
693                                     ;"JMP #R0" (000110) TO GET A
694                                     ;SCOPE LOOP AND HIT CONTINUE
695 002276 005037 177776      1S:   CLR      CC
696 002302 010700      SCOPE
697
698                                     ;*****
699                                     ;*TEST 13 TEST THAT ALL COMBINATION OF "TRAP" WILL CAUSE A TRAP
700                                     ;*****
701 002304 005237 000304  TST13: INC      #0TESTN
702 002310 012737 104400 002330  MOV      #TRAP,R01    ;INITIALIZE BASE TRAP INSTRUCTION
703 002316 012737 002344 000034  MOV      #RA1,34      ;RETURN FROM TRAP TO RA1
704 002324 012706 021774  RC1:   MOV      #BUFF,#6     ;SET UP STACK POINTER

```

T13 TEST THAT ALL COMBINATION OF "TRAP" WILL CAUSE A TRAP

```
705 002330 104400      RR1:  TRAP                ;TRAP INST WILL BE MODIFIED TO TRAP+377
706
707 002332 005212      INC      (R2)                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
708 002334 012713 000017  MOV      #17,(R3)            ;MOVE TO MAIL BOX ($FATAL) THE ERROR
709 002340 000000      HLT
710
711 002342 000210      LOOP
712
713
714 002344 005237 002330  FA1:  INC      RR1                ;REPLACE THIS INSTRUCTION BY A
715 002350 022737 104777 002330  CWP      #104777,RR1        ;"JMP 000" (000110) TO GET A
716 002356 103362      BHS      RC1                ;SCOPE LOOP AND HIT CONTINUE
717
718
719 002360 012737 000136 000034  MOV      #36,TRAPVEC        ;INCREMENT TRAP INSTRUCTION
720 002366 005037 000136  CLR      TRAPVEC+2          ;TRAP+377 TO UPPER LIMIT
721 002372 010700      SCOPE                       ;HAVE WF TESTED ALL
                                ;YES
                                ;RESTORE TRAP VECTOR TO
                                ;HALT AT 36
```

T14 TEST THAT A TRAP OCCURES ON AN "IOT" INSTRUCTION

```
722
723
724
725 002374 005237 000304      TST14: INC      ##TESTN          ;*****
726 002400 012706 021774      MOV      #BUFF,%6          ;*TEST 14 TEST THAT A TRAP OCCURES ON AN "IOT" INSTRUCTION
727 002404 012737 002432 000020  MOV      #PETA7,IOTVEC      ;STACK POINTER SETUP
728 002412 005037 000022      CLR      IOTVEC+2          ;RETURN LOCATION
729 002416 000004      IOT
730
731 002420 005212      INC      (R2)                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
732 002422 012713 000020  MOV      #20,(R3)            ;MOVE TO MAIL BOX ($FATAL) THE ERROR
733 002426 000000      HLT
734 002430 000210      LOOP
735
736
737 002432 010700      RETA2: SCOPE                ;"IOT" INSTRUCTION FAILED TO TRAP
738
739
740
741
742
743 002434 005237 000304      TST15: INC      ##TESTN          ;*****
744 002440 012706 021774      MOV      #BUFF,%6          ;*TEST 15 TEST DECREMENT OF STACK POINTER ON AN "IOT" TRAP
745 002444 012737 002454 000020  MOV      #RETB7,IOTVEC      ;STACK POINTER SETUP
746 002452 000004      IOT                          ;RETURN POINTER
747 002460 001405      RETB2: CMP      %6,#BUFF-4    ;TEST DECREMENT OF %6
748
749
750
751 002462 005212      INC      (R2)                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
752 002464 012713 000021  MOV      #21,(R3)            ;MOVE TO MAIL BOX ($FATAL) THE ERROR
753 002470 000000      HLT
754
755
756 002472 000210      LOOP
757
758
759
760
761 002474 010700      IS:   SCOPE                ;STACK POINTER DID NOT DECREMENT BY
762
763
764
765
766
767
768
769 002476 005237 000304      TST16: INC      ##TESTN          ;*****
770 002502 012706 021774      MOV      #BUFF,%6          ;*TEST 16 TEST THAT PROPER P.C. IS SAVED ON AN "IOT" TRAP
771 002506 012737 002516 000020  MOV      #RETC2,IOTVEC      ;STACK POINTER SETUP
772 002514 000004      IOT                          ;RETURN FROM TRAP POINTER
773 002516 022737 002516 021770  RETC2: CMP      #,,BUFF-4    ;TRAP ON THIS INSTRUCTION
774 002524 001105      BFC      IS                ;CHECK FOR INCREMENTED P.C.
775
776
777
778
779
780
781 002526 005212      INC      (R2)                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
782 002530 012713 000022  MOV      #22,(R3)            ;MOVE TO MAIL BOX ($FATAL) THE ERROR
783 002534 000000      HLT
784
785
786
787
788
789
790
791
792
793
794
795 002536 000210      LOOP
796
797
798
799 002540 010700      IS:   SCOPE                ;WRONG "OLD" PC WAS SAVED ON
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
```

```
778 ;*TEST 17 TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK, ON AN "IOT" TRAP
779 ;*****
780 002542 005237 000304 TST17: INC 005TESTN
781 002546 012706 021774 MOV #BUFF,%6 ;SET UP
782 002552 012737 002566 000020 MOV #RETD?,IOTVEC ;SET UP
783 002560 005037 177776 CLR CC ;CLEAR CC AND PRIORITY
784 002564 000004 IOT ;TRAP
785 002566 013700 021772 RETD2: MOV #UFF-2,%0 ;GET & TEST SAVED STATUS
786 002572 001405 BEQ 1$
787
788 002574 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
789 002576 012713 000023 MOV #23,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
790 002602 000000 HLT ;WRONG "OLD" PSW SAVED ON STACK
791 ;WHEN "IOT" TRAPPED, EXPECT 0
792 002604 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
793 ;"JMP #R0" (000110) TO GET A
794 ;SCOPE LOOP AND HIT CONTINUE
795
796
797 002610 012706 021774 MOV #BUFF,%6 ;SET UP
798 002614 012737 002632 000020 MOV #RETE2,IOTVEC ;SET UP
799 002622 012737 000357 177776 MOV #357,CC ;PRE SET STATUS
800 002630 000004 IOT
801 002632 013700 021772 RETE2: MOV #UFF-2,%0 ;GET SAVED STATUS
802 002636 022700 000357 CMP #357,%0 ;SAVED STATUS CORRECT?
803 002642 001404 BEQ 1$ ;BRANCH IF CORRECT
804
805 002644 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
806 002646 012713 000024 MOV #24,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
807 002652 000000 HLT ;WRONG "OLD" PSW SAVED ON STACK
808 ;WHEN "IOT" TRAPPED, EXPECT 357
809 002654 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
810 ;"JMP #R0" (000110) TO GET A
811 ;SCOPE LOOP AND HIT CONTINUE
812
813
814
815 ;*****
816 ;*TEST 20 TEST THAT "NEW" STATUS IS CORRECT ON AN "IOT" TRAP
817 ;*****
818 002660 005237 000304 TST20: INC 005TESTN
819 002664 012706 021774 MOV #BUFF,%6
820 002670 012737 002712 000020 MOV #RETF2,IOTVEC
821 002676 005037 000022 CLR IOTVEC+2 ;CLEAR FUTURE PRIORITY AND CC
822 002702 012737 000357 177776 MOV #30357,CC ;PRE SET STATUS
823 002710 000004 IOT
824 002712 013700 177776 RETF2: MOV CC,%0 ;GET & TEST "NEW" STATUS
825 002716 001405 BEQ 1$ ;BRANCH IF CORRECT
826
827 002720 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
828 002722 012713 000025 MOV #25,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
829 002726 000000 HLT ;WRONG "NEW" PSW LOADED WHEN
830 ;"IOT" TRAPPED, EXPECT 0
831 002730 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
832 ;"JMP #R0" (000110) TO GET A
833 ;SCOPE LOOP AND HIT CONTINUE
834
835
836
837
838
839
840
841
842
843
844
845 003000 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
846 003002 012713 000026 MOV #26,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
847 003006 000000 HLT ;WRONG "NEW" PSW LOADED WHEN
848 ;"IOT" TRAPPED, EXPECT 357
849 003010 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
850 ;"JMP #R0" (000110) TO GET A
851 ;SCOPE LOOP AND HIT CONTINUE
852
853
854 003012 012737 000022 000020 MOV #22,IOTVEC ;RESTORE IOT TRAP VECTOR
855 003020 005037 000022 CLR IOTVEC+2 ;TO HALT AT 22
856 003024 010700 SCOPE
```

```
834 002736 010700 SCOPE
835
836 002740 012706 021774 MOV #BUFF,%6
837 002744 012737 002766 000020 MOV #RETG2,IOTVEC
838 002752 012737 000357 000022 MOV #357,IOTVEC+2 ;LOAD "NEW" STATUS
839 002760 005037 177776 CLR CC ;PRE SET STATUS
840 002764 000004 IOT
841 002766 013700 177776 RETG2: MOV CC,%0 ;GET THE "NEW" STATUS
842 002772 022700 000357 CMP #357,%0 ;IS "NEW" STATUS CORRECT
843 002776 001405 BEQ 1$ ;BRANCH IF CORRECT
844
845 003000 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
846 003002 012713 000026 MOV #26,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
847 003006 000000 HLT ;WRONG "NEW" PSW LOADED WHEN
848 ;"IOT" TRAPPED, EXPECT 357
849 003010 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
850 ;"JMP #R0" (000110) TO GET A
851 ;SCOPE LOOP AND HIT CONTINUE
852
853
854 003012 012737 000022 000020 MOV #22,IOTVEC ;RESTORE IOT TRAP VECTOR
855 003020 005037 000022 CLR IOTVEC+2 ;TO HALT AT 22
856 003024 010700 SCOPE
```

T33 TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK ON A "BPT" TRAP

```
1067
1068
1069
1070 003700 005237 000304 TST32: INC #0$TESTN
1071 003704 012706 021774 MOV #BUFF,06
1072 003710 012737 003724 000014 MOV #RTE4,BPTVEC ;LOAD VECTOP
1073 003716 005037 177776 CLR CC ;CLEAR CC AND PRIORITY
1074 003722 000003 BPT
1075 003724 011700 021772 RFTD4: MOV #UFF-2,00 ;GET SAVED STATUS OFF STACK
1076 003730 001405 REU 10 ;BRANCH IF ALL 0'S
1077
1078 003732 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
1079 003734 012713 000042 MOV #42,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
1080 003740 000000 HLT ;WRONG "OLD" PSW SAVED ON STACK
1081 ;ON "BPT" TRAP, EXPECT 0
1082 003742 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1083 ;"JMP 000" (000110) TO GET A
1084 ;SCOPE LOOP AND HIT CONTINUE
1085 003744 010700 10: SCOPE
1086
1087 003746 012706 021774 MOV #BUFF,06 ;SET UP
1088 003752 012737 003774 000014 MOV #RTE4,BPTVEC ;SET UP
1089 003760 012737 000357 177776 MOV #357,CC ;PKE SET STATUS
1090 003766 000003 BPT
1091 003770 013700 021772 HETE4: MOV #UFF-2,00 ;GET THE SAVED STATUS
1092 003774 022700 000357 CMP #357,00 ;IS IT CORRECT?
1093 003780 001405 BEQ 10 ;BRANCH IF CORRECT
1094
1095 004002 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
1096 004004 012713 000042 MOV #43,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
1097 004010 000000 HLT ;WRONG "OLD" PSW SAVED ON STACK
1098 ;ON "BPT" TRAP, EXPECT 357
1099 004012 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1100 ;"JMP 000" (000110) TO GET A
1101 ;SCOPE LOOP AND HIT CONTINUE
1102 004014 010700 10: SCOPE
1103
1104
1105 ;*****
1106 ;TEST 33 TEST THAT "NEW" STATUS IS CORRECT ON A "BPT" TRAP
1107 ;*****
1108 004016 005237 000304 TST33: INC #0$TESTN
1109 004022 012706 021774 MOV #BUFF,06
1110 004026 012737 004050 000014 MOV #RTE4,BPTVEC
1111 004034 005037 000016 CLR BPTVEC+2 ;CLEAR FUTURE PRIORITY AND CC
1112 004040 012737 000357 177776 MOV #357,CC ;PKE SET STATUS
1113 004046 000003 BPT
1114 004050 013700 177776 RETF4: MOV CC,00 ;GET & TEST "NEW" STATUS
1115 004054 001405 BEQ 10 ;BRANCH IF ALL 0'S
1116
1117 004056 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
1118 004060 012713 000042 MOV #44,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
1119 004064 000000 HLT ;WRONG "NEW" PSW LOADED ON
1120 ;"BPT" TRAP, EXPECT 0
1121 004066 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1122 ;"JMP 000" (000110) TO GET A
;SCOPE LOOP AND HIT CONTINUE
```

T33 TEST THAT "NEW" STATUS IS CORRECT ON A "BPT" TRAP

```
1123 004070 005037 177776 10: CLR CC
1124 004074 010700 SCOPE
1125
1126 004076 012706 021774 MOV #BUFF,06
1127 004102 012737 004124 000014 MOV #RTE4,BPTVEC
1128 004110 012737 000357 000016 MOV #357,BPTVEC+2 ;LOAD "NEW" STATUS
1129 004116 005037 177776 CLR CC ;PKE SET STATUS
1130 004122 000003 BPT
1131 004124 013700 177776 HETG4: MOV CC,00 ;GET THE "NEW" STATUS
1132 004130 022700 000357 CMP #357,00 ;IS IT CORRECT
1133 004134 001405 BEQ 10 ;BRANCH IF CORRECT
1134
1135 004136 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
1136 004140 012713 000042 MOV #45,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
1137 004144 000000 HLT ;WRONG "NEW" PSW LOADED ON
1138 ;"BPT" TRAP, EXPECT 357
1139 004146 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1140 ;"JMP 000" (000110) TO GET A
1141 ;SCOPE LOOP AND HIT CONTINUE
1142 004150 005037 177776 10: CLR CC
1143
1144 004154 012737 000016 000014 MOV #16,BPTVEC ;RESTORE "IRT" TRAP
1145 004162 005037 000016 CLR 16 ;TO HALT AT 16
1146 004166 010700 SCOPE
1147
```

```

1148
1149
1150 ;*****
1151 ;*TEST 34 TEST THAT A TRAP OCCURS ON AN ILLEGAL INSTRUCTION - JMP R0
1152 ;*****
1152 004170 005237 000304 TST34: INC 00$TESTN
1153 004174 012706 021774 MOV 0BUFF,%6 ;STACK POINTER SETUP
1154 004200 012737 004226 000004 MOV 0RETAS,ERRVFC ;RETURN LOCATION
1155 004206 005037 000006 CLR ERRVEC+2
Z 1156 004212 000100 JMP %0 ;ILLEGAL INSTRUCTION, SHOULD TRAP
1157
1158 004214 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1159 004216 012713 000046 MOV 046,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1160 004222 000000 HLT ;ILLEGAL INSTR. DID NOT TRAP
1161 004224 000230 LOOP ;REPLACE THIS INSTRUCTION BY A
; "JMP 0R0" (000110) TO GET A
; SCOPE LOOP AND HIT CONTINUE
1162
1163 RETAS: SCOPE
1164 004226 010700
1165
1166 ;*****
1167 ;*TEST 35 TEST DECREMENT OF STACK POINTER ON AN ILLEGAL INSTRUCTION TRAP
1168 ;*****
1169 004230 005237 000304 TST35: INC 00$TESTN
1170 004234 012706 021774 MOV 0BUFF,%6 ;STACK POINTER SETUP
1171 004240 012737 004250 000004 MOV 0RETB5,ERRVEC ;RETURN POINTER
Z 1172 004246 000100 JMP %0 ;RESERVED INSTRUCTION
1173 004250 020627 021770 RETB5: CMP %6,0BUFF-4 ;TEST DECREMENT OF %6
1174 004254 001405 BEQ 1$
1175
1176 004256 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1177 004260 012713 000047 MOV 047,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1178 004264 000000 HLT ;STACK POINTER DID NOT DECREMENT
1179
1180 004266 000240 LOOP ;BY TWO WORDS ON ILLEGAL INSTR. TRAP
; REPLACE THIS INSTRUCTION BY A
; "JMP 0R0" (000110) TO GET A
; SCOPE LOOP AND HIT CONTINUE
1181
1182 1$: SCOPE
1183 004270 010700
1184
1185 ;*****
1186 ;*TEST 36 TEST THAT PROPER P.C. IS SAVED ON AN ILLEGAL INSTRUCTION TRAP
1187 ;*****
1188 004272 005237 000304 TST36: INC 00$TESTN
1189 004276 012706 021774 MOV 0BUFF,%6 ;STACK POINTER SETUP
1190 004302 012737 004312 000004 MOV 0RETC5,ERRVEC ;RETURN FROM TRAP POINTER
Z 1191 004310 000100 JMP %0 ;TRAP ON THIS INSTRUCTION
1192 004312 022737 004312 021770 RETC5: CMP %1,0BUFF-4 ;CHECK FOR INCREMENTED P.C.
1193 004320 001405 BEQ 1$
1194
1195 004322 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1196 004324 012713 000050 MOV 050,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1197 004330 000000 HLT ;WRONG "OLD" PC SAVED ON STACK
1198
1199 004332 000240 LOOP ;ON ILLEGAL INSTR. TRAP
; REPLACE THIS INSTRUCTION BY A
; "JMP 0R0" (000110) TO GET A
; SCOPE LOOP AND HIT CONTINUE
1200
1201 1$: SCOPE
1202 004334 010700
1203

```

```

1204
1205 ;*****
1206 ;*TEST 37 TEST THAT "OLD" CC AND PRIORITY ARE SAVED ON AN ILLEGAL INSTR. TRAP
1207 ;*****
1207 004336 005237 000304 TST37: INC 00$TESTN
1208 004342 012706 021774 MOV 0BUFF,%6 ;SET UP
1209 004346 012737 004362 000004 MOV 0RETD5,ERRVEC ;SET UP
Z 1210 004354 005037 177776 CLR CC ;CLEAR CC AND PRIORITY
1211 004360 000100 JMP %0 ;TRAP
1212 004362 013700 021772 RETD5: MOV 0BUFF-2,%0 ;GET THE SAVED STATUS
1213 004366 001405 BEQ 1$ ;BRANCH IF ALL 0'S
1214
1215 004370 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1216 004372 012713 000051 MOV 051,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1217 004376 000000 HLT ;ERROR! SAVED PSW IS INCORRECT
1218
1219 004400 000240 LOOP ;ON ILLEGAL INSTR. TRAP, EXPECT 0
; REPLACE THIS INSTRUCTION BY A
; "JMP 0R0" (000110) TO GET A
; SCOPE LOOP AND HIT CONTINUE
1220
1221 1$: SCOPE
1222 004402 010700
1223
1224 004404 012706 021774 MOV 0BUFF,%6 ;SET UP
1225 004410 012737 004426 000004 MOV 0RETE5,ERRVEC ;SET UP
Z 1226 004416 012737 000357 177776 MOV 0357,CC ;PRE SET STATUS
1227 004424 000100 JMP %1
1228 004426 013700 021772 RETE5: MOV 0BUFF-2,%0 ;GET THE SAVED STATUS
1229 004432 022700 000357 CMP 0357,%0 ;IS IT CORRECT
1230 004436 001405 BEQ 1$ ;BRANCH IF SAVED STATUS IS CORRECT
1231
1232 004440 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1233 004442 012713 000052 MOV 052,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1234 004446 000000 HLT ;ERROR! SAVED PSW IS INCORRECT
1235
1236 004450 000240 LOOP ;ON ILLEGAL INSTR. TRAP, EXPECT 357
; REPLACE THIS INSTRUCTION BY A
; "JMP 0R0" (000110) TO GET A
; SCOPE LOOP AND HIT CONTINUE
1237
1238 1$: SCOPE
1239 004452 010700
1240
1241 ;*****
1242 ;*TEST 40 TEST THAT "NEW" STATUS IS CORRECT ON AN ILLEGAL INSTR. TRAP
1243 ;*****
1244 004454 005237 000304 TST40: INC 00$TESTN
1245 004460 012706 021774 MOV 0BUFF,%6
1246 004464 012737 004506 000004 MOV 0RETF5,ERRVEC
1247 004472 005037 000006 CLR ERRVEC+2 ;CLEAR FUTURE PRIORITY AND CC
Z 1248 004476 012737 000357 177776 MOV 0357,CC ;PRE SET STATUS
1249 004504 000100 JMP %0
1250 004506 013700 177776 RETF5: MOV CC,%0 ;GET & TEST "NEW" STATUS
1251 004512 001405 BEQ 1$
1252
1253 004514 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1254 004516 012713 000053 MOV 053,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1255 004522 000000 HLT ;ERROR! "NEW" PSW IS INCORRECT
1256
1257 004524 000240 LOOP ;ON ILLEGAL INSTR. TRAP, EXPECT 0
; REPLACE THIS INSTRUCTION BY A
; "JMP 0R0" (000110) TO GET A
; SCOPE LOOP AND HIT CONTINUE
1258
1259

```

```

1260 004526 005037 177776 1S: CLR CC
1261 004532 010700 SCOPE
1262
1263 004534 012706 021774 MOV #BUFF,%6
1264 004540 012737 004562 000004 MOV #RETG5,ERRVEC
1265 004546 012737 000357 000006 MOV #357,FRPVFC+2 ;LOAD "NEW" STATUS
1266 004554 005037 177776 CLR CC ;PRE SET STATUS
Z 1267 004560 000100 JMP %R
1268 004562 013700 177776 RETG5: MOV CC,%0 ;GET THE "NEW" STATUS
1269 004566 022700 000357 CMP #357,%0 ;IS IT CORRECT
1270 004572 001405 BFC 1S ;BRANCH IF CORRECT
1271
1272 004574 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1273 004576 012713 000054 MOV #54,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1274 004602 000000 HLT ;ERROR: "NEW" STATUS IS INCORRECT
1275 ;ON ILLEGAL INSTR. TRAP, EXPECT 357
1276 004604 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1277 ;"JMP #R0" (000110) TO GET A
1278 ;SCOPE LOOP AND HIT CONTINUE
1279 004606 005037 177776 1S: CLR CC
1280 004612 010700 SCOPE
  
```

```

1281
1282 ;*****
1283 ;*TEST 41 TEST THAT A TRAP OCCURS ON ALL ILLEGAL INSTRUCTION - JSR R0,R0
1284 ;*****
1285 004614 005237 000304 TST41: INC #TESTN
1286 004620 012706 021774 MOV #BUFF,%6 ;STACK POINTER SETUP
1287 004624 012737 004652 000004 MOV #RETH5,ERRVEC ;RETURN LOCATION
1288 004632 005037 000006 CLR ERVFC+2
Z 1289 004636 004000 JSR %0,%0 ;ILLEGAL INSTRUCTION (SHOULD TRAP)
1290
1291 004640 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1292 004642 012713 000055 MOV #55,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1293 004646 000000 HLT ;ILLEGAL INSTRUCTION FAILED TO TRAP
1294 004650 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1295 ;"JMP #R0" (000110) TO GET A
1296 ;SCOPE LOOP AND HIT CONTINUE
1297 004652 010700 RETH5: SCOPE
1298
1299 ;TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
1300 004654 012706 021774 MOV #BUFF,%6 ;STACK POINTER SETUP
1301 004660 012737 004670 000004 MOV #RETJ,ERRVEC ;RETURN POINTER
Z 1302 004666 004000 JSR %0,%0 ;RESEVED INSTRUCTION
1303 004670 020627 021770 HETJ: CMP %6,%BUFF-4 ;TEST DECREMENT OF %6
1304 004674 001405 BEQ 1S
1305
1306 004676 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1307 004700 012713 000056 MOV #56,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1308 004704 000000 HLT ;R6 DID NOT DECREM BY 2 WORDS
1309 ;ON ILLEGAL INSTR. TRAP
1310 004706 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1311 ;"JMP #R0" (000110) TO GET A
1312 ;SCOPE LOOP AND HIT CONTINUE
1313 004710 010700 1S: SCOPE
1314
1315 ;*****
1316 ;*TEST 42 TEST THAT PROPER P.C. IS SAVED ON AN ILLEGAL INSTR. TRAP
1317 ;*****
1318 004712 005237 000304 TST42: INC #TESTN
1319 004716 012706 021774 MOV #BUFF,%6 ;STACK POINTER SETUP
1320 004722 012737 004732 000004 MOV #RETJ,ERRVEC ;RETURN FROM TRAP POINTER
Z 1321 004730 004000 JSR %0,%0 ;TRAP ON THIS INSTRUCTION
1322 004732 022737 004732 021770 RETK: CMP #INSTK+2,BUFF-4 ;CHECK FOR INCREMENTED P.C.
1323 004740 001405 BEQ 1S
1324
1325 004742 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1326 004744 012713 000057 MOV #57,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1327 004750 000000 HLT ;WRONG "OLD" PC SAVED ON STACK
1328 ;ON ILLEGAL INSTR. TRAP
1329 004752 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1330 ;"JMP #R0" (000110) TO GET A
1331 ;SCOPE LOOP AND HIT CONTINUE
1332 004754 010700 1S: SCOPE
1333
1334 ;*****
1335 ;*TEST 43 TEST THAT "OLD" CC AND PRIORITY ARE SAVED ON AN ILLEGAL INSTR. TRAP
1336 ;*****
  
```

```
1337 004756 005237 000304 TST43: INC #0STESTN
1338 004762 012706 021774 MOV #RUFF,#6 ;SET UP
1339 004766 012737 005402 000004 MOV #RETN,ERRVEC ;SET UP
1340 004774 005037 177776 CLR CC ;CLEAR CC AND PRIORITY
Z 1341 005400 004000 JSR #0,#0
1342 005002 013700 021772 RETL: MOV #RUFF-2,#0 ;GET & TEST SAVED STATUS
1343 005006 001405 BEQ IS ;BRANCH IF ALL 0'S
1344
1345 005010 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1346 005012 012713 000004 MOV #0,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1347 005016 000000 HLT ;ERROR! INCORRECT PSW SAVED ON STACK
1348
1349 005020 000700 LOOP ;ON ILLEGAL INSTR. TRAP, EXPECT 0
1350 ;REPLACE THIS INSTRUCTION BY A
1351 ;"JMP #R0" (000110) TO GET A
1352 005022 010700 IS: SCOPE ;SCOPE LOOP AND HIT CONTINUE
1353
1354 005024 012706 021774 MOV #RUFF,#6 ;SET UP
1355 005030 012737 005406 000004 MOV #RETN,ERRVEC ;SET UP
1356 005036 012737 000357 177776 MOV #357,CC ;PRE SFT STATUS
Z 1357 005044 000000 JSR #0,#0
1358 005046 013700 021772 RETM: MOV #RUFF-2,#0 ;GET SAVED STATUS
1359 005052 022700 000357 CMP #357,#0 ;IS IT CORRECT
1360 005056 001405 BFC IS
1361
1362 005060 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1363 005062 012713 000004 MOV #0,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1364 005066 000000 HLT ;ERROR! INCORRECT PSW SAVED ON STACK
1365
1366 005070 000200 LOOP ;ON ILLEGAL INSTR. TRAP, EXPECT 357
1367 ;REPLACE THIS INSTRUCTION BY A
1368 ;"JMP #R0" (000110) TO GET A
1369 005072 010700 IS: SCOPE ;SCOPE LOOP AND HIT CONTINUE
1370
1371 ;*****
1372 ;*TEST 44 TEST THAT "NEW" STATUS IS CORRECT ON AN ILLEGAL INSTR. TRAP
1373 ;*****
1374 005074 005237 000304 TST44: INC #0STESTN
1375 005100 012706 021774 MOV #RUFF,#6
1376 005104 012737 005126 000004 MOV #RETN,ERRVEC
1377 005112 005037 000006 CLR #ERRVEC+2 ;CLEAR FUTURE PRIORITY AND CC
Z 1378 005116 012737 000357 177776 MOV #357,CC ;PRE SET STATUS
1379 005124 004000 JSR #0,#0
1380 005126 013700 021772 RETN: MOV #RUFF-2,#0 ;GET & TEST "NEW" STATUS
1381 005132 001405 BEQ IS ;BRANCH IF ALL 0'S
1382
1383 005134 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1384 005136 012713 000004 MOV #0,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1385 005142 000000 HLT ;ERROR! "NEW" PSW IS INCORRECT
1386
1387 005144 000200 LOOP ;ON ILLEGAL INSTR. TRAP
1388 ;REPLACE THIS INSTRUCTION BY A
1389 ;"JMP #R0" (000110) TO GET A
1390 005146 005037 177776 IS: CLR CC ;SCOPE LOOP AND HIT CONTINUE
1391 005152 010700 SCOPE
1392
```

```
1393 005154 012706 021774 MOV #RUFF,#6
1394 005160 012737 005202 000004 MOV #RETN,ERRVEC
1395 005166 012737 000357 000006 MOV #357,ERRVEC+2 ;LOAD "NEW" STATUS
Z 1396 005174 005037 177776 CLR CC ;PRE SET STATUS
1397 005200 004000 JSR #0,#0
1398 005202 013700 177776 RETO: MOV #RUFF-2,#0 ;GET THE "NEW" STATUS
1399 005206 022700 000357 CMP #357,#0 ;IS IT CORRECT
1400 005212 001405 BFC ;BRANCH IF CORRECT
1401
1402 005214 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1403 005216 012713 000004 MOV #0,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1404 005222 000000 HLT ;ERROR! "NEW" PSW IS INCORRECT
1405
1406 005224 000200 LOOP ;ON ILLEGAL INSTR. TRAP
1407 ;REPLACE THIS INSTRUCTION BY A
1408 ;"JMP #R0" (000110) TO GET A
1409 005226 005037 177776 IS: CLR CC ;SCOPE LOOP AND HIT CONTINUE
1410 005232 010700 SCOPE
1411
1412 ;*****
1413 ;*TEST 45 TEST THAT A TRAP OCCURS ON AN ILLEGAL ADDRESS (ODD)
1414 ;*****
1415
1416 005234 005237 000304 TST45: INC #0STESTN
1417 005240 012706 021774 MOV #RUFF,#6 ;STACK POINTER SETUP
1418 005244 012737 005274 000004 MOV #RETN,ERRVEC ;RETURN LOCATION
1419 005252 005037 000006 CLP #ERRVEC+2
1420 005256 005737 000001 TST 1 ;ILL. ADPS. (ODD ADDRESS ON WORD INST.)
1421
1422 005262 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1423 005264 012713 000004 MOV #0,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1424 005270 000000 HLT ;ILLEGAL ADDRESS DID NOT TRAP
1425 005272 000200 LOOP ;REPLACE THIS INSTRUCTION BY A
1426 ;"JMP #R0" (000110) TO GET A
1427
1428 005274 010700 RETP: SCOPE ;SCOPE LOOP AND HIT CONTINUE
1429
```



```

1430
1431
1432
1433
1434 005276 005237 000304 TST46: INC 005TESTN
1435 005302 012706 021774 MOV 005BUFF,06 ;STACK POINTER SETUP
1436 005306 012737 005324 000004 MOV 005PETS,ERRVEC ;RETURN POINTER
1437 005314 005737 000001 TST 1 ;RESERVED INSTRUCTION
1438 005320 020627 021770 RETQ: CMP 06,005BUFF-4 ;TEST DECREMENT OF 06
1439 005324 001405 REG 15
1440
1441 005326 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
1442 005330 012713 000065 MOV 065,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
1443 005334 000000 HLT ;R6 DID NOT DECREMENT BY 2 WORDS
1444
1445 005336 000214 LOOP ;ON AN OCTAL ADDRESS TRAP
1446 ;REPLACE THIS INSTRUCTION BY A
1447 ;"JMP 000" (000110) TO GET A
1448 005340 010700 15: SCOPE ;SCOPE LOOP AND HIT CONTINUE
1449
1450
1451
1452
1453 005342 005237 000304 TST47: INC 005TESTN
1454 005346 012706 021774 MOV 005BUFF,06 ;STACK POINTER SETUP
1455 005352 012737 005364 000004 MOV 005PETS,ERRVEC ;RETURN FROM TRAP POINTER
1456 005360 005737 000001 TST 1 ;TRAP ON THIS INSTRUCTION
1457 005364 022737 005364 021770 RETR: CMP 0,005BUFF-4 ;CHECK FOR INCREMENTED P.C.
1458 005372 001405 REG 15
1459
1460 005374 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
1461 005376 012713 000066 MOV 066,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
1462 005402 000000 HLT ;WRONG "OLD" PC SAVED ON STACK
1463 ;ON ODD ADDRESS TRAP
1464 005404 000214 LOOP ;REPLACE THIS INSTRUCTION BY A
1465 ;"JMP 000" (000110) TO GET A
1466 ;SCOPE LOOP AND HIT CONTINUE
1467 005406 010700 15: SCOPE
1468
1469
1470
1471
1472 005410 005237 000304 TST50: INC 005TESTN
1473 005414 012706 021774 MOV 005BUFF,06 ;SET UP
1474 005420 012737 005436 000004 MOV 005PETS,ERRVEC ;SET UP
1475 005426 005037 177776 CLR CC ;CLEAR CC AND PRIORITY
1476 005432 005737 000001 TST 1
1477 005436 013700 021772 RETS: MOV 005BUFF-2,06 ;GET & TEST SAVED STATUS ON STACK
1478 005442 001405 REG 15 ;BRANCH IF SAVED STATUS IS CORRECT
1479
1480 005444 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
1481 005446 012713 000067 MOV 067,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
1482 005452 000000 HLT ;ERROR! SAVED PSW IS INCORRECT
1483 005454 000214 LOOP ;REPLACE THIS INSTRUCTION BY A
1484 ;"JMP 000" (000110) TO GET A
1485 ;SCOPE LOOP AND HIT CONTINUE
    
```

```

1486 005456 010700 15: SCOPE
1487
1488 005460 012706 021774 MOV 005BUFF,06 ;SET UP
1489 005464 012737 005504 000004 MOV 005RETT,ERRVEC ;SET UP
1490 005472 012737 000357 177776 MOV 005357,CC ;PRE SET STATUS
1491 005500 005737 000001 TST 1
1492 005504 013700 021772 RETT: MOV 005BUFF-2,06 ;GET THE SAVED STATUS OFF STACK
1493 005510 022700 000357 CMP 005357,00 ;IS IT CORRECT
1494 005514 001405 REG 15 ;BRANCH IF CORRECT
1495
1496 005516 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
1497 005520 012713 000070 MOV 070,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
1498 005524 000000 HLT ;ERROR! SAVED PSW ON STACK IS INCORRECT
1499 005526 000214 LOOP ;REPLACE THIS INSTRUCTION BY A
1500 ;"JMP 000" (000110) TO GET A
1501 ;SCOPE LOOP AND HIT CONTINUE
1502 005530 010700 15: SCOPE
1503
1504
1505
1506
1507 005532 005237 000304 TST51: INC 005TESTN
1508 005536 012706 021774 MOV 005BUFF,06 ;SET UP
1509 005542 012737 005566 000004 MOV 005PETS,ERRVEC ;SET UP
1510 005550 005037 000006 CLR ERRVEC+2 ;CLEAR FUTURE PRIORITY AND CC
1511 005554 012737 000357 177776 MOV 005357,CC ;PRE SET STATUS
1512 005562 005737 000001 TST 1
1513 005566 013700 177776 RETIU: MOV 005CC,06 ;GET & TEST THE "NEW" STATUS
1514 005572 001405 REG 15 ;BRANCH IF CORRECT
1515
1516 005574 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
1517 005576 012713 000071 MOV 071,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
1518 005602 000000 HLT ;ERROR! "NEW" PSW IS INCORRECT
1519 005604 000214 LOOP ;REPLACE THIS INSTRUCTION BY A
1520 ;"JMP 000" (000110) TO GET A
1521 ;SCOPE LOOP AND HIT CONTINUE
1522 005606 005037 177776 15: CLR CC
1523 005612 010700 SCOPE
1524
1525 005614 012706 021774 MOV 005BUFF,06 ;LOAD "NEW" STATUS
1526 005620 012737 005640 000004 MOV 005RETT,ERRVEC
1527 005626 012737 000357 000006 MOV 005357,ERRVEC+2
1528 005634 005737 000001 TST 1
1529 005640 013700 177776 RETIV: MOV 005CC,06 ;GET THE "NEW" STATUS
1530 005644 022700 000357 CMP 005357,00 ;IS IT CORRECT
1531 005650 001405 REG 15 ;BRANCH IF CORRECT
1532
1533 005652 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
1534 005654 012713 000072 MOV 072,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
1535 005660 000000 HLT ;ERROR! "NEW" STATUS IS INCORRECT
1536 005662 000214 LOOP ;REPLACE THIS INSTRUCTION BY A
1537 ;"JMP 000" (000110) TO GET A
1538 ;SCOPE LOOP AND HIT CONTINUE
1539 005664 005037 177776 15: CLR CC
1540 005670 010700 SCOPE
1541
    
```

T52 TEST THAT AN ODDSOURCE INTERMEDIATE ADDRESS CAUSES AN ODD ADDRESS TRAP

```

1542
1543 ;*****
1544 ;*TEST 52 TEST THAT AN ODDSOURCE INTERMEDIATE ADDRESS CAUSES AN ODD ADDRESS TRAP
1545 ;*****
1545 005672 005237 000304 TST52: INC #STESTN
1546 005676 012706 021774 MOV #BUFF,%6 ;SET UP STACK POINTER
1547 005702 012737 005736 000004 MOV #OAE0,4 ;LOAD ERROR VECTOR
1548 005713 005037 000006 CLP 6
1549 005714 012701 000001 MOV #1,%1 ;LOAD INDEX REGISTER
1550 005720 067100 000000 ADD #0(1),%0 ;SRC ADPS (R1) IS ODD
1551
1552 005724 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1553 005726 012713 000073 MOV #73,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1554 005732 000000 HLT ;ERROR! ODD ADRES ERROR FAILED TO TRAP
1555 005734 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1556 ;"JMP #R0" (000110) TO GET A
1557 ;SCOPE LOOP AND HIT CONTINUE
1558 005736 010700 OAE0: SCOPE
1559
1560 005740 012706 021774 MOV #BUFF,%6
1561 005744 012737 005774 000004 MOV #OAE1,4
1562 005752 012701 021773 MOV #BUFF-1,%1
1563 005756 147100 000000 BICR #0(1),%0 ;SRC INT ADRS (0(1)) IS ODD
1564
1565 005762 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1566 005764 012713 000074 MOV #74,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1567 005770 000000 HLT ;ERROR! ODD ADPS IN DST FAILED TO TRAP
1568 005772 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1569 ;"JMP #R0" (000110) TO GET A
1570 ;SCOPE LOOP AND HIT CONTINUE
1571 005774 010700 OAE1: SCOPE
1572
1573 ;*****
1574 ;*TEST 53 TEST THAT AN ODD SOURCE FINAL ADDRESS WILL CAUSE AN ODD ADDRESS TRAP
1575 ;*****
1576 005776 005237 000304 TST53: INC #STESTN
1577 006002 012706 021774 MOV #BUFF,%6
1578 006006 012737 006042 000004 MOV #OAE2,4
1579 006014 012737 000001 000100 MOV #1,%0
1580 006022 005001 CLR %1
1581 006024 017100 000100 MOV #100(1),%0 ;SRC FINAL ADPS IS ODD
1582
1583 006030 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1584 006032 012713 000075 MOV #75,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1585 006036 000000 HLT ;ERROR! ODD FINAL SRC ADPS FAILED TO TRAP
1586 006040 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1587 ;"JMP #R0" (000110) TO GET A
1588 ;SCOPE LOOP AND HIT CONTINUE
1589 006042 017000 OAE2: SCOPE
1590
1591 ;*****
1592 ;*TEST 54 TEST THAT AN ODD DEST INTERMEDIATE ADDRESS CAUSES AN ODD ADPS TRAP
1593 ;*****
1594 006044 005237 000304 TST54: INC #STESTN
1595 006050 012706 021774 MOV #BUFF,%6
1596 006054 012737 006104 000004 MOV #OAE3,4
1597 006062 012701 000001 MOV #1,%1

```

T54 TEST THAT AN ODD DEST INTERMEDIATE ADDRESS CAUSES AN ODD ADPS TRAP

```

1598 006066 074071 000000 XOR %0,%0(1) ;DST INT ADPS (0(1)) IS ODD
1599
1600 006072 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1601 006074 012713 000076 MOV #76,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1602 006100 000000 HLT ;ERROR! ODD ADPS ERROR FAILED TO TRAP
1603 006102 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1604 ;"JMP #R0" (000110) TO GET A
1605 ;SCOPE LOOP AND HIT CONTINUE
1606 006104 010700 OAE3: SCOPE
1607
1608 006106 012706 021774 MOV #BUFF,%6
1609 006112 012737 006136 000004 MOV #OAE4,4
1610 006120 005001 CLP %1
1611 006122 122131 CMPB (1)+,%(1)+ ;DST INT ADPS IS ODD ((R1) =1)
1612
1613 006124 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1614 006126 012713 000077 MOV #77,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1615 006132 000000 HLT ;ERROR! ODD ADPS ERROR FAILED TO TRAP
1616 006134 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1617 ;"JMP #R0" (000110) TO GET A
1618 ;SCOPE LOOP AND HIT CONTINUE
1619 006136 010700 OAE4: SCOPE
1620
1621 ;*****
1622 ;*TEST 55 TEST THAT AN ODD DEST FINAL ADPS WILL CAUSE AN ODD ADPS TRAP
1623 ;*****
1624 006140 005237 000304 TST55: INC #STESTN
1625 006144 012706 021774 MOV #BUFF,%6
1626 006150 012737 006204 000004 MOV #OAE5,4
1627 006156 012737 000001 000100 MOV #1,%0
1628 006164 005001 CLR %1
1629 006166 006771 000100 SXT #100(1) ;DST FINAL ADPS IS ODD
1630
1631 006172 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1632 006174 012713 000100 MOV #100,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1633 006200 000000 HLT ;ERROR! DST ODD ADPS ERR FAILED TO TRAP
1634 006202 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
1635 ;"JMP #R0" (000110) TO GET A
1636 ;SCOPE LOOP AND HIT CONTINUE
1637 006204 010700 OAE5: SCOPE
1638
1639 ;*****
1640 ;*TEST 56 TEST THAT CMPB #(R)+,%(R)+ DOES NOT FAIL
1641 ;*****
1642 006206 005237 000304 TST56: INC #STESTN
1643 006212 012706 021774 MOV #BUFF,%6 ;INITIALIZE THE STACK POINTER
1644 006216 012737 006324 000004 MOV #OAE6,4
1645 006224 005001 CLP %1
1646 006226 012737 021415 000000 MOV #TEMP+1,%0
1647 006234 012737 021416 000002 MOV #TEMP+2,%0
1648 006242 012737 123000 021414 MOV #123000,TEMP
1649 006250 012737 177246 021416 MOV #177246,TEMP+2
1650 006256 123131 CMPB #(1)+,%(1)+ ;COMP. LOC. (MYTF) TEMP+1 & TEMP+2
1651 006260 001405 BEQ 15 ;
1652
1653 006262 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR

```

```
1654 006264 012713 000101      MOV      #101,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1655 006270 000000      HLT                      ;COMPARE INSTRUCTION FAILED
1656 006272 000240      LOOP                     ;REPLACF THIS INSTRUCTION BY A
1657                                     ;"JMP R0" (000110) TO GET A
1658                                     ;SCOPE LOOP AND HIT CONTINUE
1659 006274 022701 000004      1$:  CMP      #4,#1      ;DID REGISTER INCREMENT PROPERLY
1660 006300 001405      BEQ                      ;
1661                                     ;
1662 006302 005212      INC      (R2)           ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1663 006304 012713 000102      MOV      #102,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1664 006310 000000      HLT                      ;ERROR! R1 DID NOT INC. BY +2 TWICE
1665 006312 000240      LOOP                     ;REPLACE THIS INSTRUCTION BY A
1666                                     ;"JMP R0" (000110) TO GET A
1667                                     ;SCOPE LOOP AND HIT CONTINUE
1668 006314 000405      2$:  BR      0AER0
1669                                     ;
1670 006316 005212      INC      (R2)           ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1671 006320 012713 000103      MOV      #103,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1672 006324 000000      0AER6: HLT                     ;ERROR! ODD ADDRESS ERROR
1673 006326 000240      LOOP                     ;REPLACE THIS INSTRUCTION BY A
1674                                     ;"JMP R0" (000110) TO GET A
1675                                     ;SCOPE LOOP AND HIT CONTINUE
1676 006330 010700      0AER8: SCOPE
1677                                     ;
1678                                     ;*****
1679                                     ;*TEST 57 TEST THAT SWAB ODD ADDRESS CAUSES AN ODD ADDRESS TRAP
1680                                     ;*****
1681 006332 005237 000304      TST57: INC      #0$TESTN
1682 006336 012706 021774      MOV      #RUFF,#6
1683 006342 012737 006366 000004      MOV      #0AER7,4
1684 006350 000337 021415      SWAB     TEMP+1          ;DO SWAB USING AN ODD ADDRESS
1685                                     ;
1686 006354 005212      INC      (R2)           ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1687 006356 012713 000104      MOV      #104,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1688 006362 000000      HLT                      ;ERROR! FAILED TO TRAP
1689 006364 000240      LOOP                     ;REPLACE THIS INSTRUCTION BY A
1690                                     ;"JMP R0" (000110) TO GET A
1691                                     ;SCOPE LOOP AND HIT CONTINUE
1692 006366      0AER7:
1693                                     ;
1694 006366 012737 000006 000004      MOV      #ERRVEC+2,ERRVEC;RESTORE ODD ADDRESS ERROR TO
1695 006374 005037 000006      CLR      ERVVEC+2        ;HALT AT ERRVEC+2
1696 006400 010700      SCOPE
1697
```

```
1698
1699
1700                                     ;*****
1701                                     ;*TEST 60 TEST THAT SETTING THE 'T' BIT WILL CAUSE A TRAP TO 14
1702                                     ;*****
1703 006402 005237 000304      TST60: INC      #0$TESTN
1704 006406 012706 021774      MOV      #RUFF,#6
1705 006412 005001      CLR      K1
1706 006414 012737 006454 000014      MOV      #RETAT,TRTVEC ;SET UP TO TRAP TO 14
1707 006422 005037 000016      CLR      TRTVFC+2
1708 006426 012746 000020      MOV      #20,-(6)        ;PUSH 'T' BIT ON THE STACK
1709 006432 012746 006440      MOV      #,+6,-(6)       ;PUSH PC ON THE STACK
1710 006436 000002      RTI                      ;SET 'T' BIT
1711 006440 005201      INC      R1              ;TRAP HERE
1712                                     ;
1713 006442 005212      INC      (R2)           ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1714 006444 012713 000105      MOV      #105,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1715 006450 000000      HLT                      ;TRACE HIT DOES NOT TRAP
1716 006452 000240      LOOP                     ;REPLACE THIS INSTRUCTION BY A
1717                                     ;"JMP R0" (000110) TO GET A
1718 006454 005701      RETAT: TST      R1        ;SCOPE LOOP AND HIT CONTINUE
1719 006456 001405      BEQ                      ;DID T BIT TRAP RIGHT AFTER RTI?
1720                                     ;YES
1721 006460 005212      INC      (R2)           ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1722 006462 012713 000106      MOV      #106,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1723 006466 000000      HLT                      ;T BIT TRAP DID NOT OCCUR
1724                                     ;RIGHT AFTER RTI (ONE INSTR
1725                                     ;WAS ALLOWED AFTER RTI)
1726 006470 000240      LOOP                     ;REPLACE THIS INSTRUCTION BY A
1727                                     ;"JMP R0" (000110) TO GET A
1728 006472 010700      1$:  SCOPE
1729                                     ;
1730                                     ;*****
1731                                     ;*TEST 61 TEST STACK POINTER DECREMENTS ON A T BIT TRAP
1732                                     ;*****
1733 006474 005237 000304      TST61: INC      #0$TESTN
1734 006500 012706 021774      MOV      #BUFF,#6
1735 006504 012737 006524 000014      MOV      #RETBT,TRTVEC
1736 006512 012746 000020      MOV      #20,-(6)        ;PUSH 'T' BIT ON THE STACK
1737 006516 012746 006524      MOV      #,+6,-(6)       ;PUSH PC ON THE STACK
1738 006522 000002      RTI                      ;SET 'T' BIT
1739 006524 020627 021770      RETBT: CMP      #6,#BUFF-4
1740 006530 001405      BEQ      1$
1741                                     ;
1742 006532 005212      INC      (R2)           ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1743 006534 012713 000107      MOV      #107,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1744 006540 000000      HLT                      ;STACK POINTER WAS NOT PUSHED TWICE BY T BIT TRAP
1745 006542 000240      LOOP                     ;REPLACE THIS INSTRUCTION BY A
1746                                     ;"JMP R0" (000110) TO GET A
1747                                     ;SCOPE LOOP AND HIT CONTINUE
1748 006544 010700      1$:  SCOPE
1749                                     ;
1750                                     ;*****
1751                                     ;*TEST 62 TEST FOR PROPER PC ON STACK ON A T BIT TRAP
1752                                     ;*****
1753 006546 005237 000304      TST62: INC      #0$TESTN
1754 006552 012706 021774      MOV      #BUFF,#6
```

T62 TEST FOR PROPER PC ON STACK ON A T BIT TRAP

```
1754 006556 012737 006610 000014      MOV    #RETCT,THTVEC
1755 006564 012746 000020      MOV    #20,-(6)          ;PUSH 'T' BIT ON THE STACK
1756 006570 012746 006576      MOV    #,+6,-(6)        ;PUSH PC ON THE STACK
1757 006574 000002      RTI          ;SET 'T' BIT
1758 006576
1759
1760 006576 005212      INC    (R2)             ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1761 006600 012713 000110      MOV    #110,(R3)       ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1762 006604 000000      HLT
1763
1764 006606 000710      LOOP
1765
1766
1767 006610 022737 006576 021770 RETCT:  CMP    #RETCT3,BUFF-4
1768 006616 001405      HFU    15
1769
1770
1771 006620 005212      INC    (R2)             ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1772 006622 012713 000111      MOV    #111,(R3)       ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1773 006626 000000      HLT
1774
1775
1776
1777 006632 022737 000020 021772 18:  CMP    #20,BUFF-2
1778 006640 001405      HFU    26
1779
1780
1781 006642 005212      INC    (R2)             ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1782 006644 012713 000112      MOV    #112,(R3)       ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1783 006650 000000      HLT
1784
1785
1786
1787 006654 010700      LOOP
1788
1789
1790
1791
1792
1793
1794 006656 005237 000104      TST63: INC    #STESTN
1795 006662 012700 021774      MOV    #BUFF,PC        ;INITIALIZE THE STACK POINTER
1796 006666 012737 000724 000014      MOV    #RETCT1,IHTVEC ;SET UP 'T' BIT TRAP VECTOR
1797 006674 005001      CLP    R1
1798 006676 012746 000020      MOV    #20,-(6)        ;PUSH 'T' BIT ON THE STACK
1799 006702 012746 000710      MOV    #,+6,-(6)        ;PUSH PC ON THE STACK
1800 006706 000006      RTT
1801 006710 005201      INC    R1                ;RETURN FROM INTERRUPT
1802
1803
1804 006712 005212      INC    (R2)             ;'T' BIT TRAP SHOULD OCCUR HERE
1805 006714 012713 000113      MOV    #113,(R3)       RETCT2:
1806 006720 000000      HLT
1807 006722 000720      LOOP
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819 006746 005731      18:  TST    R1
1820
1821 006750 001005      HNE    28
1822
1823 006752 005212      INC    (R2)             ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1824 006754 012713 000115      MOV    #115,(R3)       ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1825 006760 000000      HLT
1826
1827
1828 006762 000720      LOOP
1829
1830
1831
1832
1833
1834 006764 012737 000016 000014      MOV    #16,THTVEC      ;RESTORE 'T' TRAP
1835 006772 005037 000016      CLP    THTVEC+2        ;TO HALT AT 16
1836 006776 010700      SCOPE
```

T63 TEST FOR PROPER PC ON THE STACK, ON A T BIT TRAP (RTT)

```
1810 006724 022737 006712 021770 RETCT1:  CMP    #RETCT2,BUFF-4
1811 006732 001405      HFU    18
1812
1813 006734 005212      INC    (R2)             ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1814 006736 012713 000114      MOV    #114,(R3)       ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1815 006742 000000      HLT
1816 006744 000720      LOOP
1817
1818
1819 006746 005731      18:  TST    R1
1820
1821 006750 001005      HNE    28
1822
1823 006752 005212      INC    (R2)             ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
1824 006754 012713 000115      MOV    #115,(R3)       ;MOVE TO MAIL BOX ($FATAL) THE ERROR
1825 006760 000000      HLT
1826
1827
1828 006762 000720      LOOP
1829
1830
1831
1832
1833
1834 006764 012737 000016 000014      MOV    #16,THTVEC      ;RESTORE 'T' TRAP
1835 006772 005037 000016      CLP    THTVEC+2        ;TO HALT AT 16
1836 006776 010700      SCOPE
```

```
1837  
1838 ;*****  
1839 ;*TEST 64 TEST NO STACK OVERFLOW TRAP ON A DATI TO STACK LOC. LESS THAN 400  
1840 ;*****  
1841 #07007 005237 000304 TST64: INC #STESTN  
1842 #07004 012736 000376 MOV #376,#6 ;SET STACK POINTER LESS THAN 400  
1843 #07012 012737 000334 #00004 MOV #TDEC1,ERRVEC ;LOAD TRAP VECTOR  
1844 #07016 005037 000006 CLR ERRVEC+2  
1845 #07022 005716 TST (6) ;DATI FROM LOC 376  
1846 #07024 000405 RP TDFC1A  
1847  
1848 #07026 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1849 #07030 012713 #00116 MOV #116,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1850 #07034 000000 TDEC1: HLT ;ERROR! STACK OVERFLOW TRAP OCCURED  
1851 ;ON A DATI TO LOC LESS THAN 400  
1852 #07036 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
1853 ;"JMP #R0" (000110) TO GET A  
1854 ;SCOPE LOOP AND HIT CONTINUE  
1855 #07040 010700 TDFC1A: SCOPE  
1856 ;*****  
1857 ;*TEST 65 TEST STACK OVERFLOW TRAP ON A DATIP/DATO TO A STACK LOC. LESS THAN 400  
1858 ;*****  
1859  
1860 #07042 005237 000304 TST65: INC #STESTN  
1861 #07046 012706 000376 MOV #376,#6  
1862 #07052 012737 000376 #00004 MOV #TDEC7,ERRVEC ;DATIP/DATO TO BE 376  
1863 #07060 005016 CLR (6)  
1864 #07062 000240 NOP  
1865  
1866 #07064 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1867 #07066 012713 #00117 MOV #117,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1868 #07072 000000 HLT ;ERROR! NO STACK OVERFLOW TRAP (YELLOW)  
1869 #07074 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
1870 ;"JMP #R0" (000110) TO GET A  
1871 ;SCOPE LOOP AND HIT CONTINUE  
1872 #07076 010700 TDEC2: SCOPE  
1873 ;*****  
1874 ;*TEST 66 TEST THAT A DATIP/DATOR CAUSES AN OVERFLOW TRAP  
1875 ;*****  
1876  
1877 #07100 005237 000304 TST66: INC #STESTN  
1878 #07104 012706 000376 MOV #376,#6  
1879 #07110 012737 000134 #00004 MOV #TDEC2,ERRVEC  
1880 #07116 152716 177777 R1SB #-1,(6)  
1881  
1882 #07122 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1883 #07124 012713 000120 MOV #120,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1884 #07130 000000 HLT ;ERROR! NO STACK OVERFLOW TRAP (YELLOW ZONE)  
1885 #07132 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
1886 ;"JMP #R0" (000110) TO GET A  
1887 ;SCOPE LOOP AND HIT CONTINUE  
1888 #07134 010700 TDEC2A: SCOPE  
1889 ;*****  
1890 ;*TEST 67 TEST NO STACK OVERFLOW TRAP ON A DATI (BYTE) TO STACK LOC. LESS THAN 4  
1891 ;*****  
1892 ;*****
```

```
1893 #07136 005237 000304 TST67: INC #STESTN  
1894 #07142 012705 001400 MOV #1000,#5  
1895 #07146 012706 000376 MOV #376,#6  
1896 #07152 012737 000172 #00004 MOV #TDEC3,ERRVEC  
1897 #07160 124645 CNPM -(6),-(5)  
1898 #07162 000405 HP TDEC6  
1899  
1900 #07164 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1901 #07166 012713 000121 MOV #121,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1902 #07172 000000 TDEC3: HLT ;ERROR! STACK OVERFLOW TRAP OCCURED  
1903 ;ON A DATI TO STACK LOC UNDER 400  
1904 #07174 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
1905 ;"JMP #R0" (000110) TO GET A  
1906 ;SCOPE LOOP AND HIT CONTINUE  
1907 #07176 010700 TDEC6: SCOPE  
1908  
1909 #07200 012706 000400 MOV #400,#6  
1910 #07204 012737 000224 #00004 MOV #TDEC4,ERRVEC  
1911 #07212 134546 H1R -(5),-(6)  
1912 #07214 000405 RP TDEC6A  
1913  
1914 #07216 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1915 #07220 012713 000122 MOV #122,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1916 #07224 000000 TDEC4: HLT ;ERROR! STACK OVERFLOW TRAP OCCURED  
1917 ;ON A DATI TO STACK LOC UNDER 400  
1918 #07226 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
1919 ;"JMP #R0" (000110) TO GET A  
1920 ;SCOPE LOOP AND HIT CONTINUE  
1921 #07230 010700 TDEC6A: SCOPE  
1922 ;*****  
1923 ;*TEST 70 TEST THAT OVERFLOW TRAP DOES NOT LOSE INFORMATION (OLD PC & PS)  
1924 ;*****  
1925  
1926 #07232 005237 000304 TST70: INC #STESTN  
1927 #07236 012706 000400 MOV #400,#6  
1928 #07242 005037 000376 CLR 376 ;STATUS WORD OF LOC 10  
1929 #07246 005037 000374 CLR 374  
1930 #07252 012737 000304 #00004 MOV #TDEC5,ERRVEC ;RETURN TO LOC 4  
1931 #07260 012737 000017 177776 MOV #17,CC ;PRE SET STATUS  
1932 #07266 005246 INC -(6)  
1933  
1934 #07270 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1935 #07272 012713 000123 MOV #123,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1936 #07276 000000 TDEC5A: HLT ;ERROR! NO STACK OVERFLOW TRAP (YELLOW)  
1937 #07300 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
1938 ;"JMP #R0" (000110) TO GET A  
1939 ;SCOPE LOOP AND HIT CONTINUE  
1940 #07302 000415 BP TDEC5B  
1941 #07304 022737 000001 000376 TDEC5: CMP #1,376 ;GO TO SCOPE  
1942 #07312 001405 BEQ #8 ;WAS INC -(6) EXECUTED  
1943  
1944 #07314 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1945 #07316 012713 000124 MOV #124,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1946 #07322 000000 HLT ;ERROR! INSTRUCTION CAUSING YELLOW  
1947 ;ZONE VIOLATION WAS ABORTED. IT  
1948 ;SHOULD HAVE BEEN EXECUTED BEFORE
```

```
1949  
1950 007324 000240 LOOP ;TRAPPING  
1951 ;REPLACE THIS INSTRUCTION BY A  
1952 ;"JMP 000" (000110) TO GET A  
1953 007326 022737 000001 000374 10: CMP #1,374 ;SCOPE LOOP AND HIT CONTINUE  
1954 ;WAS STATUS SAVED NOTE: INC DOES NOT  
1955 007334 001400 HEO TDEC5R ;AFFECT 'C' BIT IN STATUS.  
1956 007336 010700 TDEC5R: SCOPE  
1957  
1958  
1959 ;*****  
1960 ;*TEST 71 TEST THAT A RESERVED INST CAUSES AN OVERFLOW TRAP  
1961 ;*****  
1962 007340 005237 000304 TST71: INC #0$TESTN  
1963 007344 012706 000400 MOV #400,06 ;SET UP STACK TO OVERFLOW  
1964 007350 012737 007404 000010 MOV #VDEC2,10 ;SET UP INST VECTOR  
1965 007356 012737 007410 000004 MOV #VDEC1,4 ;SET UP OVERFLOW VECTOR  
1966 007364 075040 75040 ;THIS TRAP SHOULD CAUSE OVERFLOW  
1967  
1968 007366 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1969 007370 012713 000125 MOV #125,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1970 007374 000000 HLT ;NO TRAP OCCURRED  
1971  
1972 007376 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1973 007400 012713 000126 MOV #126,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1974 007404 000000 VDEC2: HLT ;TRAP FLAG OVERFLOW DID NOT OCCUR  
1975 007406 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
1976 ;"JMP 000" (000110) TO GET A  
1977 ;SCOPE LOOP AND HIT CONTINUE  
1978 007410 010700 VDEC1: SCOPE ;NORMAL OVERFLOW RETURN  
1979  
1980 ;*****  
1981 ;*TEST 72 TEST THAT AN "IOT" CAUSES AN OVERFLOW TRAP  
1982 ;*****  
1983 007412 005237 000304 TST72: INC #0$TESTN  
1984 007416 012706 000400 MOV #400,06 ;SET UP STACK TO OVERFLOW  
1985 007422 012737 007456 000020 MOV #VDEC4,20 ;SET UP INST VECTOR  
1986 007430 012737 007462 000004 MOV #VDEC3,4 ;SET UP OVERFLOW VECTOR  
1987 007436 000004 IOT ;THIS TRAP SHOULD CAUSE OVERFLOW  
1988  
1989 007440 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1990 007442 012713 000127 MOV #127,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1991 007446 000000 HLT ;NO TRAP OCCURRED  
1992  
1993 007450 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
1994 007452 012713 000130 MOV #130,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
1995 007456 000000 VDEC4: HLT ;TRAP FLAG OVERFLOW DID NOT OCCUR  
1996 007460 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
1997 ;"JMP 000" (000110) TO GET A  
1998 ;SCOPE LOOP AND HIT CONTINUE  
1999 007462 010700 VDEC3: SCOPE ;NORMAL OVERFLOW RETURN  
2000  
2001 ;*****  
2002 ;*TEST 73 TEST THAT AN "EMT" CAUSES AN OVERFLOW TRAP  
2003 ;*****  
2004 007464 005237 000304 TST73: INC #0$TESTN  
2005 007470 012706 000400 MOV #400,06 ;SET UP STACK TO OVERFLOW
```

```
2005 007474 012737 007530 000030 MOV #VDEC6,10 ;SET UP INST VECTOR  
2006 007502 012737 007534 000004 MOV #VDEC5,4 ;SET UP OVERFLOW VECTOR  
2007 007510 104000 EMT ;THIS TRAP SHOULD CAUSE OVERFLOW  
2008  
2009 007512 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
2010 007514 012713 000131 MOV #131,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
2011 007520 000000 HLT ;NO TRAP OCCURRED  
2012  
2013 007522 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
2014 007524 012713 000132 MOV #132,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
2015 007530 000000 VDEC6: HLT ;TRAP FLAG OVERFLOW DID NOT OCCUR  
2016 007532 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
2017 ;"JMP 000" (000110) TO GET A  
2018 ;SCOPE LOOP AND HIT CONTINUE  
2019 007534 010700 VDEC5: SCOPE ;NORMAL OVERFLOW RETURN  
2020  
2021 ;*****  
2022 ;*TEST 74 TEST THAT AN "TRAP" CAUSES AN OVERFLOW TRAP  
2023 ;*****  
2024 007536 005237 000304 TST74: INC #0$TESTN  
2025 007542 012706 000400 MOV #400,06 ;SET UP STACK TO OVERFLOW  
2026 007546 012737 007602 000034 MOV #VDEC8,34 ;SET UP INST VECTOR  
2027 007554 012737 007606 000004 MOV #VDEC7,4 ;SET UP OVERFLOW VECTOR  
2028 007562 104400 TRAP ;THIS TRAP SHOULD CAUSE OVERFLOW  
2029  
2030 007564 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
2031 007566 012713 000133 MOV #133,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
2032 007572 000000 HLT ;NO TRAP OCCURRED  
2033  
2034 007574 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
2035 007576 012713 000134 MOV #134,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
2036 007602 000000 VDEC8: HLT ;TRAP FLAG OVERFLOW DID NOT OCCUR  
2037 007604 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
2038 ;"JMP 000" (000110) TO GET A  
2039 ;SCOPE LOOP AND HIT CONTINUE  
2040 007606 010700 VDEC7: SCOPE ;NORMAL OVERFLOW RETURN  
2041  
2042 ;*****  
2043 ;*TEST 75 TEST THAT AN BPT CAUSES AN OVERFLOW TRAP  
2044 ;*****  
2045 007610 005237 000304 TST75: INC #0$TESTN  
2046 007614 012706 000400 MOV #400,06 ;SET UP STACK TO OVERFLOW  
2047 007620 012737 007654 000014 MOV #VDEC10,14 ;SET UP INST VECTOR  
2048 007626 012737 007660 000004 MOV #VDEC9,4 ;SET UP OVERFLOW VECTOR  
2049 007634 000003 BPT ;THIS TRAP SHOULD CAUSE OVERFLOW  
2050  
2051 007636 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
2052 007640 012713 000135 MOV #135,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
2053 007644 000000 HLT ;NO TRAP OCCURRED  
2054  
2055 007646 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR  
2056 007650 012713 000136 MOV #136,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR  
2057 007654 000000 VDEC10: HLT ;TRAP FLAG OVERFLOW DID NOT OCCUR  
2058 007656 000240 LOOP ;REPLACE THIS INSTRUCTION BY A  
2059 ;"JMP 000" (000110) TO GET A  
2060 ;SCOPE LOOP AND HIT CONTINUE
```

```

2061 007660 010700          VDEC9: SCOPE                ;NORMAL OVERFLOW RETURN
2062
2063 ;*****
2064 ;*TEST 76 TEST THAT AN ILLEGAL INSTR. CAUSES AN OVERFLOW TRAP
2065 ;*****
2066 007662 005237 000304      TST76: INC 005TESTN
2067 007666 012706 000400      MOV 0400,%6 ;SET UP STACK TO OVERFLOW
2068 007672 012737 007726 000004  MOV 0VDEC12,4 ;SET UP INST VECTOR
2069 007700 012737 007732 000004  MOV 0VDEC11,4 ;SET UP OVERFLOW VECTOR
2070 007706 004700          ILLA ;THIS TRAP SHOULD CAUSE OVERFLOW
2071
2072 007710 005212          INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2073 007712 012713 000137      MOV 0137,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2074 007716 000000          HLT ;NO TRAP OCCURRED
2075
2076 007720 005212          INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2077 007722 012713 000140      MOV 0140,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2078 007726 000000          HLT ;TRAP FLAG OVERFLOW DID NOT OCCUR
2079 007730 000240          LOOP ;REPLACE THIS INSTRUCTION BY A
2080 ;"JMP 0R0" (000110) TO GET A
2081 ;SCOPE LOOP AND HIT CONTINUE
2082 007732 010700          VDEC11: SCOPE                ;NORMAL OVERFLOW RETURN
2083 007734 020627 000370      CMP %6,0370 ;STACY PUSHED FOUR WORDS?
2084 007740 001405          BEQ 18
2085
2086 007742 005212          INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2087 007744 012713 000141      MOV 0141,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2088 007750 000000          HLT ;TRAP OVERFLOW DID NOT OCCUR
2089 007752 000240          LOOP ;REPLACE THIS INSTRUCTION BY A
2090 ;"JMP 0R0" (000110) TO GET A
2091 ;SCOPE LOOP AND HIT CONTINUE
2092 007754 010700          18: SCOPE
2093
2094 ;*****
2095 ;*TEST 77 TEST THAT AN ILLEGAL INSTR. CAUSES AN OVERFLOW TRAP
2096 ;*****
2097 007756 005237 000304      TST77: INC 005TESTN
2098 007762 012706 000400      MOV 0400,%6 ;SET UP STACK TO OVERFLOW
2099 007766 012737 010022 000004  MOV 0VDEC14,4 ;SET UP INST VECTOR
2100 007774 012737 010026 000004  MOV 0VDEC13,4 ;SET UP OVERFLOW VECTOR
2101 010002 000100          ILLR ;THIS TRAP SHOULD CAUSE OVERFLOW
2102
2103 010004 005212          INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2104 010006 012713 000142      MOV 0142,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2105 010012 000000          HLT ;NO TRAP OCCURRED
2106
2107 010014 005212          INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2108 010016 012713 000143      MOV 0143,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2109 010022 000000          HLT ;TRAP FLAG OVERFLOW DID NOT OCCUR
2110 010024 000240          LOOP ;REPLACE THIS INSTRUCTION BY A
2111 ;"JMP 0R0" (000110) TO GET A
2112 ;SCOPE LOOP AND HIT CONTINUE
2113 010026 010700          VDEC13: SCOPE                ;NORMAL OVERFLOW RETURN
2114
2115 ;INSTRUCTION EQUATE STATEMENTS
2116

```

```

2117 ; 4510 =JSR 5,(0)
2118 ; 005046 =CLR -(6)
2119 ; 010046 =MOV %0,-(6)
2120 ; 006746 =SXT -(6)
2121 ; 074046 =XOP %0,-(6)
2122
2123 ;*****
2124 ;*TEST 100 TEST THAT THE INSTRUCTION (4510) CAUSES STACK OVERFLOW CONDITION
2125 ;*****
2126 010030 005237 000304      TST100: INC 005TESTN
2127 010034 012706 000400      MOV 0400,%6 ;SET STACK POINTER
2128 010040 005000          CLR %0 ;PRE SET R0
2129 010042 012737 010064 000004  MOV 0VDEC15,4 ;LOAD ERROR VECTOR
2130 010050 004510          4510 ;CAUSE OVERFLOW
2131
2132 010052 005212          INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2133 010054 012713 000144      MOV 0144,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2134 010060 000000          HLT ;ERROR! OVERFLOW FAILED TO TRAP
2135 010062 000740          LOOP ;REPLACE THIS INSTRUCTION BY 15
2136 ;"JMP 0R0" (000110) TO GET 15
2137 ;SCOPE LOOP AND HIT CONTINUE
2138 010064 022706 000372      VDEC15: CMP 0372,%6 ;HAS STACK POINTER MOVED BY 6
2139 010070 001405          BFC 18 ;(2 FOR THE AUTO DECREMENT + 4 FOR
2140
2141 010072 005212          INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2142 010074 012713 000145      MOV 0145,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2143 010100 000000          HLT ;THE ERROR TRAP)
2144 ;R6 DID NOT DECREMENT BY 6 WHEN
2145 ;AN INSTRUCTION DOING AUTO-DECREMENT
2146 ;OF R6 CAUSED YELLOW ZONE VIOLATION
2147 010102 000240          LOOP ;REPLACE THIS INSTRUCTION BY 15
2148 ;"JMP 0R0" (000110) TO GET 15
2149 ;SCOPE LOOP AND HIT CONTINUE
2150 010104 010700          18: SCOPE
2151
2152 ;*****
2153 ;*TEST 101 TEST THAT THE INSTRUCTION (005046) CAUSES STACK OVERFLOW CONDITION
2154 ;*****
2155 010106 005237 000304      TST101: INC 005TESTN
2156 010112 012706 000400      MOV 0400,%6 ;SET STACK POINTER
2157 010116 005000          CLR %0 ;PRE SET R0
2158 010120 012737 010142 000004  MOV 0VDEC16,4 ;LOAD ERROR VECTOR
2159 010126 005046          005046 ;CAUSE OVERFLOW
2160
2161 010130 005212          INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2162 010132 012713 000146      MOV 0146,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2163 010136 000000          HLT ;ERROR! OVERFLOW FAILED TO TRAP
2164 010140 000240          LOOP ;REPLACE THIS INSTRUCTION BY 16
2165 ;"JMP 0R0" (000110) TO GET 16
2166 ;SCOPE LOOP AND HIT CONTINUE
2167 010142 022706 000372      VDEC16: CMP 0372,%6 ;HAS STACK POINTER MOVED BY 6
2168 010146 001405          BEQ 18 ;(2 FOR THE AUTO DECREMENT + 4 FOR
2169
2170 010150 005212          INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2171 010152 012713 000147      MOV 0147,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2172 010156 000000          HLT ;THE ERROR TRAP)

```

```
2173 ;R6 DID NOT DECREMENT BY 6 WHEN
2174 ;AN INSTRUCTION DOING AUTO-DECREMENT
2175 ;OF R6 CAUSED YELLOW ZONE VIOLATION
2176 010160 000240 LOOP ;REPLACE THIS INSTRUCTION BY 16
2177 ;"JMP BR0" (000110) TO GET 16
2178 ;SCOPE LOOP AND HIT CONTINUE
2179 010162 010700 18: SCOPE
2180
2181 ;*****
2182 ;*TEST 102 TEST THAT THE INSTRUCTION (010046) CAUSES STACK OVERFLOW CONDITION
2183 ;*****
2184 010164 005237 000300 TST102: INC 005237 ;SET STACK POINTER
2185 010170 012706 000400 MOV 0400,R6 ;PRE SET R0
2186 010174 005000 CLR R0 ;LOAD ERROR VECTOR
2187 010176 012737 010220 000004 MOV 0VDEC17,4 ;CAUSE OVERFLOW
2188 010204 010046
2189
2190 010206 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2191 010210 012713 000150 MOV 0150,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2192 010214 000000 HLT ;ERROR! OVERFLOW FAILED TO TRAP
2193 010216 000240 LOOP ;REPLACE THIS INSTRUCTION BY 17
2194 ;"JMP BR0" (000110) TO GET 17
2195 ;SCOPE LOOP AND HIT CONTINUE
2196 010220 022706 000372 VDEC17: CMP 0372,R6 ;HAS STACK POINTER MOVED BY 6
2197 010224 001405 BEQ 18 ;(2 FOR THE AUTO DECREMENT + 4 FOR
2198
2199 010226 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2200 010230 012713 000151 MOV 0151,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2201 010234 000000 HLT ;THE ERROR TRAP)
2202 ;R6 DID NOT DECREMENT BY 6 WHEN
2203 ;AN INSTRUCTION DOING AUTO-DECREMENT
2204 ;OF R6 CAUSED YELLOW ZONE VIOLATION
2205 010236 000240 LOOP ;REPLACE THIS INSTRUCTION BY 17
2206 ;"JMP BR0" (000110) TO GET 17
2207 ;SCOPE LOOP AND HIT CONTINUE
2208 010240 010700 18: SCOPE
2209
2210 ;*****
2211 ;*TEST 103 TEST THAT THE INSTRUCTION (006746) CAUSES STACK OVERFLOW CONDITION
2212 ;*****
2213 010242 005237 000304 TST103: INC 005237
2214 010246 012736 000400 MOV 0400,R6 ;SET STACK POINTER
2215 010252 005000 CLR R0 ;PRE SET R0
2216 010254 012737 010276 000004 MOV 0VDEC18,4 ;LOAD ERROR VECTOR
2217 010262 006746 006746 ;CAUSE OVERFLOW
2218
2219 010264 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2220 010266 012713 000152 MOV 0152,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2221 010272 000000 HLT ;ERROR! OVERFLOW FAILED TO TRAP
2222 010274 000240 LOOP ;REPLACE THIS INSTRUCTION BY 18
2223 ;"JMP BR0" (000110) TO GET 18
2224 ;SCOPE LOOP AND HIT CONTINUE
2225 010276 022706 000372 VDEC18: CMP 0372,R6 ;HAS STACK POINTER MOVED BY 6
2226 010302 001405 BEQ 18 ;(2 FOR THE AUTO DECREMENT + 4 FOR
2227
2228 010304 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
```

```
2229 010306 012713 000153 MOV 0153,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2230 010312 000000 HLT ;THE ERROR TRAP)
2231 ;R6 DID NOT DECREMENT BY 6 WHEN
2232 ;AN INSTRUCTION DOING AUTO-DECREMENT
2233 ;OF R6 CAUSED YELLOW ZONE VIOLATION
2234 010314 000240 LOOP ;REPLACE THIS INSTRUCTION BY 18
2235 ;"JMP BR0" (000110) TO GET 18
2236 ;SCOPE LOOP AND HIT CONTINUE
2237 010316 010700 18: SCOPE
2238
2239 ;*****
2240 ;*TEST 104 TEST THAT THE INSTRUCTION (074046) CAUSES STACK OVERFLOW CONDITION
2241 ;*****
2242 010320 005237 000304 TST104: INC 005237 ;SET STACK POINTER
2243 010324 012706 000400 MOV 0400,R6 ;PRE SET R0
2244 010330 005000 CLR R0 ;LOAD ERROR VECTOR
2245 010332 012737 010354 000004 MOV 0VDEC19,4 ;CAUSE OVERFLOW
2246 010340 074046 074046
2247
2248 010342 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2249 010344 012713 000154 MOV 0154,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2250 010350 000000 HLT ;ERROR! OVERFLOW FAILED TO TRAP
2251 010352 000240 LOOP ;REPLACE THIS INSTRUCTION BY 19
2252 ;"JMP BR0" (000110) TO GET 19
2253 ;SCOPE LOOP AND HIT CONTINUE
2254 010354 022706 000372 VDEC19: CMP 0372,R6 ;HAS STACK POINTER MOVED BY 6
2255 010360 001405 BEQ 18 ;(2 FOR THE AUTO DECREMENT + 4 FOR
2256
2257 010362 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2258 010364 012713 000155 MOV 0155,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2259 010370 000000 HLT ;THE ERROR TRAP)
2260 ;R6 DID NOT DECREMENT BY 6 WHEN
2261 ;AN INSTRUCTION DOING AUTO-DECREMENT
2262 ;OF R6 CAUSED YELLOW ZONE VIOLATION
2263 010372 000240 LOOP ;REPLACE THIS INSTRUCTION BY 19
2264 ;"JMP BR0" (000110) TO GET 19
2265 ;SCOPE LOOP AND HIT CONTINUE
2266 010374 010700 18: SCOPE
2267
2268
```


T104 TEST THAT THE INSTRUCTION (074046) CAUSES STACK OVERFLOW CONDITION

```
2269
2270
2271 ;*****
2272 ;TEST 105 TEST FOR FALSE OVERFLOW TRAP
2273 ;*****
2273 210376 005237 000104 TST105: INC 005237N
2274 ;PROGRAM MAY HAVE RELOADED IF OVERFLOW FAILS
2275
2276 010402 012737 010500 000104 MOV #FOVFR,4 ;SET UP OVERFLOW POINTER
2277 010410 012736 000402 MOV #402,R6
2278 010414 010605 MOV R6,PS
2279 010416 005746 TST =(6) ;SHOULD NOT OVERFLOW
2280 010420 012736 001002 MOV #1002,R6
2281 010424 010605 MOV R6,PS
2282 010426 005746 TST =(6) ;SHOULD NOT OVERFLOW
2283 010430 012736 002002 MOV #2002,R6
2284 010434 010605 MOV R6,PS
2285 010436 005746 TST =(6) ;SHOULD NOT OVERFLOW
2286 010440 012736 004002 MOV #4002,R6
2287 010444 010605 MOV R6,PS
2288 010446 005746 TST =(6) ;SHOULD NOT OVERFLOW
2289 010450 012736 010002 MOV #10002,R6
2290 010454 010605 MOV R6,PS
2291 010456 005746 TST =(6)
2292 010460 012736 020000 MOV #20000,R6 ;SHOULD NOT OVERFLOW
2293 010464 010605 MOV R6,PS
2294 010466 005746 TST =(6)
2295 010470 000105 RP FOV1
2296
2297 010472 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2298 010474 012713 000156 MOV #156,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2299 010500 000000 FOVFR: HLT ;FALSE STACK OVERFLOW TRAP OCCURRED
2300 ;ON DOING AN AUTO-DECREMENT OF R6
2301 ;THE INITIAL R6 IS IN SAVED IN R5
2302 ;CHECK STACK TO FIND WHERE
2303 010502 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2304 ;"JMP #R0" (000110) TO GET A
2305 ;SCOPE LOOP AND HIT CONTINUE
2306 010504 010700 FOV1: SCOPE
2307
2308 ;*****
2309 ;TEST 106 TEST FOR REPEATED TRAPS AND OVERFLOW OF STACK
2310 ;*****
2311 010506 005237 000104 IST106: INC 005237N
2312 ;WHAT THE STACK LOOKS LIKE WHEN INSTRUCTION AT "04" IS BEING EXECUTED
2313 ; OVFLW PC=05,STAT=0 04 342 BOTTOM OF STACK
2314 ; 4 344
2315 ; TRT PC=04,STAT=4 03A 346
2316 ; 0 350
2317 ; OVFLW PC=05,STAT=0 03 352
2318 ; 3 354
2319 ; IOT PC=03,STAT=3 02A 356
2320 ; 0 360
2321 ; OVFLW PC=05,STAT=0 02 362
2322 ; 2 364
2323 ; TRAP PC=02,STAT=2 01A 366
2324 ; 0 370
```

T106 TEST FOR REPEATED TRAPS AND OVERFLOW OF STACK

```
2325 ; OVFLW PC=05,STAT=0 01 372
2326 ; 1 374
2327 ; EMT PC=01,STAT=1 00A 376
2328 ; 17 400
2329 ; 402 INITIAL STACK POINTER
2330 010512 012706 000402 MOV #402,R6 ;INITIALIZE STACK POINTER
2331 010516 012737 010632 000030 MOV #01,EMTVEC ;NEW PC POINTS TO NEXT INSTR.
2332 010524 012737 000001 000032 MOV #1,EMTVEC+2 ;NEW PSW
2333 010532 012737 010646 000034 MOV #02,TRAPVEC ;NEW PC POINTS TO NEXT INSTR
2334 010540 012737 000002 000036 MOV #2,TRAPVEC+2 ;NEW PSW
2335 010546 012737 010662 000020 MOV #03,IOTVEC ;NEW PC POINTS TO NEXT INSTR
2336 010554 012737 000003 000022 MOV #3,IOTVEC+2 ;NEW PSW
2337 010562 012737 010676 000014 MOV #04,TRIVEC ;NEW PC POINTS TO NEXT INSTR
2338 010570 012737 000004 000016 MOV #4,TRIVEC+2 ;NEW PSW
2339 010576 012737 011324 000004 MOV #05,ERRVEC ;NEW PC POINTS TO NEXT INSTR
2340 010604 005037 000006 CLR FPPVEC+2 ;NEW PSW
2341 010610 012737 000017 177776 OW: EMT ;PRESENT PSW
2342 010616 104000 OW1:
2343 010622
2344
2345 010620 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2346 010622 012713 000157 MOV #157,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2347 010626 000000 OWA: HLT ;EMT NOT EXECUTED
2348 010630 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2349 ;"JMP #R0" (000110) TO GET A
2350 ;SCOPE LOOP AND HIT CONTINUE
2351 010632 104400 O1: TRAP ;RETURN HERE FROM PREVIOUS YELLOW ZONE
2352 ;STACK VIOLATION
2353 010634 O11:
2354
2355 010634 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2356 010636 012713 000160 MOV #160,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2357 010642 000000 O1A: HLT ;TRAP NOT EXECUTED
2358 010644 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2359 ;"JMP #R0" (000110) TO GET A
2360 ;SCOPE LOOP AND HIT CONTINUE
2361 010646 000004 O2: IOT ;RETURN HERE FROM PREVIOUS YELLOW ZONE
2362 ;STACK VIOLATION
2363 010650 O21:
2364
2365 010650 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2366 010652 012713 000161 MOV #161,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2367 010656 000000 O2A: HLT ;IOT NOT EXECUTED
2368 010660 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2369 ;"JMP #R0" (000110) TO GET A
2370 ;SCOPE LOOP AND HIT CONTINUE
2371 010662 000003 O3: BPT ;RETURN HERE FROM PREVIOUS YELLOW ZONE
2372 ;STACK VIOLATION
2373 010664 O31:
2374
2375 010664 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2376 010666 012713 000162 MOV #162,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2377 010672 000000 O3A: HLT ;BPT NOT EXECUTED
2378 010674 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2379 ;"JMP #R0" (000110) TO GET A
2380 ;SCOPE LOOP AND HIT CONTINUE
```

2301	010676	022706	000342	04:	CMP	#342,(R6)	;IS STACK POINTER
2302	010702	001405			BEQ	200	;POSITIONED PROPEKLY
2303							
2304	010704	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2305	010706	012713	000163		MOV	#163,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2306	010712	000000			HLT		;EYHOP1 INCOORRECT STACK POINTER AFTER
2307							;4 REPETITIVE STACK OVERFLOW TRAPS
2308	010714	000210			LOOP		;REPLACE THIS INSTRUCTION BY A
2309							; "JMP 000" (000110) TO GET A
2390							;SCOPE LOOP AND HIT CONTINUE
2391	010716	012720	000402	208:	MOV	#402,(R0)	
2392	010722	022740	000017		CMP	#17,-(R)	;CORRECT OLD PSW SAVED?
2393	010726	001405			REQ	15	
2394							
2395	010730	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2396	010732	012713	000164		MOV	#164,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2397	010736	000000			HLT		;CORRECT PSW WAS NOT SAVED
2398							;ON EMT TRAP
2399	010740	000240			LOOP		;REPLACE THIS INSTRUCTION BY A
2400							; "JMP 000" (000110) TO GET A
2401							;SCOPE LOOP AND HIT CONTINUE
2402	010742	022740	010620	16:	CMP	#001,-(R)	
2403	010746	001405			REQ	28	
2404							
2405	010750	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2406	010752	012713	000165		MOV	#165,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2407	010756	000000			HLT		;CORRECT PC WAS NOT SAVED ON
2408							;FMT TRAP
2409	010760	000240			LOOP		;REPLACE THIS INSTRUCTION BY A
2410							; "JMP 000" (000110) TO GET A
2411							;SCOPE LOOP AND HIT CONTINUE
2412	010762	022740	000001	26:	CMP	#1,-(R)	
2413	010766	001405			REQ	36	
2414							
2415	010770	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2416	010772	012713	000166		MOV	#166,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2417	010776	000000			HLT		;CORRECT PSW WAS NOT SAVED ON STACK
2418							;OVERFLOW TRAP FOLLOWING EMT
2419	011000	000240			LOOP		;REPLACE THIS INSTRUCTION BY A
2420							; "JMP 000" (000110) TO GET A
2421							;SCOPE LOOP AND HIT CONTINUE
2422	011002	022740	010632	36:	CMP	#01,-(R)	
2423	011006	001405			BEQ	48	
2424							
2425	011010	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2426	011012	012713	000167		MOV	#167,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2427	011016	000000			HLT		;CORRECT PC WAS NOT SAVED ON STACK
2428							;OVERFLOW TRAP FOLLOWING EMT
2429	011020	000240			LOOP		;REPLACE THIS INSTRUCTION BY A
2430							; "JMP 000" (000110) TO GET A
2431							;SCOPE LOOP AND HIT CONTINUE
2432	011022	022740	000000	46:	CMP	#0,-(R)	
2433	011026	001405			BEQ	58	
2434							
2435	011030	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2436	011032	012713	000170		MOV	#170,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR

2437	011036	000000			HLT		;CORRECT PSW WAS NOT SAVED ON
2438							; "TRAP" TRAP
2439	011040	000240			LOOP		;REPLACE THIS INSTRUCTION BY A
2440							; "JMP 000" (000110) TO GET A
2441							;SCOPE LOOP AND HIT CONTINUE
2442	011042	022740	010634	56:	CMP	#011,-(R)	
2443	011046	001405			BEQ	68	
2444							
2445	011050	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2446	011052	012713	000171		MOV	#171,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2447	011056	000000			HLT		;CORRECT PC WAS NOT SAVED ON
2448							; "TRAP" TRAP
2449	011060	000240			LOOP		;REPLACE THIS INSTRUCTION BY A
2450							; "JMP 000" (000110) TO GET A
2451							;SCOPE LOOP AND HIT CONTINUE
2452	011062	022740	000002	66:	CMP	#2,-(R)	
2453	011066	001405			BEQ	78	
2454							
2455	011070	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2456	011072	012713	000172		MOV	#172,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2457	011076	000000			HLT		;CORRECT PSW WAS NOT SAVED ON STACK
2458							;OVERFLOW TRAP FOLLOWING "TRAP"
2459	011100	000240			LOOP		;REPLACE THIS INSTRUCTION BY A
2460							; "JMP 000" (000110) TO GET A
2461							;SCOPE LOOP AND HIT CONTINUE
2462	011102	022740	010646	76:	CMP	#02,-(R)	
2463	011106	001405			BEQ	88	
2464							
2465	011110	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2466	011112	012713	000173		MOV	#173,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2467	011116	000000			HLT		;CORRECT PC WAS NOT SAVED ON STACK
2468							;OVERFLOW TRAP FOLLOWING "TRAP"
2469	011120	000240			LOOP		;REPLACE THIS INSTRUCTION BY A
2470							; "JMP 000" (000110) TO GET A
2471							;SCOPE LOOP AND HIT CONTINUE
2472	011122	022740	000000	86:	CMP	#0,-(R)	
2473	011126	001405			BEQ	98	
2474							
2475	011130	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2476	011132	012713	000174		MOV	#174,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2477	011136	000000			HLT		;CORRECT PSW WAS NOT SAVED ON
2478							; "IOT" TRAP
2479	011140	000240			LOOP		;REPLACE THIS INSTRUCTION BY A
2480							; "JMP 000" (000110) TO GET A
2481							;SCOPE LOOP AND HIT CONTINUE
2482	011142	022740	010650	96:	CMP	#021,-(R)	
2483	011146	001405			BEQ	108	
2484							
2485	011150	005212			INC	(R2)	;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2486	011152	012713	000175		MOV	#175,(R3)	;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2487	011156	000000			HLT		;CORRECT PC WAS NOT SAVED ON
2488							; "IOT" TRAP
2489	011160	000240			LOOP		;REPLACE THIS INSTRUCTION BY A
2490							; "JMP 000" (000110) TO GET A
2491							;SCOPE LOOP AND HIT CONTINUE
2492	011162	022740	000003	106:	CMP	#3,-(R)	

```

2493 011166 001405          RFQ      118
2494
2495 011170 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2496 011172 012713 000176    MOV      #176,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2497 011176 000000          HLT
2498                                ;CORRECT PSW WAS NOT SAVED ON STACK
2499 011200 000240          LOOP
2500                                ;OVERFLOW FOLLOWING IOT
2501                                ;REPLACE THIS INSTRUCTION BY A
2502                                ;"JMP #R0" (000110) TO GET A
2503                                ;SCOPE LOOP AND HIT CONTINUE
2502 011202 022740 010662    118:    CMP      #03,-(0)
2503 011206 001405          BFQ      128
2504
2505 011210 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2506 011212 012713 000177    MOV      #177,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2507 011216 000000          HLT
2508                                ;CORRECT PC WAS NOT SAVED ON STACK
2509                                ;OVERFLOW FOLLOWING IOT
2509 011220 000240          LOOP
2510                                ;REPLACE THIS INSTRUCTION BY A
2511                                ;"JMP #R0" (000110) TO GET A
2512                                ;SCOPE LOOP AND HIT CONTINUE
2512 011222 022740 000000    128:    CMP      #0,-(0)
2513 011226 001405          BEQ      138
2514
2515 011230 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2516 011232 012713 000200    MOV      #200,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2517 011236 000000          HLT
2518                                ;CORRECT PSW WAS NOT SAVED
2519                                ;ON BPT TRAP
2519 011240 000240          LOOP
2520                                ;REPLACE THIS INSTRUCTION BY A
2521                                ;"JMP #R0" (000110) TO GET A
2522                                ;SCOPE LOOP AND HIT CONTINUE
2522 011242 022740 010664    138:    CMP      #031,-(0)
2523 011246 001405          BEQ      148
2524
2525 011250 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2526 011252 012713 000201    MOV      #201,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2527 011256 000000          HLT
2528                                ;CORRECT PC WAS NOT SAVED
2529                                ;ON BPT TRAP
2529 011260 000240          LOOP
2530                                ;REPLACE THIS INSTRUCTION BY A
2531                                ;"JMP #R0" (000110) TO GET A
2532                                ;SCOPE LOOP AND HIT CONTINUE
2532 011262 022740 000004    148:    CMP      #4,-(0)
2533 011266 001405          BFQ      158
2534
2535 011270 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2536 011272 012713 000202    MOV      #202,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2537 011276 000000          HLT
2538                                ;CORRECT PSW WAS NOT SAVED ON STACK
2539                                ;OVERFLOW FOLLOWING BPT
2539 011300 000240          LOOP
2540                                ;REPLACE THIS INSTRUCTION BY A
2541                                ;"JMP #R0" (000110) TO GET A
2542                                ;SCOPE LOOP AND HIT CONTINUE
2542 011302 022740 010676    158:    CMP      #04,-(0)
2543 011306 001405          BEQ      07
2544
2545 011310 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2546 011312 012713 000203    MOV      #203,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2547 011316 000000          STAERR: HLT
2548                                ;CORRECT PC WAS NOT SAVED ON STACK
2549                                ;OVERFLOW FOLLOWING BPT
  
```

```

2549
2550 011320 000240          LOOP
2551                                ;R0 GIVES LOCATION OF STACK ERROR
2552                                ;REPLACE THIS INSTRUCTION BY A
2553                                ;"JMP #R0" (000110) TO GET A
2554                                ;SCOPE LOOP AND HIT CONTINUE
2553 011322 000402          07:    BP      06
2554 011324 000176 000000    05:    JMP      #6
2555 011330 010700          06:    SCOPE
2556                                ;RETURN FROM STACK
2557                                ;OVERFLOW TRAP
2558
2559                                ;*****
2560                                ;TEST 107 TEST TRACE TRAP WHEN STACK IS IN YELLOW AREA
2561                                ;*****
2560 011332 005237 000304    TST107: INC      #0;TESTIN
2561 011336 005037 021414    CLR      TEMP
2562 011342 012706 000400    MOV      #400,#6
2563 011346 012737 011532 000014    MOV      #TRTC,TRTVEC ;SET UP STACK POINTER
2564 011354 005037 000016    CLP      TRTVEC+2
2565 011360 012737 011416 000004    MOV      #TRTB,ERRVEC ;LOAD TRT VECTOR
2566 011366 005037 000006    CLR      EPRVEC+2
2567 011372 013737 011410 021416    TRTA,TEMP+2
2568 011400 012737 000003 011410    MOV      #RPT,TRTA ;LOAD OVFLW. ERROR VECTOR
2569 011406 000277 021414    SCC
2570 011410 005137 000006    COM
2571 011414 000535 000017 000376 TRTA:  TRTAA: BR      TRTF
2572 011416 022737 000017 000376 TRTB:  CMP      #17,376
2573 011424 001405          RFQ      18
2574
2575 011426 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2576 011430 012713 000204    MOV      #204,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2577 011434 000000          HLT
2578                                ;ERROR! CONDITION CODFS NOT SAVED
2579                                ;REPLACE THIS INSTRUCTION BY A
2580                                ;"JMP #R0" (000110) TO GET A
2581                                ;SCOPE LOOP AND HIT CONTINUE
2581 011440 022737 011412 000374 18:    CMP      #TRTA+2,374
2582 011446 001406          BEQ      28
2583
2584 011450 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2585 011452 012713 000205    MOV      #205,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2586 011456 000000          HLT
2587 011460 000240          LOOP
2588                                ;ERROR! RETURN PC NOT SAVED
2589                                ;REPLACE THIS INSTRUCTION BY A
2590                                ;"JMP #R0" (000110) TO GET A
2591                                ;SCOPE LOOP AND HIT CONTINUE
2590 011462 000523          BR      TRTEX
2591 011464 005737 000372 28:    TST     372
2592 011470 001405          BEQ      38
2593
2594 011472 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2595 011474 012713 000206    MOV      #206,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2596 011500 000000          HLT
2597 011502 000240          LOOP
2598                                ;ERROR! TRT STATUS (TRTVEC+2) NOT SAVED
2599                                ;REPLACE THIS INSTRUCTION BY A
2600                                ;"JMP #R0" (000110) TO GET A
2601                                ;SCOPE LOOP AND HIT CONTINUE
2600 011504 022737 011532 000370 38:    CMP      #TRTTC,370
2601 011512 001406          RFQ      48
2602
2603 011514 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2604 011516 012713 000207    MOV      #207,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
  
```

2605 011522 000000 HLT ;ERROR! THE VECTOR (TRTVFC) NOT SAVED
2606 011524 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2607 ;"JMP BR0" (000110) TO GET A
2608 ;SCOPE LOOP AND HIT CONTINUE
2609 011526 000000 BR TRTEX ;EXIT TEST
2610 011530 000000 48: RTI ;RETURN FROM OVERFLOW TRAP
2611 ;
2612 011532 010005 IPTC: MOV #0,45 ;GET STACK POINTER
2613 011534 162715 SUB #2,(5) ;POINT PC TO TRAPPED INSTRUCTION
2614 011540 013735 TEMP+2,(5)+ ;RESTORE ORIGINAL INSTRUCTION (COM TEMP)
2615 011544 052715 000000 LHS #2,(5) ;SET "T" BIT ON STACK
2616 011550 012737 011566 000000 MOV #TRTD,ERRVEC ;CHANGE ERROR VECTOR
2617 011556 012737 011702 000014 MOV #TRTE,TRTVFC ;LOAD "T" BIT VECTOR
2618 011564 000000 RTI ;GO DO ORIGINAL INSTRUCTION AND TRAP
2619 ;("T" BIT TRAP) WHEN FINISHED
2620 ;
2621 ;CHECK THE STACK
2622 011566 022737 000031 000376 ;CHECK: CMP #31,376 ;IS "N","C","T" BITS SET ON THE STACK
2623 011574 001405 TRTD: BEQ 15 ;(RESULT OF COM TEMP IN TRTA
2624 ;
2625 011576 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2626 011600 012713 000210 MOV #210,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2627 011604 000000 HLT ;ERROR! STATUS NOT SAVED
2628 011606 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2629 ;"JMP BR0" (000110) TO GET A
2630 ;SCOPE LOOP AND HIT CONTINUE
2631 011610 022737 011414 000374 18: CMP #TRTAA,374 ;WAS RETURN PC FROM "T" TRAP SAVED?
2632 011616 001406 BEQ 25 ;
2633 ;
2634 011620 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2635 011622 012713 000211 MOV #211,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2636 011626 000000 HLT ;ERROR! RETURN PC NOT SAVED
2637 011630 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2638 ;"JMP BR0" (000110) TO GET A
2639 ;SCOPE LOOP AND HIT CONTINUE
2640 ;EXIT TEST
2641 011632 000437 BR TRTEX ;"T" BIT TRAP STATUS SAVED?
2642 011634 005737 28: TST 372 ;
2643 011640 001405 BEQ 35 ;
2644 ;
2645 011642 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2646 011644 012713 000212 MOV #212,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2647 011650 000000 HLT ;ERROR! "T" BIT STATUS (TRTVFC+2) NOT SAVED
2648 011652 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2649 ;"JMP BR0" (000110) TO GET A
2650 ;SCOPE LOOP AND HIT CONTINUE
2651 011654 022737 011702 000370 36: CMP #TRTE,370 ;"T" BIT VECTOR SAVED (TRTVFC)
2652 011662 001406 BEQ 48 ;
2653 ;
2654 011664 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2655 011666 012713 000213 MOV #213,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2656 011672 000000 HLT ;ERROR! "T" BIT VECTOR NOT SAVED
2657 011674 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2658 ;"JMP BR0" (000110) TO GET A
2659 ;SCOPE LOOP AND HIT CONTINUE
2660 011676 000415 BR TRTEX ;EXIT TEST
2661 011700 000000 48: RTI ;RETURN FROM ERROR TRAP

2661 ;
2662 011702 005037 000376 TRTE: CLR 376 ;CLEAR SAVED "T" BIT
2663 011706 000000 RTI ;RETURN FROM "T" BIT TRAP
2664 ;
2665 011710 022737 177777 021414 TRTF: CMP #0,TEMP ;WAS ORIGINAL INSTRUCTION EXECUTED?
2666 011716 001405 TRTEX BEQ 1 ;
2667 ;
2668 011720 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2669 011722 012713 000214 MOV #214,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2670 011726 000000 18: HLT ;TRAPPED INSTRUCTION (COM TEMP) NOT EXEC.
2671 011730 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2672 ;"JMP BR0" (000110) TO GET A
2673 ;SCOPE LOOP AND HIT CONTINUE
2674 011732 012737 000016 000014 IPTIEX: MOV #TRTVFC+2,TRTVEC ;RESTORE TRAP
2675 011740 012737 000006 000004 MOV #0,4 ;VECTORS
2676 011746 010700 SCOPE ;
2677 ;
2678 ;*****
2679 ;TEST 11X TEST YELLOW ZONE STACK VIOLATION, USING INDEX MODE (6) & R6. (R6.GT.4
2680 ;*****
2681 011750 005237 000304 TST110: INC #0,TESTN ;
2682 011754 012706 001000 INDEX: MOV #1000,#6 ;SET UP STACK POINTER
2683 011760 012737 012010 000004 MOV #IND1,4 ;
2684 011766 005037 000006 CLR 6 ;
2685 011772 005066 177340 CLP -40(6) ;FINAL ADDRESS IS 340
2686 ;
2687 011776 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2688 012000 012713 000215 MOV #215,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2689 012004 000000 HLT ;ERROR! STACK OVERFLOW TRAP DID NOT OCCUR
2690 012006 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2691 ;"JMP BR0" (000110) TO GET A
2692 ;SCOPE LOOP AND HIT CONTINUE
2693 012010 010700 IND1: SCOPE ;
2694 ;
2695 ;TEST THAT THE TRAP SEQUENCE IS EXECUTED PROPERLY ON OVERFLOW.
2696 012012 012706 001000 MOV #1000,#6 ;
2697 012016 012737 012064 000004 MOV #IND2,4 ;
2698 012024 005037 000776 CLR 776 ;
2699 012030 005037 000774 CLR 774 ;
2700 012034 012737 000001 000176 MOV #1,376 ;PRE SET DEST ADDRESS
2701 012042 012766 000000 177376 MOV #0,-40(6) ;FINAL ADDRESS IS 376
2702 012050 ;
2703 ;
2704 012050 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2705 012052 012713 000216 MOV #216,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2706 012056 000000 HLT ;ERROR! FAILED TO TRAP
2707 012060 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2708 ;"JMP BR0" (000110) TO GET A
2709 ;SCOPE LOOP AND HIT CONTINUE
2710 012062 000442 BR IND2X ;GO TO SCOPE
2711 012064 023727 000776 000004 IND2: CMP 776,#4 ;STATUS SAVED ON THE STACK (Z BIT SET)
2712 012072 001405 BEQ 18 ;(RESULT OF MOV #0,-40(6))
2713 ;
2714 012074 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
2715 012076 012713 000217 MOV #217,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
2716 012102 000000 HLT ;ERROR! INCORRECT STATUS SAVED

```

2717 012104 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
2718                                     ;"JMP 000" (000110) TO GET A
2719                                     ;SCOPE LOOP AND HIT CONTINUE
2720 012106 022737 012050 000774 18:  CMP      #IND2R,774 ;RETURN PC SAVED?
2721 012114 001405          BEQ      28          ;
2722                                     ;
2723 012116 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2724 012120 012713 000220  MOV      #220,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2725 012124 000000          HLT     ;ERROR! RETURN PC NOT SAVED
2726 012126 000240          LOOP     ;REPLACE THIS INSTRUCTION BY A
2727                                     ;"JMP 000" (000110) TO GET A
2728                                     ;SCOPE LOOP AND HIT CONTINUE
2729 012130 020627 000774 28:  CMP      #6,774    ;CORRECT STACK POINTER?
2730 012134 001405          BEQ      38          ;
2731                                     ;
2732 012136 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2733 012140 012713 000221  MOV      #221,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2734 012144 000000          HLT     ;ERROR! INCORRECT STACK POINTER
2735 012146 000240          LOOP     ;REPLACE THIS INSTRUCTION BY A
2736                                     ;"JMP 000" (000110) TO GET A
2737                                     ;SCOPE LOOP AND HIT CONTINUE
2738 012150 005737 000376 38:  TST      376       ;WAS MOVE INST. EXECUTED
2739 012154 001405          BEQ      1002A     ;
2740                                     ;
2741 012156 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2742 012160 012713 000222  MOV      #222,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2743 012164 000000          HLT     ;ERROR! MOVE NOT EXECUTED
2744 012166 000240          LOOP     ;REPLACE THIS INSTRUCTION BY A
2745                                     ;"JMP 000" (000110) TO GET A
2746                                     ;SCOPE LOOP AND HIT CONTINUE
2747 012170 010700          IND2X:  SCOPE
2748                                     ;
2749                                     ;
2750                                     ;*****
2751                                     ;*TST 111 TEST YELLOW ZONE STACK VIOLATION, USING INDEX MODE (340.LT.R6.LT.400
2752                                     ;*****
2753 012172 005237 000304          TST111: INC      #0$TESTN
2754 012176 012706 000376  MOV      #376,#6   ;SET UP STACK IN "YELLOW" AREA
2755 012202 012737 012236  MOV      #IND3,4   ;SET UP ERROR RETURN
2756 012210 005037 000371  CLP      374
2757 012214 005037 000372  CLR      372
2758 012220 005066 177742          CLR      -36(6)   ;FINAL ADDRESS IS 340 (YELLOW)
2759 012224          IND3B:
2760 012226 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2761 012228 012713 000223  MOV      #223,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2762 012232 000000          HLT     ;FAILED TO TRAP
2763 012234 000240          LOOP     ;REPLACE THIS INSTRUCTION BY A
2764                                     ;"JMP 000" (000110) TO GET A
2765                                     ;SCOPE LOOP AND HIT CONTINUE
2766 012236 022737 000004 000374  IND3:  CMP      #1,374   ;SAVED STATUS CORRECT?
2767 012244 001405          BEQ      18          ;
2768                                     ;
2769 012246 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2770 012250 012713 000224  MOV      #224,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2771 012254 000000          HLT     ;SAVED STATUS WAS INCORRECT (Z SET)
2772 012256 000240          LOOP     ;REPLACE THIS INSTRUCTION BY A

```

```

2773                                     ;"JMP 000" (000110) TO GET A
2774                                     ;SCOPE LOOP AND HIT CONTINUE
2775 012260 022737 012224 000372 18:  CMP      #IND3R,372 ;SAVED RETURN ADDRESS CORRECT?
2776 012266 001405          BEQ      28          ;
2777                                     ;
2778 012270 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2779 012272 012713 000225  MOV      #225,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2780 012276 000000          HLT     ;ERROR! SAVED RETURN ADDRESS INCORRECT
2781 012300 000240          LOOP     ;REPLACE THIS INSTRUCTION BY A
2782                                     ;"JMP 000" (000110) TO GET A
2783                                     ;SCOPE LOOP AND HIT CONTINUE
2784 012302 022706 000372 28:  CMP      #372,#6   ;STACK POINTER CORRECT
2785 012306 001405          BEQ      38          ;
2786                                     ;
2787 012310 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2788 012312 012713 000226  MOV      #226,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2789 012316 000000          HLT     ;ERROR! STACK POINTER INCORRECT
2790 012320 000240          LOOP     ;REPLACE THIS INSTRUCTION BY A
2791                                     ;"JMP 000" (000110) TO GET A
2792                                     ;SCOPE LOOP AND HIT CONTINUE
2793 012322 005737 000340 38:  TST      340       ;WAS CLEAR INSTRUCTION EXECUTED?
2794 012326 001405          BEQ      48          ;
2795                                     ;
2796 012330 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2797 012332 012713 000227  MOV      #227,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2798 012336 000000          HLT     ;ERROR! CLEAR NOT EXECUTED
2799 012340 000240          LOOP     ;REPLACE THIS INSTRUCTION BY A
2800                                     ;"JMP 000" (000110) TO GET A
2801                                     ;SCOPE LOOP AND HIT CONTINUE
2802 012342 010700          48:  SCOPE
2803                                     ;
2804                                     ;
2805                                     ;*****
2806                                     ;*TST 112 TEST RED ZONE STACK VIOLATION & ABORT OF THE VIOLATING INSTRUCTION
2807                                     ;*****
2808 012344 005237 000304          TST112: INC      #0$TESTN
2809 012350 012706 001000  MOV      #1000,#6
2810 012354 012737 012436  MOV      #IND4,4
2811 012362 012737 177777  MOV      #-1,344   ;PRE SET SAVED STATUS LOCATION
2812 012370 012737 177777  MOV      #-1,#
2813 012404 012737 177777  MOV      #-1,2
2814 012410 005037 000776  CLR      776
2815 012414 005037 177776  CLR      CC
2816 012422 005066 177777  MOV      #-1,336   ;PRE SET 336
2817 012426 005066 177336  CLR      -442(6)  ;CLEAR "RED" LOCATION (336)
2818                                     ;
2819                                     ;
2820 012426 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2821 012430 012713 000230  MOV      #230,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2822 012434 000000          HLT     ;ERROR! FAILED TO TRAP
2823 012436 005706 001405  IND4:  TST      #6
2824 012440 001405          BEQ      18          ;STACK POINTER CORRECT?
2825                                     ;
2826 012442 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2827 012444 012713 000231  MOV      #231,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2828 012450 000000          HLT     ;ERROR! INCORRECT STACK POINTER
2829 012452 000240          LOOP     ;REPLACE THIS INSTRUCTION BY A

```

```

2020 ;"JMP BR0" (000110) TO GET A
2030 ;SCOPE LOOP AND HIT CONTINUE
2031 012454 022737 177777 000336 18: CMP #=1,336
2032 012462 001405 BEQ 28 ;WAS CLEAR INSTRUCTION ABORTED?
2033
2034 012464 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2035 012466 012713 000232 MOV #232,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2036 012472 000000 HLT ;ERROR! CLEAR INSTRUCTION NOT ABORTED
2037 012474 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2038 ;"JMP BR0" (000110) TO GET A
2039 ;SCOPE LOOP AND HIT CONTINUE
2040 012476 022737 000004 000002 28: CMP #4,2
2041 012504 001405 BEQ 36 ;WAS PROPER STATUS SAVED? (Z SET)
2042 ;RESULT OF CLR -442(6)
2043 012506 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2044 012510 012713 000233 MOV #233,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2045 012514 000000 HLT ;ERROR! INCORRECT STATUS WAS SAVED
2046 ;THOUGH THE INSTR. (CLR) WAS ABORTED
2047 ;THE Z BIT WAS SET BECAUSE OF CLEAR
2048 ;HENCE, ON THE STACK Z BIT WILL BE SAVED
2049 012516 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2050 ;"JMP BR0" (000110) TO GET A
2051 ;SCOPE LOOP AND HIT CONTINUE
2052 012520 022737 012426 000000 38: CMP #IND4R,0
2053 012526 001405 BEQ 48 ;WAS RETURN PC SAVED?
2054
2055 012530 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2056 012532 012713 000234 MOV #234,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2057 012536 000000 HLT ;ERROR! TRAP DID NOT STACK AT 2 & 0
2058 012540 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2059 ;"JMP BR0" (000110) TO GET A
2060 ;SCOPE LOOP AND HIT CONTINUE
2061 012542 005737 000776 48: TST 776
2062 012546 001405 BEQ 56 ;WAS NOTHING PUT ON THE OLD STACK?
2063
2064 012550 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2065 012552 012713 000235 MOV #235,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2066 012556 000000 HLT ;ERROR! TRAP DID NOT STACK AT 2 & 0
2067 012560 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2068 ;"JMP BR0" (000110) TO GET A
2069 ;SCOPE LOOP AND HIT CONTINUE
2070 012562 010700 56: SCOPE
2071
2072 ;*****
2073 ;*TEST 113 TEST THAT INDEX MODE USING R6 IN SOURCE DOES NOT TRAP
2074 ;*****
2075 012564 005237 000304 TST113: INC #0$TESTN
2076 012570 012706 001000 MOV #1000,06 ;SET UP STACK
2077 012574 012737 012626 000004 MOV #IND5,4 ;SET UP ERROR TRAP
2078 012602 012737 177777 000350 MOV #=-1,340
2079 012610 016637 177350 021414 MOV -430(6),TEMP
2080 012616 000405 BR INDR
2081
2082 012620 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2083 012622 012713 000236 MOV #236,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2084 012626 000000 HLT ;ERROR! TRAPPED
  
```

```

2085 012630 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2086 ;"JMP BR0" (000110) TO GET A
2087 012632 010700 IND8: SCOPE ;SCOPE LOOP AND HIT CONTINUE
2088
2089 ;*****
2090 ;*TEST 114 THIS TEST IS THE SAME AS ABOVE EXCEPT THAT LOCATION IS 'RED'
2091 ;*****
2092 012634 005237 000304 TST114: INC #0$TESTN
2093 012640 012706 001000 MOV #1000,06
2094 012644 012737 012670 000004 MOV #IND6,4
2095 012652 016637 177336 021414 MOV -442(6),TEMP
2096 012660 000405 BR INDR
2097
2098 012662 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2099 012664 012713 000237 MOV #237,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2100 012670 000000 HLT ;ERROR! TRAPPED
2101 012672 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2102 ;"JMP BR0" (000110) TO GET A
2103 ;SCOPE LOOP AND HIT CONTINUE
2104 012674 010700 IND9: SCOPE
2105
2106 ;*****
2107 ;*TEST 115 TEST THAT TRAP DOES NOT OCCUR IF INDEX USING R6 IN DEST IS DEFERRED
2108 ;*****
2109 012676 005237 000304 TST115: INC #0$TESTN
2110 012702 012706 001000 MOV #1000,06
2111 012706 012737 012736 000004 MOV #IND7,4
2112 012714 012737 021414 000350 MOV #TEMP,350
2113 012722 005076 177350 CLR #=-430(6) ;CLEAR TEMP
2114 012726 000405 BR IND11
2115
2116 012730 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2117 012732 012713 000240 MOV #240,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2118 012736 000000 HLT ;ERROR! TRAPPED
2119 012740 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2120 ;"JMP BR0" (000110) TO GET A
2121 ;SCOPE LOOP AND HIT CONTINUE
2122 012742 010700 IND11: SCOPE
2123
2124 ;*****
2125 ;*TEST 116 THIS TEST IS THE SAME AS ABOVE EXCEPT THE ADDRESS IS 'RED'
2126 ;*****
2127 012744 005237 000304 TST116: INC #0$TESTN
2128 012750 012706 001000 MOV #1000,06
2129 012754 012737 013006 000004 MOV #IND10,4
2130 012762 012737 021414 000336 MOV #TEMP,336
2131 012770 012776 177777 177336 MOV #=-1,-442(6) ;MOVE #-1 TO TEMP
2132 012776 000405 BR IND12
2133
2134 012780 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
2135 012782 012713 000241 MOV #241,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2136 012786 000000 HLT ;ERROR! TRAPPED
2137 012790 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2138 ;"JMP BR0" (000110) TO GET A
2139 ;SCOPE LOOP AND HIT CONTINUE
2140
  
```

```

2941 013012 010700 IND12: SCOPE
2942
2943 ;*****
2944 ;*TEST 117 TEST RFD ZONE VIOLATION USING EMT INSTRUCTION
2945 ;*****
2946 013014 005237 000304 TST117: INC 005TESTN
2947 013020 012706 000340 MOV #340,%6 ;SET UP STACK POINTER IN "RED" AREA
2948 013024 012737 013112 000030 MOV #RED1,EMTVEC
2949 013032 005037 000032 CLR EMTVEC+2
2950 013036 012737 013116 000004 MOV #RFD1A,4
2951 013044 005037 000006 CLR 6
2952 013050 005037 000000 CLR 0
2953 013054 012737 136336 000336 MOV #136336,336 ;PRESFT "RED" LOCATION
2954 013062 012737 177777 000002 MOV #-1,2 ;N HIT SET
2955 013070 104000 EMT
2956 013072
2957
2958 013072 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
2959 013074 012713 000242 MOV #242,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2960 013100 000000 RED1B: HLT
2961 013102 000240 LOOP
2962 ;REPLACE THIS INSTRUCTION BY A
2963 ;"JMP #R0" (000110) TO GET A
2964 ;SCOPE LOOP AND HIT CONTINUE
2965 013104 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
2966 013106 012713 000243 MOV #243,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2967 013112 000000 RED1: HLT ;DID NOT ABORT EMT
2968 013114 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2969 ;"JMP #R0" (000110) TO GET A
2970 ;SCOPE LOOP AND HIT CONTINUE
2971 013116 022737 000000 000002 RED1A: CMP #0,2 ;WAS (NEW) STATUS SAVED?
2972 013124 001405 BEQ 18
2973
2974 013126 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
2975 013130 012713 000244 MOV #244,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2976 013134 000000 HLT ;CORRECT (NEW) PSW WAS NOT SAVED
2977 ;ON A RFD ZONE TRAP (CAUSED BY EMT)
2978 ;UPON EXECUTING EMT RED ZONE OCCURED
2979 ;THE 11/60 SAVES THE "NEW" PSW WHICH
2980 ;WOULD HAVE BEEN LOADED BY THE EMT.
2981 ;11/40 SAVES THE PSW THAT WAS CURRENT AT THE
2982 ;TIME EMT WAS EXECUTED.)
2983 013136 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2984 ;"JMP #R0" (000110) TO GET A
2985 ;SCOPE LOOP AND HIT CONTINUE
2986 013140 022737 013072 000000 18: CMP #RED1C,0 ;WAS RETURN PC SAVED
2987 013146 001405 BEQ 28 ;(EMTVEC)
2988
2989 013150 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
2990 013152 012713 000245 MOV #245,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
2991 013156 000000 HLT
2992 013160 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
2993 ;"JMP #R0" (000110) TO GET A
2994 ;SCOPE LOOP AND HIT CONTINUE
2995 013162 005706 28: TST %6 ;STACK POINTER=0?
2996 013164 001405 BEQ 38
  
```

```

2997
2998 013166 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
2999 013170 012713 000246 MOV #246,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3000 013174 000000 HLT
3001 013176 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3002 ;"JMP #R0" (000110) TO GET A
3003 ;SCOPE LOOP AND HIT CONTINUE
3004 013200 022737 136336 000336 38: CMP #136336,336 ;WAS "RFD" LOCATION UNDISTURBED?
3005 013206 001405 BEQ 48
3006
3007 013210 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3008 013212 012713 000247 MOV #247,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3009 013216 000000 HLT
3010 013220 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3011 ;"JMP #R0" (000110) TO GET A
3012 ;SCOPE LOOP AND HIT CONTINUE
3013 013222 012737 000032 000030 48: MOV #EMTVEC+2,EMTVEC
3014 013230 010700 SCOPE
3015
3016 ;*****
3017 ;*TEST 120 TEST RFD ZONE VIOLATION USING "TRAP" INSTRUCTION
3018 ;*****
3019 013232 005237 000304 TST120: INC 005TESTN
3020 013236 012706 000200 MOV #200,%6 ;SET UP STACK IN "RED" AREA
3021 013242 012737 176176 000176 MOV #176176,176 ;PRF SET "RFD" LOCATION
3022 013250 012737 013326 000034 MOV #RED2,TRAPVEC
3023 013256 005037 000036 CLR TRAPVEC+2
3024 013262 005037 000006 CLR 6
3025 013266 005037 000000 CLR 0
3026 013272 012737 013332 000004 MOV #RED2A,4
3027 013300 005037 000002 CLR 2 ;Z BIT SET
3028 013304 104000 TRAP
3029 013306
3030
3031 013306 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3032 013310 012713 000250 MOV #250,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3033 013314 000000 RED2B: HLT ;ERROR! FAILED TO TRAP
3034 013316 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3035 ;"JMP #R0" (000110) TO GET A
3036 ;SCOPE LOOP AND HIT CONTINUE
3037
3038 013320 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3039 013322 012713 000251 MOV #251,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3040 013326 000000 RED2: HLT ;DID NOT ABORT TRAP
3041 013330 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3042 ;"JMP #R0" (000110) TO GET A
3043 ;SCOPE LOOP AND HIT CONTINUE
3044 013332 022737 000000 000002 RED2A: CMP #0,2 ;WAS (NEW) STATUS SAVED?
3045 013340 001405 BEQ 18
3046
3047 013342 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3048 013344 012713 000252 MOV #252,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3049 013350 000000 HLT ;CORRECT STATUS WORD WAS NOT SAVED
3050 ;UPON A RED ZONE VIOLATION CAUSED BY
3051 ;"TRAP". 11/60 SAVES THE NEW PSW
3052 ;WHICH WOULD HAVE BEEN LOADED BY "TRAP"
  
```

3053 ;11/40 SAVES THE "OLD" PSW THAT WAS CURRENT
3054 ;AT THE TIME "TRAP" WAS EXECUTED
3055 R13352 000210 LOOP ;REPLACE THIS INSTRUCTION BY A
3056 ;"JMP BR0" (000110) TO GET A
3057 ;SCOPE LOOP AND HIT CONTINUE
3058 R13354 022737 R13306 000000 18: CMP #RED2C,0
3059 R13362 001405 ;(TRAPVEC)
3060 ;
3061 R13364 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
3062 R13366 012713 000253 MOV #254,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
3063 R13372 000000 HLT ;
3064 R13374 000210 LOOP ;REPLACE THIS INSTRUCTION BY A
3065 ;"JMP BR0" (000110) TO GET A
3066 ;SCOPE LOOP AND HIT CONTINUE
3067 R13376 005706 25: TST #6
3068 R13400 001405 ;
3069 ;
3070 R13402 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
3071 R13404 012713 000254 MOV #254,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
3072 R13410 000000 HLT ;
3073 R13412 000210 LOOP ;REPLACE THIS INSTRUCTION BY A
3074 ;"JMP BR0" (000110) TO GET A
3075 ;SCOPE LOOP AND HIT CONTINUE
3076 R13414 022737 170176 000176 35: CMP #170176,176
3077 R13422 001405 ;WAS "RED" LOCATION LEFT UNDISTURBED?
3078 ;
3079 R13424 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
3080 R13426 012713 000255 MOV #255,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
3081 R13432 000000 HLT ;ERROR! "RED" LOCATION WAS CHANGED
3082 R13434 000210 LOOP ;REPLACE THIS INSTRUCTION BY A
3083 ;"JMP BR0" (000110) TO GET A
3084 ;SCOPE LOOP AND HIT CONTINUE
3085 R13436 012737 000036 000034 48: MOV #TRAPVEC+2,TRAPVEC
3086 R13444 010700 SCOPE ;
3087 ;
3088 ;*****
3089 ;*TFST 121 TEST RED ZONE VIOLATION USING "IOT" INSTRUCTION
3090 ;*****
3091 R13446 005237 000104 TST121: INC #TESTM
3092 R13452 012706 000100 MOV #100,#6
3093 R13456 012737 013534 000020 MOV #RED3,IOTVEC
3094 R13464 005037 000022 CLR IOTVEC+2
3095 R13470 005037 000006 CLR 6
3096 R13474 012737 013540 000004 MOV #RED3A,4
3097 R13502 005037 000000 CLR 0
3098 R13506 005037 000002 CLR 2
3099 R13512 000000 IOT
3100 R13514 RED3C:
3101 ;
3102 R13514 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
3103 R13516 012713 000256 MOV #256,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
3104 R13522 000000 HLT ;DID NOT OVERFLOW
3105 R13524 000210 LOOP ;REPLACE THIS INSTRUCTION BY A
3106 ;"JMP BR0" (000110) TO GET A
3107 ;SCOPE LOOP AND HIT CONTINUE
3108 ;

3109 R13526 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
3110 R13530 012713 000257 MOV #257,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
3111 R13534 000000 HLT ;DID NOT ABORT IOT
3112 R13536 000210 LOOP ;REPLACE THIS INSTRUCTION BY A
3113 ;"JMP BR0" (000110) TO GET A
3114 ;SCOPE LOOP AND HIT CONTINUE
3115 R13540 022737 000000 000002 RFD3A: CMP #0,2
3116 R13546 001405 ;WAS STATUS SAVED?
3117 ;
3118 R13550 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
3119 R13552 012713 000260 MOV #260,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
3120 R13556 000000 HLT ;CORRECT STATUS WORD WAS NOT SAVED
3121 ;UPON A RED ZONE VIOLATION CAUSED BY
3122 ;"IOT". 11/40 SAVES THE NEW PSW
3123 ;WHICH WOULD HAVE BEEN LOADED BY "IOT"
3124 ;11/40 SAVES THE "OLD" PSW THAT WAS CURRENT
3125 ;AT THE TIME "TRAP" WAS EXECUTED
3126 R13560 000210 LOOP ;REPLACE THIS INSTRUCTION BY A
3127 ;"JMP BR0" (000110) TO GET A
3128 ;SCOPE LOOP AND HIT CONTINUE
3129 R13562 022737 013514 000000 15: CMP #RED3C,0
3130 R13570 001405 ;WAS RETURN PC SAVED?
3131 ;(IOTVEC)
3132 R13572 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
3133 R13574 012713 000261 MOV #261,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
3134 R13600 000000 HLT ;
3135 R13602 000210 LOOP ;REPLACE THIS INSTRUCTION BY A
3136 ;"JMP BR0" (000110) TO GET A
3137 ;SCOPE LOOP AND HIT CONTINUE
3138 R13604 005706 25: TST #6
3139 R13606 001405 ;
3140 ;
3141 R13610 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
3142 R13612 012713 000262 MOV #262,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
3143 R13616 000000 HLT ;
3144 R13620 000210 LOOP ;REPLACE THIS INSTRUCTION BY A
3145 ;"JMP BR0" (000110) TO GET A
3146 ;SCOPE LOOP AND HIT CONTINUE
3147 R13622 012737 000022 000020 35: MOV #IOTVEC+2,IOTVEC
3148 R13630 010700 SCOPE ;
3149 ;
3150 ;*****
3151 ;*TFST 122 TEST RED ZONE VIOLATION UPON TRACE TRAP
3152 ;*****
3153 R13632 005237 000104 TST122: INC #TESTM
3154 R13636 015006 CLR #6
3155 R13640 012717 013716 000014 MOV #RED4,IOTVEC
3156 R13646 005037 000016 CLR IOTVEC+2
3157 R13652 005037 000006 CLR 6
3158 R13656 012737 013722 000004 MOV #RED4A,4
3159 R13664 005037 000000 CLR 0
3160 R13670 005037 000002 CLR 2
3161 R13674 000003 BPT ;TRACE TRAP
3162 R13676 RED4C:
3163 ;
3164 R13676 005212 INC (R2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR


```

3165 013700 012713 000263      MOV      #263,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3166 013704 000000      HLT      ;DID NOT OVERFLOW
3167 013706 000240      LOOP    ;REPLACE THIS INSTRUCTION BY A
3168                ;"JMP @R0" (000110) TO GET A
3169                ;SCOPE LOOP AND HIT CONTINUE
3170
3171 013710 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3172 013712 012713 000264      MOV      #264,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3173 013716 000000      HLT      ;DID NOT ABORT TRI
3174 013720 000240      LOOP    ;REPLACE THIS INSTRUCTION BY A
3175                ;"JMP @R0" (000110) TO GET A
3176                ;SCOPE LOOP AND HIT CONTINUE
3177 013722 022737 000000 000002 RED4A:  CMP      #0,2          ;WAS CORRECT PSW SAVED?
3178 013730 001405      BFO     1$
3179
3180 013732 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3181 013734 012713 000265      MOV      #265,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3182 013740 000000      HLT      ;CORRECT STATUS WORD WAS NOT SAVED
3183                ;UPON A RED ZONE TRAP CAUSED BY A BPT
3184                ;NEW PSW TAG WOULD BE LOADED BY BPT
3185                ;SHOULD BE SAVED.
3186 013742 000240      LOOP    ;REPLACE THIS INSTRUCTION BY A
3187                ;"JMP @R0" (000110) TO GET A
3188                ;SCOPE LOOP AND HIT CONTINUE
3189 013744 022737 013076 000000 1$:  CMP      #RED4C,0
3190 013752 001405      BEQ     2$
3191
3192 013754 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3193 013756 012713 000266      MOV      #266,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3194 013762 000000      HLT      ;REPLACE THIS INSTRUCTION BY A
3195 013764 000240      LOOP    ;"JMP @R0" (000110) TO GET A
3196                ;SCOPE LOOP AND HIT CONTINUE
3197
3198 013766 005706      2$:  TST     #6
3199 013770 001405      BEQ     3$
3200
3201 013772 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3202 013774 012713 000267      MOV      #267,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3203 014000 000000      HLT      ;REPLACE THIS INSTRUCTION BY A
3204 014002 000240      LOOP    ;"JMP @R0" (000110) TO GET A
3205                ;SCOPE LOOP AND HIT CONTINUE
3206
3207 014004 012737 000016 000014 3$:  MOV      #TRPVEC+2,TRPVEC
3208 014012 010700      SCOPE
3209
3210                ;*****
3211                ;*TST 123 TEST TRANSITION FROM "YELLOW" TO "RED" AREAS
3212                ;*****
3213 014014 005237 000304      TST123: INC      #STESTV
3214                ;THE TRANSITION OCCURS AFTER THE EMT HAS PUSHED ITS RETURN ADDRESS AND STATUS.
3215 014020 012706 000344      MOV      #344,#6        ;SET UP STACK TO ALLOW 2 "PUSHES"
3216 014024 012737 014124 000030      MOV      #REDS,EMTVEC  ;LOAD EMT VECTOR
3217 014032 005037 000032      CLP     EMTVEC+2      ;AND STATUS
3218 014036 012737 014132 000004      MOV      #REDS+4       ;LOAD OVERFLOW VECTOR
3219 014044 012737 000001 000006      MOV      #1,6
3220 014052 012737 000017 000002      MOV      #17,2
  
```

```

3221 014060 005037 000000      CLR      0
3222 014064 012737 136336 000336      MOV      #136336,336   ;PHE SET "RED" LOCATION
3223 014072 012737 000004 177776      MOV      #4,CC
3224 014100 104000      EMT
3225 014102
3226 RED5C:
3227 014102 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3228 014104 012713 000270      MOV      #270,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3229 014110 000000      HLT      ;REPLACE THIS INSTRUCTION BY A
3230 014112 000240      LOOP    ;"JMP @R0" (000110) TO GET A
3231                ;SCOPE LOOP AND HIT CONTINUE
3232
3233 014114 000240      NOP
3234
3235 014116 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3236 014120 012713 000271      MOV      #271,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3237 014124 000000      HLT      ;REPLACE THIS INSTRUCTION BY A
3238 014126 000240      LOOP    ;"JMP @R0" (000110) TO GET A
3239                ;SCOPE LOOP AND HIT CONTINUE
3240
3241 014130 000240      NOP
3242 014132 022737 000004 000342 RED5A:  CMP      #4,342
3243 014140 001405      BEQ     1$
3244
3245 014142 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3246 014144 012713 000272      MOV      #272,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3247 014150 000000      HLT      ;ERROR! EMT DID NOT SAVE STATUS
3248 014152 000240      LOOP    ;REPLACE THIS INSTRUCTION BY A
3249                ;"JMP @R0" (000110) TO GET A
3250                ;SCOPE LOOP AND HIT CONTINUE
3251
3252 014154 022737 014102 000340 1$:  CMP      #RED5C,340
3253 014162 001405      BEQ     2$
3254
3254 014164 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3255 014166 012713 000273      MOV      #273,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3256 014172 000000      HLT      ;ERKOP! RETURN PC NOT SAVED
3257 014174 000240      LOOP    ;REPLACE THIS INSTRUCTION BY A
3258                ;"JMP @R0" (000110) TO GET A
3259                ;SCOPE LOOP AND HIT CONTINUE
3260
3260 014176 022737 136336 000336 2$:  CMP      #136336,336   ;WAS "RED" LOCATION LEFT UNDISTURBED?
3261 014204 001405      BEQ     3$
3262
3263 014206 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3264 014210 012713 000274      MOV      #274,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3265 014214 000000      HLT      ;ERKOP! "RED" AREA WAS CHANGED
3266 014216 000240      LOOP    ;REPLACE THIS INSTRUCTION BY A
3267                ;"JMP @R0" (000110) TO GET A
3268                ;SCOPE LOOP AND HIT CONTINUE
3269
3269 014220 022737 000001 000002 3$:  CMP      #1,2
3270 014226 001405      BEQ     4$
3271
3272 014230 005212      INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3273 014232 012713 000275      MOV      #275,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3274 014236 000000      HLT      ;CORRECT PSW WAS NOT SAVED WHEN
3275                ;A RED ZONE OCCURED WHILE TRYING
3276                ;TO SERVICE YELLOW ZONE. AFTER THE EMT
  
```

```

3277                                     ;IS EXECUTED THE PENDING YELLOW ZONE OCCURS
3278                                     ;(AND YELLOW ZONE PSW IS LOADED), BUT TRYING
3279                                     ;TO PUSH THE STACK CAUSES RED ZONE AND
3280                                     ;THE NEW PSW (YELLOW ZONE'S) SHOULD
3281                                     ;BE SAVED ON THE STACK
3282 014240 000240                               LOOP      ;REPLACE THIS INSTRUCTION BY A
3283                                     ;"JMP 000" (000110) TO GET A
3284                                     ;SCOPE LOOP AND HIT CONTINUE
3285 014242 022737 014124 000000 48:    CMP      #RED5,0      ;WAS FMT'S VECTOR SAVED?
3286 014250 001405                               RFQ      58
3287
3288 014252 005212                               INC      (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3289 014254 012713 000276                               MOV      #276,(R3)   ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3290 014260 000000                               HLT
3291 014262 000240                               LOOP      ;ERROR! OVERFLOW'S VECTOR (4) NOT SAVED
3292                                     ;REPLACE THIS INSTRUCTION BY A
3293                                     ;"JMP 000" (000110) TO GET A
3294 014264 005706                               58:      TST      B6         ;SCOPE LOOP AND HIT CONTINUE
3295 014266 001405                               BEQ      B5
3296
3297 014270 005212                               INC      (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3298 014272 012713 000277                               MOV      #277,(R3)   ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3299 014276 000000                               HLT
3300 014300 000240                               LOOP      ;REPLACE THIS INSTRUCTION BY A
3301                                     ;"JMP 000" (000110) TO GET A
3302                                     ;SCOPE LOOP AND HIT CONTINUE
3303
3304
3305 014302                               66:
3306
3307 014302 012737 000032 000030          MOV      #EMTVEC+2,EMTVEC
3308 014310 005037 000032          CLR      EMTVEC+2
3309 014314 010700          SCOPE
3310
3311                                     ;*****
3312                                     ;TEST 124 TEST TRANSITION FROM "YELLOW" TO "RED" ZONES
3313                                     ;*****
3314 TST124: INC      #000TESTN
3315                                     ;THE TRANSITION OCCURS AFTER THE JSR HAS "PUSHED" ITS OLD R5.
3316
3317 014322 012706 000342          MOV      #342,B6 ;SET UP STACK ON THE "HAIRY EDGE"
3318 014326 012737 014424          MOV      #RED6A,4   ;LOAD OVERFLOW VECTOR
3319 014334 012737 000357 000006          MOV      #357,6     ;AND OVERFLOW STATUS
3320 014342 005037 000000          CLR      0
3321 014346 005037 000002          CLR      2
3322 014352 005037 000340          CLR      340
3323 014356 012705 000007          MOV      #7,05     ;PRE SET R5
3324 014362 012737 000017 177776          MOV      #17,CC    ;PRE SET THE STATUS WORD
3325 014370 004537 014416          JSR      5,RED6    ;SHOULD CAUSE OVERFLOW TRAP
3326
3327 RFD6C:
3328 014374
3329
3330 014374 005212                               INC      (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3331 014376 012713 000300                               MOV      #300,(R3)   ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3332 014402 000000                               HLT          ;ERROR! DID NOT OVERFLOW TRAP
3333 014404 000240                               LOOP      ;REPLACE THIS INSTRUCTION BY A
3334                                     ;"JMP 000" (000110) TO GET A
3335                                     ;SCOPE LOOP AND HIT CONTINUE
3336
3337 NOP
3338
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369
3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388
  
```

```

3333
3334 014410 005212                               INC      (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3335 014412 012713 000301                               MOV      #301,(R3)   ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3336 014416 000000                               HLT          ;ERROR! DID NOT TRAP BEFORE JSR
3337 014420 000240                               LOOP      ;REPLACE THIS INSTRUCTION BY A
3338                                     ;"JMP 000" (000110) TO GET A
3339                                     ;SCOPE LOOP AND HIT CONTINUE
3340
3341 014422 000240                               NOP
3342 014424 022737 000357 177776 RED6A:  CMP      #357,CC
3343 014432 001405                               BEQ      66
3344
3345 014434 005212                               INC      (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3346 014436 012713 000302                               MOV      #302,(R3)   ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3347 014442 000000                               HLT          ;WRONG PSW ON RED ZONE TRAP
3348 014444 000240                               LOOP      ;REPLACE THIS INSTRUCTION BY A
3349                                     ;"JMP 000" (000110) TO GET A
3350                                     ;SCOPE LOOP AND HIT CONTINUE
3351 014446 022737 000007 000340 68:    CMP      #7,340
3352 014454 001405                               BEQ      18
3353
3354 014456 005212                               INC      (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3355 014460 012713 000303                               MOV      #303,(R3)   ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3356 014464 000000                               HLT          ;ERROR! R5 NOT SAVED ON THE STACK
3357 014466 000240                               LOOP      ;REPLACE THIS INSTRUCTION BY A
3358                                     ;"JMP 000" (000110) TO GET A
3359                                     ;SCOPE LOOP AND HIT CONTINUE
3360 014470 022705 014374                               18:      CMP      #RED6C,*5
3361 014474 001405                               RFQ      28
3362
3363 014476 005212                               INC      (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3364 014500 012713 000304                               MOV      #304,(R3)   ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3365 014504 000000                               HLT          ;ERROR! R5 DID NOT GET RETURN ADRS.
3366 014506 000240                               LOOP      ;REPLACE THIS INSTRUCTION BY A
3367                                     ;"JMP 000" (000110) TO GET A
3368                                     ;SCOPE LOOP AND HIT CONTINUE
3369 014510 022737 000357 000002 28:    CMP      #357,2
3370 014516 001405                               BEQ      38
3371
3372 014520 005212                               INC      (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3373 014522 012713 000305                               MOV      #305,(R3)   ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3374 014526 000000                               HLT          ;CORRECT PSW WAS NOT SAVED WHEN
3375                                     ;A RED ZONE OCCURED WHILE TRYING
3376                                     ;TO SERVICE YELLOW ZONE. AFTER THE JSR
3377                                     ;IS EXECUTED THE PENDING YELLOW ZONE OCCURS
3378                                     ;(AND YELLOW ZONE PSW IS LOADED), BUT TRYING
3379                                     ;TO PUSH THE STACK CAUSES RED ZONE AND
3380                                     ;THE NEW PSW (YELLOW ZONE'S) SHOULD
3381                                     ;BE SAVED ON THE STACK
3382                                     ;REPLACE THIS INSTRUCTION BY A
3383                                     ;"JMP 000" (000110) TO GET A
3384                                     ;SCOPE LOOP AND HIT CONTINUE
3385 014530 000240                               LOOP      ;WAS RETURN ADDRESS SAVED ON THE STACK
3386
3387 014532 022737 014416 000000 38:    CMP      #RED6,0
3388 014540 001405                               BEQ      48
3389
3390 014542 005212                               INC      (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3391 014544 012713 000306                               MOV      #306,(R3)   ;MOVE TO MAIL BOX ($FATAL) THE ERROR
  
```

```

3389 014550 000000          HLT          ;OVERFLOW TRAP DID NOT SAVE RETURN PC
3390 014552 000240          LOOP         ;REPLACE THIS INSTRUCTION BY A
3391                                     ;"JMP 000" (000110) TO GET A
3392                                     ;SCOPE LOOP AND HIT CONTINUE
3393 014554 022737 136336 000336 48:  CMP      #136336,336
3394 014562 001405          BEQ      58          ;WAS "RED" LOCATION UNDISTURBED?
3395
3396 014564 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3397 014566 012713 000307    MOV      #307,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3398 014572 000000          HLT          ;ERROR! "RED" LOCATION WAS CHANGED
3399 014574 000240          LOOP         ;REPLACE THIS INSTRUCTION BY A
3400                                     ;"JMP 000" (000110) TO GET A
3401                                     ;SCOPE LOOP AND HIT CONTINUE
3402 014576 010700          58:      SCOPE
3403
3404
3405
3406
3407 014600 005237 000304    TST125: INC      #0$TESTN
3408                                     ;THIS TEST IS THE SAME AS ABOVE EXCEPT THAT THE TRANSITION IS AFTER THE
3409 ;OVERFLOW PUSHES ONF WORD.
3410 014604 012706 000344    MOV      #344,%6
3411 014610 012737 014676 000004    MOV      #REF7A,4   ;LOAD OVERFLOW VECTOR
3412 014616 015037 000006    CLR      6
3413 014622 005037 000342    CLF      342
3414 014626 012737 136336 000336    MOV      #136336,336 ;PRE SET "RED" LOCATION
3415 014634 012705 000007    MOV      #7,%5      ;PRE SET R5
3416 014640 012737 000357 177776    MOV      #357,CC    ;PRE SET STATUS
3417 014646 004537 014672    JSR      5,RED7     ;CAUSE OVERFLOW
3418 014652
3419
3420 014652 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3421 014654 012713 000310    MOV      #310,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3422 014660 000000          HLT          ;ERROR! DID NOT OVERFLOW TRAP
3423 014662 000240          LOOP         ;REPLACE THIS INSTRUCTION BY A
3424                                     ;"JMP 000" (000110) TO GET A
3425                                     ;SCOPE LOOP AND HIT CONTINUE
3426
3427 014664 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3428 014666 012713 000311    MOV      #311,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3429 014672 000000          HLT          ;ERROR! DID NOT TRAP BEFORE JSR
3430 014674 000240          LOOP         ;REPLACE THIS INSTRUCTION BY A
3431                                     ;"JMP 000" (000110) TO GET A
3432                                     ;SCOPE LOOP AND HIT CONTINUE
3433 014676 022737 000007 000342 RED7A:  CMP      #7,342
3434 014704 001405          BEQ      18          ;WAS R5 SAVED
3435
3436 014706 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3437 014710 012713 000312    MOV      #312,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3438 014714 000000          HLT          ;ERROR! R5 NOT SAVED ON THE STACK
3439 014716 000240          LOOP         ;REPLACE THIS INSTRUCTION BY A
3440                                     ;"JMP 000" (000110) TO GET A
3441                                     ;SCOPE LOOP AND HIT CONTINUE
3442 014720 022705 014652    18:      CMP      #RED7C,%5
3443 014724 001405          BEQ      28          ;DOES R5 CONTAIN RETURN PC?
3444

```

```

3445 014726 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3446 014730 012713 000313    MOV      #313,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3447 014734 000000          HLT          ;ERROR! JSR DID NOT LOAD R5
3448 014736 000240          LOOP         ;REPLACE THIS INSTRUCTION BY A
3449                                     ;"JMP 000" (000110) TO GET A
3450                                     ;SCOPE LOOP AND HIT CONTINUE
3451 014740 022737 000357 000340 28:  CMP      #357,340
3452 014746 001405          BEQ      38          ;WAS STATUS SAVED?
3453
3454 014750 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3455 014752 012713 000314    MOV      #314,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3456 014756 000000          HLT          ;ERROR! STATUS NOT SAVED
3457 014760 000240          LOOP         ;REPLACE THIS INSTRUCTION BY A
3458                                     ;"JMP 000" (000110) TO GET A
3459                                     ;SCOPE LOOP AND HIT CONTINUE
3460 014762 022737 000000 000002 38:  CMP      #0,2
3461 014770 001425          BEQ      48          ;WAS STATUS SAVED?
3462
3463 014772 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3464 014774 012713 000315    MOV      #315,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3465 015000 000000          HLT          ;CORRECT PSW WAS NOT SAVED WHEN
3466                                     ;A RED ZONE OCCURED WHILE TRYING
3467                                     ;TO SERVICE YELLOW ZONE. AFTER THE JSR
3468                                     ;IS EXECUTED THE PENDING YELLOW ZONE OCCURS
3469                                     ;(AND YELLOW ZONE PSW IS LOADED), BUT TRYING
3470                                     ;TO PUSH THE STACK CAUSES RED ZONE AND
3471                                     ;THE NEW PSW (YELLOW ZONE'S) SHOULD
3472                                     ;BE SAVED ON THE STACK
3473 015002 000240          LOOP         ;REPLACE THIS INSTRUCTION BY A
3474                                     ;"JMP 000" (000110) TO GET A
3475                                     ;SCOPE LOOP AND HIT CONTINUE
3476 015004 022737 014672 000000 48:  CMP      #RED7,0
3477 015012 001405          BEQ      58          ;RETURN PC SAVED?
3478
3479 015014 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3480 015016 012713 000316    MOV      #316,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3481 015022 000000          HLT          ;ERROR! RETURN PC NOT SAVED
3482 015024 000240          LOOP         ;REPLACE THIS INSTRUCTION BY A
3483                                     ;"JMP 000" (000110) TO GET A
3484                                     ;SCOPE LOOP AND HIT CONTINUE
3485 015026 022737 136336 000336 58:  CMP      #136336,336
3486 015034 001405          BEQ      68          ;WAS "RED" LOCATION UNDISTURBED?
3487
3488 015036 005212          INC      (R2)       ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3489 015040 012713 000317    MOV      #317,(R3)  ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3490 015044 000000          HLT          ;ERROR! "RED" LOCATION WAS CHANGED
3491 015046 000240          LOOP         ;REPLACE THIS INSTRUCTION BY A
3492                                     ;"JMP 000" (000110) TO GET A
3493                                     ;SCOPE LOOP AND HIT CONTINUE
3494 015050 010700          68:      SCOPE
3495
3496
3497
3498
3499 015052 005237 000304    TST126: INC      #0$TESTN
3500 015056 012737 000340 177776    MOV      #340,STATUS ;LOCK OUT INTERRUPT

```

```

3501 015064 012700 000400      MOV    #000,06      ;SET UP STACK TO OVERFLOW
3502 015074 012737 015142 000004      MOV    #TDFC7,4     ;SET UP OVERFLOW TRAP
3503 015076 012737 015136 000064      MOV    #TDFC8,64    ;SET UP INTERRUPT VECTOR
3504 015104 012737 000100 177564      MOV    #100,TTCSP   ;SET INTERRUPT ENABLE
3505 015112 005037 177770      CLR    STATUS       ;ALLOW INTERRUPT TO OCCUR
3506
3507 015116 005212              INC    (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3508 015120 012713 000320      MOV    #320,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3509 015124 000000              HLT                    ;NO INTERRUPT OCCURRED
3510 015126 000240              LOOP                   ;REPLACE THIS INSTRUCTION BY A
3511                                     ;"JMP #00" (000110) TO GET A
3512                                     ;SCOPE LOOP AND HIT CONTINUE
3513
3514 015130 005212              INC    (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3515 015132 012713 000320      MOV    #320,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3516 015136 000000      TDFC8: HLT                    ;THAT FLAG OVERFLOW DID NOT OCCUR
3517 015140 000240              LOOP                   ;REPLACE THIS INSTRUCTION BY A
3518                                     ;"JMP #00" (000110) TO GET A
3519                                     ;SCOPE LOOP AND HIT CONTINUE
3520 015142 005037 177564      TDEC7: CLW    TTCSR   ;CLEAR INTERRUPT ENABLE
3521 015146 010700              SCOPE
  
```

```

3522
3523
3524
3525                                     ;*****
3526 015150 005237 000304      TST127: INC    #0TESTN      ;*****
3527 015154 012706 021774      MOV    #BUFF,06     ;SET UP STACK POINTER
3528 015160 012737 015214 000004      MOV    #R7TR1,4     ;RETURN FROM TRAP
3529 015166 012707 000001      MOV    #1,07        ;PC EQUALS ONE
3530
3531 015172 005212              INC    (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3532 015174 012713 000322      MOV    #322,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3533 015200 000000              HLT                    ;REPLACE THIS INSTRUCTION BY A
3534 015202 000240              LOOP                   ;"JMP #00" (000110) TO GET A
3535                                     ;SCOPE LOOP AND HIT CONTINUE
3536
3537
3538 015204 005212              INC    (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3539 015206 012713 000323      MOV    #323,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3540 015212 000000              HLT                    ;
3541 015214 022737 000003 021770 07TR1: CMP    #3,BUFF-4     ;
3542 015222 001405              BEQ    15            ;CORRECT PC WAS NOT SAVED ON STACK
3543
3544 015224 005212              INC    (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3545 015226 012713 000324      MOV    #324,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3546 015232 000000              HLT                    ;REPLACE THIS INSTRUCTION BY A
3547 015234 000240              LOOP                   ;"JMP #00" (000110) TO GET A
3548                                     ;SCOPE LOOP AND HIT CONTINUE
3549
3550
3551 015236 010700              16:    SCOPE
3552
3553                                     ;*****
3554                                     ;*TEST 130 TEST THAT A TRAP OCCURS WHEN THE PC IS INCREMENTED TO AN ODD VALUE
3555                                     ;*****
3556 015240 005237 000304      TST130: INC    #0TESTN      ;*****
3557 015244 012706 021774      MOV    #BUFF,06     ;STACK POINTER
3558 015250 012737 015302 000004      MOV    #R7TR2,4     ;
3559 015256 005207              INC    #7            ;PC BECOMES ODD
3560 015260      R7TR2A:
3561
3562 015260 005212              INC    (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3563 015262 012713 000325      MOV    #325,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3564 015266 000000              HLT                    ;REPLACE THIS INSTRUCTION BY A
3565 015270 000240              LOOP                   ;"JMP #00" (000110) TO GET A
3566                                     ;SCOPE LOOP AND HIT CONTINUE
3567
3568
3569 015272 005212              INC    (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3570 015274 012713 000326      MOV    #326,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3571 015300 000000              HLT                    ;
3572 015302 022737 015263 021770 07TR2: CMP    #R7TR2A+3,BUFF-4
3573 015310 001405              BEQ    15            ;CORRECT PC NOT ON STACK
3574
3575 015312 005212              INC    (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3576 015314 012713 000327      MOV    #327,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3577 015320 000000              HLT
  
```

T130 TEST THAT A TRAP OCCURS WHEN THE PC IS INCREMENTED TO AN ODD VALUE

```
3578 015322 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
3579                                     ;"JMP 000" (000110) TO GET A
3580                                     ;SCOPE LOOP AND HIT CONTINUE
3581 015324 010700          IS:   SCOPE
3582                                     ;*****
3583                                     ;*TEST 131 TEST THAT A DECREMENT OF PC TO AN ODD VALUE RESULTS IN A TRAP
3584                                     ;*****
3585
3586 015326 005237 000304          TST131: INC  #05TESTN
3587 015332 012736 021774          MOV  #RUFF,#6
3588 015336 012737 015360 000004  MOV  #R7TR3,4
3589 015344 005327          DEC  #7          ;MAKE PC ODD
3590 015346
3591
3592 015346 005212          INC  (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3593 015350 012713 000330          MOV  #330,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3594 015354 000000          HLT          ;SHOULD TRAP
3595 015356 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
3596                                     ;"JMP 000" (000110) TO GET A
3597                                     ;SCOPE LOOP AND HIT CONTINUE
3598 015360 022737 015347 021770  R7TR3: CMP  #R7TR3A+1,RUFF-4 ;CHECK VALUE OF PC ON STACK
3599 015366 001405          BEQ  15
3600
3601 015370 005212          INC  (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3602 015372 012713 000331          MOV  #331,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3603 015376 000000          HLT          ;WRONG VALUE ON STACK
3604 015400 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
3605                                     ;"JMP 000" (000110) TO GET A
3606                                     ;SCOPE LOOP AND HIT CONTINUE
3607
3608 015402 010700          IS:   SCOPE
3609                                     ;*****
3610                                     ;*TEST 132 TEST THAT "SEC, POL PC" RESULTS IN TRAP
3611                                     ;*****
3612
3613 015404 005237 000304          TST132: INC  #05TESTN
3614 015410 012706 021774          MOV  #RUFF,#6
3615 015414 012737 015450 000004  MOV  #R7TR4,4
3616 015422 000761          SEC          ;CARRY EQUALS A 1
3617 015424 006107          ROL  #7          ;PC BECOMES ODD
3618 015426
3619
3620 015426 005212          INC  (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3621 015430 012713 000332          MOV  #332,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3622 015434 000000          HLT          ;SHOULD TRAP
3623 015436 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
3624                                     ;"JMP 000" (000110) TO GET A
3625                                     ;SCOPE LOOP AND HIT CONTINUE
3626
3627 015440 005212          INC  (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3628 015442 012713 000333          MOV  #333,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3629 015446 000000          HLT          ;SHOULD TRAP
3630 015450 022737 033057 021770  R7TR4: CMP  #TR4A+TR4A+3,RUFF-4 ;CHECK FOR VALUE ON STACK
3631 015456 001405          BEQ  15
3632
3633 015460 005212          INC  (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
```

T132 TEST THAT "SEC, POL PC" RESULTS IN TRAP

```
3634 015462 012713 000334          MOV  #334,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3635 015466 000000          HLT          ;WRONG VALUE ON STACK
3636 015470 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
3637                                     ;"JMP 000" (000110) TO GET A
3638                                     ;SCOPE LOOP AND HIT CONTINUE
3639 015472 012737 000006 000004  IS:   MOV  #6,4          ;RESET UP A HALT FOR TRAP
3640
3641 015500 010700          SCOPE
3642                                     ;*****
3643                                     ;*TEST 133 TEST THAT A PENDING TTY INTERRUPT OCCURS BEFORE A "TRAP" TRAP
3644                                     ;*****
3645
3646 015502 005237 000304          TST133: INC  #05TESTN
3647 015506 012706 021774          MOV  #RUFF,#6
3648 015512 012737 000340 177776  MOV  #340,STATUS ;HIGHEST PRIORITY LEVEL
3649 015520 012737 000100 177564  MOV  #100,TTCSR  ;INTERRUPT FOR TTY PUNCH/PRINTER
3650 015526 012737 015564 000034  MOV  #TR1,34     ;TRAP VECTOR
3651 015534 012737 015570 000064  MOV  #TR2,64     ;TTY VECTOR
3652 015542 012737 000340 000036  MOV  #340,64    ;IF TRAP TRAPS, MOVE 340 TO PRIORITY
3653 015550 005037 177776          CLR  STATUS     ;SHOULD TRAP AT END OF CLR INST
3654 015554 104400          TRAP          ;TTY INTERRUPT SHOULD OVERRIDE TRAP
3655
3656 015556 005212          INC  (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3657 015560 012713 000335          MOV  #335,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3658 015564 000000          HLT          ;TRAP OCCUR FIRST
3659 015566 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
3660                                     ;"JMP 000" (000110) TO GET A
3661                                     ;SCOPE LOOP AND HIT CONTINUE
3662 015570 005037 177564          TP2:  CLR  TTCSR
3663 015574 005037 000036          CLR  36
3664 015600 010700          SCOPE
3665
3666                                     ;*****
3667                                     ;*TEST 134 TEST THAT A PENDING TTY INTERRUPT OCCURS BETWEEN "TRAP" & AN "IOT"
3668                                     ;*****
3669
3670 015602 005237 000304          TST134: INC  #05TESTN
3671 015606 012706 021774          MOV  #RUFF,#6
3672 015612 012737 000340 177776  MOV  #340,STATUS ;HIGHEST PRIORITY LEVEL
3673 015620 012737 000100 177564  MOV  #100,TTCSR  ;INTERRUPT FOR TTY PUNCH/PRINTER
3674 015626 012737 015664 000034  MOV  #TR1,34     ;TRAP VECTOR
3675 015634 005037 000036          CLR  36         ;TRAP
3676 015636 012737 015700 000064  MOV  #TR4,64     ;TTY OUTPUT
3677 015640 012737 015674 000020  MOV  #TR5,20    ;IOT
3678 015646 012737 000340 000022  MOV  #340,72    ;IOT PRIORITY
3679 015664 000004          TRAP          ;THE ACT OF TRAPPING LOWER PRIORITY
3680                                     ;INTERRUPT SHOULD OCCURE INPLACE OF IOT TRAP
3681 015666 005212          INC  (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3682 015670 012713 000336          MOV  #336,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3683 015674 000000          HLT          ;NO INTERRUPT BETWEEN TRAPS
3684 015676 000240          LOOP          ;REPLACE THIS INSTRUCTION BY A
3685                                     ;"JMP 000" (000110) TO GET A
3686                                     ;SCOPE LOOP AND HIT CONTINUE
3687 015700 005037 177564          TR4:  CLR  TTCSR
3688 015704 005037 000022          CLR  22
3689 015710 010700          SCOPE          ;CLR IOT PRIORITY
```

```

3690 ;*****
3691 ;*TEST 135 TEST THAT 'T' BIT TRAP OCCURS BEFORE AN INTERRUPT IS ACKNOWLEDGED
3692 ;*****
3693 015712 005237 000304 TST135: INC 00TESTN
3694 015716 012737 000340 177776 MOV 0340,CC ;SET PRIORITY =7
3695 015724 012737 000100 177564 MOV 0100,TTCSR ;ENABLE INTERRUPT ON TTY PRINTER
3696 015732 012737 015770 000014 MOV 0INT,TBITVLC ;LOAD 'T' BIT VECTOR
3697 015740 005037 000016 CLR TBITVEC+2
3698 015744 012737 016002 000064 MOV 0INT2,64 ;LOAD TTY INT. VECTOR
3699 015752 012706 021774 MOV 0BUFF,06 ;SET STACK POINTER
3700 015756 012746 000020 MOV 020,-(6) ;PUSH 'T'BIT ON THE STACK
3701 015762 012746 015770 MOV 0,+6,-(6) ;PUSH PC ON THE STACK
3702 015766 000002 RTI ;SET 'T' BIT
3703 015770 INT:
3704
3705 015770 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3706 015772 012713 000337 MOV 0337,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3707 015776 000000 HLT ;ERROR! TTY INTERRUPTED
3708 016000 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3709 ;"JMP 000" (000110) TO GET A
3710 ;SCOPE LOOP AND HIT CONTINUE
3711 016002 005037 177564 INT2: CLR TTCSR
3712 016006 022737 000020 021772 CMP 020,BUFF-2
3713 016014 001405 BEQ 1$
3714
3715 016016 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3716 016020 012713 000340 MOV 0340,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3717 016024 000000 HLT
3718 016026 000240 LOOP
3719 ;REPLACE THIS INSTRUCTION BY A
3720 ;"JMP 000" (000110) TO GET A
3721 ;SCOPE LOOP AND HIT CONTINUE
3721 016030 022737 015770 021770 1$: CMP 0INT,BUFF-4
3722 016036 001405 BEQ 2$
3723
3724 016040 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3725 016042 012713 000341 MOV 0341,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3726 016046 000000 HLT
3727 016050 000240 LOOP
3728 ;REPLACE THIS INSTRUCTION BY A
3729 ;"JMP 000" (000110) TO GET A
3730 ;SCOPE LOOP AND HIT CONTINUE
3730 016052 022737 000000 021766 2$: CMP 00,BUFF-6
3731 016060 001405 BEQ 3$
3732
3733 016062 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3734 016064 012713 000342 MOV 0342,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3735 016070 000000 HLT
3736 016072 000240 LOOP
3737 ;REPLACE THIS INSTRUCTION BY A
3738 ;"JMP 000" (000110) TO GET A
3739 ;SCOPE LOOP AND HIT CONTINUE
3739 016074 022737 015770 021764 3$: CMP 0INT,BUFF-10
3740 016102 001405 BEQ INT1
3741
3742 016104 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3743 016106 012713 000343 MOV 0343,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3744 016112 000000 HLT
3745 016114 000240 LOOP ;REPLACE THIS INSTRUCTION BY A

```

```

3746 ;"JMP 000" (000110) TO GET A
3747 ;SCOPE LOOP AND HIT CONTINUE
3748 016116 010700 INT1: SCOPE
3749
3750 ;*****
3751 ;*TEST 136 TEST THAT "RESET" GOES TO OUTSIDE WORLD
3752 ;*****
3753 016120 005237 000304 TST136: INC 00TESTN
3754 016124 012706 021774 MOV 0BUFF,06
3755 016130 012737 000340 177776 MOV 0340,CC ;LOCK OUT INTERRUPTS
3756 016136 012737 016206 000064 MOV 0TR6,64 ;LOAD TELEPRINTER VECTOR
3757 016144 012737 000100 177564 MOV 0100,TTCSR ;SET INTERRUPT ENABLE
3758 016152 000005 RESET ;SHOULD CLEAR INTERRUPT ENABLE
3759 016154 032737 000100 177564 BIT 0100,TTCSR ;TEST FOR CLEAR
3760 016162 001405 BEQ 1$
3761
3762 016164 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3763 016166 012713 000344 MOV 0344,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3764 016172 000000 HLT ;RESET FAILED TO CLEAR TTCSR
3765 016174 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3766 ;"JMP 000" (000110) TO GET A
3767 ;SCOPE LOOP AND HIT CONTINUE
3768 016176 000406 1$: BR TR6X ;GO TO SCOPE
3769
3770 016200 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3771 016202 012713 000345 TP6: MOV 0345,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3772 016206 000000 HLT ;ERROR! TELETYPE INTERRUPTED WHEN
3773 ;PROCESSOR WAS AT LEVEL 7
3774 016210 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3775 ;"JMP 000" (000110) TO GET A
3776 ;SCOPE LOOP AND HIT CONTINUE
3777 016212 022626 TR6X: CMP (6)+,(6)+
3778 016214 005037 177564 CLR TTCSR
3779 016220 010700 SCOPE
3780
3781 ;*****
3782 ;*TEST 137 TEST THAT RESET DOES NOT HANG THE SYSTEM
3783 ;*****
3784 016222 005237 000304 TST137: INC 00TESTN
3785 016226 012706 021774 MOV 0BUFF,06 ;SET STACK
3786 016232 005037 177776 CLR STATUS ;ALLOW INTERRUPT
3787 016236 012737 016252 000064 MOV 0RESET,64 ;TTY INTERRUPT VECTOR
3788 016244 052737 000100 177564 BIS 0100,TTCSR ;SET INTERRUPT ENABLE
3789 016252 000005 RESET1: RESET ;IF THIS HANGS CHECK SACK
3790 016254 012737 000066 000064 MOV 066,64 ;FOR FALSE INTERRUPT
3791 016262 010700 SCOPE
3792 ;*****
3793 ;*TEST 140 TEST RESET WITH TRACE ON
3794 ;*****
3795 016264 005237 000304 TST140: INC 00TESTN
3796 016270 012706 021774 MOV 0BUFF,06 ;SET STACK
3797 016274 012737 016336 000014 MOV 0RESET2,TRITVEC ;SET UP TRACE VECTOR
3798 016302 005037 000016 CLR TBITVEC+2
3799 016306 012746 000020 MOV 020,-(6) ;PUSH 'T'BIT ON THE STACK
3800 016312 012746 016320 MOV 0,+6,-(6) ;PUSH PC ON THE STACK
3801 016316 000002 RTI ;SET 'T' BIT

```

```

3802 016320 000005          RESET
3803 016322 000005          RESET                ;SHOULD HAVE NO EFFECT
3804                                     ;NO EFFECT
3805 016324 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3806 016326 012713 000346  MOV     #346,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3807 016332 000000          HLT                                     ;TRACE TRAP FAILED
3808 016334 000240          LOOP                                ;REPLACE THIS INSTRUCTION BY A
3809                                     ;"JMP #R0" (000110) TO GET A
3810                                     ;SCOPE LOOP AND HIT CONTINUE
3811 016336 005037 177776  RESET2: CLR     STATUS
3812 016342 005037 000016  CLR     16              ;CLEAR TRACK
3813 016346 010700          SCOPE                                ;TRACE STATUS
3814
3815                                     ;*****
3816                                     ;*TFST 141 TEST THAT WHEN ITY INTERRUPTS IT POPS NEW STATUS
3817                                     ;*****
3818 016350 005237 000304  TST141: INC     #1$TESTN
3819 016354 000005          HRESET
3820 016356 012706 021774  MOV     #BUFF,%6        ;SET UP STACK
3821 016362 012737 016406 000064  MOV     #TTY3,%6        ;INTERRUPT VECTOR
3822 016370 005037 177776  CLR     STATUS          ;DPOP PROCESSOR PRIORITY
3823 016374 012737 000357 000066  MOV     #357,%6        ;HIGH PRIORITY ON INTERRUPT
3824 016402 005137 177564  COM     TTCSR          ;SHOULD SET INTERRUPT ENABLE & INTERRUPT
3825
3826 016406 013727 177776  ITY3:  MOV     STATUS,(PC)+
3827 016412 000000          .WORD
3828 016414 022737 000357 016412  CMP     #357,-2
3829 016422 001405          BEQ     1$
3830
3831 016424 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3832 016426 012713 000347  MOV     #347,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3833 016432 000000          HLT                                     ;INTERRUPT DID NOT POP CORRECT STATUS
3834 016434 000240          LOOP                                ;REPLACE THIS INSTRUCTION BY A
3835                                     ;"JMP #R0" (000110) TO GET A
3836                                     ;SCOPE LOOP AND HIT CONTINUE
3837 016436 000005          1$:  HRESET
3838 016440 010700          SCOPE                                ;CLR INTERRUPT ENABLE
3839 016442 012706 021774  MOV     #BUFF,%6        ;STACK SET UP
3840 016446 012737 016472 000064  MOV     #TTY4,%6        ;INTERRUPT VECTOR
3841 016454 005037 000066  CLR     %6              ;CLR NEW STATUS
3842 016460 012737 000157 177776  MOV     #157,STATUS    ;PROCESSOR STATUS
3843 016466 005137 177564  COM     TTCSR          ;SET INTERRUPT ENABLE
3844 016472 013727 177776  ITY4:  MOV     STATUS,(PC)+
3845 016476 000000          .WORD
3846 016500 005737 016476  TST     -2
3847 016504 001405          BEQ     1$
3848
3849 016506 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3850 016510 012713 000350  MOV     #350,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3851 016514 000000          HLT                                     ;INTERRUPT DID NOT POP CORRECT STATUS
3852 016516 000240          LOOP                                ;REPLACE THIS INSTRUCTION BY A
3853                                     ;"JMP #R0" (000110) TO GET A
3854                                     ;SCOPE LOOP AND HIT CONTINUE
3855 016520 005037 177564  1$:  CLR     TTCSR
3856 016524 010700          SCOPE
3857                                     ;*****

```

```

3859                                     ;*TEST 142 CONTIGUOUS MEMORY ADDRESS TEST
3860 016526 005237 000304  TST142: INC     #1$TESTN
3861                                     ;THIS TEST CHECKS THAT ALL MEMORY (UP TO 28K) IS CONTIGUOUS
3862 016532 012706 021774  MOV     #BUFF,%6        ;SET STACK POINTER
3863 016536 012737 016552 000004  MOV     #MEMEND,%04    ;SET TIME OUT TRAP VECTOR
3864 016544 005000          CLR     %0              ;SET STARTING ADDRESS FOR TEST
3865 016546 005720          TST     (0)+           ;BEGIN
3866 016550 000776          BR      -2              ;LOOP UNTIL TIMEOUT OCCURS
3867 016552 022626          MEMEND: CMP     (6)+,(6)+ ;RESTORE STACK POINTER
3868 016554 022700 160002  CMP     #160002,%0    ;AT END OF 28K MEMORY??
3869 016560 001406          BEQ     MEMEX          ;YES - EXIT TEST
3870 016562 005720          TST     (0)+           ;SHOULD TIME OUT HERE UP TO 28K
3871
3872 016564 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
3873 016566 012713 000351  MOV     #351,(R3)      ;MOVE TO MAIL BOX ($FATAL) THE ERROR
3874 016572 000000          HLT                                     ;ERROR - FAILED TO TIME OUT!
3875 016574 000240          LOOP                                ;REPLACE THIS INSTRUCTION BY A
3876                                     ;"JMP #R0" (000110) TO GET A
3877                                     ;SCOPE LOOP AND HIT CONTINUE
3878 016576 012737 000006 000004  MEMEX: MOV     #6,%04
3879 016604 005037 000006  CLR     #6
3880 016610 010700          SCOPE

```

```
3P01 ;SPECIAL CASE OF ODD, EVEN ,BYTE AND REGISTER 6
3P02
3P03 ;*****
3P04 ;*TEST 141 TEST AUTO INCREMENT & DECREMENT OF R6 - WORD & BYTE
3P05 ;*****
3P06 016612 045237 000104 TST141: INC R6,TESTN
3P07
3P08 016616 012706 021774 ADR6: MOV #RUFF,%6 ;SET UP THE STACK
3P09 016627 112627 MOVR (6)+,(PC)+ ;SIX SHOULD INCREMENT BY TWO
3P10 016624 000000 ,WORD
3P11 016626 020627 021776 CMP %6,#BUFF+2
3P12 016632 001405 BEQ 15
3P13
3P14 016634 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3P15 016636 012713 000352 MOV #352,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
3P16 016642 000000 HLT ;R6 DID NOT AUTO INCREMENT BY TWO
3P17 016644 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3P18 ;"JMP #R0" (000110) TO GET A
3P19 ;SCOPE LOOP AND HIT CONTINUE
3P20 18: SCOPE
3P21
3P22 016650 012706 001000 MOV #1000,%6
3P23 016654 114627 MOVR -(6),(PC)+ ;SHOULD DECREMENT BY TWO
3P24 016656 000000 ,WORD
3P25 016660 020627 000776 CMP %6,#776
3P26 016664 001405 BEQ 25
3P27
3P28 016666 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3P29 016670 012713 000353 MOV #353,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
3P30 016674 000000 HLT ;R6 DID NOT AUTO DECREMENT BY 2
3P31 016676 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3P32 ;"JMP #R0" (000110) TO GET A
3P33 ;SCOPE LOOP AND HIT CONTINUE
3P34 25: SCOPE
3P35
3P36 016700 010700 35: SCOPE
3P37
3P38 016702 012706 021774 MOV #RUFF,%6
3P39 016706 112626 MOVR (6)+,(6)+ ;DOUBLE AUTO INCREMENT OF R6
3P40 016710 020627 022000 CMP %6,#BUFF+4
3P41 016714 001405 BEQ 35
3P42
3P43 016716 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3P44 016720 012713 000354 MOV #354,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
3P45 016724 000000 HLT ;WRONG AUTO INCREMENT OF R6
3P46 016726 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3P47 ;"JMP #R0" (000110) TO GET A
3P48 ;SCOPE LOOP AND HIT CONTINUE
3P49 35: SCOPE
3P50
3P51 016730 010700 35: SCOPE
3P52
3P53 016732 012706 021774 MOV #RUFF,%6
3P54 016736 005004 CLR %4
3P55 016740 122624 CMPL (6)+,(4)+ ;TEST INCREMENT OF R6
3P56 016742 020627 021776 CMP %6,#BUFF+2
3P57 016746 001405 BEQ 45
3P58
3P59 016750 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3P60 016752 012713 000355 MOV #355,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
```

```
3P61 016756 000000 HLT ;WRONG INCREMENT OF R6
3P62 016760 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3P63 ;"JMP #R0" (000110) TO GET A
3P64 ;SCOPE LOOP AND HIT CONTINUE
3P65 45: SCOPE
3P66
3P67 016764 012706 021774 MOV #RUFF,%6
3P68 016770 005004 CLR %4
3P69 016772 122426 CMPL (4)+,(6)+ ;TEST INCREMENT OF R6
3P70 016774 020627 021776 CMP %6,#BUFF+2
3P71 017000 001405 BEQ 55
3P72
3P73 017002 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3P74 017004 012713 000356 MOV #356,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
3P75 017010 000000 HLT ;WRONG INCREMENT OF R6
3P76 017012 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3P77 ;"JMP #R0" (000110) TO GET A
3P78 ;SCOPE LOOP AND HIT CONTINUE
3P79 55: SCOPE
3P80
3P81 017016 012706 021774 MOV #RUFF,%6
3P82 017022 005004 CLR %4
3P83 017024 122624 CMPL (6)+,(4)+ ;TEST INCREMENT OF R4
3P84 017026 020427 000001 CMP %4,#1
3P85 017032 001405 BEQ 65
3P86
3P87 017034 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3P88 017036 012713 000357 MOV #357,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
3P89 017042 000000 HLT ;WRONG INCREMENT OF R4
3P90 017044 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3P91 ;"JMP #R0" (000110) TO GET A
3P92 ;SCOPE LOOP AND HIT CONTINUE
3P93 65: SCOPE
3P94
3P95 017046 010700 75: SCOPE
3P96
3P97 017048 012706 021774 MOV #RUFF,%6
3P98 017054 005004 CLR %4
3P99 017056 122426 CMPL (4)+,-(6) ;TEST DECREMENT OF R6
3P100 017060 020627 021772 CMP %6,#BUFF-2
3P101 017064 001405 BEQ 75
3P102
3P103 017066 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3P104 017070 012713 000360 MOV #360,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
3P105 017074 000000 HLT ;WRONG INCREMENT OF R6
3P106 017076 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
3P107 ;"JMP #R0" (000110) TO GET A
3P108 ;SCOPE LOOP AND HIT CONTINUE
3P109 75: SCOPE
3P110
3P111 017100 010700 75: SCOPE
3P112
3P113 017102 012706 021774 MOV #RUFF,%6
3P114 017106 005004 CLR %4
3P115 017110 122426 CMPL (4)+,(6)+ ;TEST INCREMENT OF R4
3P116 017112 020427 000001 CMP %4,#1
3P117 017116 001405 BEQ 85
3P118
3P119 017120 005212 INC (R2) ;SET MESSAGE TYPE (MSGTY) TO FATAL ERROR
3P120 017122 012713 000361 MOV #361,(R3) ;MOVE TO MAIL BOX (SFATAL) THE ERROR
3P121 017126 000000 HLT ;WRONG INCREMENT OF R4
```



```

3993 017130 000210 LOOP
3994
3995 ;REPLACE THIS INSTRUCTION BY A
3996 017132 010700 RS: SCOPE ;"JMP 000" (000110) TO GET A
3997 ;SCOPE LOOP AND HIT CONTINUE
3998 017134 012700 001000 MOV #1070,06
3999 017140 174627 CMPR -(6),(PC)+ ;TEST DECREMENT OF R6
4000 017142 027000 000770 CMP #770,06
4001 017146 001405 BFO 98
4002
4003 017150 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4004 017152 012713 000362 MOV #362,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4005 017156 000000 HLT ;WHONG DECREMENT OF R6
4006 017160 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
4007 ;"JMP 000" (000110) TO GET A
4008 ;SCOPE LOOP AND HIT CONTINUE
4009 017162 010700 98: SCOPE
4010
4011 017164 010700 SCOPE ;5 JAN 71
4012 017166 012700 021774 MOV #00FF,06 ;INITIAL POSITION OF R6
4013 017172 012737 177777 021772 MOV #-1,00FF-2 ;HIGH BYTE WILL BE DATA
4014 017200 012737 025252 001000 MOV #25252,K1 ;INITIAL SET UP OF DESTINATION
4015 017206 114637 001000 MOV #-(6),K1+1 ;HIGH BYTE OF BUFF-2, TO LOW BYTE OF K1
4016 017212 027337 177652 001000 MOV #177652,K1 ;MOV TO OPD ADDRESS
4017 017220 001405 BFO 108 ;DID MOVW WORK CORRECTLY
4018
4019 017222 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4020 017224 012713 000363 MOV #363,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4021 017230 000000 HLT ;MOVW -(6), ODD: FAILED
4022 017232 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
4023 ;"JMP 000" (000110) TO GET A
4024 ;SCOPE LOOP AND HIT CONTINUE
4025 017234 010700 108: SCOPE
4026
4027 017236 012700 021774 MOV #00FF,06 ;INITIAL POSITION OF R6
4028 017242 012737 177777 021772 MOV #-1,00FF-2 ;HIGH BYTE SOURCE DATA
4029 017250 012737 125252 001000 MOV #125252,K1 ;INITIAL SET UP OF DESTINATION
4030 017256 114637 001000 MOV #-(6),K1 ;HIGH BYTE OF BUFF-2, TO LOW BYTE OF K1
4031 017262 027337 125377 001000 CMP #125377,K1 ;TEST RESULTS
4032 017270 001405 BFO 118
4033
4034 017272 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4035 017274 012713 000364 MOV #364,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4036 017300 000000 HLT ;MOVW -(6), EVEN, FAILED
4037 017302 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
4038 ;"JMP 000" (000110) TO GET A
4039 ;SCOPE LOOP AND HIT CONTINUE
4040 017304 010700 118: SCOPE

```

```

4041 ;*****
4042 ;TEST 144 TEST TRANSFER OF .BYTE USING R6
4043 ;*****
4044 017306 005237 000304 IST144: INC #005237
4045 017312 012737 123456 001010 BXFP: MOV #123456,K5
4046 017320 012737 050505 001000 MOV #050505,K1
4047 017326 012705 001000 MOV #K1,05 ;R5=(050505)K1
4048 017332 012706 001010 MOV #K5,06 ;R6(123456)R5
4049 017336 114625 MOV #-(6),(5)+ ;LOW .BYTE OF R6 TO R5
4050 017340 027337 050456 001000 CMP #050456,K1
4051 017346 001405 BEQ 18
4052
4053 017350 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4054 017352 012713 000365 MOV #365,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4055 017356 000000 HLT ;FALSE TRANSFER OF .BYTE
4056 017360 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
4057 ;"JMP 000" (000110) TO GET A
4058 ;SCOPE LOOP AND HIT CONTINUE
4059 017362 010700 18: SCOPE
4060
4061 017364 012737 123456 001010 MOV #123456,K5
4062 017372 012737 050505 001000 MOV #050505,K1
4063 017400 012705 001000 MOV #K1,05 ;R5(050505)K1
4064 017404 012706 001010 MOV #K6,06 ;R6(123456)K5
4065 017410 114625 MOV #-(6),(5)+ ;LOW .BYTE OF R6 TO R5 (DECREMENT)
4066 017412 027372 001000 050456 CMP #1,050456
4067 017420 001405 BFO 28
4068
4069 017422 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4070 017424 012713 000366 MOV #366,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4071 017430 000000 HLT ;FALSE R6 .BYTE TRANSFER
4072 017432 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
4073 ;"JMP 000" (000110) TO GET A
4074 ;SCOPE LOOP AND HIT CONTINUE
4075 017434 010700 28: SCOPE
4076
4077 017436 012737 123456 001000 MOV #123456,K1
4078 017444 012737 050505 001010 MOV #050505,K5
4079 017452 012705 001000 MOV #K1,05 ;(123456)
4080 017456 012706 001010 MOV #K5,06 ;(050505)
4081 017462 112526 MOV #-(5),(6)+ ;LOW OF R5 TO LOW OF R6
4082 017464 027372 050456 001010 CMP #050456,K5
4083 017472 001405 BEQ 38
4084
4085 017474 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4086 017476 012713 000367 MOV #367,(R3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4087 017502 000000 HLT ;FALSE R6 .BYTE TRANSFER
4088 017504 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
4089 ;"JMP 000" (000110) TO GET A
4090 ;SCOPE LOOP AND HIT CONTINUE
4091 017506 010700 38: SCOPE
4092
4093 017510 012737 123456 001000 MOV #123456,K1
4094 017516 012737 050505 001010 MOV #050505,K5
4095 017524 012705 001001 MOV #K1+1,05 ;123456
4096 017530 012706 001010 MOV #K5,06 ;050505

```

```

4097 017534 112526          MOVB (5)+,(6)+      ;HIGH OF R5 TO LOW OF R6
4098 017536 023727 001010 050647  CMP      K5,#050647
4099 017544 001405          BEQ
4100
4101 017546 005212          INC      (R2)
4102 017550 012713 000370  MOV      #370,(R3)      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4103 017554 000000          HLT
4104 017556 000240          LOOP
4105
4106
4107 017560 010700          48:  SCOPE
4108
4109 017562 012737 123456 001000  MOV      #123456,K1
4110 017570 012737 050505 001010  MOV      #050505,K5
4111 017576 012705 001001  MOV      #K1+1,#5
4112 017602 012706 001010  MOV      #K5,#6
4113 017606 112625  MOVBR   (6)+,(5)+
4114 017610 022737 042456 001000  CMP      #042456,K1
4115 017616 001405          BEQ      55
4116
4117 017620 005212          INC      (R2)
4118 017622 012713 000371  MOV      #371,(R3)      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4119 017626 000000          HLT
4120 017630 000240          LOOP
4121
4122
4123 017632 010700          54:  SCOPE
4124
4125
4126
4127
4128 017634 045237 000304  TSTI45: INC      #05237N
4129 017640 123737 001014 001015  BYTOP:  CMPB   K7,K7+1
4130 017646 001405          BEQ      18
4131
4132 017650 005212          INC      (R2)
4133 017652 012713 000372  MOV      #372,(R3)      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4134 017656 000000          HLT
4135 017660 000240          LOOP
4136
4137
4138 017662 010700          18:  SCOPE
4139
4140 017664 123737 001015 001014  CMPR   K7+1,K7
4141 017672 001405          BEQ      28
4142
4143 017674 005212          INC      (R2)
4144 017676 012713 000373  MOV      #373,(R3)      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4145 017702 000000          HLT
4146 017704 000240          LOOP
4147
4148
4149 017706 010700          28:  SCOPE
4150
4151 017710 123737 001017 001014  CMPR   K10+1,K7
4152 017716 001405          BEQ      38
;SAME ,WORD LOW TO HIGH
;COMPARE ODD TO ,EVEN SAME ,WORD
;DIFFERENT ,WORDS

```

```

4153
4154 017720 005212          INC      (R2)
4155 017722 012713 000374  MOV      #374,(R3)      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4156 017726 000000          HLT
4157 017730 000240          LOOP
4158
4159
4160 017732 010700          38:  SCOPE
4161
4162 017734 123737 001016 001012  CMPR   K10,K6
4163 017742 001405          BEQ      48
4164
4165 017744 005212          INC      (R2)
4166 017746 012713 000375  MOV      #375,(R3)      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4167 017752 000000          HLT
4168 017754 000240          LOOP
4169
4170
4171 017756 010700          48:  SCOPE
4172
4173 017760 123737 001015 001017  CMPR   K7+1,K10+1
4174 017766 001405          BEQ      58
4175
4176 017770 005212          INC      (R2)
4177 017772 012713 000376  MOV      #376,(R3)      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4178 017776 000000          HLT
4179 020000 000240          LOOP
4180
4181
4182 020002 010700          58:  SCOPE
4183
4184 020004 123737 001016 001017  CMPR   K10,K10+1
4185 020012 001005          BNE      68
4186
4187 020014 005212          INC      (R2)
4188 020016 012713 000377  MOV      #377,(R3)      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4189 020022 000000          HLT
4190 020024 000240          LOOP
4191
4192
4193 020026 010700          68:  SCOPE
4194
4195 020030 123737 001017 001016  CMPR   K10+1,K10
4196 020036 001005          BNE      78
4197
4198 020040 005212          INC      (R2)
4199 020042 012713 000400  MOV      #400,(R3)      ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4200 020046 000000          HLT
4201 020050 000240          LOOP
4202
4203
4204 020052 010700          78:  SCOPE
4205
4206 020054 123737 001016 001015  CMPB   K10,K7+1
4207 020062 001005          RNE      88
4208

```

```
4209 020064 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4210 020066 012713 000401    MOV      #401,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4211 020072 000000          HLT                     ;,EVEN TO ODD FAILED
4212 020074 000240          LOOP                    ;REPLACE THIS INSTRUCTION BY A
4213                                     ;"JMP #R0" (000110) TO GET A
4214                                     ;SCOPE LOOP AND HIT CONTINUE
4215 020076 010700          86:   SCOPE
4216
```

```
4217
4218
4219                                     ;*****
4220                                     ;*TEST 146 TEST THAT MOV B $R MOVES ONLY THE LSH OF THE REGISTER*
4221                                     ;*****
4221 020100 005237 000304          IST146: INC      #0,TESTM
4222 020104 012737 177777 001020  RLOBYT: MOV      #-1,K11
4223 020112 012700 000125          MOV      #125,#0
4224 020116 110037 001020          MOV B   $0,K11          ;LOAD R0
4225 020122 023727 001020 177525  CMP      K11,#177525    ;WAS ONLY LSH MOVED?
4226 020130 001405          BEQ      18
4227
4228 020132 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4229 020134 012713 000402    MOV      #402,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4230 020140 000000          HLT                     ;ERROR! MOV B $R FAILED
4231 020142 000240          LOOP                    ;REPLACE THIS INSTRUCTION BY A
4232                                     ;"JMP #R0" (000110) TO GET A
4233                                     ;SCOPE LOOP AND HIT CONTINUE
4234 020144 010700          18:   SCOPE
4235
4236 020146 012700 001020          MOV      #K11,$0
4237 020152 010037 001020          MOV      $0,K11
4238 020156 110020          MOVR    $0,(0)+
4239 020160 022737 001021 001020  CMP      #K11+1,K11
4240 020166 001405          BEQ      28
4241
4242 020170 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4243 020172 012713 000403    MOV      #403,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4244 020176 000000          HLT                     ;REPLACE THIS INSTRUCTION BY A
4245 020200 000240          LOOP                    ;"JMP #R0" (000110) TO GET A
4246                                     ;SCOPE LOOP AND HIT CONTINUE
4247
4248 020202 010700          28:   SCOPE
4249
4250 020204 012706 001020          MOV      #K11,$6
4251 020210 010637 001020          MOV      $6,K11
4252 020214 110626          MOVR    $6,(6)+
4253 020216 023727 001020 001022  CMP      K11,#K11+2
4254 020224 001405          BEQ      38
4255
4256 020226 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4257 020230 012713 000404    MOV      #404,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4258 020234 000000          HLT                     ;FAILED MOVR $6,(6)+
4259 020236 000240          LOOP                    ;REPLACE THIS INSTRUCTION BY A
4260                                     ;"JMP #R0" (000110) TO GET A
4261                                     ;SCOPE LOOP AND HIT CONTINUE
4262 020240 010700          38:   SCOPE
4263
4264 020242 012706 001020          MOV      #K11,$6
4265 020246 010626          MOV      $6,(6)+
4266 020250 023727 001020 001022  CMP      K11,#K11+2
4267 020256 001405          BEQ      48
4268
4269 020260 005212          INC      (R2)          ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4270 020262 012713 000405    MOV      #405,(R3)     ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4271 020266 000000          HLT                     ;FAILED MOV $6,(6)+
4272 020270 000240          LOOP                    ;REPLACE THIS INSTRUCTION BY A
```



```

4385 020570 000000      HLT                ;V NOT SET
4386 020572 000240      LOOP              ;REPLACE THIS INSTRUCTION BY A
4387                                ;"JMP BR0" (000110) TO GET A
4388                                ;SCOPE LOOP AND HIT CONTINUE
4389 020574 100405      158: RMI        168
4390                                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4391 020576 005212      INC (R2)         ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4392 020600 012713 000421  MOV #421,(R3)    ;N NOT SET
4393 020604 000000      HLT              ;REPLACE THIS INSTRUCTION BY A
4394 020606 000240      LOOP              ;"JMP BR0" (000110) TO GET A
4395                                ;SCOPE LOOP AND HIT CONTINUE
4396
4397 020610 100405      168: RCS        178
4398                                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4399 020612 005212      INC (R2)         ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4400 020614 012713 000422  MOV #422,(R3)    ;C NOT SET
4401 020620 000000      HLT              ;REPLACF THIS INSTRUCTION BY A
4402 020622 000240      LOOP              ;"JMP BR0" (000110) TO GET A
4403                                ;SCOPE LOOP AND HIT CONTINUE
4404
4405
4406 020624 010700      178: SCOPE
4407 020626 005037 021776  CLR #0           ;ZEPU TO STATUS
4408 020632 012706 021774  MOV #0,00000000 ;
4409 020636 012737 020650 021774  MOV #0,00000000 ;
4410 020644 000277      SCC              ;SET CONDITION CODES
4411 020646 000000      BIT              ;RETURN
4412 020650 000005      BNE 198
4413
4414 020652 005212      INC (R2)         ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4415 020654 012713 000423  MOV #423,(R3)    ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4416 020660 000000      HLT              ;2 NOT CLEARED
4417 020662 000240      LOOP              ;REPLACE THIS INSTRUCTION BY A
4418                                ;"JMP BR0" (000110) TO GET A
4419                                ;SCOPE LOOP AND HIT CONTINUE
4420 020664 100405      188: RVC        198
4421                                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4422 020666 005212      INC (R2)         ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4423 020670 012713 000424  MOV #424,(R3)    ;V NOT CLEARED
4424 020674 000000      HLT              ;REPLACE THIS INSTRUCTION BY A
4425 020676 000240      LOOP              ;"JMP BR0" (000110) TO GET A
4426                                ;SCOPE LOOP AND HIT CONTINUE
4427
4428 020700 100405      198: RPL        208
4429                                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4430 020702 005212      INC (R2)         ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4431 020704 012713 000425  MOV #425,(R3)    ;N NOT CLEARED
4432 020710 000000      HLT              ;REPLACE THIS INSTRUCTION BY A
4433 020712 000240      LOOP              ;"JMP BR0" (000110) TO GET A
4434                                ;SCOPE LOOP AND HIT CONTINUE
4435
4436 020714 100405      208: BCC        218
4437                                ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4438 020716 005212      INC (R2)         ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4439 020720 012713 000426  MOV #426,(R3)    ;C NOT CLEARED
4440 020724 000000      HLT

```

```

4441 020726 000240      LOOP              ;REPLACE THIS INSTRUCTION BY A
4442                                ;"JMP BR0" (000110) TO GET A
4443                                ;SCOPE LOOP AND HIT CONTINUE
4444 020730 010700      218: SCOPE
4445 020732      RLOAY1:

```

T146 TEST THAT MOVES ARE MOVES ONLY THE LSH OF THE REGISTER

```
4446  
4447  
4448  
4449  
4450 020732 005237 000304  
4451 020736 012700 137777  
4452 020742 076600  
4453 020744 000352  
4454 020746 012700 021344  
4455 020752 012002  
4456 020754 012001  
4457 020756 020237 021400  
4458 020762 001453  
4459 020764 010237 021402  
4460 020770 012737 021012  
4461 020776 012706 021774  
4462 021002 025037 177776  
4463 021006 000137 021402  
4464  
4465  
4466  
4467 021012 020627 021770  
4468 021016 001404  
4469  
4470 021020 005212  
4471 021022 012713 000427  
4472 021026 000000  
4473 021030 023727 021770 021404  
4474 021036 001405  
4475  
4476 021040 005212  
4477 021042 012713 000430  
4478 021046 000000  
4479 021050 000240  
4480  
4481  
4482 021052 005737 021772  
4483 021056 001405  
4484  
4485 021060 005212  
4486 021062 012713 000431  
4487 021066 000000  
4488 021070 000240  
4489  
4490  
4491 021072 005237 021402  
4492 021076 005202  
4493 021100 023701 021402  
4494 021104 001722  
4495 021106 000137 020764  
4496  
4497 021112 010700  
4498  
4499  
4500  
4501
```

;TEST 147 TEST THAT ALL RESERVED INSTRUCTIONS TRAP

TST147: INC #02FF51N
MOV #137777,PC ;DISABLE WCS IF PRESENT
MED ;BY DOING "INIT", XFC SHOULD
352 ;TRAP AA A RESERVED INSTR.
MOV #TABLF,TAR ;TABLE POINTER
GIN1: MOV (TAB)+,FIRST ;FIRST OR CURRENT INSTRUCTION
(TAB)+,LAST ;LAST INSTRUCTION OF GROUP
CMP FIRST,FINISH ;TESTED ALL
BEQ #FT3A ;GO TO MULT TRAPS TEST IF FIN.
GIN2: MOV FIRST,INST ;SET UP INST
MOV #RET,RESVFC ;LOAD RESERVED INST. TRAP VECTOR
MOV #BUFF,%6 ;SET UP STACK POINTER
CLR CC ;CLEAR PRIORITY
JMP INST ;EXECUTE RESERVED INSTRUCTION
;TRAP SHOULD OCCUR

;TRAPPING SHOULD SEND YOU HERE
RET: CMP #6,#00FF-4 ;TEST DECREMENT OF %6
REQ RET1

PET1: CMP #BUFF-4,#INST+2
BEQ RET2 ;WRONG DECREMENT
;LOC OF INST UNINCREMENTED

RET2: TST #430,(R3) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
REQ ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
;INST INC ON TRAP
;REPLACE THIS INSTRUCTION BY A
;"JMP #00" (000110) TO GET A
;SCOPE LOOP AND HIT CONTINUE

RET3: INC INST ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
INC FIRST ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
CMP INST,LAST ;CONDITION CODES SET ON TRAP
HFO GIN1 ;REPLACE THIS INSTRUCTION BY A
JMP GIN2 ;"JMP #00" (000110) TO GET A
;SCOPE LOOP AND HIT CONTINUE

RET3A: SCOPE ;END OF INSTRUCTION GROUP

;TEST 150 CHECK MULTIPLE TRAP CONDITIONS

T150 CHECK MULTIPLE TRAP CONDITIONS

```
4502 021114 005237 000304  
4503  
4504  
4505 021120 012706 021774  
4506 021124 012737 021176 000014  
4507 021132 012737 021222 000004  
4508 021140 012737 021205 000030  
4509 021146 005037 000032  
4510 021152 005000  
4511 021154 012746 000020  
4512 021160 012746 021166  
4513 021164 000006  
4514 021166 104000  
4515  
4516 021170 005212  
4517 021172 012713 000432  
4518 021176 000000  
4519 021200 000240  
4520  
4521  
4522 021202 000455  
4523 021204 000240  
4524  
4525 021206 005212  
4526 021210 012713 000433  
4527 021214 000000  
4528 021216 000240  
4529  
4530  
4531 021220 000446  
4532 021222 012737 021246 021764  
4533 021230 000006  
4534  
4535 021232 005212  
4536 021234 012713 000434  
4537 021240 000000  
4538 021242 000240  
4539  
4540  
4541 021244 000434  
4542 021246 012737 021276 021770  
4543 021254 012737 021316 000014  
4544 021262 000006  
4545  
4546 021264 005212  
4547 021266 012713 000435  
4548 021272 000000  
4549 021274 000240  
4550  
4551  
4552 021276 012700 000001  
4553  
4554 021302 005212  
4555 021304 012713 000436  
4556 021310 000000  
4557 021312 000240
```

TST150: INC #02FF51N
;THE FOLLOWING TESTS ARE WRITTEN TO CHECK MULTIPLE TRAP CONDITIONS
;TO TEST THAT TRAPS ARE PROCESSED PROPERLY.
SINS1: MOV #BUFF,%6 ;SET UP STACK POINTER
MOV #SINS0,TBITVEC ;LOAD "T" BIT VECTOR
MOV #SINSE,4 ;LOAD ERROR VECTOR
MOV #SINSF+1,EMTVEC ;LOAD EMT VECTOR (ODD)
CLR EMTVEC+2
CLR #0 ;PRE SET RN
MOV #20,-(6) ;SET UP
MOV #SINSDD,-(6) ;TO SET "T" BIT
PTI ;SET "T" BIT
SINSDD: EMT ;NO EMT PICK UP ODD PC

SINS1: MOV (P2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
MOV #432,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
HLT ;ERROR! "T" TRAP OCCURRED BEFORE ERROR
LOOP ;REPLACE THIS INSTRUCTION BY A
;"JMP #00" (000110) TO GET A
;SCOPE LOOP AND HIT CONTINUE
;GO TO SCOPE

SINSF: BP SINS1
NOP ;THE EMT TRIES TO COME HERE

SINS1: INC (P2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
MOV #433,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
HLT ;ERROR! NO ERROR TRAP ON ODD EMT VECTOR
LOOP ;REPLACE THIS INSTRUCTION BY A
;"JMP #00" (000110) TO GET A
;SCOPE LOOP AND HIT CONTINUE
;GO TO SCOPE

SINSE: BP SINS1
MOV #SINS0,BUFF-10 ;CORRECT BAD EMT VECTOR ON THE STACK
RTI ;GO TO SINSG (SERVICE EMT)

SINS1: INC (P2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
MOV #434,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
HLT ;DIDN'T GO
LOOP ;REPLACE THIS INSTRUCTION BY A
;"JMP #00" (000110) TO GET A
;SCOPE LOOP AND HIT CONTINUE
;GO TO SCOPE

SINSG: BP SINS1
MOV #SINS0,BUFF-4 ;RESTORE RETURN FROM EMT (ON STACK)
MOV #SINSJ,TBITVEC ;CHANGE "T" HIT VECTOR
RTI ;GO TO SINSH

SINSH: MOV #1,%0 ;LOAD INDICATOR

SINS1: INC (P2) ;SET MESSAGE TYPE (\$MSGTY) TO FATAL ERROR
MOV #436,(R3) ;MOVE TO MAIL BOX (\$FATAL) THE ERROR
HLT ;ERROR! SHOULD HAVE "T" TRAPPED
LOOP ;REPLACE THIS INSTRUCTION BY A

```

4558 ;"JMP R00" (000110) TO GET A
4559 ;SCOPE LOOP AND HIT CONTINUE
4560 #21314 000410 SINSJ: BR SINSI ;GO TO SCOPE
4561 #21316 022700 000001 CMP #1,00 ;WAS INST. AT SINSH EXECUTED?
4562 #21322 001405 REQ SINSI
4563
4564 #21324 005212 INC (R2) ;SET MESSAGE TYPE ($MSGTY) TO FATAL ERROR
4565 #21326 012713 000437 MOV #437,(P3) ;MOVE TO MAIL BOX ($FATAL) THE ERROR
4566 #21332 000000 HLT ;ERROR! INST AT SINSH WAS NOT EXECUTED
4567 #21334 000240 LOOP ;REPLACE THIS INSTRUCTION BY A
4568 ;"JMP R00" (000110) TO GET A
4569 ;SCOPE LOOP AND HIT CONTINUE
4570 #21336 010700 SINSI: SCOPE
4571
4572 #21340 000137 021420 JMP FNDOT ;END OF PROGRAM, GOT TO
4573 ;END OF TEST LINKAGE
4574
4575 ;XFC= 076XXX EXCEPT 0766XX
4576 #21344 000007 TABLE: 7
4577 ;MED=076600
4578 #21346 000077 77
4579 #21350 000210 210
4580 #21352 000227 227
4581 #21354 007000 7000
4582 #21356 007777 7777
4583 #21360 075040 75040
4584 #21362 076577 76577
4585
4586 #21364 076601 76601
4587 #21366 076677 76677
4588
4589 #21370 106400 106400
4590 #21372 106477 106477
4591
4592 #21374 106700 106700
4593 #21376 107777 107777
4594 #21400 021400 FINISH:
4595 #21402 000000 INST: HALT ;END FLAG
4596 #21404 000000 HALT ;WILL CONTAIN RESERVED INST
4597 #21406 000000 HALT ;SHOULD TRAP TO LOC 10
4598 #21410 000000 HALT ;LOC 10 SHOULD SEND YOU TO
4599 #21412 000000 HALT ;RET
4600 #21414 000000 TEMP: 0
4601 #21420 071420 .,+.2
4602
4603 ;END OF PASS LINKAGE
4604
4605
4606 #21420 005237 000306 ENDOT: INC ##$PASS
4607 #21424 105237 021546 INCR PASSPT ;SHOULD PRINT THIS PASS?
4608 #21430 001027 BNE ACT ;NO
4609 #21432 132737 000040 #00321 BTR #40,$ENVM ;WILL APT ALLOW PRINTING?
4610 #21440 001023 BNE ACT ;NO
4611 #21442 023727 000042 #21520 CMP #42,$ENDAD
4612 #21450 001417 BEQ ACT
4613 #21452 012700 021550 MOV #MSG,R0 ;GET MSG ADDR.

```

```

4614 #21456 105737 177564 WAIT: TSTB #TPS ;TTY READY
4615 #21462 100375 BPL WAIT ;NO WAIT
4616 #21464 112037 177566 MOVB (R0)+,##TPB ;PRINT CHARACTER
4617 #21470 001372 BNE WAIT ;NEXT IF NOT DONE.
4618 #21472 105737 177564 WAIT1: TSTR #TPS
4619 #21476 100375 BPL WAIT1
4620 #21500 000005 RFSET
4621 #21502 012737 177761 021546 MOV #177761,PASSPT ;DO IT ABOUT 15 DECIMAL TIMES
4622 #21510 013700 000042 ACT: MOV ##42,R0 ;CHECK ACT
4623 #21514 001405 BEQ GOAGIN ;KEEP GOING
4624 #21516 000005 RESET
4625 #21520 004710 $ENDAD: JSR PC,(R0) ;ACT HOOKS
4626 #21522 000240 NOP
4627 #21524 000240 NOP
4628 #21526 000240 NOP
4629 #21530 012737 000012 000010 GOAGIN: MOV #12,10
4630 #21536 005037 000012 CLR 12
4631 #21542 000137 001110 JMP BFGIN1 ;DO NEXT PASS
4632 #21546 177777 PASSPT: -1
4633 #21550 005015 047105 020104 MSG: .ASCIZ <15><12>/END OF PASS/
4634 #21556 043117 020040 040520
4635 #21564 051523 000
4636 #21570 021570 .EVFN
4637
4638 ;POWER FAIL, POWER UP ROUTINE
4639
4640 #21570 012737 021600 000024 PWRDWN: MOV #PWRUP,24
4641 #21576 000000 HALT
4642
4643 #21600 012737 021570 000024 PWRUP: MOV #PWRDWN,24
4644 #21606 012706 021774 MOV #RUFF,SP
4645 #21612 132737 000040 000321 BTR #40,$ENVM ;WILL APT ALLOW PRINTING?
4646 #21620 001013 BNE PFRFS ;NO
4647 #21622 012700 021654 MOV #MSGPWF,R0 ;GET MSG ADDR.
4648 #21626 105737 177564 PWAIT: TSTB #TPS ;TTY READY
4649 #21632 100375 BPL PWAIT ;NO WAIT
4650 #21634 112037 177566 MOVB (R0)+,##TPB ;PRINT CHARACTER
4651 #21640 001372 BNE PWAIT ;NEXT IF NOT DONE.
4652 #21642 105737 177564 PWAIT1: TSTB #TPS
4653 #21646 100375 BPL PWAIT1
4654 #21650 000137 001102 PFRFS: JMP BFGIN
4655 #21654 005015 047520 042527 MSGPWF: .ASCIZ <15><12>.POWER FAILED!.
4656 #21662 020122 040506 046111
4657 #21670 042105 000041
4658
4659 #21774 000000 .,+.100
4660 #21774 000000 BUFF: 0
4661 #21774 000001 .END

```

ARASE = 000000	310			
ACDW1 = 000000	310			
ACDW2 = 000000	310			
ACPUOP = 000000	310			
ACT = 021510	4600	333	4612	4622*
ADDW0 = 000000	310			
ADDW1 = 000000	310			
ADDW10 = 000000	310			
ADDW11 = 000000	310			
ADDW12 = 000000	310			
ADDW13 = 000000	310			
ADDW14 = 000000	310			
ADDW15 = 000000	310			
ADDW2 = 000000	310			
ADDW3 = 000000	310			
ADDW4 = 000000	310			
ADDW5 = 000000	310			
ADDW6 = 000000	310			
ADDW7 = 000000	310			
ADDW8 = 000000	310			
ADDW9 = 000000	310			
ADEVCT = 000000	310	374		
ADEVN = 000000	310			
AFNV = 000000	310	329		
AFNVN = 000000	310	330		
AFATAL = 000000	310	321		
ATDP6 = 016616	3000*			
AMADR1 = 000000	310			
AMADR2 = 000000	310			
AMADR3 = 000000	310			
AMADR4 = 000000	310			
AMAMS1 = 000000	310			
AMAMS2 = 000000	310			
AMAMS3 = 000000	310			
AMAMS4 = 000000	310			
AMSGAD = 000000	310	326		
AMSGLG = 000000	310	327		
AMSGTY = 000000	310	320		
AMTYP1 = 000000	310			
AMTYP2 = 000000	310			
AMTYP3 = 000000	310			
AMTYP4 = 000000	310			
APASS = 000000	310	323		
APRIOR = 000000	310			
ASWPEG = 000000	310	331		
ATESTN = 000000	310	322		
AUNIT = 000000	310	325		
AUSWR = 000000	310	332		
AVECT1 = 000000	310			
AVECT2 = 000000	310			
BEGIN = 001102	300	397*	4654	
BFGIN1 = 001113	4010	4631		
BFLI = 000240	2860			
BGN2 = 001202	4100	419	421	
BGN2A = 001216	4230	424		
BGN3 = 001224	413	416	425*	

AEPOR	259#	436	454	474	493	511	533	553	575	593	612	631	648	669	687	
	706	730	748	767	787	804	825	844	865	883	902	922	939	960	978	
	996	1024	1038	1057	1077	1094	1115	1134	1157	1175	1194	1214	1231	1252	1271	
	1290	1305	1324	1344	1361	1382	1401	1421	1440	1459	1479	1495	1515	1532	1551	
	1564	1582	1599	1612	1630	1652	1661	1689	1685	1711	1720	1741	1759	1769	1779	
	1803	1812	1822	1847	1865	1881	1899	1913	1933	1943	1967	1971	1988	1992	2008	
	2012	2029	2033	2050	2054	2071	2075	2085	2102	2106	2131	2140	2160	2169	2189	
	2198	2218	2227	2247	2256	2296	2344	2354	2364	2374	2383	2394	2404	2414	2424	
	2434	2444	2454	2464	2474	2484	2494	2504	2514	2524	2534	2544	2574	2583	2593	
	2642	2624	2633	2643	2652	2667	2686	2703	2713	2722	2731	2740	2759	2768	2777	
	2786	2795	2818	2824	2833	2842	2854	2863	2881	2899	2916	2934	2957	2964	2973	
	2988	2997	3006	3030	3037	3046	3060	3069	3078	3101	3108	3117	3131	3140	3163	
	3170	3179	3191	3200	3226	3234	3244	3253	3262	3271	3287	3296	3325	3333	3343	
	3352	3361	3370	3386	3395	3419	3426	3435	3444	3453	3462	3478	3487	3506	3513	
	3530	3537	3543	3561	3568	3574	3591	3600	3619	3626	3632	3655	3680	3704	3714	
	3723	3732	3741	3761	3769	3804	3830	3848	3871	3893	3907	3920	3934	3948	3962	
	3975	3989	4002	4018	4033	4052	4068	4084	4100	4116	4131	4142	4153	4164	4175	
	4186	4197	4208	4227	4241	4255	4268	4280	4288	4296	4304	4317	4325	4333	4341	
	4358	4374	4382	4390	4398	4413	4421	4429	4437	4469	4475	4484	4515	4524	4534	
	4545	4553	4563													
APTFRO	257#	436	454	474	493	511	533	553	575	593	612	631	648	669	687	
	706	730	748	767	787	804	825	844	865	883	902	922	939	960	978	
	996	1020	1030	1057	1077	1094	1115	1134	1157	1175	1194	1214	1231	1252	1271	
	1290	1305	1324	1344	1361	1382	1401	1421	1440	1459	1479	1495	1515	1532	1551	
	1564	1582	1599	1612	1630	1652	1661	1689	1685	1711	1720	1741	1759	1769	1779	
	1803	1812	1822	1847	1865	1881	1899	1913	1933	1943	1967	1971	1988	1992	2008	
	2012	2029	2033	2050	2054	2071	2075	2085	2102	2106	2131	2140	2160	2169	2189	
	2198	2218	2227	2247	2256	2296	2344	2354	2364	2374	2383	2394	2404	2414	2424	
	2434	2444	2454	2464	2474	2484	2494	2504	2514	2524	2534	2544	2574	2583	2593	
	2602	2624	2633	2643	2652	2667	2686	2703	2713	2722	2731	2740	2759	2768	2777	
	2786	2795	2818	2824	2833	2842	2854	2863	2881	2899	2916	2934	2957	2964	2973	
	2988	2997	3006	3030	3037	3046	3060	3069	3078	3101	3108	3117	3131	3140	3163	
	3170	3179	3191	3200	3226	3234	3244	3253	3262	3271	3287	3296	3325	3333	3343	
	3352	3361	3370	3386	3395	3419	3426	3435	3444	3453	3462	3478	3487	3506	3513	
	3530	3537	3543	3561	3568	3574	3591	3600	3619	3626	3632	3655	3680	3704	3714	
	3723	3732	3741	3761	3769	3804	3830	3848	3871	3893	3907	3920	3934	3948	3962	
	3975	3989	4002	4018	4033	4052	4068	4084	4100	4116	4131	4142	4153	4164	4175	
	4186	4197	4208	4227	4241	4255	4268	4280	4288	4296	4304	4317	4325	4333	4341	
	4358	4374	4382	4390	4398	4413	4421	4429	4437	4469	4475	4484	4515	4524	4534	
	4545	4553	4563													
HDR	261#	420	445	465	483	522	567	584	603	621	658	698	722	739	758	
	777	814	857	874	893	912	949	988	1012	1029	1048	1067	1104	1149	1166	
	1185	1204	1241	1282	1315	1334	1371	1413	1431	1450	1469	1504	1542	1573	1591	
	1621	1639	1678	1699	1730	1749	1790	1838	1857	1874	1890	1923	1959	1980	2000	
	2021	2042	2063	2094	2123	2152	2181	2210	2239	2270	2308	2557	2678	2749	2804	
	2872	2890	2907	2925	2943	3016	3088	3150	3210	3309	3404	3496	3523	3553	3583	
	3610	3643	3666	3690	3750	3781	3792	3815	3857	3883	4041	4125	4218	4447	4499	
	SFTTBI	259#	1707	1736	1755	3700	3799									
		258#	307	317	344	346	353									
		264#	428	445	465	483	522	567	584	603	621	658	698	722	739	758
777		814	857	874	893	912	949	988	1012	1029	1048	1067	1104	1149	1166	
1185		1204	1241	1282	1315	1334	1371	1413	1431	1450	1469	1504	1542	1573	1591	
1621		1639	1678	1699	1730	1749	1790	1838	1857	1874	1890	1923	1959	1980	2000	
2021		2042	2063	2094	2123	2152	2181	2210	2239	2270	2308	2557	2678	2749	2804	
2872		2890	2907	2925	2943	3016	3088	3150	3210	3309	3404	3496	3523	3553	3583	
3610		3643	3666	3690	3750	3781	3792	3815	3857	3883	4041	4125	4218	4447	4499	
THDR		261#	420	445	465	483	522	567	584	603	621	658	698	722	739	758
	777	814	857	874	893	912	949	988	1012	1029	1048	1067	1104	1149	1166	
	1185	1204	1241	1282	1315	1334	1371	1413	1431	1450	1469	1504	1542	1573	1591	
	1621	1639	1678	1699	1730	1749	1790	1838	1857	1874	1890	1923	1959	1980	2000	
	2021	2042	2063	2094	2123	2152	2181	2210	2239	2270	2308	2557	2678	2749	2804	
	2872	2890	2907	2925	2943	3016	3088	3150	3210	3309	3404	3496	3523	3553	3583	
	3610	3643	3666	3690	3750	3781	3792	3815	3857	3883	4041	4125	4218	4447	4499	

VTRP0	2115#	2123	2152	2181	2210	2239
.#ACT1	300#	305				
.#APT0	300#	315				
.#APTH	300#	342				

. ABS. #21776 000

% ERRORS DETECTED: 0 HARD 17 SOFT
 DEFAULT GLOBALS GENERATED: 0

DSKZ:DOKDBA,DSKZ:DOKDBA,SEQ/CRF/SOL=DSKZ:DOKDBA.P11
 RUN-TIME: 14 12 1 SECONDS
 RUN-TIME RATIO: 423/28=14.9
 CORE USED: 10K (20 PAGES)