

LSI-11

TRAP TEST (30K)
CVKALAO

AH-F013A-MC
COPYRIGHT © 75-78
FICHE 1 OF 1

DEC 1978
digital
MADE IN USA

TEST #	TEST NAME	TEST DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
1	TEST 1	TEST 1 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
2	TEST 2	TEST 2 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
3	TEST 3	TEST 3 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
4	TEST 4	TEST 4 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
5	TEST 5	TEST 5 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
6	TEST 6	TEST 6 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
7	TEST 7	TEST 7 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
8	TEST 8	TEST 8 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
9	TEST 9	TEST 9 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
10	TEST 10	TEST 10 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
11	TEST 11	TEST 11 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
12	TEST 12	TEST 12 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
13	TEST 13	TEST 13 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
14	TEST 14	TEST 14 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
15	TEST 15	TEST 15 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
16	TEST 16	TEST 16 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
17	TEST 17	TEST 17 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
18	TEST 18	TEST 18 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
19	TEST 19	TEST 19 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
20	TEST 20	TEST 20 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
21	TEST 21	TEST 21 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
22	TEST 22	TEST 22 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
23	TEST 23	TEST 23 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
24	TEST 24	TEST 24 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
25	TEST 25	TEST 25 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
26	TEST 26	TEST 26 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
27	TEST 27	TEST 27 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
28	TEST 28	TEST 28 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
29	TEST 29	TEST 29 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS
30	TEST 30	TEST 30 DESCRIPTION	TEST TYPE	TEST STATUS	TEST RESULT	TEST COMMENTS

.REM !

4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46

IDENTIFICATION

PRODUCT CODE: AC-F012A-MC
PRODUCT NAME: CVKALAO LSI-11 TRAP (30K+FIS)
DATE: AUG. 1978
MAINTAINER: DIAGNOSTIC GROUP

COPYRIGHT (C) 1975,1978 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

THIS SOFTWARE IS FURNISHED TO PURCHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DEC'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DEC.

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION.

DEC ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DEC.

48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73

1. ABSTRACT

THIS PROGRAM IS A COPY OF CVKADB WITH MINOR CHANGES. THE CHANGES
ENABLE THE PROGRAM TO RUN WITH A 30K MEMORY SYSTEM. THE PROGRAM
ALSO DEFAULTS TO RUNNING WITH 'FIS' OPTIONS.
THIS IS A TEST OF ALL OPERATIONS AND INSTRUCTIONS THAT CAUSE
TRAPS, ODDITIES OF REGISTER 6, INTERRUPTS, THE RESET
AND WAIT INSTRUCTIONS.

2. REQUIREMENTS

2.1 EQUIPMENT

LSI-11 STANDARD COMPUTER WITH AN SLU UNIT
AND 4K OF MEMORY

2.2 STORAGE

2.2.1 PROGRAM STORAGE - THE ROUTINE USES 4K MEMORY

3. LOADING PROCEDURE

3.1 METHOD

PROCEDURE FOR NORMAL ABSOLUTE TAPES SHOULD BE FOLLOWED.

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

4. STARTING PROCEDURE

THE PROGRAM STARTS AT 200. IF THIS PROGRAM RUNS UNDER APT NO CHANGE IS NECESSARY. IF THIS PROGRAM RUNS ALONE THE OPERATOR HAS THE FOLLOWING OPTIONS BY SETTING THE SOFTWARE SWITCH REGISTER (LOCATION 422)

BIT 6=1 (100 OCTAL) INHIBIT EIS/FIS OPTION TESTS.

BIT 5=1 (40 OCTAL) IF WE WANT TO SUPRESS 'END OF PASS' TYPEOUT

BIT 4=1 (20 OCTAL) WILL NOT ALLOW OPCODES 75400-76777 TO DO RESERVED INSTRUCTIONS TRAPS IN THE LAST TEST OF THIS DIAGNOSTIC.

BIT 3=1 (10 OCTAL) WILL NOT ALLOW OPCODES 170000-177777 TO DO RESERVED INSTRUCTIONS TRAPS IN THE LAST TEST OF THIS DIAGNOSTIC.

BIT 2=1 (4 OCTAL) WILL NOT ALLOW OPCODES 76030-76057 (DIS RESERVED OPCODE SPACE) NOR EIS OPCODES TO DO RESERVED INSTRUCTION TRAPS IN THE LAST TEST OF THIS DIAGNOSTIC

THE PROGRAM STARTS AT 200.
IF IT IS DESIRED TO RESET THE PASS COUNT BACK TO ZERO START AT LOCATION 210.

4.3 PROGRAM AND/OR OPERATOR ACTION

LOAD PROGRAM INTO MEMORY. (BOTTOM 4K)
SET THE DESIRED SWITCH REGISTER BITS, IF ANY.
SET 'LTC' SWITCH TO THE 'OFF' POSITION.
LOAD ADDRESS.
START.

THE PROGRAM WILL PRINT END OF PASS AFTER THE 1ST ITERATION AND THEN PRINT IT EVERY 15 TIMES; APROXIMATELY 2 MINUTES.

114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146

5. OPERATION PROCEDURE

5.2 SUBROUTINE ABSTRACTS

5.2.1 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 PRINCIPLE 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.

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 TRAP OCCURRED. (MEMORY AS SPECIFIED BY R6 CONTAINS THE PC OF THE INSTRUCTION FOLLOWING THE INSTRUCTION WHERE THE TRAP OCCURRED. ALSO THE CONTENTS OF '\$TESTN' CONTAIN THE TEST NUMBER THAT IT WAS DOING BEFORE IT TRAPPED.

5.3 PROGRAM AND/OR OPERATOR ACTION

5.3.1 LOADING AND STARTING AT 200 STARTS THE TEST. IF AN ERROR IS DETECTED, THERE WILL BE A HALT.

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

6. ERRORS

6.1 ALL ERRORS WILL CAUSE A HALT.
THE PC+2 OF THE HALT INSTRUCTION IS PRINTED
ON THE CONSOLE DEVICE BY THE LSI-11.

6.1.1 THE PROGRAM CHECKS TO SEE THAT THE P.C. DOESN'T JUMP
WITHIN THE TESTS, BY A SEQUENCE COUNT CALLED '\$TSTN'
THIS TEST IS A SEQUENTIAL INCREMENT AND COMPARE COUNT.

EX: CODE

```
INC    @#$TESTN    ;UPDATE TEST NUMBER
CMP    #N,@#$TESTN ;SEQUENCE ERROR?
BNE    SOME LOCATION ;BRANCH TO ERROR HALT ON SEQ ERROR
IMPORTANT
```

IF AN ERROR IS DETECTED ;IT COULD BE BECAUSE OF TWO REASONS.
A) WRONG TEST NUMBER
B) ERROR IN THE PRESENT TEST.

```
////////////////////////////////////  
THE TEST SEQUENCE LOCATION 'TESTN' SHOULD BE CHECKED FIRST  
TO SEE IF IT MATCHES THE PRESENT TEST.  
IF IT DOESN'T MATCH ; THEN THE CONTENTS OF THIS LOCATION  
TELL YOU WHICH TEST IT WAS DOING BEFORE IT HALTED.  
////////////////////////////////////
```

6.2 ERROR RECOVERY
ON TRAP ERRORS - RESTART AT STARTING ADDRESS

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

7. RESTRICTIONS

7.1 STARTING RESTRICTION

'LTC' SWITCH MUST BE 'OFF'.

7.2 OPERATIONAL RESTRICTION

'LTC' SWITCH MUST BE 'OFF'.

8. MISCELLANEOUS

THERE IS A TEST THAT WILL CHECK THAT ODD ADDRESSING
WILL IGNORE BIT '0'

8.1 EXECUTION TIME

FOR ONE PASS APROXIMATELY 8 SECONDS; THEN IT TYPES
'END OF PASS' APROXIMATELY EVERY 2 MINUTES.

9. 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 ARE THE RTT AND THE RTI INSTRUCTIONS AND THAT ALL
RESTRICTED INSTRUCTIONS WILL TRAP. VERIFICATION OF THE 'BPT' 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.
SPECIAL CHECKS ARE MADE TO SEE IF BUS
ERROR TRAPS OCCUR ON NON-EXISTENT MEMORY.

```
311
312
313          .NLIST MD,CND,MC
314          .LIST  ME
315
316
317          000007          PC=%7
318          000006          SP=%6
319          000003          TAB=%3
320          000000          RO=%0
321          000001          LAST=%1
322          000002          FIRST=%2
323          000000          HLT=HALT
324          104400          TRAP=104400
325          104000          EMT=104000
326          000003          TRT=3
327          000004          ITRAP5=4
328          000004          RTRAP5=4          ;RESERVED INST AND ILLEGAL ADDRESSES
329          000014          RTRAP4=14          ;FOR TRACE TRAP
330          000030          RTRAP3=30          ;FOR EMULATOR TRAP
331          000020          RTRAP2=20          ;FOR IOT TRAP
332          000034          RTRAP1=34          ;FOR TRAP INST
333          177564          TTCSR=177564
334          177560          TRCSR=177560
335          177564          TPS=177564
336          177566          TPB=177566
337          000240          BELL=240
338          000240          NOP=240
339          000007          TRAPA=000007
340          000010          RTRAP=10
341          004700          ILLA=004700
342          000100          ILLB=100
343          000404          $TSTNM=$TESTN
344          000402          $ERROR=$FATAL
345          .ABS
417          .MCALL .SAPTHDR
418          .MCALL .SAPTBL5
419          .MCALL, .SACT11
```



```
421
422
423      000400
424      .SBTTL  .=400
          ACT11 HOOKS
(1)
(2)      ;*****
(1)      ;HOOKS REQUIRED BY ACT11
(1)      $SVPC=.          ;SAVE PC
(1)      000046          .=46
(1)      000046 013232  $ENDAD          ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
(1)      000052 000052  .=52
(1)      000052 000000  .WORD 0          ;;2)SET LOC.52 TO ZERO
(1)      000400          .=$SVPC          ;; RESTORE PC
425      .SBTTL  APT MAILBOX-ETABLE
(1)
(2)      ;*****
(1)      .EVEN
(1)      000400 $MAIL:          ;;APT MAILBOX
(1)      000400 000000 $MSGTY: .WORD  AMSGTY  ;;MESSAGE TYPE CODE
(1)      000402 000000 $FATAL: .WORD  AFATAL  ;;FATAL ERROR NUMBER
(1)      000404 000000 $TESTN: .WORD  ATESTN  ;;TEST NUMBER
(1)      000406 000000 $PASS:  .WORD  APASS   ;;PASS COUNT
(1)      000410 000000 $DEVCT: .WORD  ADEVCT  ;;DEVICE COUNT
(1)      000412 000000 $UNIT:  .WORD  AUNIT   ;;I/O UNIT NUMBER
(1)      000414 000000 $MSGAD: .WORD  AMSGAD  ;;MESSAGE ADDRESS
(1)      000416 000000 $MSGLG: .WORD  AMSGLG  ;;MESSAGE LENGTH
(1)      000420 $ETABLE:          ;;APT ENVIRONMENT TABLE
(1)      000420      000  $ENV:  .BYTE  AENV    ;;ENVIRONMENT BYTE
(1)      000421      000  $ENVM: .BYTE  AENVM   ;;ENVIRONMENT MODE BITS
(1)      000422 000000 $$WREG: .WORD  ASWREG  ;;APT SWITCH REGISTER
(1)      000424 000000 $USWR:  .WORD  AUSWR   ;;USER SWITCHES
(1)      000426 000000 $CPUOP: .WORD  ACPUOP  ;;CPU TYPE,OPTIONS
(1)      ;*          BITS 15-11=CPU TYPE
(1)      ;*          11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
(1)      ;*          11/70=06,PDQ=07,Q=10
(1)      ;*          BIT 10=REAL TIME CLOCK
(1)      ;*          BIT 9=FLOATING POINT PROCESSOR
(1)      ;*          BIT 8=MEMORY MANAGEMENT
(1)      000430      000  $MAMS1: .BYTE  AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
(1)      000431      000  $MTYP1: .BYTE  AMTYP1  ;;MEM. TYPE,BLK#1
(1)      ;*          MEM.TYPE BYTE  -- (HIGH BYTE)
(1)      ;*          900 NSEC CORE=001
(1)      ;*          300 NSEC BIPOLAR=002
(1)      ;*          500 NSEC MOS=003
(1)      000432 000000 $MADR1: .WORD  AMADR1  ;;HIGH ADDRESS,BLK#1
(1)      ;*          MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF 'TYPE' ABOVE
(1)      000434      000  $MAMS2: .BYTE  AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
(1)      000435      000  $MTYP2: .BYTE  AMTYP2  ;;MEM.TYPE,BLK#2
(1)      000436 000000 $MADR2: .WORD  AMADR2  ;;MEM.LAST ADDRESS,BLK#2
(1)      000440      000  $MAMS3: .BYTE  AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
(1)      000441      000  $MTYP3: .BYTE  AMTYP3  ;;MEM.TYPE,BLK#3
(1)      000442 000000 $MADR3: .WORD  AMADR3  ;;MEM.LAST ADDRESS,BLK#3
(1)      000444      000  $MAMS4: .BYTE  AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
(1)      000445      000  $MTYP4: .BYTE  AMTYP4  ;;MEM.TYPE,BLK#4
```

(1) 000446 000000
(1) 000450
(1)
426
(1)
(2)
(1)
(2)
(1) 000450
(1) 000024 000024
(1) 000024 000200
(1) 000044 000044
(1) 000044 000450
(1) 000450
(2)
(1)
(1)
(1)
(1) 000450
(1) 000450 000000
(1) 000452 000400
(1) 000454 000011
(1) 000456 000011
(1) 000460 000000
(1) 000462 000024

```
$MADR4: .WORD AMADR4 ;;MEM.LAST ADDRESS,BLK#4
$ETEND:
.MEXIT
.SBTTL APT PARAMETER BLOCK

:*****
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
:*****
.$X=    ;;SAVE CURRENT LOCATION
.=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
200     ;;FOR APT START UP
.=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
$APTHDR ;;POINT TO APT HEADER BLOCK
.=.$X   ;;RESET LOCATION COUNTER

:*****
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
:INTERFACE SPEC.

$APTHD:
$SHIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MBADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
$TSTM: .WORD 11 ;;RUN TIM OF LONGEST TEST
$PASTM: .WORD 11 ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
$UNITM: .WORD 0 ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
        .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
```

```

428
429
430      . =200
431 000200 000167 000276      JMP      START
432      . =210
433 000210 005037 000406      CLR      @#$PASS      ;CLEAR THE PASS COUNT
434 000214 000167 000262      JMP      START
435      . =500
436 000500 000000      BUFF: 00000
437 000502 012767 013302 177314  START: MOV      #PWRDWN,24      ;SET UP THE POWER DOWN VECTOR
438 000510 012767 000340 177310      MOV      #340,26      ;SET UP POWER DOWN PRIORITY
439 000516 105767 177676      TSTB     $ENV          ;ARE WE UNDER APT?
440 000522 001023      BNE      BEGIN        ;YES
441 000524 005067 177671      CLR      $ENVM
442 000530 005067 177672      CLR      $CPUOP
443 000534 132767 000040 177660      BITB     #40,$SWREG    ;DO WE PRINT END OF PASS
444 000542 001403      BEQ      1$           ;YES
445 000544 152767 000040 177647      BISB     #40,$ENVM
446
447 000552 016700 177644      1$:  MOV      $SWREG,R0      ;GET CONTENT OF $SWREG
448 000556 032700 000100      BIT      #100,R0       ;TEST IF INHIBIT EIS/FIS SW SET
449 000562 001403      BEQ      BEGIN        ;BR IF YES
450 000564 052767 000300 177634      BIS      #300,$CPUOP   ;NO SET UP LOCATION #CPUOP
451
452 000572 012737 177777 013262  BEGIN: MOV      #-1,@#PASSPT
453 000600 012702 000400      RESTRT: MOV      # $MSGTY,%2
454 000604 005067 177570      CLR      $MSGTY
455 000610 005067 177570      CLR      $TSTNM
456 000614 005067 177562      CLR      $ERROR
457 000620 000167 000026      JMP      TST1
458 000624 000000      K1:  0
459 000626 000000      K2:  0
460 000630 000000      K3:  0
461 000632 000000      K4:  0
462 000634 000000      K5:  0
463 000636 000000      K6:  0
464 000640 052525      K7:  052525
465 000642 052400      K10: 052400
466 000644 000000      K11: 0
467 000646 000000      K12: 0
468 000650 000000      HERE: 0

```

```

470
471
(2)
(3)
(2) 000652 005237 000404
(2) 000656 022737 000001 000404
(2) 000664 001124
472 000666 005006
473 000670 112667 177754
474 000674 020627 000002
475 000700 001405
(3) 000702 012737 000001 000402
(2) 000710 005212
(2) 000712 000000
(4)
(4)
476
477 000714 012706 001000
478 000720 114667 177724
479 000724 020627 000776
480 000730 001405
(3) 000732 012737 000002 000402
(2) 000740 005212
(2) 000742 000000
(4)
(4)
481
482 000744 005006
483 000746 112626
484 000750 020627 000004
485 000754 001405
(3) 000756 012737 000003 000402
(2) 000764 005212
(2) 000766 000000
(4)
(4)
486
487 000770 005006
488 000772 005004
489 000774 122624
490 000776 020627 000002
491 001002 001405
(3) 001004 012737 000004 000402
(2) 001012 005212
(2) 001014 000000
(4)
(4)
492
493 001016 005006
494 001020 005004
495 001022 122426
496 001024 020627 000002
497 001030 001405
(3) 001032 012737 000005 000402

```

```

:*****
:TEST 1 TEST AUTO INCREMENT AND DECREMENT OF R6 FOR WORD AND BYTES
:*****
TST1:  INC @#$TESTN ;UPDATE TEST NUMBER
      CMP #1,@#$TESTN ;SEQUENCE ERROR?
      BNE TST2-12 ;BR TO ERROR HALT ON SEQ ERROR
R6TST: CLR %6
      MOVB (6)+,HERE ;SIX SHOULD INCREMENT BY TWO
      CMP %6,#2
      BEQ 1$
      MOV #1,@#$FATAL ;MOVE TO MAILBOX # ***** 1 *****
      INC (R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;R6 DID NOT AUTO INCREMENT BY TWO
           ; TO SCOPE REPLACE HALT W/ 240
           ; AND REPLACE NEXT INST W/ 764
1$:   MOV #1000,%6
      MOVB -(6),HERE ;SHOULD DECREMENT BY TWO
      CMP %6,#776
      BEQ 2$
      MOV #2,@#$FATAL ;MOVE TO MAILBOX # ***** 2 *****
      INC (R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;R6 DID NOT AUTO DECREMENT BY 2
           ; TO SCOPE REPLACE HALT W/ 240
           ; AND REPLACE NEXT INST W/ 750
2$:   CLR %6
      MOVB (6)+,(6)+ ;DOUBLES AUTO INCREMENT OF R6
      CMP %6,#4
      BEQ 3$
      MOV #3,@#$FATAL ;MOVE TO MAILBOX # ***** 3 *****
      INC (R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;WRONG AUTO INCREMENT OF R6
           ; TO SCOPE REPLACE HALT W/ 240
           ; AND REPLACE NEXT INST W/ 736
3$:   CLR %6
      CLR %4
      CMPB (6)+,(4)+ ;TEST INCREMENT OF R6
      CMP %6,#2
      BEQ 4$
      MOV #4,@#$FATAL ;MOVE TO MAILBOX # ***** 4 *****
      INC (R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;WRONG INCREMENT OF R6
           ; TO SCOPE REPLACE HALT W/ 240
           ; AND REPLACE NEXT INST W/ 723
4$:   CLR %6
      CLR %4
      CMPB (4)+,(6)+ ;TEST INCREMENT OF R6
      CMP %6,#2
      BEQ 5$
      MOV #5,@#$FATAL ;MOVE TO MAILBOX # ***** 5 *****

```


516
517
(2)
(3)
(2)
(2)
(2)
518
519
520
521
522
523
524
(3)
(2)
(2)
(4)
(4)
525
526
527
528
529
530
531
532
(3)
(2)
(2)
(4)
(4)
533
534
535
536
537
538
539
540
(3)
(2)
(2)
(4)
(4)
541
542
543
544
545
546
547
548
(3)

001150 005237 000404
001154 022737 000002 000404
001162 001137
001164 012767 123456 177442
001172 012767 050505 177424
001200 012705 000624
001204 012706 000634
001210 112625
001212 022767 050456 177404
001220 001405
001222 012737 000011 000402
001230 005212
001232 000000

001234 012767 123456 177372 1\$:
001242 012767 050505 177354
001250 012705 000624
001254 012706 000636
001260 114625
001262 026727 177336 050456
001270 001405
001272 012737 000012 000402
001300 005212
001302 000000

001304 012767 123456 177312 2\$:
001312 012767 050505 177314
001320 012705 000624
001324 012706 000634
001330 112526
001332 022767 050456 177274
001340 001405
001342 012737 000013 000402
001350 005212
001352 000000

001354 012767 123456 177242 3\$:
001362 012767 050505 177244
001370 012705 000625
001374 012706 000634
001400 112526
001402 026727 177226 050647
001410 001405
001412 012737 000014 000402

```

:*****
:TEST 2 TEST TRANSFER OF BYTE USING R6
:*****
TST2:  INC    @#$TESTN      :UPDATE TEST NUMBER
      CMP    #2,@#$TESTN   :SEQUENCE ERROR?
      BNE    TST3-12 ;BR TO ERROR HALT ON SEQ ERROR
      MOV    #123456,K5
      MOV    #050505,K1
      MOV    #K1,%5        :%5=(050505)K1
      MOV    #K5,%6        :%6=(123456)K5
      MOVB  (6)+,(5)+      :LOW .BYTE OF R6 TO R5
      CMP    #050456,K1
      BEQ    1$
      MOV    #11,@#$FATAL  :MOVE TO MAILBOX # ***** 11 *****
      INC    (R2)          :SET MSGTYP TO FATAL ERROR
      HALT                :FALSE TRANSFER OF .BYTE
                        : TO SCOPE REPLACE HALT W/ 240
                        : AND REPLACE NEXT INST W/ 753

      MOV    #123456,K5
      MOV    #050505,K1
      MOV    #K1,%5        :%5(050505)K1
      MOV    #K6,%6        :%6(123456)K5
      MOVB  -(6),(5)+      :LOW .BYTE OF R6 TO R5 (DECREMENT)
      CMP    K1,#050456
      BEQ    2$
      MOV    #12,@#$FATAL  :MOVE TO MAILBOX # ***** 12 *****
      INC    (R2)          :SET MSGTYP TO FATAL ERROR
      HALT                :FALSE R6 .BYTE TRANSFER
                        : TO SCOPE REPLACE HALT W/ 240
                        : AND REPLACE NEXT INST W/ 727

      MOV    #123456,K1
      MOV    #050505,K5
      MOV    #K1,%5        :(123456)
      MOV    #K5,%6        :(050505)
      MOVB  (5)+,(6)+      :LOW OF R5 TO LOW OF R6
      CMP    #050456,K5
      BEQ    3$
      MOV    #13,@#$FATAL  :MOVE TO MAILBOX # ***** 13 *****
      INC    (R2)          :SET MSGTYP TO FATAL ERROR
      HALT                :FALSE R6 .BYTE TRANSFER
                        : TO SCOPE REPLACE HALT W/ 240
                        : AND REPLACE NEXT INST W/ 703

      MOV    #123456,K1
      MOV    #050505,K5
      MOV    #K1+1,%5      :123456
      MOV    #K5,%6        :050505
      MOVB  (5)+,(6)+      :HIGH OF R5 TO LOW OF R6
      CMP    K5,#050647
      BEQ    4$
      MOV    #14,@#$FATAL  :MOVE TO MAILBOX # ***** 14 *****

```



```
558
559 :*****
(2) :TEST 3 TEST BYTE OPERATION WITH SEQUENTIAL ODD; .EVEN ADDRESS
(3) :*****
(2) 001474 005237 000404 TST3: INC @#STESTN ;UPDATE TEST NUMBER
(2) 001500 022737 000003 000404 CMP #3,@#STESTN ;SEQUENCE ERROR?
(2) 001506 001103 BNE TST4-12 ;BR TO ERROR HALT ON SEQ ERROR
560 001510 126767 177124 177123 CMPB K7,K7+1 ;SAME .WORD LOW TO HIGH
561 001516 001405 BEQ 1$
(3) 001520 012737 000016 000402 MOV #16,@#$FATAL ;MOVE TO MAILBOX # ***** 16 *****
(2) 001526 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 001530 000000 HALT ;SHOULD COMPARE LOW TO HIGH
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 766
562
563 001532 126767 177103 177100 1$: CMPB K7+1,K7 ;COMPARE ODD TO .EVEN SAME .WORD
564 001540 001405 BEQ 2$
(3) 001542 012737 000017 000402 MOV #17,@#$FATAL ;MOVE TO MAILBOX # ***** 17 *****
(2) 001550 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 001552 000000 HALT ;ODD TO .EVEN .BYTE FAILURE
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 755
565
566 001554 126767 177063 177056 2$: CMPB K10+1,K7 ;SEQUENTIAL .BYTES
567 BEQ TST4 ;DIFFERENT .WORDS
568 001562 001462 BEQ TST4
(4) 001564 012737 000020 000402 MOV #20,@#$FATAL ;MOVE TO MAILBOX # ***** 20 *****
(3) 001572 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 001574 000000 HALT ;ODD TO .EVEN FAILED
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 744
569
570 001576 126767 177040 177032 CMPB K10,K6
571 001604 001405 BEQ 3$
(3) 001606 012737 000021 000402 MOV #21,@#$FATAL ;MOVE TO MAILBOX # ***** 21 *****
(2) 001614 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 001616 000000 HALT ;.EVEN TO EVEN FAILED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 733
572 001620 126767 177015 177015 3$: CMPB K7+1,K10+1
573 001626 001405 BEQ 4$
(3) 001630 012737 000022 000402 MOV #22,@#$FATAL ;MOVE TO MAILBOX # ***** 22 *****
(2) 001636 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 001640 000000 HALT ;ODD TO ODD FAILED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 722
574
575 001642 126767 176774 176773 4$: CMPB K10,K10+1
576 001650 001005 BNE 5$
(3) 001652 012737 000023 000402 MOV #23,@#$FATAL ;MOVE TO MAILBOX # ***** 23 *****
(2) 001660 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 001662 000000 HALT ;LOW TO HIGH IN SAME .WORD FAILED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 711
```



```
577
578 001664 126767 176753 176751 5$:  CMPB  K10+1,K:10+1
579 001672 001405  BEQ   6$
(3) 001674 012737 000024 000402  MOV   #24,@#$FATAL ;MOVE TO MAILBOX # ***** 24 *****
(2) 001702 005212  INC   (R2) ;SET MSGTYP TO FATAL ERROR
(2) 001704 000000  HALT  ;HIGH TO LOW IN SAME .WORD FAILED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 700
580
581 001706 126767 176730 176725 6$:  CMPB  K10,K7+1
582 001714 001005  BNE   TST4
(4) 001716 012737 000025 000402  MOV   #25,@#$FATAL ;MOVE TO MAILBOX # ***** 25 *****
(3) 001724 005212  INC   (R2) ;SET MSGTYP TO FATAL ERROR
(3) 001726 000000  HALT  ;.EVEN TO ODD FAILED,OR WRONG $TESTN,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 667
```

```

584
585
(2)
(3)
(2) 001730 005237 000404
(2) 001734 022737 000004 000404
(2) 001742 001070
586 001744 000277
587 001746 005067 011144
588 001752 106437
(1) 001754 013116
589 001756 103005
(3) 001760 012737 000026 000402
(2) 001766 005212
(2) 001770 000000
(4)
(4)
590 001772
(1) 001772 102005
(3) 001774 012737 000027 000402
(2) 002002 005212
(2) 002004 000000
(4)
(4)
591 002006
(1) 002006 001005
(3) 002010 012737 000030 000402
(2) 002016 005212
(2) 002020 000000
(4)
(4)
592 002022
(1) 002022 100005
(3) 002024 012737 000031 000402
(2) 002032 005212
(2) 002034 000000
(4)
(4)
593
594 002036 000257
595 002040 106737
(1) 002042 013116
596 002044 052767 000017 011044
597 002052 106437
(1) 002054 013116
598
599 002056 103405
(3) 002060 012737 000032 000402
(2) 002066 005212
(2) 002070 000000
(4)
(4)
600 002072
(1) 002072 102405

```

:TEST 4 TEST THE CC BITS

TST4: INC @#\$TESTN ;UPDATE TEST NUMBER
 CMP #4,@#\$TESTN ;SEQUENCE ERROR?
 BNE TST5-12 ;BR TO ERROR HALT ON SEQ ERROR
 SCC ;SET STATUS
 CLR STATUS ;CLEAR STATUS
 .WORD 106437
 .WORD STATUS
 BCC 1\$
 MOV #26,@#\$FATAL ;MOVE TO MAILBOX # ***** 26 *****
 INC (R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;C NOT CLEAR
 ; TO SCOPE REPLACE HALT W/ 240
 ; AND REPLACE NEXT INST W/ 764

1\$: BVC 2\$
 MOV #27,@#\$FATAL ;MOVE TO MAILBOX # ***** 27 *****
 INC (R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;V NOT CLEAR
 ; TO SCOPE REPLACE HALT W/ 240
 ; AND REPLACE NEXT INST W/ 756

2\$: BNE 3\$
 MOV #30,@#\$FATAL ;MOVE TO MAILBOX # ***** 30 *****
 INC (R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;Z NOT CLEAR
 ; TO SCOPE REPLACE HALT W/ 240
 ; AND REPLACE NEXT INST W/ 750

3\$: BPL 4\$
 MOV #31,@#\$FATAL ;MOVE TO MAILBOX # ***** 31 *****
 INC (R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;N NOT CLEAR
 ; TO SCOPE REPLACE HALT W/ 240
 ; AND REPLACE NEXT INST W/ 742

4\$: CCC ;CLEAR CONDITION CODES
 .WORD 106737
 .WORD STATUS
 BIS #17,STATUS ;SET STATUS TO ONES
 .WORD 106437
 .WORD STATUS

5\$: BCS 5\$
 MOV #32,@#\$FATAL ;MOVE TO MAILBOX # ***** 32 *****
 INC (R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;C NOT SET
 ; TO SCOPE REPLACE HALT W/ 240
 ; AND REPLACE NEXT INST W/ 724

6\$: BVS 6\$


```

604
605 ;*****
606 ;TEST 5 TEST THAT A TRAP OCCURES ON A RESERVED INSTRUCTION
607 ;*****
608 TST5: INC @#$TESTN ;UPDATE TEST NUMBER
609 (2) 002136 005237 000404 CMP #5,@#$TESTN ;SEQUENCE ERROR?
610 (2) 002142 022737 000005 000404 BNE TST6-12 ;BR TO ERROR HALT ON SEQ ERROR
611 (2) 002150 001006 MOV #BUFF,SP ;STACK POINTER SETUP
612 606 002152 012706 000500 MOV #RETA,RTRAP ;RETURN LOCATION
613 607 002156 012767 002200 175624 TRAPA ;RESERVED INSTRUCTION, SHOULD TRAP
614 608 002164 000007 MOV #36,@#$FATAL ;MOVE TO MAILBOX # ***** 36 *****
615 609 002166 012737 000036 000402 INC (R2) ;SET MSGTYP TO FATAL ERROR
616 (2) 002174 005212 HALT ;DID NOT TRAP OR WRONG $TESTN
617 (2) 002176 000000 ; TO SCOPE REPLACE HALT W/ 240
618 (4) ; AND REPLACE NEXT INST W/ 764
619 (4)
620 610 002200 RETA:
621 611 ;*****
622 (2) ;TEST 6 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
623 (3) ;*****
624 (2) 002200 005237 000404 TST6: INC @#$TESTN ;UPDATE TEST NUMBER
625 (2) 002204 022737 000006 000404 CMP #6,@#$TESTN ;SEQUENCE ERROR?
626 (2) 002212 001011 BNE TST7-12 ;BR TO FRROR HALT ON SEQ ERROR
627 612 002214 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
628 613 002220 012767 002230 175562 MOV #RETB,RTRAP ;RETURN POINTER
629 614 002226 000007 TRAPA ;RESERVED INSTRUCTION
630 615 002230 020627 000474 RETB: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
631 616 002234 001405 BEQ TST7
632 (4) 002236 012737 000037 000402 MOV #37,@#$FATAL ;MOVE TO MAILBOX # ***** 37 *****
633 (3) 002244 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
634 (3) 002246 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $TESTN
635 (5) ; TO SCOPE REPLACE HALT W/ 240
636 (5) ; AND REPLACE NEXT INST W/ 761
    
```



```

643
644
(2)
(3)
(2) 002460 005237 000404
(2) 002464 022737 000011 000404
(2) 002472 001125
645 002474 012706 000500
646 002500 012767 002514 175302
647 002506 005067 175300
648 002512 000007
649 002514
650 002514 100005
(3) 002516 012737 000043 000402
(2) 002524 005212
(2) 002526 000000
(4)
(4)
651 002530
(1) 002530 001005
(3) 002532 012737 000044 000402
(2) 002540 005212
(2) 002542 000000
(4)
(4)
652 002544
(1) 002544 102005
(3) 002546 012737 000045 000402
(2) 002554 005212
(2) 002556 000000
(4)
(4)
653 002560
(1) 002560 103005
(3) 002562 012737 000046 000402
(2) 002570 005212
(2) 002572 000000
(4)
(4)
654 002574
(1) 002574 106737
(1) 002576 013116
655 002600 032767 000340 010310
656 002606 001405
(3) 002610 012737 000047 000402
(2) 002616 005212
(2) 002620 000000
(4)
(4)
657 002622 012706 000500
658 002626 012767 002644 175154
659 002634 012767 000357 175150
660 002642 000007
661 002644

```

```

:*****
:TEST 11 TEST THAT 'NEW' STATUS IS CORRECT
:*****
TST11: INC @#STESTN ;UPDATE TEST NUMBER
CMP #11,@#STESTN ;SEQUENCE ERROR?
BNE RSTP1 ;BR TO ERROR HALT ON SEQ ERROR
MOV #BUFF,SP
MOV #RETF,RTRAP
CLR RTRAP+2 ;CLEAR FUTURE PRIORITY AND CC
TRAPA

RETF: BPL 1$
MOV #43,@#$FATAL ;MOVE TO MAILBOX # ***** 43 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;C NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 761

1$: BNE 2$
MOV #44,@#$FATAL ;MOVE TO MAILBOX # ***** 44 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;Z NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 753

2$: BVC 3$
MOV #45,@#$FATAL ;MOVE TO MAILBOX # ***** 45 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;V NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 745

3$: BCC 4$
MOV #46,@#$FATAL ;MOVE TO MAILBOX # ***** 46 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;C NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 737

4$: .WORD 106737
.WORD STATUS
BIT #340,STATUS ;TEST PRIORITY
BEQ 5$
MOV #47,@#$FATAL ;MOVE TO MAILBOX # ***** 47 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;PRIORITY NOT ZERO
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 724

5$: MOV #BUFF,SP
MOV #RETF,RTRAP
MOV #357,RTRAP+2 ;SET NEW 'CC' AND PRIORITY
TRAPA ;TRAP HERE

RETF:

```

```

662 002644 100405      BMI      1$
(3) 002646 012737 000050 000402  MOV      #50,@#$FATAL ;MOVE TO MAILBOX # ***** 50 *****
(2) 002654 005212      INC      (R2)          ;SET MSGTYP TO FATAL ERROR
(2) 002656 000000      HALT                    ;N NOT SET
(4)                                     ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 705

663 002660 001405      1$:
(1) 002660 001405      BEQ      2$
(3) 002662 012737 000051 000402  MOV      #51,@#$FATAL ;MOVE TO MAILBOX # ***** 51 *****
(2) 002670 005212      INC      (R2)          ;SET MSGTYP TO FATAL ERROR
(2) 002672 000000      HALT                    ;Z NOT SET
(4)                                     ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 677

664 002674 102405      2$:
(1) 002674 102405      BVS      3$
(3) 002676 012737 000052 000402  MOV      #52,@#$FATAL ;MOVE TO MAILBOX # ***** 52 *****
(2) 002704 005212      INC      (R2)          ;SET MSGTYP TO FATAL ERROR
(2) 002706 000000      HALT                    ;V NOT SET
(4)                                     ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 671

665 002710 103405      3$:
(1) 002710 103405      BCS      4$
(3) 002712 012737 000053 000402  MOV      #53,@#$FATAL ;MOVE TO MAILBOX # ***** 53 *****
(2) 002720 005212      INC      (R2)          ;SET MSGTYP TO FATAL ERROR
(2) 002722 000000      HALT                    ;C NOT SET
(4)                                     ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 663

666 002724 106737      4$:
(1) 002724 106737      .WORD    106737
(1) 002726 013116      .WORD    STATUS
667 002730 016700 010162  MOV      STATUS,SP
668 002734 042706 000017  BIC      #17,SP
669 002740 022706 000340  CMP      #340,SP
670 002744 001405      BEQ      RST1
(1) 002746 001405      RSTP1:
(3) 002746 012737 000054 000402  MOV      #54,@#$FATAL ;MOVE TO MAILBOX # ***** 54 *****
(2) 002754 005212      INC      (R2)          ;SET MSGTYP TO FATAL ERROR
(2) 002756 000000      HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TESTN
(4)                                     ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 645

671 002760 012767 000012 175022  RST1:  MOV      #12,10
672 002766 005067 175020      CLR      12
    
```

```

674
675
(2)
(3)
(2) 002772 005237 000404
(2) 002776 022737 000012 000404
(2) 003004 001006
676 003006 012706 000500
677 003012 012767 003034 175014
678 003020 104400
679 003022 012737 000055 000402
(2) 003030 005212
(2) 003032 000000
(4)
(4)
680 003034
681
(2)
(3)
(2) 003034 005237 000404
(2) 003040 022737 000013 000404
(2) 003046 001011
682 003050 012706 000500
683 003054 012767 003064 174752
684 003062 104400
685 003064 020627 000474
686 003070 001405
(4) 003072 012737 000056 000402
(3) 003100 005212
(3) 003102 000000
(5)
(5)
687
(2)
(3)
(2) 003104 005237 000404
(2) 003110 022737 000014 000404
(2) 003116 001012
688 003120 012706 000500
689 003124 012767 003134 174702
690 003132 104400
691 003134 022767 003134 175332
692 003142 001405
(4) 003144 012737 000057 000402
(3) 003152 005212
(3) 003154 000000
(5)
(5)
:*****
:TEST 12 TEST THAT A TRAP OCCURES FOR A 'TRAP' INSTRUCTION
:*****
TST12: INC @#$TESTN ;UPDATE TEST NUMBER
        CMP #12,@#$TESTN ;SEQUENCE ERROR?
        BNE TST13-12 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #BUFF,SP ;STACK POINTER SETUP
        MOV #RETA1,RTRAP1 ;RETURN LOCATION
        TRAP ;RESERVED INSTRUCTION, SHOULD TRAP
        MOV #55,@#$FATAL ;MOVE TO MAILBOX # ***** 55 *****
        INC (R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;DID NOT TRAP,OR WRONG $TESTN
        ; TO SCOPE REPLACE HALT W/ 240
        ; AND REPLACE NEXT INST W/ 764

RETA1:
:*****
:TEST 13 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
:*****
TST13: INC @#$TESTN ;UPDATE TEST NUMBER
        CMP #13,@#$TESTN ;SEQUENCE ERROR?
        BNE TST14-12 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #BUFF,SP ;STACK POINTER SETUP
        MOV #RETB1,RTRAP1 ;RETURN POINTER
        TRAP ;RESERVED INSTRUCTION
RETB1: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
        BEQ TST14
        MOV #56,@#$FATAL ;MOVE TO MAILBOX # ***** 56 *****
        INC (R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $TESTN
        ; TO SCOPE REPLACE HALT W/ 240
        ; AND REPLACE NEXT INST W/ 761

:*****
:TEST 14 TEST THAT PROPER P.C. IS SAVED
:*****
TST14: INC @#$TESTN ;UPDATE TEST NUMBER
        CMP #14,@#$TESTN ;SEQUENCE ERROR?
        BNE TST15-12 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #BUFF,SP ;STACK POINTER SETUP
        MOV #RETC1,RTRAP1 ;RETURN FROM TRAP POINTER
        TRAP ;TRAP ON THIS INSTRUCTION
RETC1: CMP #,BUFF-4 ;CHECK INCREMENTED P.C.
        BEQ TST15
        MOV #57,@#$FATAL ;MOVE TO MAILBOX # ***** 57 *****
        INC (R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;INCORRECT P.C.,OR WRONG $TESTN
        ; TO SCOPE REPLACE HALT W/ 240
        ; AND REPLACE NEXT INST W/ 760
    
```



```

694
695
(2)
(3)
(2) 003156 005237 000404
(2) 003162 022737 000015 000404
(2) 003170 001043
696 003172 012706 000500
697 003176 012767 003220 174630
698 003204 005067 007706
699 003210 106437
(1) 003212 013116
700 003214 000257
701 003216 104400
702 003220 026727 175252 000000 RETD1:
703 003226 001405
(3) 003230 012737 000060 000402
(2) 003236 005212
(2) 003240 000000
(4)
(4)
704 003242 012706 000500 1$: MOV #BUFF, SP
705 003246 012767 003270 174560 MOV #RETE1, RTRAP1
706 003254 012767 000357 007634 MOV #357, STATUS
707 003262 106437 .WORD 106437
(1) 003264 013116 .WORD STATUS
708 003266 104400 TRAP
709 003270 026727 175202 000357 RETE1: CMP BUFF-2, #357
710 003276 001405 BEQ TST16
(4) 003300 012737 000061 000402 MOV #61, @#$FATAL
(3) 003306 005212 INC (R2)
(3) 003310 000000 HALT
    
```

```

:*****
:TEST 15 TEST THAT 'OLD' STATUS AND PRIORITY ARE PLACED ON STACK
:*****
TST15: INC @#$TESTN ;UPDATE TEST NUMBER
CMP #15,@#$TESTN ;SEQUENCE ERROR?
BNE TST16-12 ;BR TO ERROR HALT ON SEQ ERROR
MOV #BUFF, SP ;SET UP
MOV #RETD1, RTRAP1 ;SET UP
CLR STATUS ;CLEAR STATUS AND PRIORITY
.WORD 106437
.WORD STATUS
CCC
TRAP ;TRAP
CMP BUFF-2, #0 ;TEST THAT OLD STATUS WENT TO STACK
BEQ 1$
MOV #60, @#$FATAL ;MOVE TO MAILBOX # ***** 60 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INCORRECT STATUS
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 753
1$: MOV #BUFF, SP ;SET UP
MOV #RETE1, RTRAP1 ;SET UP
MOV #357, STATUS ;SET PRIORITY
.WORD 106437
.WORD STATUS
TRAP ;SET CC
CMP BUFF-2, #357 ;COMPARES STATUS ON STACK
BEQ TST16
MOV #61, @#$FATAL ;MOVE TO MAILBOX # ***** 61 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INCORRECT STATUS ON STACK
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 727
    
```

```

712
713
(2)
(3)
(2) 003312 005237 000404
(2) 003316 022737 000016 000404
(2) 003324 001125
714 003326 012706 000500
715 003332 012767 003346 174474
716 003340 005067 174472
717 003344 104400
718 003346
719 003346 100005
(3) 003350 012737 000062 000402
(2) 003356 005212
(2) 003360 000000
(4)
(4)
720 003362
(1) 003362 001005
(3) 003364 012737 000063 000402
(2) 003372 005212
(2) 003374 000000
(4)
(4)
721 003376
(1) 003376 102005
(3) 003400 012737 000064 000402
(2) 003406 005212
(2) 003410 000000
(4)
(4)
722 003412
(1) 003412 103005
(3) 003414 012737 000065 000402
(2) 003422 005212
(2) 003424 000000
(4)
(4)
723 003426
(1) 003426 106737
(1) 003430 013116
724 003432 032767 000340 007456
725 003440 001405
(3) 003442 012737 000066 000402
(2) 003450 005212
(2) 003452 000000
(4)
(4)
726 003454 012706 000500
727 003460 012767 003476 174346
728 003466 012767 000357 174342
729 003474 104400
730 003476
    
```

```

:*****
:TEST 16 TEST THAT 'NEW' STATUS IS CORRECT
:*****
TST16: INC @#STESTN ;UPDATE TEST NUMBER
CMP #16,@#STESTN ;SEQUENCE ERROR?
BNE TST17-12 ;BR TO ERROR HALT ON SEQ ERROR
MOV #BUFF,SP
MOV #RETF1,RTRAP1
CLR RTRAP1+2 ;CLEAR FUTURE PRIORITY AND CC
TRAP

RETF1: BPL 1$
MOV #62,@#SFATAL ;MOVE TO MAILBOX # ***** 62 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;C NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 761

1$: BNE 2$
MOV #63,@#SFATAL ;MOVE TO MAILBOX # ***** 63 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;Z NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 753

2$: BVC 3$
MOV #64,@#SFATAL ;MOVE TO MAILBOX # ***** 64 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;V NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 745

3$: BCC 4$
MOV #65,@#SFATAL ;MOVE TO MAILBOX # ***** 65 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;C NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 737

4$: .WORD 106737
.WORD STATUS
BIT #340,STATUS ;TEST PRIORITY
BEQ 5$
MOV #66,@#SFATAL ;MOVE TO MAILBOX # ***** 66 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;PRIORITY NOT ZERO
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 724

5$: MOV #BUFF,SP
MOV #RETG1,RTRAP1
MOV #357,RTRAP1+2 ;SET NEW 'CC' AND PRIORITY
TRAP ;TRAP HERE

RETG1:
    
```

731	003476	100405			BMI	1\$		
(3)	003500	012737	000067	000402	MOV	#67,@#\$FATAL	:MOVE TO MAILBOX # ***** 67 *****	
(2)	003506	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	003510	000000			HALT		:N NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 705	
732	003512					1\$:		
(1)	003512	001405			BEQ	2\$		
(3)	003514	012737	000070	000402	MOV	#70,@#\$FATAL	:MOVE TO MAILBOX # ***** 70 *****	
(2)	003522	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	003524	000000			HALT		:Z NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 677	
733	003526					2\$:		
(1)	003526	102405			BVS	3\$		
(3)	003530	012737	000071	000402	MOV	#71,@#\$FATAL	:MOVE TO MAILBOX # ***** 71 *****	
(2)	003536	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	003540	000000			HALT		:V NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 671	
734	003542					3\$:		
(1)	003542	103405			BCS	4\$		
(3)	003544	012737	000072	000402	MOV	#72,@#\$FATAL	:MOVE TO MAILBOX # ***** 72 *****	
(2)	003552	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	003554	000000			HALT		:C NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 663	
735	003556					4\$:		
(1)	003556	106737			.WORD	106737		
(1)	003560	013116			.WORD	STATUS		
736	003562	016706	007330		MOV	STATUS,SP		
737	003566	042706	000017		BIC	#17,SP		
738	003572	022706	000340		CMP	#340,SP		
739	003576	001405			BEQ	TST17		
(4)	003600	012737	000073	000402	MOV	#73,@#\$FATAL	:MOVE TO MAILBOX # ***** 73 *****	
(3)	003606	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(3)	003610	000000			HALT		:PRIORITY WAS CHANGED,OR WRONG \$TESTN	
(5)							: TO SCOPE REPLACE HALT W/ 240	
(5)							: AND REPLACE NEXT INST W/ 645	

```
741
742 :*****
(2) :TEST 17 TEST THAT ALL COMBINATION OF 'TRAP' WILL CAUSE A TRAP
(3) :*****
(2) 003612 005237 000404 TST17: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 003616 022737 000017 000404 CMP #17,@#$TESTN ;SEQUENCE ERROR?
(2) 003624 001011 BNE RB1AA ;BR TO ERROR HALT ON SEQ ERROR
743 003626 012767 104400 000012 MOV #TRAP,RB1 ;INITIALIZE BASE TRAP INSTRUCTION
744 003634 012767 003662 174172 MOV #RA1,34 ;RETURN FROM TRAP TO RA1
745 003642 012706 000500 RC1: MOV #BUFF,SP ;SET UP STACK POINTER
746 003646 104400 RB1: TRAP ;TRAP INST WILL BE MODIFIED TO TRAP+377
747 003650 RB1AA:
(3) 003650 012737 000074 000402 MOV #74,@#$FATAL ;MOVE TO MAILBOX # ***** 74 *****
(2) 003656 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 003660 000000 HALT ;PREVIOUS INST FAILED TO TRAP,OR WRONG $TESTN
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 761
748 003662 005267 177760 RA1: INC RB1
749 003666 022767 104777 177752 CMP #104777,RB1 ;TRAP+377 TO UPPER LIMIT
750 003674 103362 BHIS RC1 ;HAVE WE TESTED ALL
751 003676 012767 000036 174130 MOV #36,34
752 003704 005067 174126 CLR 36
753 :*****
(2) :TEST 20 TEST THAT A TRAP OCCURES ON AN 'IOT' INSTRUCTION
(3) :*****
(2) 003710 005237 000404 TST20: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 003714 022737 000020 000404 CMP #20,@#$TESTN ;SEQUENCE ERROR?
(2) 003722 001006 BNE TST21-12 ;BR TO ERROR HALT ON SEQ ERROR
754 003724 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
755 003730 012767 003752 174062 MOV #RETA2,RTRAP2 ;RETURN LOCATION
756 003736 000004 IOT ;RESERVE INSTRUCTION, SHOULD TRAP
757 003740 012737 000075 000402 MOV #75,@#$FATAL ;MOVE TO MAILBOX # ***** 75 *****
(2) 003746 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 003750 000000 HALT ;IOT DID NOT TRAP,OR WRONG $TESTN
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 764
758 003752 RETA2:
759 :*****
(2) :TEST 21 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
(3) :*****
(2) 003752 005237 000404 TST21: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 003756 022737 000021 000404 CMP #21,@#$TESTN ;SEQUENCE ERROR?
(2) 003764 001011 BNE TST22-12 ;BR TO ERROR HALT ON SEQ ERROR
760 003766 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
761 003772 012767 004002 174020 MOV #RETB2,RTRAP2 ;RETURN POINTER
762 004000 000004 IOT ;RESERVED INSTRUCTION
763 004002 020627 000474 RETB2: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
764 004006 001405 BEQ TST22
(4) 004010 012737 000076 000402 MOV #76,@#$FATAL ;MOVE TO MAILBOX # ***** 76 *****
(3) 004016 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 004020 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 761
```

```
766
767
(2)
(3)
(2) 004022 005237 000404
(2) 004026 022737 000022 000404
(2) 004034 001012
768 004036 012706 000500
769 004042 012767 004052 173750
770 004050 000004
771 004052 022767 004052 174414 RETC2:
772 004060 001405
(4) 004062 012737 000077 000402
(3) 004070 005212
(3) 004072 000000
(5)
(5)
773
(2)
(3)
(2) 004074 005237 000404
(2) 004100 022737 000023 000404
(2) 004106 001044
774 004110 012706 000500
775 004114 012767 004136 173676
776 004122 005067 006770
777 004126 106437
(1) 004130 013116
778 004132 000257
779 004134 000004
780 004136 026727 174334 000000 RETD2:
781 004144 001405
(3) 004146 012737 000100 000402
(2) 004154 005212
(2) 004156 000000
(4)
(4)
782 004160 012706 000500 1$:
783 004164 012767 004210 173626
784 004172 012767 000357 006716
785 004200 106437
(1) 004202 013116
786 004204 000277
787 004206 000004
788 004210 026727 174262 000357 RETE2:
789 004216 001405
(4) 004220 012737 000101 000402
(3) 004226 005212
(3) 004230 000000
(5)
(5)
```

:TEST 22 TEST THAT PROPER P.C. IS SAVED

TST22: INC @#\$TESTN ;UPDATE TEST NUMBER
CMP #22,@#\$TESTN ;SEQUENCE ERROR?
BNE TST23-12 ;BR TO ERROR HALT ON SEQ ERROR
MOV #BUFF,SP ;STACK POINTER SETUP
MOV #RETC2,RTRAP2 ;RETURN FROM TRAP POINTER
IOT ;TRAP ON THIS INSTRUCTION
RETC2: CMP #.,BUFF-4 ;CHECK FOR INCREMENTED P.C.
BEQ TST23
MOV #77,@#\$FATAL ;MOVE TO MAILBOX # ***** 77 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INCORRECT P.C.,OR WRONG \$TESTN
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 760

:TEST 23 TEST THAT 'OLD' STATUS AND PRIORITY ARE PLACED ON STACK

TST23: INC @#\$TESTN ;UPDATE TEST NUMBER
CMP #23,@#\$TESTN ;SEQUENCE ERROR?
BNE TST24-12 ;BR TO ERROR HALT ON SEQ ERROR
MOV #BUFF,SP ;SET UP
MOV #RETD2,RTRAP2 ;SET UP
CLR STATUS ;CLEAR STATUS AND PRIORITY
.WORD 106437
.WORD STATUS
IOT ;TRAP
RETD2: CMP BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK
BEQ 1\$
MOV #100,@#\$FATAL ;MOVE TO MAILBOX # ***** 100 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INCORRECT STATUS
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 753
1\$: MOV #BUFF,SP ;SET UP
MOV #RETE2,RTRAP2 ;SET UP
MOV #357,STATUS ;SET PRIORITY
.WORD 106437
.WORD STATUS
SCC ;SET CC
IOT ;TRAP
RETE2: CMP BUFF-2,#357 ;COMPARES STATUS ON STACK
BEQ TST24
MOV #101,@#\$FATAL ;MOVE TO MAILBOX # ***** 101 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INCORRECT STATUS ON STACK,OR WRONG \$TESTN
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 726

```

791
792
(2)
(3)
(2) 004232 005237 000404
(2) 004236 022737 000024 000404
(2) 004244 001125
793 004246 012706 000500
794 004252 012767 004266 173540
795 004260 005067 173536
796 004264 000004
797 004266
798 004266 100005
(3) 004270 012737 000102 000402
(2) 004276 005212
(2) 004300 000000
(4)
(4)
799 004302
(1) 004302 001005
(3) 004304 012737 000103 000402
(2) 004312 005212
(2) 004314 000000
(4)
(4)
800 004316
(1) 004316 102005
(3) 004320 012737 000104 000402
(2) 004326 005212
(2) 004330 000000
(4)
(4)
801 004332
(1) 004332 103005
(3) 004334 012737 000105 000402
(2) 004342 005212
(2) 004344 000000
(4)
(4)
802 004346
(1) 004346 106737
(1) 004350 013116
803 004352 032767 000340 006536
804 004360 001405
(3) 004362 012737 000106 000402
(2) 004370 005212
(2) 004372 000000
(4)
(4)
805 004374 012706 000500
806 004400 012767 004416 173412
807 004406 012767 000357 173406
808 004414 000004
809 004416

```

```

:*****
:TEST 24 TEST THAT 'NEW' STATUS IS CORRECT
:*****
TST24: INC @#STESTN ;UPDATE TEST NUMBER
CMP #24,@#STESTN ;SEQUENCE ERROR?
BNE STP ;BR TO ERROR HALT ON SEQ ERROR
MOV #BUFF,SP
MOV #RETF2,RTRAP2
CLR RTRAP2+2 ;CLEAR FUTURE PRIORITY AND CC
IOT

RETF2: BPL 1$
MOV #102,@#$FATAL ;MOVE TO MAILBOX # ***** 102 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;C NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 761

1$: BNE 2$
MOV #103,@#$FATAL ;MOVE TO MAILBOX # ***** 103 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;Z NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 753

2$: BVC 3$
MOV #104,@#$FATAL ;MOVE TO MAILBOX # ***** 104 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;V NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 745

3$: BCC 4$
MOV #105,@#$FATAL ;MOVE TO MAILBOX # ***** 105 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;C NOT CLEARED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 737

4$: .WORD 106737
.WORD STATUS
BIT #340,STATUS ;TEST PRIORITY
BEQ 5$
MOV #106,@#$FATAL ;MOVE TO MAILBOX # ***** 106 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;PRIORITY NOT ZERO
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 724

5$: MOV #BUFF,SP
MOV #RETF2,RTRAP2
MOV #357,RTRAP2+2 ;SET NEW 'CC' AND PRIORITY
IOT ;TRAP HERE

RETG2:

```

810	004416	100405			BMI	1\$		
(3)	004420	012737	000107	000402	MOV	#107,@#\$FATAL	:MOVE TO MAILBOX # ***** 107 *****	
(2)	004426	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	004430	000000			HALT		:N NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 705	
811	004432					1\$:		
(1)	004432	001405			BEQ	2\$		
(3)	004434	012737	000110	000402	MOV	#110,@#\$FATAL	:MOVE TO MAILBOX # ***** 110 *****	
(2)	004442	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	004444	000000			HALT		:Z NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 677	
812	004446					2\$:		
(1)	004446	102405			BVS	3\$		
(3)	004450	012737	000111	000402	MOV	#111,@#\$FATAL	:MOVE TO MAILBOX # ***** 111 *****	
(2)	004456	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	004460	000000			HALT		:V NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 671	
813	004462					3\$:		
(1)	004462	103405			BCS	4\$		
(3)	004464	012737	000112	000402	MOV	#112,@#\$FATAL	:MOVE TO MAILBOX # ***** 112 *****	
(2)	004472	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	004474	000000			HALT		:C NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 663	
814	004476					4\$:		
(1)	004476	106737			.WORD	106737		
(1)	004500	013116			.WORD	STATUS		
815	004502	016706	006410		MOV	STATUS,SP		
816	004506	042706	000017		BIC	#17,SP		
817	004512	022706	000340		CMP	#340,SP		
818	004516	001405			BEQ	STPA		
(1)	004520					STP:		
(3)	004520	012737	000113	000402	MOV	#113,@#\$FATAL	:MOVE TO MAILBOX # ***** 113 *****	
(2)	004526	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	004530	000000			HALT		:PRIORITY WAS CHANGED,OR WRONG \$TESTN	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 645	
819	004532	012767	000022	173260	STPA:	MOV	#22,20	
820	004540	005067	173256		CLR	22		

```
822
823 :*****
(2) :TEST 25 TEST THAT A TRAP OCCURS ON AN EMT RESTRICTED INSTRUCTION
(3) :*****
(2) 004544 005237 000404 TST25: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 004550 022737 000025 000404 CMP #25,@#$TESTN ;SEQUENCE ERROR?
(2) 004556 001006 BNE TST26-12 ;BR TO ERROR HALT ON SEQ ERROR
824 004560 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
825 004564 012767 004606 173236 MOV #RETA3,RTRAP3 ;RETURN LOCATION
826 004572 104000 EMT ;RESERVE INSTRUCTION, SHOULD TRAP
827 004574 012737 000114 000402 MOV #114,@#$FATAL ;MOVE TO MAILBOX # ***** 114 *****
(2) 004602 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 004604 000000 HALT ;EMT DID NOT TRAP,OR WRONG $TESTN
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 764
828 004606 RETA3:
829 :*****
(2) :TEST 26 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
(3) :*****
(2) 004606 005237 000404 TST26: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 004612 022737 000026 000404 CMP #26,@#$TESTN ;SEQUENCE ERROR?
(2) 004620 001011 BNE TST27-12 ;BR TO ERROR HALT ON SEQ ERROR
830 004622 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
831 004626 012767 004636 173174 MOV #RETB3,RTRAP3 ;RETURN POINTER
832 004634 104000 EMT ;RESERVED INSTRUCTION
833 004636 020627 000474 RETB3: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
834 004642 001405 BEQ TST27
(4) 004644 012737 000115 000402 MOV #115,@#$FATAL ;MOVE TO MAILBOX # ***** 115 *****
(3) 004652 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 004654 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 761
835 :*****
(2) :TEST 27 TEST THAT PROPER P.C. IS SAVED
(3) :*****
(2) 004656 005237 000404 TST27: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 004662 022737 000027 000404 CMP #27,@#$TESTN ;SEQUENCE ERROR?
(2) 004670 001012 BNE TST30-12 ;BR TO ERROR HALT ON SEQ ERROR
836 004672 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
837 004676 012767 004706 173124 MOV #RETC3,RTRAP3 ;RTURN FROM TRAP POINTER
838 004704 104000 EMT ;TRAP ON THIS INSTRICION
839 004706 022767 004706 173560 RETC3: CMP #,BUFF-4 ;CHECK FOR INCREMENTED P.C.
840 004714 001405 BEQ TST30
(4) 004716 012737 000116 000402 MOV #116,@#$FATAL ;MOVE TO MAILBOX # ***** 116 *****
(3) 004724 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 004726 000000 HALT ;INCORRECT P.C.,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 760
```



```
842  
843  
(2) :*****  
(3) :TEST 30 TEST THAT 'OLD' STATUS AND PRIORITY ARE PLACED ON STACK  
(2) :*****  
(2) 004730 005237 000404 TST30: INC @#$TESTN ;UPDATE TEST NUMBER  
(2) 004734 022737 000030 000404 CMP #30,@#$TESTN ;SEQUENCE ERROR?  
(2) 004742 001044 BNE TST31-12 ;BR TO ERROR HALT ON SEQ ERROR  
844 004744 012706 000500 MOV #BUFF,SP ;SET UP  
845 004750 012767 004772 173052 MOV #RETD3,RTRAP3 ;SET UP  
846 004756 005067 006134 CLR STATUS ;CLEAR STATUS AND PRIORITY  
847 004762 106437 .WORD 106437  
(1) 004764 013116 .WORD STATUS  
848 004766 000257 CCC  
849 004770 104000 EMT ;TRAP  
850 004772 026727 173500 000000 RETD3: CMP BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK  
851 005000 001405 BEQ 1$  
(3) 005002 012737 000117 000402 MOV #117,@#$FATAL ;MOVE TO MAILBOX # ***** 117 *****  
(2) 005010 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR  
(2) 005012 000000 HALT ;INCORRECT STATUS  
(4) ; TO SCOPE REPLACE HALT W/ 240  
(4) ; AND REPLACE NEXT INST W/ 753  
852 005014 012706 000500 1$: MOV #BUFF,SP ;SET UP  
853 005020 012767 005044 173002 MOV #RETE3,RTRAP3 ;SET UP  
854 005026 012767 000357 006062 MOV #357,STATUS ;SET PRIORITY  
855 005034 106437 .WORD 106437  
(1) 005036 013116 .WORD STATUS  
856 005040 000277 SCC ;SET CC  
857 005042 104000 EMT ;TRAP  
858 005044 026727 173426 000357 RETE3: CMP BUFF-2,#357 ;COMPARES STATUS ON STACK  
859 005052 001405 BEQ TST31  
(4) 005054 012737 000120 000402 MOV #120,@#$FATAL ;MOVE TO MAILBOX # ***** 120 *****  
(3) 005062 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR  
(3) 005064 000000 HALT ;INCORRECT STATUS ON STACK,OR WRONG $TESTN  
(5) ; TO SCOPE REPLACE HALT W/ 240  
(5) ; AND REPLACE NEXT INST W/ 726
```

```
861
862 :*****
(2) :TEST 31 TEST THAT 'NEW' STATUS IS CORRECT
(3) :*****
(2) 005066 005237 000404 TST31: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 005072 022737 000031 000404 CMP #31,@#$TESTN ;SEQUENCE ERROR?
(2) 005100 001125 BNE TST32-12 ;BR TO ERROR HALT ON SEQ ERROR
863 005102 012706 000500 MOV #BUFF,SP
864 005106 012767 005122 172714 MOV #RETF3,RTRAP3
865 005114 005067 172712 CLR RTRAP3+2 ;CLEAR FUTURE PRIORITY AND CC
866 005120 104000 EMT
867 005122 RETF3: ;TEST FOR 'C' CLEARED
868 005122 100005 BPL 1$
(3) 005124 012737 000121 000402 MOV #121,@#$FATAL ;MOVE TO MAILBOX # ***** 121 *****
(2) 005132 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 005134 000000 HALT ;C NOT CLEARED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 761
869 005136 1$:
(1) 005136 001005 BNE 2$
(3) 005140 012737 000122 000402 MOV #122,@#$FATAL ;MOVE TO MAILBOX # ***** 122 *****
(2) 005146 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 005150 000000 HALT ;Z NOT CLEARED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 753
870 005152 2$:
(1) 005152 102005 BVC 3$
(3) 005154 012737 000123 000402 MOV #123,@#$FATAL ;MOVE TO MAILBOX # ***** 123 *****
(2) 005162 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 005164 000000 HALT ;V NOT CLEARED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 745
871 005166 3$:
(1) 005166 103005 BCC 4$
(3) 005170 012737 000124 000402 MOV #124,@#$FATAL ;MOVE TO MAILBOX # ***** 124 *****
(2) 005176 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 005200 000000 HALT ;C NOT CLEARED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 737
872 005202 4$:
(1) 005202 106737 .WORD 106737
(1) 005204 013116 .WORD STATUS
873 005206 032767 000340 005702 BIT #340,STATUS ;TEST PRIORITY
874 005214 001405 BEQ 5$
(3) 005216 012737 000125 000402 MOV #125,@#$FATAL ;MOVE TO MAILBOX # ***** 125 *****
(2) 005224 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 005226 000000 HALT ;PRIORITY NOT ZERO
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 724
875 005230 5$: MOV #BUFF,SP
876 005234 012767 005252 172566 MOV #RETF3,RTRAP3
877 005242 012767 000357 172562 MOV #357,RTRAP3+2 ;SET NEW 'CC' AND PRIORITY
878 005250 104000 EMT ;TRAP HERE
879 005252 RETG3:
```

880	005252	100405			BMI	1\$		
(3)	005254	012737	000126	000402	MOV	#126,@#\$FATAL	:MOVE TO MAILBOX # ***** 126 *****	
(2)	005262	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	005264	000000			HALT		:N NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 705	
881	005266					1\$:		
(1)	005266	001405			BEQ	2\$		
(3)	005270	012737	000127	000402	MOV	#127,@#\$FATAL	:MOVE TO MAILBOX # ***** 127 *****	
(2)	005276	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	005300	000000			HALT		:Z NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 677	
882	005302					2\$:		
(1)	005302	102405			BVS	3\$		
(3)	005304	012737	000130	000402	MOV	#130,@#\$FATAL	:MOVE TO MAILBOX # ***** 130 *****	
(2)	005312	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	005314	000000			HALT		:V NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 671	
883	005316					3\$:		
(1)	005316	103405			BCS	4\$		
(3)	005320	012737	000131	000402	MOV	#131,@#\$FATAL	:MOVE TO MAILBOX # ***** 131 *****	
(2)	005326	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	005330	000000			HALT		:C NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 663	
884	005332					4\$:		
(1)	005332	106737			.WORD	106737		
(1)	005334	013116			.WORD	STATUS		
885	005336	016706	005554		MOV	STATUS,SP		
886	005342	042706	000017		BIC	#17,SP		
887	005346	022706	000340		CMF	#340,SP		
888	005352	001405			BEQ	TST32		
(4)	005354	012737	000132	000402	MOV	#132,@#\$FATAL	:MOVE TO MAILBOX # ***** 132 *****	
(3)	005362	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(3)	005364	000000			HALT		:PRIORITY WAS CHANGED,OR WRONG \$TESTN	
(5)							: TO SCOPE REPLACE HALT W/ 240	
(5)							: AND REPLACE NEXT INST W/ 645	

```

890
891      :*****
(2)      :TEST 32      TEST THAT ALL COMBINATION OF EMT WILL CAUSE A TRAP
(3)      :*****
(2) 005366 005237 000404 TST32: INC @#$TESTN      ;UPDATE TEST NUMBER
(2) 005372 022737 000032 000404      CMP #32,@#$TESTN      ;SEQUENCE ERROR?
(2) 005400 001011      BNE RBBB      ;BR TO ERROR HALT ON SEQ ERROR
892 005402 012767 104000 000012      MOV #EMT,RB      ;INITIALIZE BASE EMT INSTRUCTION
893 005410 012767 005436 172412      MOV #RA,30      ;RETURN FROM TRAP TO RA
894 005416 012706 000500      RC: MOV #BUFF,SP      ;SET UP STACK POINTER
895 005422 104000      RB: EMT      ;TRAP INST. WILL BE MODIFIED TO EMT+377
896 005424      RBBB:
(3) 005424 012737 000133 000402      MOV #133,@#$FATAL      ;MOVE TO MAILBOX # ***** 133 *****
(2) 005432 005212      INC (R2)      ;SET MSGTYP TO FATAL ERROR
(2) 005434 000000      HALT      ;PREVIOUS INST FAILED TO TRAP,OR WRONG $TESTN
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 761
897 005436 005267 177760      RA: INC RB
898 005442 022767 104377 177752      CMP #104377,RB      ;EMT+377 TO EMT?
899 005450 103362      BHIS RC      ;HAVE WE TESTED ALL
900 005452 012767 000032 172350      MOV #32,30
901 005460 005067 172346      CLR 32      ;HALT
902      :*****
(2)      :TEST 33      TEST THAT A TRAP OCCURES ON AN 'BPT' INSTRUCTION
(3)      :*****
(2) 005464 005237 000404 TST33: INC @#$TESTN      ;UPDATE TEST NUMBER
(2) 005470 022737 000033 000404      CMP #33,@#$TESTN      ;SEQUENCE ERROR?
(2) 005476 001006      BNE TST34-12      ;BR TO ERROR HALT ON SEQ ERROR
903 005500 012706 000500      MOV #BUFF,SP      ;STACK POINTER SETUP
904 005504 012767 005526 172302      MOV #RETA4,RTRAP4      ;RETURN LOCATION
905 005512 000003      TRT      ;RESERVED INSTRUCTION, SHOULD TRAP
906 005514 012737 000134 000402      MOV #134,@#$FATAL      ;MOVE TO MAILBOX # ***** 134 *****
(2) 005522 005212      INC (R2)      ;SET MSGTYP TO FATAL ERROR
(2) 005524 000000      HALT      ;DID NOT TRAP,OR WRONG $TESTN
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 764
907 005526      RETA4:
    
```

```

909
910      :*****
(2)      :TEST 34      TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
(3)      :*****
(2) 005526 005237 000404      TST34: INC      @#$TESTN      :UPDATE TEST NUMBER
(2) 005532 022737 000034 000404      CMP      #34,@#$TESTN      :SEQUENCE ERROR?
(2) 005540 001011      BNE      TST35-12      :BR TO ERROR HALT ON SEQ ERROR
911 005542 012706 000500      MOV      #BUFF,SP      :STACK POINTER SETUP
912 005546 012767 005556 172240      MOV      #RETB4,RTRAP4      :RETURN POINTER
913 005554 000003      TRT      :RESERVED INSTRUCTION
914 005556 020627 000474      RETB4: CMP      SP,#BUFF-4      :TEST DECREMENT OF SP
915 005562 001405      BEQ      TST35
(4) 005564 012737 000135 000402      MOV      #135,@#$FATAL      :MOVE TO MAILBOX # ***** 135 *****
(3) 005572 005212      INC      (R2)      :SET MSGTYP TO FATAL ERROR
(3) 005574 000000      HALT      :NOT DECREMENTED TWO WORDS,OR WRONG $TESTN
(5)      : TO SCOPE REPLACE HALT W/ 240
(5)      : AND REPLACE NEXT INST W/ 761
916      :*****
(2)      :TEST 35      TEST THAT PROPER P.C. IS SAVED
(3)      :*****
(2) 005576 005237 000404      TST35: INC      @#$TESTN      :UPDATE TEST NUMBER
(2) 005602 022737 000035 000404      CMP      #35,@#$TESTN      :SEQUENCE ERROR?
(2) 005610 001012      BNE      TST36-12      :BR TO ERROR HALT ON SEQ ERROR
917 005612 012706 000500      MOV      #BUFF,SP      :STACK POINTER SETUP
918 005616 012767 005626 172170      MOV      #RETC4,RTRAP4      :RETURN FROM TRAP POINTER
919 005624 000003      TRT      :TRAP ON THIS INSTRUCTION
920 005626 022767 005626 172640      RETC4: CMP      #,BUFF-4      :CHECK FOR INCREMENTED P.C.
921 005634 001405      BEQ      TST36
(4) 005636 012737 000136 000402      MOV      #136,@#$FATAL      :MOVE TO MAILBOX # ***** 136 *****
(3) 005644 005212      INC      (R2)      :SET MSGTYP TO FATAL ERROR
(3) 005646 000000      HALT      :INCORRECT P.C.,OR WRONG $TESTN
(5)      : TO SCOPE REPLACE HALT W/ 240
(5)      : AND REPLACE NEXT INST W/ 760
    
```

```

923
924
(2)
(3)
(2) 005650 005237 000404
(2) 005654 022737 000036 000404
(2) 005662 001044
925 005664 012706 000500
926 005670 012767 005712 172116
927 005676 005067 005214
928 005702 106437
(1) 005704 013116
929 005706 000257
930 005710 000003
931 005712 026727 172560 000000 RETD4:
932 005720 001405
(3) 005722 012737 000137 000402
(2) 005730 005212
(2) 005732 000000
(4)
(4)
933 005734 012706 000500 1$:
934 005740 012767 005764 172046
935 005746 012767 000357 005142
936 005754 106437
(1) 005756 013116
937 005760 000277
938 005762 000003
939 005764 026727 172506 000357 RETE4:
940 005772 001405
(4) 005774 012737 000140 000402
(3) 006002 005212
(3) 006004 000000
(5)
(5)
:*****
:TEST 36 TEST THAT 'OLD' STATUS AND PRIORITY ARE PLACED ON STACK
:*****
TST36: INC @#$TESTN ;UPDATE TEST NUMBER
CMP #36,@#$TESTN ;SEQUENCE ERROR?
BNE TST37-12 ;BR TO ERROR HALT ON SEQ ERROR
MOV #BUFF,SP ;SET UP
MOV #RETD4,RTRAP4 ;SET UP
CLR STATUS ;CLEAR STATUS AND PRIORITY
.WORD 106437
.WORD STATUS
CCC
TRT ;TRAP
CMP BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK
BEQ 1$
MOV #137,@#$FATAL ;MOVE TO MAILBOX # ***** 137 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INCORRECT STATUS
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 753
MOV #BUFF,SP ;SET UP
MOV #RETE4,RTRAP4 ;SET UP
MOV #357,STATUS ;SET PRIORITY
.WORD 106437
.WORD STATUS
SCC ;SET-SET CC
TRT ;TRAP
CMP BUFF-2,#357 ;COMPARES STATUS ON STACK
BEQ TST37
MOV #140,@#$FATAL ;MOVE TO MAILBOX # ***** 140 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INCORRECT STATUS ON STACK,OR WRONG $TESTN
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 726
    
```

```

942
943
(2)
(3)
(2) 006006 005237 000404
(2) 006012 022737 000037 000404
(2) 006020 001125
944 006022 012706 000500
945 006026 012767 006042 171760
946 006034 005067 171756
947 006040 000003
948 006042
949 006042 100005
(3) 006044 012737 000141 000402
(2) 006052 005212
(2) 006054 000000
(4)
(4)
950 006056
(1) 006056 001005
(3) 006060 012737 000142 000402
(2) 006066 005212
(2) 006070 000000
(4)
(4)
951 006072
(1) 006072 102005
(3) 006074 012737 000143 000402
(2) 006102 005212
(2) 006104 000000
(4)
(4)
952 006106
(1) 006106 103005
(3) 006110 012737 000144 000402
(2) 006116 005212
(2) 006120 000000
(4)
(4)
953 006122
(1) 006122 106737
(1) 006124 013116
954 006126 032767 000340 004762
955 006134 001405
(3) 006136 012737 000145 000402
(2) 006144 005212
(2) 006146 000000
(4)
(4)
956 006150 012706 000500
957 006154 012767 006172 171632
958 006162 012767 000357 171626
959 006170 000003
960 006172

```

:*****
 :TEST 37 TEST THAT 'NEW' STATUS IS CORRECT
 :*****
 TST37: INC @#\$TESTN ;UPDATE TEST NUMBER
 CMP #37,@#\$TESTN ;SEQUENCE ERROR?
 BNE RSTP2 ;BR TO ERROR HALT ON SEQ ERROR
 MOV #BUFF,SP
 MOV #RETF4,RTRAP4
 CLR RTRAP4+2 ;CLEAR FUTURE PRIORITY AND CC
 TRT
 RETF4:
 BPL 1\$
 MOV #141,@#\$FATAL ;MOVE TO MAILBOX # ***** 141 *****
 INC (R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;C NOT CLEARED
 ; TO SCOPE REPLACE HALT W/ 240
 ; AND REPLACE NEXT INST W/ 761
 1\$:
 BNE 2\$
 MOV #142,@#\$FATAL ;MOVE TO MAILBOX # ***** 142 *****
 INC (R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;Z NOT CLEARED
 ; TO SCOPE REPLACE HALT W/ 240
 ; AND REPLACE NEXT INST W/ 753
 2\$:
 BVC 3\$
 MOV #143,@#\$FATAL ;MOVE TO MAILBOX # ***** 143 *****
 INC (R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;V NOT CLEARED
 ; TO SCOPE REPLACE HALT W/ 240
 ; AND REPLACE NEXT INST W/ 745
 3\$:
 BCC 4\$
 MOV #144,@#\$FATAL ;MOVE TO MAILBOX # ***** 144 *****
 INC (R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;C NOT CLEARED
 ; TO SCOPE REPLACE HALT W/ 240
 ; AND REPLACE NEXT INST W/ 737
 4\$:
 .WORD 106737
 .WORD STATUS
 BIT #340,STATUS ;TEST PRIORITY
 BEQ 5\$
 MOV #145,@#\$FATAL ;MOVE TO MAILBOX # ***** 145 *****
 INC (R2) ;SET MSGTYP TO FATAL ERROR
 HALT ;PRIORITY NOT ZERO
 ; TO SCOPE REPLACE HALT W/ 240
 ; AND REPLACE NEXT INST W/ 724
 5\$:
 MOV #BUFF,SP
 MOV #RETG4,RTRAP4
 MOV #357,RTRAP4+2 ;SET NEW 'CC' AND PRIORITY
 TRT ;TRAP HERE
 RETG4:


```

974
975 ;PDP-11 ILLEGAL AND ADDRESS INSTRUCTION TEST
976 ;ALL INSTRUCTIONS THAT ARE RESERVED
977 ;SHOULD TRAP TO LOCATION 4, AND THE
978 ;PC THAT POINTS TO THE TRAPPING INSTRUCTION
979 ;SHOULD BE PLACED ON THE STACK
980
981 ;*****
981 (2) ;TEST 40 TEST THAT A TRAP OCCURS ON AN ILLEGAL INSTRUCTION
981 (3) ;*****
(2) 006320 005237 000404 TST40: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 006324 022737 000040 000404 CMP #40,@#$TESTN ;SEQUENCE ERROR?
(2) 006332 001006 BNE TST41-12 ;BR TO ERROR HALT ON SEQ ERROR
982 006334 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
983 006340 012767 006362 171436 MOV #RETA5,RTRAP5 ;RETURN LOCATION
984 006346 000100 JMP %0 ;ILLEGAL INSTRUCTION, SHOULD TRAP
985 006350 012737 000153 000402 MOV #153,@#$FATAL ;MOVE TO MAILBOX # ***** 153 *****
(2) 006356 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 006360 000000 HALT ;DID NOT TRAP,OR WRONG $TESTN
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 764
986 006362 RETA5:
987 ;*****
(2) ;TEST 41 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
(3) ;*****
(2) 006362 005237 000404 TST41: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 006366 022737 000041 000404 CMP #41,@#$TESTN ;SEQUENCE ERROR?
(2) 006374 001011 BNE TST42-12 ;BR TO ERROR HALT ON SEQ ERROR
988 006376 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
989 006402 012767 006412 171374 MOV #RETB5,RTRAP5 ;RETURN POINTER
990 006410 000100 JMP %0 ;RESERVED INSTRUCTION
991 006412 020627 000474 RETB5: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
992 006416 001405 BEQ TST42
(4) 006420 012737 000154 000402 MOV #154,@#$FATAL ;MOVE TO MAILBOX # ***** 154 *****
(3) 006426 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 006430 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 761
    
```

```
994
995 ;*****
(2) ;TEST 42 TEST THAT PROPER P.C. IS SAVED
(3) ;*****
(2) 006432 005237 000404 TST42: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 006436 022737 000042 000404 CMP #42,@#$TESTN ;SEQUENCE ERROR?
(2) 006444 001012 BNE TST43-12 ;BR TO ERROR HALT ON SEQ ERROR
996 006446 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
997 006452 012767 006462 171324 MOV #RETC5,RTRAP5 ;RETURN FROM TRAP POINTER
998 006460 000100 JMP %0 ;TRAP ON THIS INSTRUCTION
999 006462 022767 006462 172004 RETC5: CMP #.BUFF-4 ;CHECK FOR INCREMENTED P.C.
1000 006470 001405 BEQ TST43
(4) 006472 012737 000155 000402 MOV #155,@#$FATAL ;MOVE TO MAILBOX # ***** 155 *****
(3) 006500 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 006502 000000 HALT ;INCORRECT P.C.,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 760
1001 ;*****
(2) ;TEST 43 TEST THAT 'OLD' STATUS AND PRIORITY ARE PLACED ON STACK
(3) ;*****
(2) 006504 005237 000404 TST43: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 006510 022737 000043 000404 CMP #43,@#$TESTN ;SEQUENCE ERROR?
(2) 006516 001044 BNE TST44-12 ;BR TO ERROR HALT ON SEQ ERROR
1002 006520 012706 000500 MOV #BUFF,SP ;SET UP
1003 006524 012767 006546 171252 MOV #RETD5,RTRAP5 ;SET UP
1004 006532 005067 004360 CLR STATUS ;CLEAR STATUS AND PRIORITY
1005 006536 106437 .WORD 106437
(1) 006540 013116 .WORD STATUS
1006 006542 000257 CCC
1007 006544 000100 JMP %0 ;TRAP
1008 006546 026727 171724 000000 RETD5: CMP BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK
1009 006554 001405 BEQ 1$
(3) 006556 012737 000156 000402 MOV #156,@#$FATAL ;MOVE TO MAILBOX # ***** 156 *****
(2) 006564 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 006566 000000 HALT ;INCORRECT STATUS
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 753
1010 006570 012706 000500 1$: MOV #BUFF,SP ;SET UP
1011 006574 012767 006620 171202 MOV #RETE5,RTRAP5 ;SET UP
1012 006602 012767 000357 004306 MOV #357,STATUS ;SET PRIORITY
1013 006610 106437 .WORD 106437
(1) 006612 013116 .WORD STATUS
1014 006614 000277 SCC ;SET CC
1015 006616 000100 JMP %0 ;TRAP
1016 006620 026727 171652 000357 RETE5: CMP BUFF-2,#357 ;COMPARES STATUS ON STACK
1017 006626 001405 BEQ TST44
(4) 006630 012737 000157 000402 MOV #157,@#$FATAL ;MOVE TO MAILBOX # ***** 157 *****
(3) 006636 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 006640 000000 HALT ;INCORRECT STATUS ON STACK,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 726
```

```

1019
1020      ;*****
(2)      ;TEST 44      TEST THAT 'NEW' STATUS IS CORRECT
(3)      ;*****
(2) 006642 005237 000404      TST44: INC      @#$TESTN      ;UPDATE TEST NUMBER
(2) 006646 022737 000044 000404      CMP      #44,@#$TESTN      ;SEQUENCE ERROR?
(2) 006654 001123      BNE      TST45-12      ;BR TO ERROR HALT ON SEQ ERROR
1021 006656 012706 000500      MOV      #BUFF,SP
1022 006662 012767 006676 171114      MOV      #RET5,RTRAP5
1023 006670 005067 171112      CLR      RTRAP5+2      ;CLEAR FUTURE PRIORITY AND CC
1024 006674 000100      JMP      %0
1025      RETF5:
1026 006676 100005      BPL      1$
(3) 006700 012737 000160 000402      MOV      #160,@#$FATAL      ;MOVE TO MAILBOX # ***** 160 *****
(2) 006706 005212      INC      (R2)      ;SET MSGTYP TO FATAL ERROR
(2) 006710 000000      HALT      ;C NOT CLEARED
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 761
1027 006712      1$:
(1) 006712 001005      BNE      2$
(3) 006714 012737 000161 000402      MOV      #161,@#$FATAL      ;MOVE TO MAILBOX # ***** 161 *****
(2) 006722 005212      INC      (R2)      ;SET MSGTYP TO FATAL ERROR
(2) 006724 000000      HALT      ;Z NOT CLEARED
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 753
1028 006726      2$:
(1) 006726 102005      BVC      3$
(3) 006730 012737 000162 000402      MOV      #162,@#$FATAL      ;MOVE TO MAILBOX # ***** 162 *****
(2) 006736 005212      INC      (R2)      ;SET MSGTYP TO FATAL ERROR
(2) 006740 000000      HALT      ;V NOT CLEARED
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 745
1029 006742      3$:
(1) 006742 103005      BCC      4$
(3) 006744 012737 000163 000402      MOV      #163,@#$FATAL      ;MOVE TO MAILBOX # ***** 163 *****
(2) 006752 005212      INC      (R2)      ;SET MSGTYP TO FATAL ERROR
(2) 006754 000000      HALT      ;C NOT CLEARED
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 737
1030 006756      4$:
(1) 006756 106737      .WORD    106737
(1) 006760 013116      .WORD    STATUS
1031 006762 032767 000357 004126      BIT      #357,STATUS      ;TEST PRIORITY
1032 006770 001405      BEQ      5$
(3) 006772 012737 000164 000402      MOV      #164,@#$FATAL      ;MOVE TO MAILBOX # ***** 164 *****
(2) 007000 005212      INC      (R2)      ;SET MSGTYP TO FATAL ERROR
(2) 007002 000000      HALT      ;PRIORITY NOT ZERO
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 724
1033 007004 012706 000500      5$: MOV      #BUFF,SP
1034 007010 012767 007026 170766      MOV      #RETG5,RTRAP5
1035 007016 012767 000357 170762      MOV      #357,RTRAP5+2      ;SET NEW 'CC' AND PRIORITY
1036 007024 000100      JMP      %0      ;TRAP HERE
1037      RETG5:
    
```

1038	007026	100405			BMI	1\$		
(3)	007030	012737	000165	000402	MOV	#165,@#\$FATAL	:MOVE TO MAILBOX # ***** 165 *****	
(2)	007036	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	007040	000000			HALT		:N NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 705	
1039	007042					1\$:		
(1)	007042	001405			BEQ	2\$		
(3)	007044	012737	000166	000402	MOV	#166,@#\$FATAL	:MOVE TO MAILBOX # ***** 166 *****	
(2)	007052	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	007054	000000			HALT		:Z NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 677	
1040	007056					2\$:		
(1)	007056	102405			BVS	3\$		
(3)	007060	012737	000167	000402	MOV	#167,@#\$FATAL	:MOVE TO MAILBOX # ***** 167 *****	
(2)	007066	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	007070	000000			HALT		:V NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 671	
1041	007072					3\$:		
(1)	007072	103405			BCS	4\$		
(3)	007074	012737	000170	000402	MOV	#170,@#\$FATAL	:MOVE TO MAILBOX # ***** 170 *****	
(2)	007102	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(2)	007104	000000			HALT		:C NOT SET	
(4)							: TO SCOPE REPLACE HALT W/ 240	
(4)							: AND REPLACE NEXT INST W/ 663	
1042	007106					4\$:		
(1)	007106	106737			.WORD	106737		
(1)	007110	013116			.WORD	STATUS		
1043	007112	016706	004000		MOV	STATUS,SP		
1044	007116	022706	000357		CMP	#357,SP		
1045	007122	001405			BEQ	TST45		
(4)	007124	012737	000171	000402	MOV	#171,@#\$FATAL	:MOVE TO MAILBOX # ***** 171 *****	
(3)	007132	005212			INC	(R2)	:SET MSGTYP TO FATAL ERROR	
(3)	007134	000000			HALT		:PRIORITY WAS CHANGED,OR WRONG \$TESTN	
(5)							: TO SCOPE REPLACE HALT W/ 240	
(5)							: AND REPLACE NEXT INST W/ 647	

```

1047
1048 :*****
(2) :TEST 45 TEST THAT A TRAP OCCURES ON ALL ILLEGAL INSTRUION
(3) :*****
(2) 007136 005237 000404 TST45: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 007142 022737 000045 000404 CMP #45,@#$TESTN ;SEQUENCE ERROR?
(2) 007150 001006 BNE TST46-12 ;BR TO ERROR HALT ON SEQ ERROR
1049 007152 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
1050 007156 012767 007200 170620 MOV #RETH5,RTRAP5 ;RETURN LOCATION
1051 007164 004000 JSR %0,%0 ;RESERVED INSTRUCTION, SHOULD TRAP
1052 007166 012737 000172 000402 MOV #172,@#$FATAL ;MOVE TO MAILBOX # ***** 172 *****
(2) 007174 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 007176 000000 HALT ;DID NOT TRAP,OR WRONG $TESTN
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 764
1053 007200 RETH5:
1054 :*****
(2) :TEST 46 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
(3) :*****
(2) 007200 005237 000404 TST46: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 007204 022737 000046 000404 CMP #46,@#$TESTN ;SEQUENCE ERROR?
(2) 007212 001011 BNE TST47-12 ;BR TO ERROR HALT ON SEQ ERROR
1055 007214 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
1056 007220 012767 007230 170556 MOV #RETJ,RTRAP5 ;RETURN POINTER
1057 007226 004000 JSR %0,%0 ;RESERVED INSTRUCTION
1058 007230 020627 000474 RETJ: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
1059 007234 001405 BEQ TST47
(4) 007236 012737 000173 000402 MOV #173,@#$FATAL ;MOVE TO MAILBOX # ***** 173 *****
(3) 007244 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 007246 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 761
    
```

```
1061
1062 :*****
(2) :TEST 47 TEST THAT PROPER P.C. IS SAVED
(3) :*****
(2) 007250 005237 000404 TST47: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 007254 022737 000047 000404 CMP #47,@#$TESTN ;SEQUENCE ERROR?
(2) 007262 001012 BNE TST50-12 ;BR TO ERROR HALT ON SEQ ERROR
1063 007264 012706 000500 MOV #BUFF,SP ;STACK POINTER SETUP
1064 007270 012767 007300 170506 MOV #RETK,RTRAP5 ;RETURN FROM TRAP POINTER
1065 007276 004000 INSTK: JSR %0,%0 ;TRAP ON THIS INSTRUCTION
1066 007300 022767 007300 171166 RETK: CMP #INSTK+2,BUFF-4 ;CHECK FOR INCREMEND P.C.
1067 007306 001405 BEQ TST50
(4) 007310 012737 000174 000402 MOV #174,@#$FATAL ;MOVE TO MAILBOX # ***** 174 *****
(3) 007316 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 007320 000000 HALT ;INCORRECT P.C.,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 760
1068
1069 :*****
(2) :TEST 50 TEST THAT 'OLD' STATUS AND PRIORITY ARE PLACED ON STACK
(3) :*****
(2) 007322 005237 000404 TST50: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 007326 022737 000050 000404 CMP #50,@#$TESTN ;SEQUENCE ERROR?
(2) 007334 001044 BNE TST51-12 ;BR TO ERROR HALT ON SEQ ERROR
1070 007336 012706 000500 MOV #BUFF,SP ;SET UP
1071 007342 012767 007364 170434 MOV #RETL,RTRAP5 ;SET UP
1072 007350 005067 003542 CLR STATUS ;CLEAR STATUS AND PRIORITY
1073 007354 106437 .WORD 106437
(1) 007356 013116 .WORD STATUS
1074 007360 000257 CCC
1075 007362 004000 JSR %0,%0 ;TRAP
1076 007364 026727 171106 000000 RETL: CMP BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK
1077 007372 001405 BEQ 1$
(3) 007374 012737 000175 000402 MOV #175,@#$FATAL ;MOVE TO MAILBOX # ***** 175 *****
(2) 007402 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 007404 000000 HALT ;INCORRECT STATUS
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 753
1078 007406 012706 000500 1$: MOV #BUFF,SP ;SET UP
1079 007412 012767 007436 170364 MOV #RETM,RTRAP5 ;SET UP
1080 007420 012767 000357 003470 MOV #357,STATUS ;SET PRIORITY
1081 007426 106437 .WORD 106437
(1) 007430 013116 .WORD STATUS
1082 007432 000277 SCC ;SET CC
1083 007434 004000 JSR %0,%0 ;TRAP
1084 007436 026727 171034 000357 RETM: CMP BUFF-2,#357 ;COMPARES STATUS ON STACK
1085 007444 001405 BEQ TST51
(4) 007446 012737 000176 000402 MOV #176,@#$FATAL ;MOVE TO MAILBOX # ***** 176 *****
(3) 007454 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 007456 000000 HALT ;INCORRECT STATUS ON STACK,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 726
1086
1087 :*****
```

H 4

CVKALA LSI-11 TRAP (30K+FIS) TEST MACY11 27(654) 19-SEP-78 11:25 PAGE 34-1
 CVKALA.P11 T50 TEST THAT 'OLD' STATUS AND PRIORITY ARE PLACED ON STACK SEQ 0046

```

(2) ;TEST 51 TEST THAT 'NEW' STATUS IS CORRECT
(3) ;*****
(2) 007460 005237 000404 TST51: INC @#STESTN ;UPDATE TEST NUMBER
(2) 007464 022737 000551 000404 CMP #51,@#STESTN ;SEQUENCE ERROR?
(2) 007472 001122 BNE STP1 ;BR TO ERROR HALT ON SEQ ERROR
1088 007474 012706 000500 MOV #BUFF,SP
1089 007500 012767 007514 170276 MOV #RETN,RTRAP5
1090 007506 005067 170274 CLR RTRAP5+2 ;CLEAR FUTURE PRIORITY AND CC
1091 007512 004000 JSR %0,%0
1092 007514 RETN: ;TEST FOR 'C' CLEARED
1093 007514 100005 BPL 1$
(3) 007516 012737 000177 000402 MOV #177,@#SFATAL ;MOVE TO MAILBOX # ***** 177 *****
(2) 007524 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 007526 000000 HALT ;C NOT CLEARED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 761
1094 007530 1$:
(1) 007530 001005 BNE 2$
(3) 007532 012737 000200 000402 MOV #200,@#SFATAL ;MOVE TO MAILBOX # ***** 200 *****
(2) 007540 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 007542 000000 HALT ;Z NOT CLEARED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 753
1095 007544 2$:
(1) 007544 102005 BVC 3$
(3) 007546 012737 000201 000402 MOV #201,@#SFATAL ;MOVE TO MAILBOX # ***** 201 *****
(2) 007554 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 007556 000000 HALT ;V NOT CLEARED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 745
1096 007560 3$:
(1) 007560 103005 BCC 4$
(3) 007562 012737 000202 000402 MOV #202,@#SFATAL ;MOVE TO MAILBOX # ***** 202 *****
(2) 007570 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 007572 000000 HALT ;C NOT CLEARED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 737
1097 007574 4$:
(1) 007574 106737 .WORD 106737
(1) 007576 013116 .WORD STATUS
1098 007600 016700 003312 MOV STATUS,%0 ;TEMP STORAGE
1099 007604 001405 BEQ 5$
(3) 007606 012737 000203 000402 MOV #203,@#SFATAL ;MOVE TO MAILBOX # ***** 203 *****
(2) 007614 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 007616 000000 HALT ;PRIORITY NOT ZERO
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 725
1100 007620 012706 000500 5$: MOV #BUFF,SP
1101 007624 012767 007642 170152 MOV #RETO,RTRAP5
1102 007632 012767 000357 170146 MOV #357,RTRAP5+2 ;SET NEW 'CC' AND PRIORITY
1103 007640 004000 JSR %0,%0 ;TRAP HERE
1104 007642 RETO:
1105 007642 100405 BMI 1$
(3) 007644 012737 000204 000402 MOV #204,@#SFATAL ;MOVE TO MAILBOX # ***** 204 *****

```

```

(2) 007652 005212          INC      (R2)          ;SET MSGTYP TO FATAL ERROR
(2) 007654 000000          HALT                    ;N NOT SET
(4)                                     ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 706
1106 007656 001405          1$:          BEQ      2$
(1) 007656 001405          MOV      #205,@#$FATAL ;MOVE TO MAILBOX # ***** 205 *****
(3) 007660 012737 000205 000402  INC      (R2)          ;SET MSGTYP TO FATAL ERROR
(2) 007666 005212          HALT                    ;Z NOT SET
(2) 007670 000000          ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 700
1107 007672 102405          2$:          BVS      3$
(1) 007672 102405          MOV      #206,@#$FATAL ;MOVE TO MAILBOX # ***** 206 *****
(3) 007674 012737 000206 000402  INC      (R2)          ;SET MSGTYP TO FATAL ERROR
(2) 007702 005212          HALT                    ;V NOT SET
(2) 007704 000000          ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 672
1108 007706 103405          3$:          BCS      4$
(1) 007706 103405          MOV      #207,@#$FATAL ;MOVE TO MAILBOX # ***** 207 *****
(3) 007710 012737 000207 000402  INC      (R2)          ;SET MSGTYP TO FATAL ERROR
(2) 007716 005212          HALT                    ;C NOT SET
(2) 007720 000000          ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 664
1109 007722 106737          4$:          .WORD   106737
(1) 007722 106737          .WORD   STATUS
(1) 007724 013116          MOV      STATUS,%0
1110 007726 016700 003164    MOV      #357,%0
1111 007732 022700 000357    CMP
1112 007736 001405          BEQ      STPB
(1) 007740 001405          STP1:      MOV      #210,@#$FATAL ;MOVE TO MAILBOX # ***** 210 *****
(3) 007740 012737 000210 000402  INC      (R2)          ;SET MSGTYP TO FATAL ERROR
(2) 007746 005212          HALT                    ;PRIORITY WAS CHANGED,OR WRONG $TESTN
(2) 007750 000000          ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 650
(4)
1113 007752 012767 000006 170024 STPB:      MOV      #6,4
1114 007760 005067 170022    CLR      6
    
```



```

1116
1117
(2)
(3)
(2) 007764 005237 000404
(2) 007770 022737 000052 000404
(2) 007776 001013
1118 010000 012706 000500
1119 010004 012767 010040 170002
1120 010012 012746 000020
1121 010016 012746 010024
1122 010022 000002
1123 010024 000240
1124 010026 012737 000211 000402
(2) 010034 005212
(2) 010036 000000
(4)
(4)
1125 010040
1126
(2)
(3)
(2) 010040 005237 000404
(2) 010044 022737 000053 000404
(2) 010052 001023
1127 010054 012706 000500
1128 010060 012767 010114 167726
1129 010066 012746 000020
1130 010072 012746 010100
1131 010076 000002
1132 010100 000240
1133 010102 012737 000212 000402
(2) 010110 005212
(2) 010112 000000
(4)
(4)
1134 010114 020627 000474
1135 010120 001405
(4) 010122 012737 000213 000402
(3) 010130 005212
(3) 010132 000000
(5)
(5)
:*****
:TEST 52 TEST THAT THE TRACE TRAP; (BIT4 20(8)) WILL CAUSE A TRAP TO 14
:*****
TST52: INC @#$TESTN ;UPDATE TEST NUMBER
CMP #52,@#$TESTN ;SEQUENCE ERROR?
BNE TST53-12 ;BR TO ERROR HALT ON SEQ ERROR
MOV #BUFF,SP
MOV #RETAT,RTRAP4 ;SET UP TO TRAP TO 14
MOV #20,-(SP) ;PUSH T BIT
MOV #.+6,-(SP) ;PUSH PC
RTI ;SET T BIT
NOP ;TRAP HERE
MOV #211,@#$FATAL ;MOVE TO MAILBOX # ***** 211 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TRACE BIT DID NOT TRAP!,OR WRONG $TESTN
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 757

RETAT:
:*****
:TEST 53 TEST STACK POINTER DECREMENTS
:*****
TST53: INC @#$TESTN ;UPDATE TEST NUMBER
CMP #53,@#$TESTN ;SEQUENCE ERROR?
BNE TST54-12 ;BR TO ERROR HALT ON SEQ ERROR
MOV #BUFF,SP
MOV #RETBT,RTRAP4
MOV #20,-(SP) ;PUSH T BIT
MOV #.+6,-(SP) ;PUSH PC
RTI ;SET T BIT
NOP ;TRAP HERE
MOV #212,@#$FATAL ;MOVE TO MAILBOX # ***** 212 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TRACE BIT DID NOT TRAP!
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 757

RETBT: CMP SP,#BUFF-4
BEQ TST54
MOV #213,@#$FATAL ;MOVE TO MAILBOX # ***** 213 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;STACK POINTER WAS NOT PUSHED BY TRAP,OR WRONG $TESTN
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 747
    
```

```

1137
1138 :*****
(2) :TEST 54 TEST FOR PROPER PC ON STACK
(3) :*****
(2) 010134 005237 000404 TST54: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 010140 022737 000054 000404 CMP #54,@#$TESTN ;SEQUENCE ERROR?
(2) 010146 001016 BNE TST55-12 ;BR TO ERROR HALT ON SEQ ERROR
1139 010150 012706 000500 MOV #BUFF,SP
1140 010154 012767 010174 167632 MOV #RETCT,RTRAP4
1141 010162 012746 000020 MOV #20,-(SP) ;PUSH T BIT
1142 010166 012746 010174 MOV #.+6,-(SP) ;PUSH PC
1143 010172 000002 RTI ;SET T BIT
1144 ;TRAP HERE
1145 010174 022767 010174 170272 RETCT: CMP #.,BUFF-4
1146 010202 001405 BEQ TST55
(4) 010204 012737 000214 000402 MOV #214,@#$FATAL ;MOVE TO MAILBOX # ***** 214 *****
(3) 010212 005212 INC ;SET MSGTYP TO FATAL ERROR
(3) 010214 000000 HALT ;CORRECT PC WAS NOT SAVED ON STACK,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 754
1147
1148
1149 :*****
(2) :TEST 55 TEST THAT RTT POPS T- BIT
(3) :*****
(2) 010216 005237 000404 TST55: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 010222 022737 000055 000404 CMP #55,@#$TESTN ;SEQUENCE ERROR?
(2) 010230 001015 BNE TST56-12 ;BR TO ERROR HALT ON SEQ ERROR
1150
1151 010232 012706 000500 MOV #BUFF,SP
1152 010236 005001 CLR R1 ;CLEAR R1
1153 010240 012746 000020 MOV #20,-(SP)
1154 010244 012746 010260 MOV #RTT1,-(SP)
1155 010250 012767 010276 167536 MOV #RTT2,14
1156 010256 000006 RTT
1157 010260 000240 RTT1: NOP
1158 010262 001405 BEQ TST56
(4) 010264 012737 000215 000402 MOV #215,@#$FATAL ;MOVE TO MAILBOX # ***** 215 *****
(3) 010272 005212 INC ;SET MSGTYP TO FATAL ERROR
(3) 010274 000000 HALT ;T-BIT DID NOT TRAP,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 755
1159
1160 010276 RTT2:
    
```

```
1162
1163 :*****
(2) :TEST 56 TEST THAT RTT ALLOWS ONE INST. BEFORE TRAP
(3) :*****
(2) 010276 005237 000404 TST56: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 010302 022737 000056 000404 CMP #56,@#$TESTN ;SEQUENCE ERROR?
(2) 010310 001031 BNE TST57-12 ;BR TO ERROR HALT ON SEQ ERROR
1164 010312 012705 177777 MOV #177777,%5
1165 010316 012706 000500 RTT5: MOV #BUFF,SP
1166 010322 012746 000020 MOV #20,-(SP)
1167 010326 012746 010344 MOV #RTT3,-(SP)
1168 010332 012767 010364 167454 MOV #RTT4,14
1169 010340 005001 CLR R1 ;CLEAR R0
1170 010342 000006 RTT ;SET T-BIT
1171 010344 005201 RTT3: INC R1
1172 010346 005205 INC %5
1173 010350 001762 BEQ RTT5 ;DO THIS TEST NO MORE THAN 2 TIMES
1174 010352 012737 000216 000402 MOV #216,@#$FATAL ;MOVE TO MAILBOX # ***** 216 *****
(2) 010360 005212 INC ;SET MSGTYP TO FATAL ERROR
(2) 010362 000000 HALT ;DID NOT TRAP
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 752
1175 010364 005301 RTT4: DEC R1 ;SEE IF RTT ALLOWS 1 INST.
1176 010366 001407 BEQ RTT6
1177 010370 005205 INC %5 ;DO THIS TEST NO MORE THAN TWO TIMES
1178 010372 001751 BEQ RTT5
(3) 010374 012737 000217 000402 MOV #217,@#$FATAL ;MOVE TO MAILBOX # ***** 217 *****
(2) 010402 005212 INC ;SET MSGTYP TO FATAL ERROR
(2) 010404 000000 HALT ;RTT DID NOT ALLOW 1 INST.,OR WRONG $TESTN
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 741
1179 010406 RTT6:
```

1181
 1182
 (2)
 (3)
 (2)
 (2)
 (2)
 1183
 1184
 1185
 1186
 1187
 1188
 1189
 1190
 (2)
 (2)
 (4)
 (4)
 1191
 1192
 1193
 (4)
 (3)
 (3)
 (5)
 (5)

010406 005237 000404
 010412 022737 000057 000404
 010420 001023
 010422 012706 000500
 010426 012746 000020
 010432 012746 010450
 010436 012767 010464 167350
 010444 005001
 010446 000002
 010450 005201
 010452 012737 000220 000402
 010460 005212
 010462 000000
 010464 005701
 010466 001405
 010470 012737 000221 000402
 010476 005212
 010500 000000

```

:*****
:TEST 57      TEST THAT RTI DOES NOT ALLOW 1 INST.
:*****
TST57:  INC    @#$TESTN      ;UPDATE TEST NUMBER
        CMP    #57,@#$TESTN ;SEQUENCE ERROR?
        BNE   TST60-12      ;BR TO ERROR HALT ON SEQ ERROR
        MOV   #BUFF,SP
        MOV   #20,-(SP)
        MOV   #RTI1,-(SP)
        MOV   #RTI2,14
        CLR   R1
        RTI
        ;SET T-BIT
RTI1:   INC    R1            ;RTI SHOULD NOT ALLOW THIS
        MOV   #220,@#$FATAL ;MOVE TO MAILBOX # ***** 220 *****
        INC   (R2)          ;SET MSGTYP TO FATAL ERROR
        HALT                ;T- BIT DID NOT CAUSE TRAP
        ; TO SCOPE REPLACE HALT W/ 240
        ; AND REPLACE NEXT INST W/ 756

RTI2:   TST    R1
        ;RTI SHOULD NOT ALLOW 1 INST. BEFORE TRAP
        BEQ   TST60
        MOV   #221,@#$FATAL ;MOVE TO MAILBOX # ***** 221 *****
        INC   (R2)          ;SET MSGTYP TO FATAL ERROR
        HALT                ;RTI DID ALLOW 1 INST. BEFORE TRAP,OR WRONG $TESTN
        ; TO SCOPE REPLACE HALT W/ 240
        ; AND REPLACE NEXT INST W/ 747
    
```

```

1195
1196
(2)
(3)
(2) 010502 005237 000404
(2) 010506 022737 000060 000404
(2) 010514 001033
1197
1198
1199 010516 012705 177777
1200 010522 012706 000500
1201 010526 012767 010600 167260
1202 010534 005027 000016
1203 010540 005027 000022
1204 010544 012767 010616 167246
1205 010552 012746 000020
1206 010556 012746 010564
1207 010562 000006
1208 010564 000004
1209 010566 012737 000222 000402
(2) 010574 005212
(2) 010576 000000
(4)
(4)
1210 010600 005205
1211 010602 001747
(1) 010604
(3) 010604 012737 000223 000402
(2) 010612 005212
(2) 010614 000000
(4)
(4)
1212 010616 012767 000016 167170
1213 010624 012767 000022 167166

```

```

:*****
:TEST 60 TEST TRAP ON TRAP
:*****
TST60: INC @#$TESTN ;UPDATE TEST NUMBER
CMP #60,@#$TESTN ;SEQUENCE ERROR?
BNE TRACE ;BR TO ERROR HALT ON SEQ ERROR
;TEST THAT TRACE BIT TRAPS ARE INHIBITED ON TRAP INST

TRPTRP: MOV #177777,%5
MOV #BUFF,%6
MOV #TRACE1,14 ;TRACE TRAP
CLR #16
CLR #22
MOV #TONT1,20 ;IOT TRAP
MOV #20,-(SP) ;PUSH T BIT
MOV #.+6,-(SP) ;PUSH PC
RTT ;SET T BIT
IOT ;TRAP, NEW STATUS HAVE TRACE RESET
MOV #222,@#$FATAL ;MOVE TO MAILBOX # ***** 222 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;NO TRAP OCCURRED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 746
; IF FAILED TRY THIS TEST TWICE BUT NO MORE

TRACE1: INC %5
BEQ TRPTRP

TRACE: MOV #223,@#$FATAL ;MOVE TO MAILBOX # ***** 223 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IOT SHOULD HAVE CLEARED THE T BIT,OR WRONG $TESTN
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 737

TONT1: MOV #16,14
MOV #22,20

```

```
1215
1216      ;*****
(2)      ;TEST 61      TEST THAT THE TRACE BIT WILL CAUSE A TRAP
(3)      ;*****
(2) 010632 005237 000404 TST61: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 010636 022737 000061 000404 CMP #61,@#$TESTN ;SEQUENCE ERROR?
(2) 010644 001026 BNE TST62-12 ;BR TO ERROR HALT ON SEQ ERROR
1217 010646 012706 000500 MOV #BUFF,%6 ;SET UP STACK POINTER
1218 010652 012767 010712 167134 MOV #TRC1,14 ;TRACE TRAP RETURN
1219 010660 005067 167132 CLR 16
1220 010664 012746 000020 MOV #20,-(SP) ;PUSH T BIT
1221 010670 012746 010676 MOV #.+6,-(SP) ;PUSH PC
1222 010674 000002 RTI ;SET T BIT
1223 010676 000240 NOP
1224 010700 012737 000224 000402 MOV #224,@#$FATAL ;MOVE TO MAILBOX # ***** 224 *****
(2) 010706 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 010710 000000 HALT ;DO NOT TRAP
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 755
1225 010712 036727 167560 000020 TRC1: BIT BUFF-2,#20 ;CHECK FOR T BIT ON STACK
1226 010720 001005 BNE TST62
(4) 010722 012737 000225 000402 MOV #225,@#$FATAL ;MOVE TO MAILBOX # ***** 225 *****
(3) 010730 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(3) 010732 000000 HALT ;T BIT NOT SAVED ON STACKED,OR WRONG $TESTN
(5) ; TO SCOPE REPLACE HALT W/ 240
(5) ; AND REPLACE NEXT INST W/ 744
1227      ;*****
(2)      ;TEST 62      TEST THAT AN RTI POPS THE T BIT
(3)      ;*****
(2) 010734 005237 000404 TST62: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 010740 022737 000062 000404 CMP #62,@#$TESTN ;SEQUENCE ERROR?
(2) 010746 001020 BNE TST63-12 ;BR TO ERROR HALT ON SEQ ERROR
1228 010750 012706 000500 MOV #BUFF,%6 ;SET UP THE STACK
1229 010754 012746 000020 MOV #20,-(6) ;FUTURE T BIT ON STACK
1230 010760 012746 010774 MCV #TRC2,-(6) ;RTI RETURN
1231 010764 012767 011010 167022 MOV #TRC3,14 ;TRACE TRAP INTERRUPT POINTER
1232 010772 000002 RTI
1233
1234 010774 000240 TRC2: NOP ;TRACE IS SET SHOULD TRAP TO 14
1235 010776 012737 000226 000402 MOV #226,@#$FATAL ;MOVE TO MAILBOX # ***** 226 *****
(2) 011004 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 011006 000000 HALT ;DID NOT TRACE TRAP,OR WRONG $TESTN
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 757
1236
1237 011010 012767 000016 166776 TRC3: MOV #16,14
1238 011016 005067 166774 CLR 16
```

```
1240
1241
(2)
(3)
(2) 011022 005237 000404
(2) 011026 022737 000063 000404
(2) 011034 001052
1242 011036 032767 000001 167354
1243 011044 001403
1244 011046 005767 167334
1245 011052 001052
1246 011054
1247 011054 105737 177564
1248 011060 100375
1249 011062 012706 000500
1250 011066 012767 000340 002022
1251 011074 106437
(1) 011076 013116
1252 011100 012767 011150 166756
1253 011106 012767 000100 166450
1254 011114 012767 011162 166712
1255 011122 012767 011174 166734
1256 011130 012767 000340 166700
1257 011136 005067 001754
1258 011142 106437
(1) 011144 013116
1259 011146 104400
1260 011150
(3) 011150 012737 000227 000402
(2) 011156 005212
(2) 011160 000000
(4)
(4)
1261 011162
(3) 011162 012737 000230 000402
(2) 011170 005212
(2) 011172 000000
(4)
(4)
1262 011174 005067 166636
```

```
*****
:TEST 63 TEST THAT A PENDING INTERRUPT OCCURS BEFORE TRAP
*****
TST63: INC @#$TESTN ;UPDATE TEST NUMBER
CMP #63,@#$TESTN ;SEQUENCE ERROR?
BNE TR1 ;BR TO ERROR HALT ON SEQ ERROR
BIT #1,$ENV ; CHECK IF ON APT
BEQ NOAPT ; IF NOT ON APT
TST $PASS ; CHECK IF ON FIRST PASS
BNE TST64 ; IF NOT FIRST PASS

NOAPT: TSTB @#TPS
BPL -4
MOV #BUFF,%6
MOV #340,STATUS ;HIGHEST PRIORITY LEVEL
.WORD 106437
.WORD STATUS
MOV #TR0,64
MOV #100,TTCSR ;INTERRUPT FOR TTY PUNCH/PRINTER
MOV #TR1,34 ;TRAP VECTOR
MOV #TR2,64 ;TTY VECTOR
MOV #340,36 ;IF TRAP TRAPS, MOVE 340 TO PRIORITY
CLR STATUS ;SHOULD TRAP AT END OF CLR INST
.WORD 106437
.WORD STATUS
TRAP ;TTY INTERRUPT SHOULD OVERRIDE TRAP

TR0: MOV #227,@#$FATAL ;MOVE TO MAILBOX # ***** 227 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TTY SHOULDN'T HAVE INTERRUPTED
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 725

TR1: MOV #230,@#$FATAL ;MOVE TO MAILBOX # ***** 230 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INTERRUPT DID NOT OCCUR FIRST,OR WRONG $TESTN
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 720

TR2: CLR 36
```

```
1264  
1265  
(2)  
(3)  
(2) 011200 005237 000404  
(2) 011204 022737 000064 000404  
(2) 011212 001042  
1266 011214 032767 000001 167176  
1267 011222 001403  
1268 011224 005767 167156  
1269 011230 001055  
1270 011232  
1271 011232 042767 000100 166324  
1272 011240 012706 000500  
1273 011244 012767 000340 001644  
1274 011252 106437  
(1) 011254 013116  
1275 011256 012767 000100 166300  
1276 011264 012767 011316 166542  
1277 011272 012767 011332 166564  
1278 011300 012767 011320 166512  
1279 011306 012767 000340 166506  
1280 011314 104400  
1281 011316 000004  
1282 011320  
(3) 011320 012737 000231 000402  
(2) 011326 005212  
(2) 011330 000000  
(4)  
(4)  
1283 011332 005067 166464  
1284 011336 005067 166524  
1285 011342 012767 000066 166514  
1286 011350 012767 000036 166456  
1287 011356 012767 000022 166434  
1288
```

```
*****  
:TEST 64 TEST THAT A PENDING INTERRUPT; INTERRUPTS BETWEEN TRAPS  
*****  
TST64: INC @#$TESTN ;UPDATE TEST NUMBER  
CMP #64,@#$TESTN ;SEQUENCE ERROR?  
BNE TR5 ;BR TO ERROR HALT ON SEQ ERROR  
BIT #1,$ENV ;CHECK IF ON APT  
BEQ NOAPT1 ; IF NOT  
TST $PASS ; CHECK IF ON FIRST PASS  
BNE TST65 ; IF NOT  
NOAPT1: BIC #100,TTCSR  
MOV #BUFF,%6  
MOV #340,STATUS  
.WORD 106437  
.WORD STATUS  
MOV #100,TTCSR  
MOV #TR3,34 ;TRAP  
MOV #TR4,64 ;TTY OUTPUT  
MOV #TR5,20 ;IOT  
MOV #340,22 ;IOT PRIORITY  
TRAP ;THE ACT OF TRAPPING LOWER PRIORITY  
IOT ;INTERRUPT SHOULD OCCUR IN PLACE OF IOT TRAP  
TR3: TR5: MOV #231,@#$FATAL ;MOVE TO MAILBOX # ***** 231 *****  
INC (R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;NO INTERRUPT BETWEEN TRAPS,OR WRONG $TESTN  
; TO SCOPE REPLACE HALT W/ 240  
; AND REPLACE NEXT INST W/ 730  
TR4: CLR 22 ;CLR IOT PRIORITY  
CLR 66  
MOV #66,64  
MOV #36,34  
MOV #22,20
```



```
1290
1291
(2)
(3)
(2) 011364 005237 000404
(2) 011370 022737 000065 000404
(2) 011376 001026
1292 011400 106437
(1) 011402 013122
1293 011404 012767 000100 166152
1294 011412 012767 000100 166140
1295 011420 000005
1296 011422 032767 000100 166134
1297 011430 001405
(3) 011432 012737 000232 000402
(2) 011440 005212
(2) 011442 000000
(4)
(4)
1298 011444 032767 000100 166106 1$:
1299 011452 001405
(4) 011454 012737 000233 000402
(3) 011462 005212
(3) 011464 000000
(5)
(5)
```

```
*****
:TEST 65 TEST THAT 'RESET' GOES TO OUTSIDE WORLD
*****
TST65: INC @#$TESTN ;UPDATE TEST NUMBER
CMP #65,@#$TESTN ;SEQUENCE ERROR?
BNE TST66-12 ;BR TO ERROR HALT ON SEQ ERROR
.WORD 106437
.WORD K340
MOV #100,TTCSR ;SET INTERRUPT ENABLE
MOV #100,TRCSR ;SET INTERRUPT ENABLE
RESET ;SHOULD CLEAR INTERRUPT ENABLE
BIT #100,TTCSR ;TEST FOR CLEAR
BEQ 1$
MOV #232,@#$FATAL ;MOVE TO MAILBOX # ***** 232 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESET FAILED TO CLEAR TTCSR
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 755
;TEST FOR CLEAR
BIT #100,TRCSR
BEQ TST66
MOV #233,@#$FATAL ;MOVE TO MAILBOX # ***** 233 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESET FAILED TO CLEAR TRCSR,OR WRONG $TESTN
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 744
```

```

1301
1302      ;*****
(2)      ;TEST 66      TEST THAT RESET HAS NO EFFECT ON THE TRACE TRAP
(3)      ;*****
(2) 011466 005237 000404      TST66:  INC    @#$TESTN      ;UPDATE TEST NUMBER
(2) 011472 022737 000066 000404      CMP    #66,@#$TESTN      ;SEQUENCE ERROR?
(2) 011500 001014      BNE    RSTP3      ;BR TO ERROR HALT ON SEQ ERROR
1303 011502 012706 000500      MOV    #BUFF,%6      ;SET STACK
1304 011506 012767 011544 166300      MOV    #RESET2,14      ;SET UP TRACE VECTOR
1305 011514 012746 000020      MOV    #20,-(SP)      ;PUSH T BIT
1306 011520 012746 011526      MOV    #.+6,-(SP)      ;PUSH PC
1307 011524 000006      RTT      ;SET T BIT
1308 011526 000005      RESET      ;SHOULD HAVE NO EFFECT
1309 011530 000005      RESET      ;NO EFFECT
1310 011532      RSTP3:
(3) 011532 012737 000234 000402      MOV    #234,@#$FATAL      ;MOVE TO MAILBOX # ***** 234 *****
(2) 011540 005212      INC    (R2)      ;SET MSGTYP TO FATAL ERROR
(2) 011542 000000      HALT      ;TRACE TRAP FAILED,OR WRONG $TESTN
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 756
1311 011544 005067 001346      RESET2: CLR    STATUS      ;CLEAR TRACK
1312 011550 106437      .WORD 106437
(1) 011552 013116      .WORD STATUS
1313 011554 012767 000016 166232      MOV    #16,14
1314 011562 005067 166230      CLR    16      ;TRACE STATUS
    
```

```
1316
1317      ;*****
      ;TEST 67      TEST THAT WHEN TTY INTERRUPTS IT POPS NEW STATUS
      ;*****
(2)      (2) 011566 005237 000404      TST67:  INC  @#$TESTN      ;UPDATE TEST NUMBER
(2)      (2) 011572 022737 000067 000404      CMP  #67,@#$TESTN      ;SEQUENCE ERROR?
(2)      (2) 011600 001070      BNE  RSTP4      ;BR TO ERROR HALT ON SEQ ERROR
1318      (2) 011602 032767 000001 166610      BIT  #1,$ENV      ; CHECK IF ON APT
1319      (2) 011610 001403      BEQ  NOAPT2      ; IF NOT ON APT
1320      (2) 011612 005767 166570      TST  $PASS      ;CHECK IF FIRST PASS
1321      (2) 011616 001073      BNE  TST70      ; IF NOT
1322      NOAPT2:
1323      (2) 011620 000005      RESET
1324      (2) 011622 012706 000500      MOV  #BUFF,%6      ;SET UP STACK
1325      (2) 011626 012767 011670 166230      MOV  #TTY3,64      ;INTERRUPT VECTOR
1326      (1) 011634 106437      .WORD 106437
1327      (1) 011636 013120      .WORD KO
1328      (2) 011640 012767 000357 166220      MOV  #357,66      ;HIGH PRIORITY ON INTERRUPT
1329      (2) 011646 052767 000100 165710      BIS  #100,TTCSR      ;SHOULD SET INTERRUPT ENABLE & INTERRUPT
1330      (2) 011654 000240      NOP
1331      (2) 011656 012737 000235 000402      MOV  #235,@#$FATAL      ;MOVE TO MAILBOX # ***** 235 *****
(2)      (2) 011664 005212      INC  (R2)      ;SET MSGTYP TO FATAL ERROR
(2)      (2) 011666 000000      HALT      ;NO INTERRUPT
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 744
1331      (1) 011670 106737      TTY3:  .WORD 106737
(1)      (1) 011672 013116      .WORD STATUS
1332      (2) 011674 022767 000357 001214      CMP  #357,STATUS
1333      (3) 011702 001405      BEQ  1$
(2)      (2) 011704 012737 000236 000402      MOV  #236,@#$FATAL      ;MOVE TO MAILBOX # ***** 236 *****
(2)      (2) 011712 005212      INC  (R2)      ;SET MSGTYP TO FATAL ERROR
(2)      (2) 011714 000000      HALT      ;INTERRUPT DID NOT POP CORRECT STATUS
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 731
1334      (2) 011716 000005      1$:  RESET      ;CLR NTERRUPT ENABLE
1335      (2) 011720 012706 000500      MOV  #BUFF,%6      ;STACK SET UP
1336      (2) 011724 012767 011750 166132      MOV  #TTY4,64      ;INTERRUPT VECTOR
1337      (2) 011732 005067 166130      CLR  66      ;CLR NEW STATUS
1338      (1) 011736 106437      .WORD 106437
(1)      (1) 011740 013120      .WORD KO
1339      (2) 011742 052767 000100 165614      BIS  #100,TTCSR      ;SET INTERRUPT ENABLE
1340      (1) 011750 106737      TTY4:  .WORD 106737
(1)      (1) 011752 013116      .WORD STATUS
1341      (2) 011754 005767 001136      TST  STATUS
1342      (1) 011760 001405      BEQ  RST4
(1)      (1) 011762      RSTP4:
(3)      (3) 011762 012737 000237 000402      MOV  #237,@#$FATAL      ;MOVE TO MAILBOX # ***** 237 *****
(2)      (2) 011770 005212      INC  (R2)      ;SET MSGTYP TO FATAL ERROR
(2)      (2) 011772 000000      HALT      ;INTERRUPT DID NOT POP CORRECT STATUS,OR WRONG $TESTN
(4)      ; TO SCOPE REPLACE HALT W/ 240
(4)      ; AND REPLACE NEXT INST W/ 702
1343      (2) 011774 005067 165564      RST4:  CLR  TTCSR
```

1344 012000 012767 000066 166056 MOV #66.64

```

1346
1347
1348 ;THIS ROUTINE TESTS THAT NO LEGAL ADDRESS TRAPS.
1349 ;AND THAT AN ILLEGAL ADDRESS TRAPS TO LOCATION 4
1350 ;*****
(2) ;TEST 70 TEST NON-EXISTENT ADDRESS TRAPS
(3) ;*****
(2) 012006 005237 000404 000404 TST70: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 012012 022737 000070 000404 CMP #70,@#$TESTN ;SEQUENCE ERROR?
(2) 012020 001066 BNE AUTO1 ;BR TO ERROR HALT ON SEQ ERROR
1351
1352 ;THIS ROUTINE TESTS MEMORY UNTIL IT DOES A NXM STOP
1353 012022 000402 BR ADALL
1354 012024 000000 TSL: 0
1355 012026 000000 CORH: 0
1356 012030 005000 ADALL: CLR %0
1357 012032 005067 165750 CLR 6
1358 012036 012767 012072 165740 MOV #ATRAP,4 ;SET UP ADDRESS TRAP ENTRANCE
1359 012044 012706 000500 NOR: MOV #BUFF,SP
1360 012050 105720 TSTB (0)+ ;IF OUTSIDE OF CORE, TRAP TO 4
1361 012052 020027 170000 CMP %0,#170000 ;IS POINTER IN SIDE CORE
1362 012056 101772 BLOS NOR ;TEST THE REST OF CORE
1363 012060 AUTO:
(3) 012060 012737 000240 000402 MOV #240,@#$FATAL ;MOVE TO MAILBOX # ***** 240 *****
(2) 012066 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 012070 000000 HALT ;SHOULD HAVE TRAPED
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 753
1364 ;RETURN HERE ON AN ADDRESS TRAP
1365 012072 005300 ATRAP: DEC R0
1366 012074 010067 177726 MOV R0,CORH ;MOVE THE FIRST NXM LOCATION IN CORH
1367 ;THIS ROUTINE DOES NXM TRAPS UNTIL IT FINDS AN EXISTANT MEMORY LOCATION
1368 012100 012700 170001 MOV #170001,R0 ;SET UP THE HIGHEST MEM LOCATION
1369 012104 012767 012142 165672 CTRAP: MOV #BTRAP,4 ;SET UP THE VECTOR
1370 012112 012706 000500 MOV #BUFF,SP
1371 012116 105740 TSTB -(R0) ;DOES IT EXIST?
1372 012120 005200 DTRAP: INC R0 ;IF YES INCREMENT IT
1373 012122 020067 177700 CMP R0,CORH ;IS IT THE SAME LOCATION?
1374 012126 001430 BEQ TRAPB
(3) 012130 012737 000241 000402 MOV #241,@#$FATAL ;MOVE TO MAILBOX # ***** 241 *****
(2) 012136 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 012140 000000 HALT ;CONTENTS OF R0 AND CORH SHOULD HAVE BEEN EQUAL
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 727
1375 ;IF THIS COMPARISON FAILS IT MEANS
1376 ;THAT SOME LEGAL ADDRESS TRAPPEDOR
1377 ;THAT AN ILLEGAL ADDRESS DID NOT TRAP
1378 012142 BTRAP:
(1) 012142 106737 .WORD 106737
(1) 012144 013116 .WORD STATUS
1379 012146 005767 000744 TST STATUS
1380 012152 001405 BEQ 1$
(3) 012154 012737 000242 000402 MOV #242,@#$FATAL ;MOVE TO MAILBOX # ***** 242 *****
(2) 012162 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR

```

(2)	012164	000000				HALT				:NEW PSW SHOULD HAVE BEEN ZERO
(4)										: TO SCOPE REPLACE HALT W/ 240
(4)										: AND REPLACE NEXT INST W/ 715
1381	012166	026727	166302	012120	1\$:	CMP	BUFF-4,#DTRAP			
1382	012174	001743				BEQ	CTRAP			
(1)	012176				AUTO1:					
(3)	012176	012737	000243	000402		MOV	#243,@#\$FATAL			:MOVE TO MAILBOX # ***** 243 *****
(2)	012204	005212				INC	(R2)			:SET MSGTYP TO FATAL ERROR
(2)	012206	000000				HALT				:OLD PC WAS NOT SAVED OR WRONG \$TESTN
(4)										: TO SCOPE REPLACE HALT W/ 240
(4)										: AND REPLACE NEXT INST W/ 704
1383	012210	012767	000006	165566	TRAPB:	MOV	#6,4			
1384	012216	005067	165564			CLR	6			

```

1386
1387
(2)
(3)
(2) 012222 005237 000404
(2) 012226 022737 000071 000404
(2) 012234 001070
1388 012236 032767 000001 166154
1389 012244 001403
1390 012246 005767 166134
1391 012252 001066
1392 012254 042767 000100 165302 NOAPT3: BIC #100,TPS
1393 012262 012706 000500 MOV #BUFF,SP
1394 012266 012767 012362 165570 MOV #WATE,64
1395 012274 005067 165566 CLR 66
1396 012300 105767 165260 WATE1: TSTB TPS
1397 012304 100375 BPL WATE1
1398 012306 012767 000015 165252 MOV #15,TPB
1399 012314 105767 165244 WATE2: TSTB TPS
1400 012320 100375 BPL WATE2
1401 012322 012767 000015 165236 MOV #15,TPB
1402 012330 052767 000100 165226 BIS #100,TPS
1403 012336 005067 000554 CLR STATUS
1404 012342 106437 .WORD 106437
(1) 012344 013116 .WORD STATUS
1405 012346 000001 WATE3: WAIT
1406 012350 012737 000244 000402 MOV #244,@#$FATAL
(2) 012356 005212 INC (R2)
(2) 012360 000000 HALT
(4)
(4)
1407 012362 WATE:
(1) 012362 106737 .WORD 106737
(1) 012364 013116 .WORD STATUS
1408 012366 005767 000524 TST STATUS
1409 012372 001405 BEQ 1$
(3) 012374 012737 000245 000402 MOV #245,@#$FATAL
(2) 012402 005212 INC (R2)
(2) 012404 000000 HALT
(4)
(4)
1410 012406 026727 166062 012350 1$: CMP BUFF-4,#WATE3+2
1411 012414 001405 BEQ REES
(1) 012416
(3) 012416 012737 000246 000402 REES1: MOV #246,@#$FATAL
(2) 012424 005212 INC (R2)
(2) 012426 000000 HALT
(4)
(4)
1412 012430 042767 000100 165126 REES: BIC #100,TPS
1413 012436 012767 000066 165420 MOV #66,64

```

```

:*****
:TEST 71 TEST THE 'WAIT' INSTRUCTION
:*****
TST71: INC @#$TESTN ;UPDATE TEST NUMBER
CMP #71,@#$TESTN ;SEQUENCE ERROR?
BNE REES1 ;BR TO ERROR HALT ON SEQ ERROR
BIT #1,$ENV ;CHECK IF ON APT
BEQ NOAPT3 ;BR, IF NOT ON APT
TST $PASS ;CHECK IF FIRST PASS
BNE REES ;BR, IF NOT
NOAPT3: BIC #100,TPS ;CLEAR INTERRUPT ENABLE
MOV #BUFF,SP ;SET UP THE STACK
MOV #WATE,64 ;SET UP THE INTERRUPT VECTOR
CLR 66
WATE1: TSTB TPS ;WAIT FOR READY
BPL WATE1 ;TO BE UP
MOV #15,TPB ;DO A CARRIAGE RETURN
WATE2: TSTB TPS ;WAIT FOR READY TO COME UP
BPL WATE2
MOV #15,TPB ;DO ANOTHER CARRIAGE RETURN
BIS #100,TPS ;SET THE INTERRUPT ENABLE
CLR STATUS ;CLEAR THE PSW
.WORD 106437
.WORD STATUS
WATE3: WAIT ;WAIT FOR THE INTERRUPT
MOV #244,@#$FATAL ;MOVE TO MAILBOX # ***** 244 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;WAIT INSTRUCTION DID NOT LOOP
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 725
WATE:
.WORD 106737
.WORD STATUS
TST STATUS ;IS THE PSW CORRECT?
BEQ 1$
MOV #245,@#$FATAL ;MOVE TO MAILBOX # ***** 245 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;NEW PSW SHOULD HAVE BEEN ZERO
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 713
1$: CMP BUFF-4,#WATE3+2 ;IS THE OLD PC SAVED
BEQ REES
REES1: MOV #246,@#$FATAL ;MOVE TO MAILBOX # ***** 246 *****
INC (R2) ;SET MSGTYP TO FATAL ERROR
HALT ;OLD PC WAS NOT SAVED OR WRONG $TESTN
; TO SCOPE REPLACE HALT W/ 240
; AND REPLACE NEXT INST W/ 702
REES: BIC #100,TPS ;CLEAR THE INTERRUPT ENABLE
MOV #66,64

```

```
1415
1416 ;*****
(2) ;TEST 72 TEST ,THAT ODD ADDRESSING WILL IGNORE BIT 0
(3) ;*****
(2) 012444 005237 000404 TST72: INC @#$TESTN ;UPDATE TEST NUMBER
(2) 012450 022737 000072 000404 CMP #72,@#$TESTN ;SEQUENCE ERROR?
(2) 012456 001002 BNE RSTP5 ;BR TO ERROR HALT ON SEQ ERROR
1417 012460 000167 000013 JMP ODD+1
1418 012464 RSTP5:
(3) 012464 012737 000247 000402 MOV #247,@#$FATAL ;MOVE TO MAILBOX # ***** 247 *****
(2) 012472 005212 INC (R2) ;SET MSGTYP TO FATAL ERROR
(2) 012474 000000 HALT ;SHOULD HAVE JUMPED,OR WRONG $TESTN
(4) ; TO SCOPE REPLACE HALT W/ 240
(4) ; AND REPLACE NEXT INST W/ 770
1419 012476 005307 ODD: DEC PC
```


1421
1422
(2)
(3)
(2) 012500 005237 000404
(2) 012504 022737 000073 000404
(2) 012512 001136
1423 012514 010267 000540
1424 012520 010700
1425 012522 010704
1426 012524 010705
1427 012526 012703 013042
1428 012532 012302
1429 012534 012301
1430 012536 020267 000310
1431 012542 001014
1432 012544 032767 000300 165654
1433 012552 001403
1434 012554 062703 000004
1435 012560 000764
1436 012562 032767 000004 165632 1\$:
1437 012570 001401
1438 012572 000757
1439 012574 020267 000262 2\$:
1440 012600 001007
1441 012602 032767 000020 165612
1442 012610 001403
1443 012612 062703 000010
1444 012616 000745
1445 012620 020267 000242 3\$:
1446 012624 001005
1447 012626 032767 000004 165566
1448 012634 001401
1449 012636 000735
1450 012640 020267 000232 4\$:
1451 012644 001005
1452 012646 032767 000010 165546
1453 012654 001401
1454 012656 000725
1455 012660 020267 000216 5\$:
1456 012664 001002
1457 012666 000167 000250
1458 012672 010267 000206
1459 012676 005267 000202
1460 012702 012767 012730 165100
1461 012710 012706 000500
1462 012714 005067 000176
1463 012720 106437
(1) 012722 013116
1464 012724 000167 000154
1465
1466
1467 012730 010267 000104
1468 012734 016702 000320

```
*****  
:TEST 73 TEST THAT ALL RESERVED INSTRUCTIONS TRAP  
*****  
TST73: INC @#$TESTN ;UPDATE TEST NUMBER  
CMP #73,@#$TESTN ;SEQUENCE ERROR?  
BNE RET4 ;BR TO ERROR HALT ON SEQ ERROR  
MOV R2,R2STOR ;SAVE REG 2  
MOV PC,%0 ;SET THESE  
MOV PC,%4 ;REGISTERS  
MOV PC,%5 ;TO EXISTENT MEMORY LOCATIONS  
GIN1: MOV #TABLE,TAB ;TABLE POINTER  
MOV (TAB)+,FIRST ;FIRST OR CURRENT INSTRUCTION  
MOV (TAB)+,LAST ;LAST INSTRUCTION OR GROUP  
CMP FIRST,EISFIS ;IS IT THE 'EISFIS' GROUP?  
BNE 2$ ;NO  
BIT #300,$CPUOP ;DO WE HAVE EISFIS OPTION?  
BEQ 1$ ;NO  
ADD #4,TAB ;IF YES DO NO DO THE  
BR GIN1 ;EIS FIS OP CODES  
1$: BIT #4,$SWREG ;DO WE HAVE DIS INSTRUCTION SET  
BEQ 2$ ;NO  
BR GIN1 ;IF YES, DO NOT DO EIS OP CODES - DO JUST FIS  
2$: CMP FIRST,STOP ;IS IT THE STOP GROUP  
BNE 3$ ;NO  
BIT #20,$SWREG ;DO WE WANT TO DO IT?  
BEQ 3$ ;YES  
ADD #10,TAB ;SKIP ENTIRE STOP GROUP  
BR GIN1 ;NO  
3$: CMP FIRST,DIS ;IS THIS THE DIS GROUP?  
BNE 4$ ;NO  
BIT #4,$SWREG ;DO WE HAVE DIS OPTION?  
BEQ 4$ ;NO  
BR GIN1 ;IF YES, SKIP THE DIS GROUP  
4$: CMP FIRST,STOP1 ;IS IT THE STOP1 GROUP?  
BNE 5$ ;NO  
BIT #10,$SWREG ;DO WE WANT TO DO IT?  
BEQ 5$ ;YES  
BR GIN1 ;NO  
5$: CMP FIRST,FINISH ;TESTED ALL  
BNE 6$ ;NO, BRANCH  
JMP GIN3 ;YES, GO TO END OF PASS ROUTINE  
6$: MOV FIRST,INST ;SET UP INST  
GIN2: INC INST  
MOV #RET,10 ;SET UP RETURN FROM TRAP  
MOV #BUFF,SP ;SET UP STACK POINTER  
CLR STATUS ;CLEAR PRIORITY  
.WORD 106437  
.WORD STATUS  
JMP INST ;EXECUTE RESERVED INSTRUCTION  
  
;TRAPPING SHOULD SEND YOU HERE  
RET: MOV R2,R2SAVE ;SAVE REG 2  
MOV R2STOR,R2 ;RESTORE MAILBOX POINTER
```

```

1469 012740 020627 000474      CMP      SP,#BUFF-4      ;TEST DECREMENT OF SP
1470 012744 001405      BEQ      RET1
(3) 012746 012737 000250 000402  MOV      #250,@#SFATAL ;MOVE TO MAILBOX # ***** 250 *****
(2) 012754 005212      INC      (R2)           ;SET MSGTYP TO FATAL ERROR
(2) 012756 000000      HALT                    ;WRONG DECREMENT
(4)                                     ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 655
1471 012760 026727 165510 013106 RET1:    CMP      BUFF-4,#INST+2 ;LOC OF INST UNINCREMENTED
1472 012766 001405      BEQ      RET2
(3) 012770 012737 000251 000402  MOV      #251,@#SFATAL ;MOVE TO MAILBOX # ***** 251 *****
(2) 012776 005212      INC      (R2)           ;SET MSGTYP TO FATAL ERROR
(2) 013000 000000      HALT                    ;INST INC ON TRAP
(4)                                     ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 644
1473 013002 005767 165470      RET2:    TST      BUFF-2
1474 013006 001405      BEQ      RET3
(1) 013010                                     RET4:
(3) 013010 012737 000252 000402  MOV      #252,@#SFATAL ;MOVE TO MAILBOX # ***** 252 *****
(2) 013016 005212      INC      (R2)           ;SET MSGTYP TO FATAL ERROR
(2) 013020 000000      HALT                    ;CONDITION CODES SET ON TRAP,OR WRONG $TESTN
(4)                                     ; TO SCOPE REPLACE HALT W/ 240
(4)                                     ; AND REPLACE NEXT INST W/ 634
1475 013022 016702 000012      RET3:    MOV      R2SAVE,R2 ;RESTORE REG 2
1476 013026 026701 000052      CMP      INST, LAST
1477 013032 001637      BEQ      GIN1           ;SET UP NEW GROUP
1478 013034 000167 177636      JMP      GIN2           ;FINISH OLD GROUP
1479 013040 000000      R2SAVE: .WORD 0
1480                                     ;END OF INSTRUCTION GROUP
1481 013042 006777      TABLE: 6777
1482 013044 007777          7777
1483 013046 106777          106777
1484 013050 107777          107777
1485 013052 067777      EISFIS: 67777 ;IF WE HAVE THE EIS FIS OPTION
1486 013054 073777          73777 ;THEN THE EISFIS GROUP
1487 013056 074777      FIS: 74777 ;WILL BE SKIPED
1488 013060 075037          75037
1489 013062 075377      STOP: 75377
1490 013064 076026          76026
1491 013066 076027      DIS: 76027
1492 013070 076057          76057
1493 013072 076057          76057
1494 013074 076777          76777
1495 013076 167777      STOP1: 167777
1496 013100 177777          177777
1497 013102 013102      FINISH: .
1498 013104 000000      INST: HALT ;END FLAG
1499 013106 000406          BR      TERR ;WILL CONTINUE RESERVED INST
1500 013110 000405          BR      TERR
1501 013112 000404          BR      TERR
1502 013114 000403          BR      TERR
1503 013116 000000      STATUS: 0
1504 013120 000000      KO: 0
1505 013122 000340      K340: 340
1506 013124 016702 000130      TERR: MOV      R2STOR,R2 ;RESTORE R2

```

```

1507 013130 012737 000255 000402      MOV    #255,@#FATAL ; INDICATE ERROR
1508 013136 005212                    INC    (R2)
1509 013140 000000                    HALT    ;INSTRUCTION TRAP IN ERROR
1510
1511
1512 013142 005237 000406      GIN3:  INC    @#SPASS
1513 013146 105267 000110      INCB   PASSPT ; SHOULD PRINT THIS PASS?
1514 013152 001023                    BNE    ACT ; NO
1515 013154 132767 000040 165237  BITB   #40,$ENVM ; WILL APT ALLOW PRINTING?
1516 013162 001017                    BNE    ACT ; NO
1517 013164 012700 013264      MOV    #MSG,R0 ; GET MSG ADDR.
1518 013170 105737 177564      WAIT:  TSTB  @#TPS ; TTY READY
1519 013174 100375                    BPL    WAIT ; NO WAIT
1520 013176 112037 177566      MOV    (R0)+,@#TPB ; PRINT CHARACTER
1521 013202 001372                    BNE    WAIT ; NEXT IF NOT DONE.
1522 013204 105737 177564      WAIT1: TSTB  @#TPS
1523 013210 100375                    BPL    WAIT1
1524 013212 000005                    RESET
1525 013214 012767 177761 000040  MOV    #177761,PASSPT ; DO IT 15 DECIMAL TIMES
1526 013222 013700 000042      ACT:   MOV    @#42,R0 ; CHECK ACT
1527 013226 001405                    BEQ    GOAGIN ; KEEP GOING
1528 013230 000005                    RESET
1529 013232 004710      $ENDAD: JSR    PC,(R0) ; ACT HOOKS
1530 013234 000240                    NOP
1531 013236 000240                    NOP
1532 013240 000240                    NOP
1533 013242 012767 000012 164540  GOAGIN: MOV    #12,10
1534 013250 005067 164536      CLR    12
1535 013254 000167 165320      JMP    RESTRT ; DO NEXT PASS
1536 013260 000000      R2STOR: .WORD 0
1537 013262 177777      PASSPT: -1
1538 013264 005015 047105 020104  MSG:   .ASCIZ <15><12>.END OF PASS.
        013272 043117 050040 051501
        013300 000123
    
```

```

1540
1541
1542
1543
1544
1545
1546 013302 012767 013312 164514 PWRDWN: MOV #PWRUP,24
1547 013310 000000 HALT
1548
1549 013312 012767 013302 164504 PWRUP: MOV #PWRDWN,24
1550 013320 012706 000500 MOV #BUFF,SP
1551 013324 132767 000040 165067 BITB #40,$ENVM :WILL APT ALLOW PRINTING?
1552 013332 001013 BNE PFRES :NO
1553 013334 012700 013366 MOV #MSGPWF,R0 :GET MSG ADDR.
1554 013340 105737 177564 PWAIT: TSTB @#TPS :TTY READY
1555 013344 100375 BPL PWAIT :NO WAIT
1556 013346 112037 177566 MOVB (R0)+,@#TPB :PRINT CHARACTER
1557 013352 001372 BNE PWAIT :NEXT IF NOT DONE.
1558 013354 105737 177564 PWAIT1: TSTB @#TPS
1559 013360 100375 BPL PWAIT1
1560 013362 000167 165114 PFRES: JMP START
1561 013366 005015 047520 042527 MSGPWF: .ASCIZ <15><12>.POWER FAILED!.
    013374 020122 040506 046111
    013402 042105 000041
1562 000001 .END
    
```

ABASE = 000000	425		
ACDW1 = 000000	425		
ACDW2 = 000000	425		
ACPUOP = 000000	425		
ACT 013222	1514	1516	1526#
ADALL 012030	1353	1356#	
ADDW0 = 000000	425		
ADDW1 = 000000	425		
ADDW10 = 000000	425		
ADDW11 = 000000	425		
ADDW12 = 000000	425		
ADDW13 = 000000	425		
ADDW14 = 000000	425		
ADDW15 = 000000	425		
ADDW2 = 000000	425		
ADDW3 = 000000	425		
ADDW4 = 000000	425		
ADDW5 = 000000	425		
ADDW6 = 000000	425		
ADDW7 = 000000	425		
ADDW8 = 000000	425		
ADDW9 = 000000	425		
ADEVCT = 000000	425		
ADEVN = 000000	425		
AENV = 000000	425		
AENVN = 000000	425		
AFATAL = 000000	425		
AMADR1 = 000000	425		
AMADR2 = 000000	425		
AMADR3 = 000000	425		
AMADR4 = 000000	425		
AMAMS1 = 000000	425		
AMAMS2 = 000000	425		
AMAMS3 = 000000	425		
AMAMS4 = 000000	425		
AMSGAD = 000000	425		
AMSGLG = 000000	425		
AMSGTY = 000000	425		
AMTYP1 = 000000	425		
AMTYP2 = 000000	425		
AMTYP3 = 000000	425		
AMTYP4 = 000000	425		
APASS = 000000	425		
APRIOR = 000000	425		
ASWREG = 000000	425		
ATESTN = 000000	425		
ATRAP 012072	1358	1365#	
AUNIT = 000000	425		
AUSWR = 000000	425		
AUTO 012060	1363#		
AUTO1 012176	1350	1382#	
AVECT1 = 000000	425		
AVECT2 = 000000	425		
BEGIN 000572	440	449	452#

ODD	012476	1417	1419#				
PASSPT	013262	452*	1513*	1525*	1537#		
PC	=%000007	317#	1419*	1424	1425	1426	1529*
PFRES	013362	1552	1560#				
PWAIT	013340	1554#	1555	1557			
PWAIT1	013354	1558#	1559				
PWRDWN	013302	437	1546#	1549			
PWRUP	013312	1546	1549#				
RA	005436	893	897#				
RA1	003662	744	748#				
RB	005422	892*	895#	897*	898		
RBBB	005424	891	896#				
RB1	003646	743*	746#	748*	749		
RB1AA	003650	742	747#				
RC	005416	894#	899				
RC1	003642	745#	750				
REES	012430	1391	1411	1412#			
REES1	012416	1387	1411#				
RESET2	011544	1304	1311#				
RESTR1	000600	453#	1535				
RET	012730	1460	1467#				
RETA	002200	607	610#				
RETAT	010040	1119	1125#				
RETA1	003034	677	680#				
RETA2	003752	755	758#				
RETA3	004606	825	828#				
RETA4	005526	904	907#				
RETA5	006362	983	986#				
RETB	002230	613	615#				
RETB1	010114	1128	1134#				
RETB2	003064	683	685#				
RETB3	004002	761	763#				
RETB4	004636	831	833#				
RETB5	005556	912	914#				
RETB6	006412	989	991#				
RETC	002300	621	623#				
RETC1	010174	1140	1145#				
RETC2	003134	689	691#				
RETC3	004052	769	771#				
RETC4	004706	837	839#				
RETC5	005626	918	920#				
RETD	006462	997	999#				
RETD1	002364	627	632#				
RETD2	003220	697	702#				
RETD3	004136	775	780#				
RETD4	004772	845	850#				
RETD5	005712	926	931#				
RETD6	006546	1003	1008#				
RETE	002436	635	640#				
RETE1	003270	705	709#				
RETE2	004210	783	788#				
RETE3	005044	853	858#				
RETE4	005764	934	939#				
RETE5	006620	1011	1016#				

RETF	002514	646	649#											
RETF1	003346	715	718#											
RETF2	004266	794	797#											
RETF3	005122	864	867#											
RETF4	006042	945	948#											
RETF5	006676	1022	1025#											
RETG	002644	658	661#											
RETG1	003476	727	730#											
RETG2	004416	806	809#											
RETG3	005252	876	879#											
RETG4	006172	957	960#											
RETG5	007026	1034	1037#											
RETH5	007200	1050	1053#											
RETJ	007230	1056	1058#											
RETK	007300	1064	1066#											
RETL	007364	1071	1076#											
RETM	007436	1079	1084#											
RETN	007514	1089	1092#											
RETO	007642	1101	1104#											
RET1	012760	1470	1471#											
RET2	013002	1472	1473#											
RET3	013022	1474	1475#											
RET4	013010	1422	1474#											
RSTP1	002746	644	670#											
RSTP2	006274	943	969#											
RSTP3	011532	1302	1310#											
RSTP4	011762	1317	1342#											
RSTP5	012464	1416	1418#											
RST1	002760	670	671#											
RST2	006306	969	970#											
RST4	011774	1342	1343#											
RTI1	010450	1185	1189#											
RTI2	010464	1186	1191#											
RTRAP =	000010	340#	607*	613*	621*	627*	635*	646*	647*	658*	659*			
RTRAP1=	000034	332#	677*	683*	689*	697*	705*	715*	716*	727*	728*			
RTRAP2=	000020	331#	755*	761*	769*	775*	783*	794*	795*	806*	807*			
RTRAP3=	000030	330#	825*	831*	837*	845*	853*	864*	865*	876*	877*			
RTRAP4=	000014	329#	904*	912*	918*	926*	934*	945*	946*	957*	958*	1119*	1128*	1140*
RTRAP5=	000004	328#	983*	989*	997*	1003*	1011*	1022*	1023*	1034*	1035*	1050*	1056*	1064*
		1071*	1079*	1089*	1090*	1101*	1102*							
RTT1	010260	1154	1157#											
RTT2	010276	1155	1160#											
RTT3	010344	1167	1171#											
RTT4	010364	1168	1175#											
RTT5	010316	1165#	1173	1178										
RTT6	010406	1176	1179#											
RO =%	000000	320#	447*	448	1365*	1366	1368*	1371	1372*	1373	1517*	1520	1526*	1529
		1553*	1556											
R1 =%	000001	1152*	1169*	1171*	1175*	1187*	1189*	1191						
R2 =%	000002	475*	480*	485*	491*	497*	503*	509*	514*	524*	532*	540*	548*	556*
		561*	564*	568*	571*	573*	576*	579*	582*	589*	590*	591*	592*	599*
		600*	601*	602*	609*	616*	624*	633*	641*	650*	651*	652*	653*	656*
		662*	663*	664*	665*	670*	679*	686*	692*	703*	710*	719*	720*	721*
		722*	725*	731*	732*	733*	734*	739*	747*	757*	764*	772*	781*	789*

TR2	011174	1255	1262#	
TR3	011316	1276	1281#	
TR4	011332	1277	1283#	
TR5	011320	1265	1278	1282#
TSL	012024	1354#		
TST1	000652	457	471#	
TST10	002322	619	624	625#
TST11	002460	625	641	644#
TST12	002772	675#		
TST13	003034	675	681#	
TST14	003104	681	686	687#
TST15	003156	687	692	695#
TST16	003312	695	710	713#
TST17	003612	713	739	742#
TST2	001150	471	514	517#
TST20	003710	753#		
TST21	003752	753	759#	
TST22	004022	759	764	767#
TST23	004074	767	772	773#
TST24	004232	773	789	792#
TST25	004544	823#		
TST26	004606	823	829#	
TST27	004656	829	834	835#
TST3	001474	517	556	559#
TST30	004730	835	840	843#
TST31	005066	843	859	862#
TST32	005366	862	888	891#
TST33	005464	902#		
TST34	005526	902	910#	
TST35	005576	910	915	916#
TST36	005650	916	921	924#
TST37	006006	924	940	943#
TST4	001730	559	568	582
TST40	006320	981#		585#
TST41	006362	981	987#	
TST42	006432	987	992	995#
TST43	006504	995	1000	1001#
TST44	006642	1001	1017	1020#
TST45	007136	1020	1045	1048#
TST46	007200	1048	1054#	
TST47	007250	1054	1059	1062#
TST5	002136	585	602	605#
TST50	007322	1062	1067	1069#
TST51	007460	1069	1085	1087#
TST52	007764	1117#		
TST53	010040	1117	1126#	
TST54	010134	1126	1135	1138#
TST55	010216	1138	1146	1149#
TST56	010276	1149	1158	1163#
TST57	010406	1163	1182#	
TST6	002200	605	611#	
TST60	010502	1182	1193	1196#
TST61	010632	1216#		
TST62	010734	1216	1226	1227#

	1299*	1310*	1330*	1333*	1342*	1363*	1374*	1380*	1382*	1406*	1409*	1411*	1418*
\$HIBTS 000450	426#												
\$MADR1 000432	425#												
\$MADR2 000436	425#												
\$MADR3 000442	425#												
\$MADR4 000446	425#												
\$MAIL 000400	425#	426											
\$MAMS1 000430	425#												
\$MAMS2 000434	425#												
\$MAMS3 000440	425#												
\$MAMS4 000444	425#												
\$MBADR 000452	426#												
\$MSGAD 000414	425#												
\$MSGLG 000416	425#												
\$MSGTY 000400	425#	453	454*										
\$MTYP1 000431	425#												
\$MTYP2 000435	425#												
\$MTYP3 000441	425#												
\$MTYP4 000445	425#												
\$PASS 000406	425#	433*	1244	1268	1320	1390	1512*						
\$PASTM 000456	426#												
\$SVPC = 000400	424#												
\$SWR = 000000	262#												
\$SWREG 000422	425#	443	447	1436	1441	1447	1452						
\$TESTN 000404	343	425#	471*	517*	559*	585*	605*	611*	619*	625*	644*	675*	681*
	687*	695*	713*	742*	753*	759*	767*	773*	792*	823*	829*	835*	843*
	862*	891*	902*	910*	916*	924*	943*	981*	987*	995*	1001*	1020*	1048*
	1054*	1062*	1069*	1087*	1117*	1126*	1138*	1149*	1163*	1182*	1196*	1216*	1227*
\$TN = 000074	1241*	1265*	1291*	1302*	1317*	1350*	1387*	1416*	1422*				
	263#	471#	514	517#	556	559#	568	582	585#	602	605#	611#	616
	619#	624	625#	641	644#	675#	681#	686	687#	692	695#	710	713#
	739	742#	753#	759#	764	767#	772	773#	789	792#	823#	829#	834
	835#	840	843#	859	862#	888	891#	902#	910#	915	916#	921	924#
	940	943#	981#	987#	992	995#	1000	1001#	1017	1020#	1045	1048#	1054#
	1059	1062#	1067	1069#	1085	1087#	1117#	1126#	1135	1138#	1146	1149#	1158
	1163#	1182#	1193	1196#	1216#	1226	1227#	1241#	1265#	1291#	1299	1302#	1317#
	1350#	1387#	1416#	1422#									
\$TSTM 000454	426#												
\$TSTM= 000404	343#	455*											
\$UNIT 000412	425#												
\$UNITM 000460	426#												
\$USWR 000424	425#												
\$X = 012514	471#	475	480	485	491	497	503	509	514	517#	524	532	540
	548	556	559#	561	564	568	571	573	576	579	582	585#	589
	590	591	592	599	600	601	602	605#	609	611#	616	619#	624
	625#	633	641	644#	650	651	652	653	656	662	663	664	665
	670	675#	679	681#	686	687#	692	695#	703	710	713#	719	720
	721	722	725	731	732	733	734	739	742#	747	753#	757	759#
	764	767#	772	773#	781	789	792#	798	799	800	801	804	810
	811	812	813	818	823#	827	829#	834	835#	840	843#	851	859
	862#	868	869	870	871	874	880	881	882	883	888	891#	896
	902#	906	910#	915	916#	921	924#	932	940	943#	949	950	951
	952	955	961	962	963	964	969	981#	985	987#	992	995#	1000

\$XX = 177635

1001#	1009	1017	1020#	1026	1027	1028	1029	1032	1038	1039	1040	1041
1045	1048#	1052	1054#	1059	1062#	1067	1069#	1077	1085	1087#	1093	1094
1095	1096	1099	1105	1106	1107	1108	1112	1117#	1124	1126#	1133	1135
1138#	1146	1149#	1158	1163#	1174	1178	1182#	1190	1193	1196#	1209	1211
1216#	1224	1226	1227#	1235	1241#	1260	1261	1265#	1282	1291#	1297	1299
1302#	1310	1317#	1330	1333	1342	1350#	1363	1374	1380	1382	1387#	1406
1409	1411	1416#	1418	1422#	1470	1472	1474					
475#	480#	485#	491#	497#	503#	509#	514#	524#	532#	540#	548#	556#
561#	564#	568#	571#	573#	576#	579#	582#	589#	590#	591#	592#	599#
600#	601#	602#	609#	616#	624#	633#	641#	650#	651#	652#	653#	656#
662#	663#	664#	665#	670#	679#	686#	692#	703#	710#	719#	720#	721#
722#	725#	731#	732#	733#	734#	739#	747#	757#	764#	772#	781#	789#
798#	799#	800#	801#	804#	810#	811#	812#	813#	818#	827#	834#	840#
851#	859#	868#	869#	870#	871#	874#	880#	881#	882#	883#	888#	896#
906#	915#	921#	932#	940#	949#	950#	951#	952#	955#	961#	962#	963#
964#	969#	985#	992#	1000#	1009#	1017#	1026#	1027#	1028#	1029#	1032#	1038#
1039#	1040#	1041#	1045#	1052#	1059#	1067#	1077#	1085#	1093#	1094#	1095#	1096#
1099#	1105#	1106#	1107#	1108#	1112#	1124#	1133#	1135#	1146#	1158#	1174#	1178#
1190#	1193#	1209#	1211#	1224#	1226#	1235#	1260#	1261#	1282#	1297#	1299#	1310#
1330#	1333#	1342#	1363#	1374#	1380#	1382#	1406#	1409#	1411#	1418#	1470#	1472#
1474#												

\$XXX = 000634

475#	480#	485#	491#	497#	503#	509#	514#	524#	532#	540#	548#	556#
561#	564#	568#	571#	573#	576#	579#	582#	589#	590#	591#	592#	599#
600#	601#	602#	609#	616#	624#	633#	641#	650#	651#	652#	653#	656#
662#	663#	664#	665#	670#	679#	686#	692#	703#	710#	719#	720#	721#
722#	725#	731#	732#	733#	734#	739#	747#	757#	764#	772#	781#	789#
798#	799#	800#	801#	804#	810#	811#	812#	813#	818#	827#	834#	840#
851#	859#	868#	869#	870#	871#	874#	880#	881#	882#	883#	888#	896#
906#	915#	921#	932#	940#	949#	950#	951#	952#	955#	961#	962#	963#
964#	969#	985#	992#	1000#	1009#	1017#	1026#	1027#	1028#	1029#	1032#	1038#
1039#	1040#	1041#	1045#	1052#	1059#	1067#	1077#	1085#	1093#	1094#	1095#	1096#
1099#	1105#	1106#	1107#	1108#	1112#	1124#	1133#	1135#	1146#	1158#	1174#	1178#
1190#	1193#	1209#	1211#	1224#	1226#	1235#	1260#	1261#	1282#	1297#	1299#	1310#
1330#	1333#	1342#	1363#	1374#	1380#	1382#	1406#	1409#	1411#	1418#	1470#	1472#
1474#												

= 013406

347#	351	352	354	356	358	360	362	364	366	368	370	372
374	376	378	380	382	384	386	388	390	392	394	396	398
400	402	404	406	408	410	412	414	423#	424#	426#	430#	432#
435#	471	475	480	485	491	497	503	509	514	517	524	532
540	548	556	559	561	564	568	571	573	576	579	582	585
589	590	591	592	599	600	601	602	605	609	611	616	619
623	624	625	633	641	644	650	651	652	653	656	662	663
664	665	670	675	679	681	686	687	691	692	695	703	710
713	719	720	721	722	725	731	732	733	734	739	742	747
753	757	759	764	767	771	772	773	781	789	792	798	799
800	801	804	810	811	812	813	818	823	827	829	834	835
839	840	843	851	859	862	868	869	870	871	874	880	881
882	883	888	891	896	902	906	910	915	916	920	921	924
932	940	943	949	950	951	952	955	961	962	963	964	969
981	985	987	992	995	999	1000	1001	1009	1017	1020	1026	1027
1028	1029	1032	1038	1039	1040	1041	1045	1048	1052	1054	1059	1062
1067	1069	1077	1085	1087	1093	1094	1095	1096	1099	1105	1106	1107
1108	1112	1117	1121	1124	1126	1130	1133	1135	1138	1142	1145	1146
1149	1158	1163	1174	1178	1182	1190	1193	1196	1206	1209	1211	1216

1221	1224	1226	1227	1235	1241	1248	1260	1261	1265	1282	1291	1297
1299	1302	1306	1310	1317	1330	1333	1342	1350	1363	1374	1380	1382
1387	1406	1409	1411	1416	1418	1422	1470	1472	1474	1497		

.BX = 000450

426#

ERROR	229#	475	480	485	491	497	503	509	514	524	532	540	548	556	561
	564	568	571	573	576	579	582	589	590	591	592	599	600	601	602
	609	616	624	633	641	650	651	652	653	656	662	663	664	665	670
	679	686	692	703	710	719	720	721	722	725	731	732	733	734	739
	747	757	764	772	781	789	798	799	800	801	804	810	811	812	813
	818	827	834	840	851	859	868	869	870	871	874	880	881	882	883
	888	896	906	915	921	932	940	949	950	951	952	955	961	962	963
	964	969	985	992	1000	1009	1017	1026	1027	1028	1029	1032	1038	1039	1040
	1041	1045	1052	1059	1067	1077	1085	1093	1094	1095	1096	1099	1105	1106	1107
	1108	1112	1124	1133	1135	1146	1158	1174	1178	1190	1193	1209	1211	1224	1226
	1235	1260	1261	1282	1297	1299	1310	1330	1333	1342	1363	1374	1380	1382	1406
	1409	1411	1418	1470	1472	1474									
LOOP	295#	475	480	485	491	497	503	509	514	524	532	540	548	556	561
	564	568	571	573	576	579	582	589	590	591	592	599	600	601	602
	609	616	624	633	641	650	651	652	653	656	662	663	664	665	670
	679	686	692	703	710	719	720	721	722	725	731	732	733	734	739
	747	757	764	772	781	789	798	799	800	801	804	810	811	812	813
	818	827	834	840	851	859	868	869	870	871	874	880	881	882	883
	888	896	906	915	921	932	940	949	950	951	952	955	961	962	963
	964	969	985	992	1000	1009	1017	1026	1027	1028	1029	1032	1038	1039	1040
	1041	1045	1052	1059	1067	1077	1085	1093	1094	1095	1096	1099	1105	1106	1107
	1108	1112	1124	1133	1135	1146	1158	1174	1178	1190	1193	1209	1211	1224	1226
	1235	1260	1261	1282	1297	1299	1310	1330	1333	1342	1363	1374	1380	1382	1406
	1409	1411	1418	1470	1472	1474									
MFPS	225#	595	654	666	723	735	802	814	872	884	953	965	1030	1042	1097
	1109	1331	1340	1378	1407										
MTPS	221#	588	597	629	637	699	707	777	785	847	855	928	936	1005	1013
	1073	1081	1251	1258	1274	1292	1312	1326	1338	1404	1463				
NEWTST	283#	471	517	559	585	605	611	619	625	644	675	681	687	695	713
	742	753	759	767	773	792	823	829	835	843	862	891	902	910	916
	924	943	981	987	995	1001	1020	1048	1054	1062	1069	1087	1117	1126	1138
	1149	1163	1182	1196	1216	1227	1241	1265	1291	1302	1317	1350	1387	1416	1422
STARS	291#	424	425	426	471	517	559	585	605	611	619	625	644	675	681
	687	695	713	742	753	759	767	773	792	823	829	835	843	862	891
	902	910	916	924	943	981	987	995	1001	1020	1048	1054	1062	1069	1087
	1117	1126	1138	1149	1163	1182	1196	1216	1227	1241	1265	1291	1302	1317	1350
	1387	1416	1422												
\$\$ERCD	256#	475	480	485	491	497	503	509	514	524	532	540	548	556	561
	564	568	571	573	576	579	582	589	590	591	592	599	600	601	602
	609	616	624	633	641	650	651	652	653	656	662	663	664	665	670
	679	686	692	703	710	719	720	721	722	725	731	732	733	734	739
	747	757	764	772	781	789	798	799	800	801	804	810	811	812	813
	818	827	834	840	851	859	868	869	870	871	874	880	881	882	883
	888	896	906	915	921	932	940	949	950	951	952	955	961	962	963
	964	969	985	992	1000	1009	1017	1026	1027	1028	1029	1032	1038	1039	1040
	1041	1045	1052	1059	1067	1077	1085	1093	1094	1095	1096	1099	1105	1106	1107
	1108	1112	1124	1133	1135	1146	1158	1174	1178	1190	1193	1209	1211	1224	1226
	1235	1260	1261	1282	1297	1299	1310	1330	1333	1342	1363	1374	1380	1382	1406
	1409	1411	1418	1470	1472	1474									
\$\$ERNU	250#	475	480	485	491	497	503	509	514	524	532	540	548	556	561
	564	568	571	573	576	579	582	589	590	591	592	599	600	601	602
	609	616	624	633	641	650	651	652	653	656	662	663	664	665	670
	679	686	692	703	710	719	720	721	722	725	731	732	733	734	739
	747	757	764	772	781	789	798	799	800	801	804	810	811	812	813

ADD	1434	1443														
BCC	589	653	722	801	871	952	1029	1096								
BCS	599	665	734	813	883	964	1041	1108								
BEQ	444	449	475	480	485	491	497	503	509	514	524	532	540	548	556	
	561	564	568	571	573	579	601	616	624	633	641	656	663	670	686	
	692	703	710	725	732	739	764	772	781	789	804	811	818	834	840	
	851	859	874	881	888	915	921	932	940	955	962	969	992	1000	1009	
	1017	1032	1039	1045	1059	1067	1077	1085	1099	1106	1112	1135	1146	1158	1173	
	1176	1178	1193	1211	1243	1267	1297	1299	1319	1333	1342	1374	1380	1382	1389	
	1409	1411	1433	1437	1442	1448	1453	1470	1472	1474	1477	1527				
BHIS	750	899														
BIC	668	737	816	886	967	1271	1392	1412								
BIS	450	596	1328	1339	1402											
BISB	445															
BIT	448	655	724	803	873	954	1031	1225	1242	1266	1296	1298	1318	1388	1432	
	1436	1441	1447	1452												
BITB	443	1515	1551													
BLOS	1362															
BMI	602	662	731	810	880	961	1038	1105								
BNE	440	471	517	559	576	582	585	591	605	611	619	625	644	651	675	
	681	687	695	713	720	742	753	759	767	773	792	799	823	829	835	
	843	862	869	891	902	910	916	924	943	950	981	987	995	1001	1020	
	1027	1048	1054	1062	1069	1087	1094	1117	1126	1138	1149	1163	1182	1196	1216	
	1226	1227	1241	1245	1265	1269	1291	1302	1317	1321	1350	1387	1391	1416	1422	
	1431	1440	1446	1451	1456	1514	1516	1521	1552	1557						
BPL	592	650	719	798	868	949	1026	1093	1248	1397	1400	1519	1523	1555	1559	
BR	1353	1435	1438	1444	1449	1454	1499	1500	1501	1502						
BVC	590	652	721	800	870	951	1028	1095								
BVS	600	664	733	812	882	963	1040	1107								
CCC	594	630	700	778	848	929	1006	1074								
CLR	433	441	442	454	455	456	472	482	487	488	493	494	499	500	505	
	506	587	628	647	672	698	716	752	776	795	820	846	865	901	927	
	946	971	1004	1023	1072	1090	1114	1152	1169	1187	1202	1203	1219	1238	1257	
	1262	1283	1284	1311	1314	1337	1343	1356	1357	1384	1395	1403	1462	1534		
CMP	471	474	479	484	490	496	502	508	513	517	523	531	539	547	555	
	559	585	605	611	615	619	623	625	632	640	644	669	675	681	685	
	687	691	695	702	709	713	738	742	749	753	759	763	767	771	773	
	780	788	792	817	823	829	833	835	839	843	850	858	862	887	891	
	898	902	910	914	916	920	924	931	939	943	968	981	987	991	995	
	999	1001	1008	1016	1020	1044	1048	1054	1058	1062	1066	1069	1076	1084	1087	
	1111	1117	1126	1134	1138	1145	1149	1163	1182	1196	1216	1227	1241	1265	1291	
	1302	1317	1332	1350	1361	1373	1381	1387	1410	1416	1422	1430	1439	1445	1450	
	1455	1469	1471	1476												
CMPB	489	495	501	507	512	560	563	566	570	572	575	578	581			
DEC	1175	1365	1419													
EMT	826	832	838	849	857	866	878	895								
HALT	323	351	353	355	357	359	361	363	365	367	369	371	373	375	377	
	379	381	383	385	387	389	391	393	395	397	399	401	403	405	407	
	409	411	413	415	475	480	485	491	497	503	509	514	524	532	540	
	548	556	561	564	568	571	573	576	579	582	589	590	591	592	599	
	600	601	602	609	616	624	633	641	650	651	652	653	656	662	663	
	664	665	670	679	686	692	703	710	719	720	721	722	725	731	732	
	733	734	739	747	757	764	772	781	789	798	799	800	801	804	810	
	811	812	813	818	827	834	840	851	859	868	869	870	871	874	880	

	881	882	883	888	896	906	915	921	932	940	949	950	951	952	955
	961	962	963	964	969	985	992	1000	1009	1017	1026	1027	1028	1029	1032
	1038	1039	1040	1041	1045	1052	1059	1067	1077	1085	1093	1094	1095	1096	1099
	1105	1106	1107	1108	1112	1124	1133	1135	1146	1158	1174	1178	1190	1193	1209
	1211	1224	1226	1235	1260	1261	1282	1297	1299	1310	1330	1333	1342	1363	1374
INC	1380	1382	1406	1409	1411	1418	1470	1472	1474	1498	1509	1547			
	471	475	480	485	491	497	503	509	514	517	524	532	540	548	556
	559	561	564	568	571	573	576	579	582	585	589	590	591	592	599
	600	601	602	605	609	611	616	619	624	625	633	641	644	650	651
	652	653	656	662	663	664	665	670	675	679	681	686	687	692	695
	703	710	713	719	720	721	722	725	731	732	733	734	739	742	747
	748	753	757	759	764	767	772	773	781	789	792	798	799	800	801
	804	810	811	812	813	818	823	827	829	834	835	840	843	851	859
	862	868	869	870	871	874	880	881	882	883	888	891	896	897	902
	906	910	915	916	921	924	932	940	943	949	950	951	952	955	961
	962	963	964	969	981	985	987	992	995	1000	1001	1009	1017	1020	1026
	1027	1028	1029	1032	1038	1039	1040	1041	1045	1048	1052	1054	1059	1062	1067
	1069	1077	1085	1087	1093	1094	1095	1096	1099	1105	1106	1107	1108	1112	1117
	1124	1126	1133	1135	1138	1146	1149	1158	1163	1171	1172	1174	1177	1178	1182
	1189	1190	1193	1196	1209	1210	1211	1216	1224	1226	1227	1235	1241	1260	1261
	1265	1282	1291	1297	1299	1302	1310	1317	1330	1333	1342	1350	1363	1372	1374
	1380	1382	1387	1406	1409	1411	1416	1418	1422	1459	1470	1472	1474	1508	1512
INCB	1513														
IOT	756	762	770	779	787	796	808	1208	1281						
JMP	431	434	457	984	990	998	1007	1015	1024	1036	1417	1457	1464	1478	1535
	1560														
JSR	1051	1057	1065	1075	1083	1091	1103	1529							
MOV	437	438	447	452	453	475	477	480	485	491	497	503	509	511	514
	518	519	520	521	524	526	527	528	529	532	534	535	536	537	540
	542	543	544	545	548	550	551	552	553	556	561	564	568	571	573
	576	579	582	589	590	591	592	599	600	601	602	606	607	609	612
	613	616	620	621	624	626	627	633	634	635	636	641	645	646	650
	651	652	653	656	657	658	659	662	663	664	665	667	670	671	676
	677	679	682	683	686	688	689	692	696	697	703	704	705	706	710
	714	715	719	720	721	722	725	726	727	728	731	732	733	734	736
	739	743	744	745	747	751	754	755	757	760	761	764	768	769	772
	774	775	781	782	783	784	789	793	794	798	799	800	801	804	805
	806	807	810	811	812	813	815	818	819	824	825	827	830	831	834
	836	837	840	844	845	851	852	853	854	859	863	864	868	869	870
	871	874	875	876	877	880	881	882	883	885	888	892	893	894	896
	900	903	904	906	911	912	915	917	918	921	925	926	932	933	934
	935	940	944	945	949	950	951	952	955	956	957	958	961	962	963
	964	966	969	970	982	983	985	988	989	992	996	997	1000	1002	1003
	1009	1010	1011	1012	1017	1021	1022	1026	1027	1028	1029	1032	1033	1034	1035
	1038	1039	1040	1041	1043	1045	1049	1050	1052	1055	1056	1059	1063	1064	1067
	1070	1071	1077	1078	1079	1080	1085	1088	1089	1093	1094	1095	1096	1098	1099
	1100	1101	1102	1105	1106	1107	1108	1110	1112	1113	1118	1119	1120	1121	1124
	1127	1128	1129	1130	1133	1135	1139	1140	1141	1142	1146	1151	1153	1154	1155
	1158	1164	1165	1166	1167	1168	1174	1178	1183	1184	1185	1186	1190	1193	1199
	1200	1201	1204	1205	1206	1209	1211	1212	1213	1217	1218	1220	1221	1224	1226
	1228	1229	1230	1231	1235	1237	1249	1250	1252	1253	1254	1255	1256	1260	1261
	1272	1273	1275	1276	1277	1278	1279	1282	1285	1286	1287	1293	1294	1297	1299
	1303	1304	1305	1306	1310	1313	1324	1325	1327	1330	1333	1335	1336	1342	1344
	1358	1359	1363	1366	1368	1369	1370	1374	1380	1382	1383	1393	1394	1398	1401

.IFT	475	480	485	491	497	503	509	514	524	532	540	548	556	561	564
	568	571	573	576	579	582	589	590	591	592	599	600	601	602	609
	616	624	633	641	650	651	652	653	656	662	663	664	665	670	679
	686	692	703	710	719	720	721	722	725	731	732	733	734	739	747
	757	764	772	781	789	798	799	800	801	804	810	811	812	813	818
	827	834	840	851	859	868	869	870	871	874	880	881	882	883	888
	896	906	915	921	932	940	949	950	951	952	955	961	962	963	964
	969	985	992	1000	1009	1017	1026	1027	1028	1029	1032	1038	1039	1040	1041
	1045	1052	1059	1067	1077	1085	1093	1094	1095	1096	1099	1105	1106	1107	1108
	1112	1124	1133	1135	1146	1158	1174	1178	1190	1193	1209	1211	1224	1226	1235
	1260	1261	1282	1297	1299	1310	1330	1333	1342	1363	1374	1380	1382	1406	1409
	1411	1418	1470	1472	1474										
.IIF	425	471	475	480	485	491	497	503	509	514	517	524	532	540	548
	556	559	561	564	568	571	573	576	579	582	585	589	590	591	592
	599	600	601	602	605	609	611	616	619	624	625	633	641	644	650
	651	652	653	656	662	663	664	665	670	675	679	681	686	687	692
	695	703	710	713	719	720	721	722	725	731	732	733	734	739	742
	747	753	757	759	764	767	772	773	781	789	792	798	799	800	801
	804	810	811	812	813	818	823	827	829	834	835	840	843	851	859
	862	868	869	870	871	874	880	881	882	883	888	891	896	902	906
	910	915	916	921	924	932	940	943	949	950	951	952	955	961	962
	963	964	969	981	985	987	992	995	1000	1001	1009	1017	1020	1026	1027
	1028	1029	1032	1038	1039	1040	1041	1045	1048	1052	1054	1059	1062	1067	1069
	1077	1085	1087	1093	1094	1095	1096	1099	1105	1106	1107	1108	1112	1117	1124
	1126	1133	1135	1138	1146	1149	1158	1163	1174	1178	1182	1190	1193	1196	1209
	1211	1216	1224	1226	1227	1235	1241	1260	1261	1265	1282	1291	1297	1299	1302
	1310	1317	1330	1333	1342	1350	1363	1374	1380	1382	1387	1406	1409	1411	1416
	1418	1422	1470	1472	1474										
.LIST	3	265	309	314	416	425	471	475	480	485	491	497	503	509	514
	517	524	532	540	548	556	559	561	564	568	571	573	576	579	582
	585	589	590	591	592	599	600	601	602	605	609	611	616	619	624
	625	633	641	644	650	651	652	653	656	662	663	664	665	670	675
	679	681	686	687	692	695	703	710	713	719	720	721	722	725	731
	732	733	734	739	742	747	753	757	759	764	767	772	773	781	789
	792	798	799	800	801	804	810	811	812	813	818	823	827	829	834
	835	840	843	851	859	862	868	869	870	871	874	880	881	882	883
	888	891	896	902	906	910	915	916	921	924	932	940	943	949	950
	951	952	955	961	962	963	964	969	981	985	987	992	995	1000	1001
	1009	1017	1020	1026	1027	1028	1029	1032	1038	1039	1040	1041	1045	1048	1052
	1054	1059	1062	1067	1069	1077	1085	1087	1093	1094	1095	1096	1099	1105	1106
	1107	1108	1112	1117	1124	1126	1133	1135	1138	1146	1149	1158	1163	1174	1178
	1182	1190	1193	1196	1209	1211	1216	1224	1226	1227	1235	1241	1260	1261	1265
	1282	1291	1297	1299	1302	1310	1317	1330	1333	1342	1350	1363	1374	1380	1382
	1387	1406	1409	1411	1416	1418	1422	1470	1472	1474					
.MACRO	221	225	229	245	250	256	267	283	291	295	305				
.MCALL	417	418	419												
.MEXIT	425														
.NLIST	1	220	266	313	346	425	471	475	480	485	491	497	503	509	514
	517	524	532	540	548	556	559	561	564	568	571	573	576	579	582
	585	589	590	591	592	599	600	601	602	605	609	611	616	619	624
	625	633	641	644	650	651	652	653	656	662	663	664	665	670	675
	679	681	686	687	692	695	703	710	713	719	720	721	722	725	731
	732	733	734	739	742	747	753	757	759	764	767	772	773	781	789
	792	798	799	800	801	804	810	811	812	813	818	823	827	829	834

	835	840	843	851	859	862	868	869	870	871	874	880	881	882	883
	888	891	896	902	906	910	915	916	921	924	932	940	943	949	950
	951	952	955	961	962	963	964	969	981	985	987	992	995	1000	1001
	1009	1017	1020	1026	1027	1028	1029	1032	1038	1039	1040	1041	1045	1048	1052
	1054	1059	1062	1067	1069	1077	1085	1087	1093	1094	1095	1096	1099	1105	1106
	1107	1108	1112	1117	1124	1126	1133	1135	1138	1146	1149	1158	1163	1174	1178
	1182	1190	1193	1196	1209	1211	1216	1224	1226	1227	1235	1241	1260	1261	1265
	1282	1291	1297	1299	1302	1310	1317	1330	1333	1342	1350	1363	1374	1380	1382
	1387	1406	1409	1411	1416	1418	1422	1470	1472	1474					
.REM	4														
.REPT	348														
.SBTTL	424	425	426	471	517	559	585	605	611	619	625	644	675	681	687
	695	713	742	753	759	767	773	792	823	829	835	843	862	891	902
	910	916	924	943	981	987	995	1001	1020	1048	1054	1062	1069	1087	1117
	1126	1138	1149	1163	1182	1196	1216	1227	1241	1265	1291	1302	1317	1350	1387
	1416	1422													
.TITLE	2														
.WORD	424	425	426	588	595	597	629	637	654	666	699	707	723	735	777
	785	802	814	847	855	872	884	928	936	953	965	1005	1013	1030	1042
	1073	1081	1097	1109	1251	1258	1274	1292	1312	1326	1331	1338	1340	1378	1404
	1407	1463	1479	1536											

ERRORS DETECTED: 0

CVKALA LSI-11 TRAP (30K+FIS) TEST
CVKALA.P11

MACY11 27(654) 19-SEP-78^{H 7} 11:25 PAGE 50-18

SEQ 0085

*CVKALA, CVKALA/CRF/DS:ERFLZ=CVKALA.P11
RUN-TIME: 21 13 2 SECONDS
CORE USED: 11K