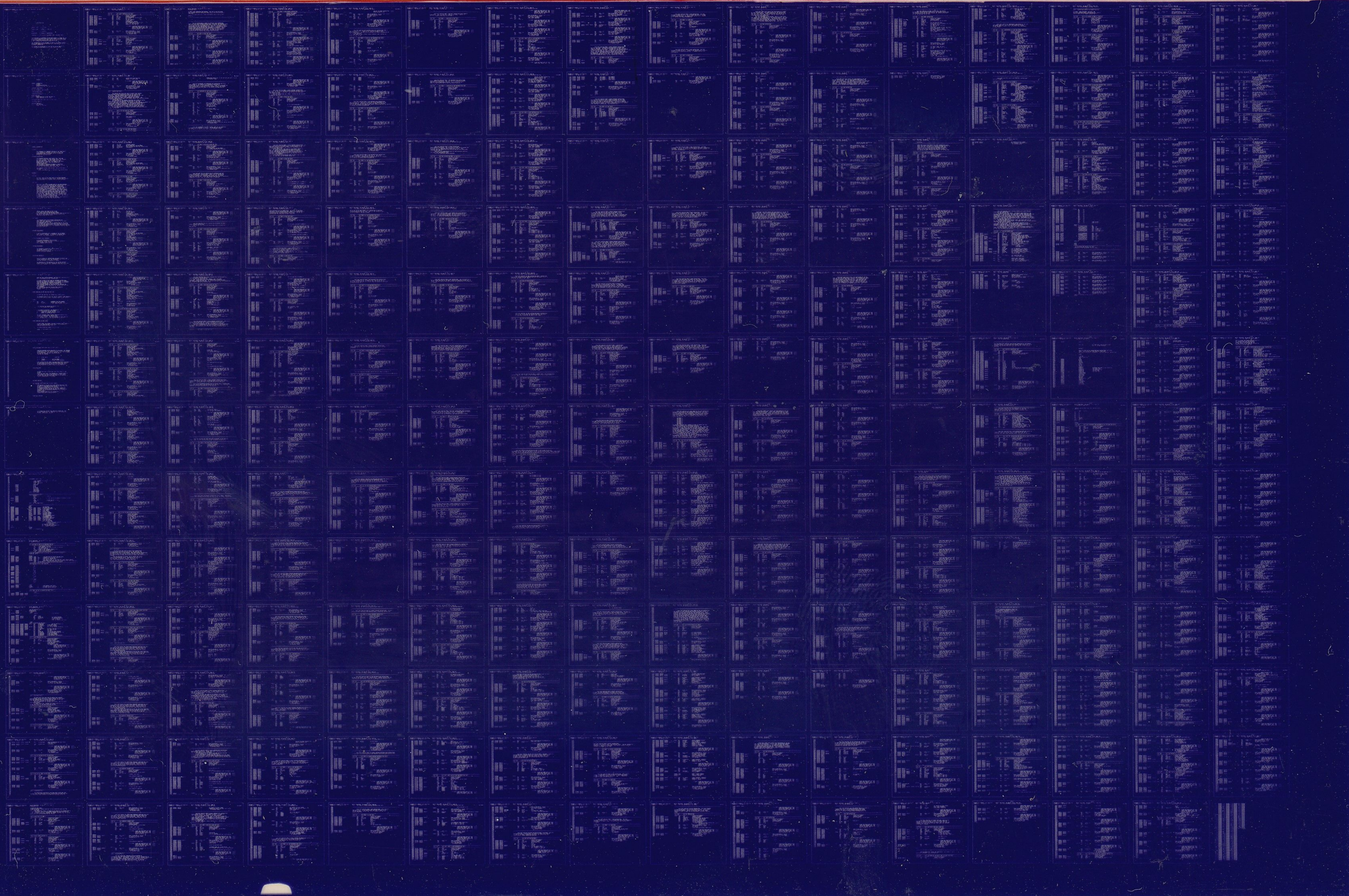


Microfiche grid containing technical diagrams and data for the PDP11/23 CPU.



000000

.REPT 0

1  
2  
3  
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

IDENTIFICATION  
-----

PRODUCT CODE: AC-F141A-MC

PRODUCT NAME: CJKDBAO F11 CPU DIAG

DATE: 24-JAN-79

MAINTAINER: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1979 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.

41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71

CONTENTS  
-----

1.0 GENERAL INFORMATION.  
1.1 HISTORY.  
1.2 PROGRAM DESCRIPTION.  
1.3 ABSTRACTS OF PART ONE, TWO AND THREE.

2.0 HARDWARE REQUIREMENT.

3.0 RELATED DOCUMENTS AND STANDARDS.

4.0 STARTING PROCEDURES.

5.0 TRAPCATCHER ABSTRACTS.

6.0 ERROR HANDLING.  
6.1 ERROR HANDLING IN PART ONE AND TWO.  
6.2 ERROR HANDLING IN PART THREE.

7.0 SWITCH SETTING (APPLICABLE ONLY TO PART THREE).

8.0 EXECUTION TIMES.

9.0 ROUTINES ABSTRACT.  
9.1 HALT ROUTINE.  
9.2 POWER FAIL ROUTINE.

72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
122  
123  
124  
125  
126  
127

1.0 GENERAL INFORMATION  
-----

1.1 HISTORY:

THIS PROGRAM IS A COMBINED VERSION OF THE THREE BASIC 11/34 DIAGNOSTIC PROGRAMS WITH MODIFICATIONS AND ENHANCEMENTS MADE TO ACCOUNT FOR THE DIFFERENCES BETWEEN THE TWO PROCESSORS.

1.2 PROGRAM DESCRIPTION:

THIS PROGRAM CONTAINS THREE PARTS: CPU, TRAP AND EIS TESTS. IN THE FIRST AND SECOND PARTS, THE PROGRAM WILL HALT ON ERROR. IN PART THREE, EIS TEST, WHEN AN ERROR IS DETECTED, THE ERROR PC AND ERROR NUMBER WILL BE TYPED, THEN THE PROGRAM WILL CONTINUE EXECUTION. LOOP ON ERROR IS PROVIDED BY MANUALLY MODIFYING SOME APPROPRIATE MEMORY LOCATIONS. SEE THE LISTING OF THAT TEST FOR DETAILS AND INSTRUCTIONS.

THIS PROGRAM ASSUMES SOME OPTIONS (FOR EIS TEST ONLY), THEY ARE:  
1. ENABLE ERROR PRINTOUTS, 2. CONTINUE EXECUTION ON ERROR.

1.3 ABSTRACT  
-----

PART ONE:

CPU TEST, THIS IS THE FIRST PART OF THE MAIN PROGRAM. THIS TEST CHECK OUT THE BASIC PDP-11 INSTRUCTIONS IN EVERY ADDRESSING MODES WITH VARIOUS TYPES OF DATA PATTERNS.

PART TWO:

TRAP TEST, THIS IS THE SECOND PART OF THE MAIN PROGRAM. THIS IS A TEST OF ALL OPERATIONS AND INSTRUCTIONS THAT CAUSE TRAPS. ALSO TESTED ARE TRAP OVERFLOW CONDITIONS, ODDITIES OF REGISTER 6, INTERRUPTS, THE RESET AND WAIT INSTRUCTIONS. THIS PROGRAM CHECKS THAT ON ALL TRAP OPERATIONS REGISTER 6 IS DECREMENTED THE CORRECT AMOUNT, THAT THE CORRECT PC IS SAVED ON THE STACK, THAT THE OLD CONDITION CODES AND PRIORITY ARE PLACED ON THE STACK AND THAT THE NEW STATUS AND CONDITION CODES ARE CORRECT. BOTH THE "TRAP" AND "EMT" TRAP INSTRUCTIONS ARE TESTED TO SEE THAT ALL COMBINATIONS WILL TRAP. CHECKED ALSO IS THAT ALL RESERVED INSTRUCTIONS WILL TRAP. THE TRACE BIT IS CHECKED TO SEE IF IT CAUSES A TRAP. THE RTI AND RTT INSTRUCTIONS ARE CHECKED. STACK OVERFLOW IS ALSO CHECKED FOR ALL THE TRAP INSTRUCTIONS.

128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167  
168  
169  
170  
171  
172  
173  
174  
175  
176  
177  
178  
179  
180  
181  
182  
183

SPECIAL CHECKS ARE MADE TO SEE IF BUS  
ERROR TRAPS OCCUR ON NON-EXISTENT MEMORY.  
ALL INSTRUCTIONS THAT ARE RESERVED SHOULD TRAP TO LOCATION 10,  
AND THE PC THAT POINTS TO THE TRAPPING INSTRUCTION  
SHOULD BE PLACED ON THE STACK.

PART THREE:

THIS PROGRAM TESTS THE EXTENDED INSTRUCTION SET  
<ASH, ASHC, MUL, AND DIV> USING REGISTERS 0-5 AT LEAST  
ONCE WITH EACH INSTRUCTION.  
THIS PROGRAM TESTS ALL THE EIS INSTRUCTIONS OF THE 11/34  
FOR ASH AND ASHC INSTRUCTIONS EVERY EVEN PASS IS EXECUTED  
WITH DESTINATION MODE 0 FOR ALL REGISTERS AND EVERY ODD PASS  
WITH DESTINATION MODE OF 67. THE DIAGNOSTIC DOES NOT MAKE A  
PASS WITH T BIT SET.

2.0 HARDWARE REQUIREMENT

A KDF11-A PROCESSOR WITH A MINIMUM OF 16K OF MEMORY  
AND A CONSOLE TERMINAL. IF PROGRAM IS RUNNING UNDER  
APT OR ACT, THE CONSOLE TERMINAL IS NOT NECESSARY.

3.0 RELATED DOCUMENTS AND STANDARDS:

ACT11/XXDP PROGRAMMING SPECIFICATION  
STANDARD APT SYSTEM TO A PDP11 DIAGNOSTIC INTERFACE  
PDP11 MAINDEC SYSMAC PACKAGE  
KDF11-A MODULE SPECIFICATION

4.0 STARTING PROCEDURES

THE PROGRAM IS STARTED BY LOADING ADDRESS 200.  
THE RESTART ADDRESS IS 1024.  
PROGRAM IDENTIFICATION WILL BE TYPED AFTER THE FIRST  
PASS OF THE WHOLE PROGRAM.

5.0 TRAPCATCHER ABSTRACTS

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 CON-  
TAIN A HALT (00000) (THIS LOCATION IS ALSO THE STATUS

184  
185  
186  
187  
188  
189  
190  
191  
192  
193  
194  
195  
196  
197  
198  
199  
200  
201  
202  
203  
204  
205  
206  
207  
208  
209  
210  
211  
212  
213  
214  
215  
216  
217  
218  
219  
220  
221  
222  
223  
224  
225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239

WORD FOR THAT VECTOR ENTRANCE. BUT THIS WILL HAVE NO EFFECT ON IT ALSO BEING THE NEXT INSTRUCTION).

IF A HALT OCCURS IN THE TRAP OR INTERRUPT VECTOR AREA, REGISTER SIX SHOULD BE EXAMINED TO DETERMINE ITS CONTENTS, THEN USE REGISTER SIX CONTENTS AS AN ADDRESS TO DETERMINE WHERE THE PROGRAM WAS. WHEN THE INTERRUPT OR TRAP OCCURRED; MEMORY AS SPECIFIED BY R6 CONTAINS THE PC OF THE INSTRUCTION FOLLOWING THE INSTRUCTION WHERE THE TRAP OCCURRED. THE CONTENTS OF LOCATION '\$TESTN'(304) CONTAINS THE TEST NUMBER THAT IT WAS DOING BEFORE IT TRAPPED.

### 6.1 ERROR HANDLING IN PART ONE AND PART TWO

-----

IN PARTS ONE AND TWO, ALL ERRORS WILL CAUSE A HALT.

THE PROGRAM CHECKS TO SEE THAT THE PC. DOESN'T JUMP ERRATICALLY WITHIN THE TESTS BY USING A SEQUENCE COUNT CALLED '\$TESTN'.

#### EXAMPLE

```
TSTA:  INC      (R2)           ;INCREMENT THE TEST NUMBER  
        CMP     #A,(R2)       ;COMPARE FOR THE RIGHT TEST  
        BNE    TSTA+1-10     ;IF NOT CORRECT BRANCH TO A HALT
```

\* R2 CONTAINS THE ADDRESS OF \$TESTN (304).  
A IS THE CURRENT TEST NUMBER.

IF AN ERROR IS DETECTED, THE PROGRAM WILL HALT IT COULD BE BECAUSE OF TWO REASONS.

- A) WRONG TEST NUMBER (SEQUENCE ERROR)
- B) ERROR IN THE PRESENT TEST.

THE TEST SEQUENCE COUNT "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 HANDLING IN PART THREE

-----

IN PART THREE, ANY ERROR, INCLUDES SEQUENCE CHECK ERROR WILL CAUSE THE ERROR MESSAGE TO BE TYPED. THE PROGRAM WILL CONTINUE EXECUTION AFTER TYPE OUT.

THE ERROR REPORTING FORMAT IS AS FOLLOWS:

```
ERROR! PC AND ERROR # ARE:  
PC #
```

240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280  
281  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295

ERROR #

7.0 SWITCH SETTINGS (APPLICABLE ONLY TO PART THREE).  
-----

SINCE NO HARDWARE SWITCH REGISTER IS AVAILABLE, THE PROGRAM AUTOMATICALLY USES THE CONTENTS OF LOC. 176 AS THE SOFTWARE SWITCH REGISTER. THE INITIAL CONTENT OF LOC. 176 IS 000000. THE USER MAY PRE-SET THIS LOCATION BEFORE STARTING THE PROGRAM.

| BIT # | OCTAL VALUE | FUNCTION               |
|-------|-------------|------------------------|
| 15    | 100000..... | HALT ON ERROR          |
| 13    | 020000..... | INHIBIT ERROR PRINTOUT |

ALSO, WITHIN THE APT TABLE, AN 8 BIT BYTE \$ENVN [LOCATION 321] HAS BEEN USED TO DEFINE THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING BIT 5 OF BYTE \$ENVN HIGH, IN OTHER WORDS BY PLACING A 20000 IN LOCATION 320.

8.0 EXECUTION TIMES  
-----

THE RUN TIME FOR A SINGLE RUN (THE FIRST PASS) IS ONE SECOND. AFTER THE FIRST PASS, THE PROGRAM WILL ITERATE EVERY 15 TIMES BEFORE THE END OF PASS MESSAGE IS TYPED AGAIN. THE RUN TIME FOR EACH ADDITIONAL END OF PASS MESSAGE TYPED IS APPROXIMATELY 15 SECONDS.

9.0 ROUTINES ABSTRACT  
-----

9.1 HALT ROUTINE (APPLICABLE ONLY TO PART THREE).  
-----

THIS ROUTINE IS CALLED VIA A JSR INSTRUCTION EACH TIME AN ERROR IS SEEN AND AN ERROR MESSAGE IS THEN TYPED OUT UNLESS IT IS SUPPRESSED BY THE SWITCHES. THE COMMENTS BESIDE THE CALL TO THE HALT SUBROUTINE TELLS WHAT WAS BEING TESTED AND WHAT WAS EXPECTED. ALL PRINTOUTS WILL BE SUPPRESSED WHEN BIT 5 OF LOCATION \$ENVN IS HIGH. WHILE RUNNING UNDER APT THE DIAGNOSTIC WILL NOT SUPPORT SPOOLING OF CONSOLE OUTPUTS.

9.2 POWER FAIL ROUTINE  
-----



296  
297  
298  
299  
300  
301  
302

IF A POWER FAIL OCCURS (FOLLOWED BY A POWER UP), THE  
MESSAGE 'POWER FAIL' IS TYPED OUT AND THE PROGRAM WILL  
RESTART EXECUTION AT 'RESTRT'.

.ENDR

303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329  
330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358

;PROGRAMMER: KIM C. LEE

.TITLE MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
.ENABLE ABS  
.NLIST CND,MC,MD  
.LIST ME  
SCOPE=NOP  
R7=27  
R6=26  
PS=177776  
TPS=177564  
TPB=177566  
USRM=140000  
PUSRM=30000  
.SBTTL ACT11 HOOKS

\*\*\*\*\*  
;HOOKS REQUIRED BY ACT11

\$SVPC=. ;SAVE PC  
.=46  
\$ENDAD ;:1)SET LOC.46 TO ADDRESS OF \$ENDAD IN .SEOP  
.=52  
.WORD 0 ;:2)SET LOC.52 TO ZERO  
.=300 ;: RESTORE PC

.SBTTL APT MAILBOX-ETABLE

\*\*\*\*\*  
;EVEN

\$MAIL: ;:APT MAILBOX  
\$MSGTY: .WORD AMSGTY ;:MESSAGE TYPE CODE  
\$FATAL: .WORD AFATAL ;:FATAL ERROR NUMBER  
\$TESTN: .WORD ATESTN ;:TEST NUMBER  
\$PASS: .WORD APASS ;:PASS COUNT  
\$DEVCT: .WORD ADEVCT ;:DEVICE COUNT  
\$UNIT: .WORD AUNIT ;:I/O UNIT NUMBER  
\$MSGAD: .WORD AMSGAD ;:MESSAGE ADDRESS  
\$MSGLG: .WORD AMSGLG ;:MESSAGE LENGTH  
\$ETABLE: ;:APT ENVIRONMENT TABLE  
\$ENV: .BYTE AENV ;:ENVIRONMENT BYTE  
\$ENVM: .BYTE AENVM ;:ENVIRONMENT MODE BITS  
\$SWREG: .WORD ASWREG ;:APT SWITCH REGISTER  
\$USWR: .WORD AUSWR ;:USER SWITCHES  
\$CPUOP: .WORD ACPUOP ;:CPU TYPE,OPTIONS  
BITS 15-11=CPU TYPE  
11/04=01,11/05=02,11/20=03,11/40=04,11/45=05  
11/70=06,PDQ=07,Q=10  
BIT 10=REAL TIME CLOCK  
BIT 9=FLOATING POINT PROCESSOR  
BIT 8=MEMORY MANAGEMENT

\$ETEND:

.MEXIT  
.SBTTL APT PARAMETER BLOCK

359  
360  
361  
362 000330  
363 000024  
364 000024 000200  
365 000044  
366 000044 000330  
367 000330  
368  
369  
370  
371  
372 000330  
373 000330 000000  
374 000332 000300  
375 000334 000010  
376 000336 000020  
377 000340 000005  
378 000342 000014  
379  
380  
381  
382 000004  
383 000004 027352  
384 000006 000000  
385 000010 027362  
386 000012 000000  
387 000014 027372  
388 000030  
389 000030 027402  
390 000032 000000  
391 000034 027412  
392 000036 000000  
393 000114  
394 000114 027422  
395 000116 000000  
396 000244  
397 000244 027432  
398 000246 000000  
399 000250 027442  
400 000252 000000  
401  
402 000172  
403 000172 000000  
404 000174 000000  
405 000176 000000  
406  
407  
408  
409  
410 000370  
411 000370 000000 000000 000000  
412 000376 000000 000000 000000  
413 000404 000001 000001 177777  
414

```
*****  
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT  
*****  
      .SX=      ;;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  
      .=.SX     ;;RESET LOCATION COUNTER  
*****  
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC  
:INTERFACE SPEC.  
$APTHD:  
$HIBTS: .WORD 0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.  
$MBADR: .WORD $MAIL  ;;ADDRESS OF APT MAILBOX (BITS 0-15)  
$TSTM:  .WORD 10     ;;RUN TIM OF LONGEST TEST  
$PASTM: .WORD 20     ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)  
$UNITM: .WORD 5      ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT  
      .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)  
*****  
:SOME POINTERS TO CPU TRAP HANDLERS  
*****  
      .=4  
      T04  
      0  
      T010  
      0  
      T014  
      .=30  
      T030  
      0  
      T034  
      0  
      .=114  
      T0114  
      0  
      .=244  
      T0244  
      0  
      T0250  
      0  
      .=172  
LPADR: 0      ;LOOP ADDRESS (EIS TEST)  
DISPREG: 0    ;SOFTWARE DISPLAY REGISTER  
SWREG: 0     ;SOFTWARE SWITCH REGISTER  
*****  
:DATA TABLE FOR USE IN ADDRESSING MODE TESTS  
*****  
      .=370  
      0,0,0,0,0,0  
      1,1,-1  
*****
```

```

415 ;SET UP STARTING ADDRESS
416 $ERROR=$FATAL
417 $STSTNM=$TESTN
418 .=1000
419 STBOT: .WORD 0 ;STACK POINTER
420
421 .=200
422 JMP START
423 MOV #STBOT,R6 ;SET STACK POINTER
424 MOV #STSTNM,R2 ;SET MAILBOX POINTER
425 JMP @PC+ ;JUMP TO SUBTEST
426 0 ;ADDR. OF SUBTEST GOES HERE
427
428
429 .=1002
430 .SBTTL **STARTING OF CPU TEST **
431 START: MOV #PWRDN,@#24 ;SET UP FOR POWER FAIL
432 MOV #0,@#SPASS ;CLEAR PASS COUNT
433 MOV #16,@#PASSPT ;SET PRINT COUNTER
434 RESTRT: MOV #18,@#4 ;SET UP FOR TIMEOUT IF NO MULTI TESTER
435 MOV #340,@#6
436 MOV #2,@#164000 ;SET BIT1 FOR MULTI TESTER
437 1$: MOV #6,@#4 ;SET TRAP CATCHER
438 CLR @#6 ;SET HALT
439 MOV #STBOT,R6 ;INITIALIZE STACK POINTER
440 MOV #STSTNM,R2 ;SET UP POINTER TO MESSAGE TYPE
441 MOV #0,@#STSTNM ;CLEAR TEST NUMBER
442 MOV #0,@#ERROR ;CLEAR ERROR NUMBER
443 MOV #0,@#MSGTY ;CLEAR MESSAGE TYPE (FOR APT)
444
445 ;*****
446 ;TEST 1 CHECK BRANCHES ON Z BIT
447 ;*****
448 TS1: INC (R2) ;UPDATE TEST NUMBER
449 CMP #1,(R2) ;SEQUENCE ERROR?
450 BNE TS2-10 ;BR TO ERROR HALT ON SEQ ERROR
451 CCC ;CLEAR ALL CONDITION CODES
452 BEQ BRA1 ;SHOULD BRANCH
453 BR BRA2 ;BAD BRANCH OF Z-BIT
454 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
455 ; BRANCH INSTRUCTION AND <====
456 ; REPLACE THE MOVE INSTRUCTION <====
457 ; FOLLOWING W/ 774 <====
458 BRA1: MOV #1,-(R2) ;MOVE TO MAILBOX # ***** 1 *****
459 INC -(R2) ;SET MSGTYP TO FATAL ERROR
460 HALT ;SHOULD HAVE BRANCHED: Z=0
461 BRA2: BNE BRA3
462 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
463 ; CONDITIONAL BRANCH INST. AND <====
464 ; REPLACE THE MOVE INSTRUCTION <====
465 ; WHICH FOLLOWS W/ 767 <====
466 MOV #2,-(R2) ;MOVE TO MAILBOX # ***** 2 *****
467 INC -(R ) ;SET MSGTYP TO FATAL ERROR
468 HALT
469 BRA3: SEZ
470 BNE BRA4
    
```

```
471 001156 000404          BR      BRAS
472
473
474
475
476 001160
477 001160 012742 000003    BRA4:  MOV      #3,-(R2)      ;MOVE TO MAILBOX # ***** 3 *****
478 001164 005242          INC      -(R2)              ;SET MSGTYP TO FATAL ERROR
479 001166 000000          HALT                       ;SHOULD NOT HAVE BRANCHED HERE ON Z=1
480 001170
481 001170 001404          BRAS:  BEQ      TS2
482
483
484
485
486 001172 012742 000004    MOV      #4,-(R2)      ;MOVE TO MAILBOX # ***** 4 *****
487 001176 005242          INC      -(R2)              ;SET MSGTYP TO FATAL ERROR
488 001200 000000          HALT                       ;SHOULD HAVE BRANCHED ON Z=1
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505 001202 005212          TS2:  INC      (R2)          ;UPDATE TEST NUMBER
506 001204 022712 000002    CMP      #2,(R2)          ;SEQUENCE ERROR?
507 001210 001006          BNE     TS3-10 ;BR TO ERROR HALT ON SEQ ERROR
508 001212 012737 000000 000000  MOV      #0,#0           ;MOVE ZEROES THRU ADDRESS LINES, DATA
509
510
511
512
513
514
515
516 001226 012742 000005    TST     #0
517 001232 005242          BEQ     TS3
518 001234 000000
519
520
521
522
523
524 001236 005212          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
525 001240 022712 000003    ;          CONDITIONAL BRANCH INST. AND <====
526 001244 001007          ;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 771 <====
;MOVE TO MAILBOX # ***** 5 *****
;SET MSGTYP TO FATAL ERROR
;DATA INCORRECT
; OR SEQUENCE ERROR
```

\*\*\*\*\*  
:SBTTL DATA PATH TESTS

: THE DATA PATH TESTS ARE USED TO VERIFY THAT VARIOUS  
: DATA PATTERNS CAN BE SUCCESSFULLY MOVED THROUGH THE DATA PATHS  
: MOVE AND COMPARE MODE 2,3 INSTRUCTIONS ARE USED TO PASS AND  
: TEST VARIOUS DATA PATTERNS IN THE DATA PATHS.  
: THE TEST EXERCISES THE INTERNAL DATA PATHS, AND THE UNIBUS  
: DATA TRANSCIEVERS.  
: IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0)  
: TO SEE WHICH BITS OF THE DATA PATH ARE FAILING.

\*\*\*\*\*  
:TEST 2 TEST OF ZEROES IN THE DATA PATH

\*\*\*\*\*  
:TEST 3 TEST OF PATTERN 125252 IN DATA PATH  
\*\*\*\*\*

```
524 001236 005212          TS3:  INC      (R2)          ;UPDATE TEST NUMBER
525 001240 022712 000003    CMP      #3,(R2)          ;SEQUENCE ERROR?
526 001244 001007          BNE     TS4-10 ;BR TO ERROR HALT ON SEQ ERROR
```

```
527 001246 012737 125252 000000      MOV      #125252,@#0      ;MOVE ALTERNATING ONES AND ZEROES
528                                     ;THRU DATA PATHS
529 001254 022737 125252 000000      CMP      #125252,@#0      ;SUCCESSFUL
530 001262 001404                                     BEQ      TS4
531                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
532                                     ;          CONDITIONAL BRANCH INST. AND <====
533                                     ;          REPLACE THE MOVE INSTRUCTION <====
534                                     ;          WHICH FOLLOWS W/ 770 <====
535 001264 012742 000006      MOV      #6,-(R2)        ;MOVE TO MAILBOX # ***** 6 *****
536 001270 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
537 001272 000000      HALT                    ;DATA INCORRECT
538                                     ; OR SEQUENCE ERROR
```

\*\*\*\*\*  
;TEST 4 TEST OF PATTERN 052525 IN DATA PATH  
\*\*\*\*\*

```
543 001274 005212      TS4:  INC      (R2)          ;UPDATE TEST NUMBER
544 001276 022712 000004      CMP      #4,(R2)        ;SEQUENCE ERROR?
545 001302 001007      BNE      TS5-10 ;BR TO ERROR HALT ON SEQ ERROR
546 001304 012737 052525 000000      MOV      #052525,@#0    ;MOVE ALTERNATING ZEROES AND ONES
547                                     ;THRU DATA PATH
548 001312 022737 052525 000000      CMP      #052525,@#0    ;SUCCESSFUL?
549 001320 001404      BEQ      TS5
550                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
551                                     ;          CONDITIONAL BRANCH INST. AND <====
552                                     ;          REPLACE THE MOVE INSTRUCTION <====
553                                     ;          WHICH FOLLOWS W/ 770 <====
554 001322 012742 000007      MOV      #7,-(R2)        ;MOVE TO MAILBOX # ***** 7 *****
555 001326 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
556 001330 000000      HALT                    ;DATA INCORRECT
557                                     ; OR SEQUENCE ERROR
```

\*\*\*\*\*  
;TEST 5 TEST OF ALL ONES IN DATA PATH  
\*\*\*\*\*

```
562 001332 005212      TS5:  INC      (R2)          ;UPDATE TEST NUMBER
563 001334 022712 000005      CMP      #5,(R2)        ;SEQUENCE ERROR?
564 001340 001007      BNE      TS6-10 ;BR TO ERROR HALT ON SEQ ERROR
565 001342 012737 177777 000000      MOV      #177777,@#0    ;MOVE ONES THRU DATA PATH
566 001350 022737 177777 000000      CMP      #177777,@#0    ;SUCCESSFUL
567 001356 001404      BEQ      TS6
568                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
569                                     ;          CONDITIONAL BRANCH INST. AND <====
570                                     ;          REPLACE THE MOVE INSTRUCTION <====
571                                     ;          WHICH FOLLOWS W/ 770 <====
572 001360 012742 000010      MOV      #10,-(R2)      ;MOVE TO MAILBOX # ***** 10 *****
573 001364 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
574 001366 000000      HALT                    ;DATA INCORRECT
575                                     ; OR SEQUENCE ERROR
```

\*\*\*\*\*  
;SBTTL B-REGISTER TEST  
\*\*\*\*\*

```
580                                     ; THE B-REGISTER (LOCATION 0) SHIFTING LOGIC TESTS ARE USED
581 ;TO TEST THAT THE B-REGISTER CAN HOLD VARIOUS DATA PATTERNS AND THAT
582 ;THE ASSOCIATED LOGIC SUPPORTS THE SHIFTING FUNCTIONS WITHIN THE
```

583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594 001370 005212  
595 001372 022712 000006  
596 001376 001012  
597 001400 000241  
598 001402 012737 000001 000000  
599 001410 006137 000000  
600 001414 022737 000002 000000  
601 001422 001404  
602  
603  
604  
605  
606 001424 012742 000011  
607 001430 005242  
608 001432 000000  
609  
610  
611  
612  
613  
614 001434 005212  
615 001436 022712 000007  
616 001442 001017  
617 001444 012737 000000 000000  
618 001452 000261  
619 001454 006137 000000  
620 001460 103014  
621  
622  
623  
624  
625 001462 012742 000012  
626 001466 005242  
627 001470 000000  
628  
629 001472 022737 000001 000000  
630 001500 001404  
631  
632  
633  
634  
635 001502 012742 000013  
636 001506 005242  
637 001510 000000  
638

:B-REGISTER AND C-BIT.  
:A ONE IS SHIFTED THROUGH EVERY BIT IN THE B-REGISTER AND C-BIT IN  
:BOTH DIRECTIONS.  
: THE B-REGISTER ITSELF IS TESTED IN ITS ABILITY AS A BUFFER AND AS  
:A SHIFT REGISTER. DATA IS ALSO PASSED THROUGH THE DATA PATH AND ALU,  
: IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0) TO SEE  
:WHICH BITS OF THE B-REGISTER MAY BE FAILING.

\*\*\*\*\*  
:TEST 6 SHIFT BIT 0 TO BIT 1  
\*\*\*\*\*

TS6: INC (R2) ;UPDATE TEST NUMBER  
CMP #6,(R2) ;SEQUENCE ERROR?  
BNE TS7-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLC ;CLEAR CARRY BIT  
MOV #1,@#0 ;LOAD A 1  
ROL @#0 ;SHIFT LEFT  
CMP #2,@#0 ;SUCCESSFUL  
BEQ TS7  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 765 <====  
MOV #11,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 11 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;BIT 1 NOT SET  
; OR SEQUENCE ERROR

\*\*\*\*\*  
:TEST 7 SHIFT CARRY INTO BIT 0  
\*\*\*\*\*

TS7: INC (R2) ;UPDATE TEST NUMBER  
CMP #7,(R2) ;SEQUENCE ERROR?  
BNE TS10-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #0,@#0 ;CLEAR LOCATION  
SEC ;SET CARRY  
ROL @#0 ;ROTATE CARRY BIT TO BIT 0  
BCC TS10  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 770 <====  
MOV #12,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 12 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CARRY CLEAR  
; OR SEQUENCE ERROR  
CMP #1,@#0 ;BIT 0 SET  
BEQ TS10  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 760 <====  
MOV #13,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 13 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;BIT 0 NOT SET  
; OR SEQUENCE ERROR

639  
640  
641  
642  
643 001512 005212  
644 001514 022712 000010  
645 001520 001014  
646 001522 012737 000001 000000  
647 001530 012700 177757  
648 001534 000241  
649 001536 005200  
650 001540 001404  
651 001542 006137 000000  
652 001546 103373  
653 001550 001404  
654  
655  
656  
657  
658 001552  
659 001552 012742 000014  
660 001556 005242  
661 001560 000000  
662  
663  
664  
665  
666  
667 001562 005212  
668 001564 022712 000011  
669 001570 001012  
670 001572 012737 100000 000000  
671 001600 000241  
672 001602 006037 000000  
673 001606 022737 040000 000000  
674 001614 001404  
675  
676  
677  
678  
679 001616 012742 000015  
680 001622 005242  
681 001624 000000  
682  
683  
684  
685  
686  
687 001626 005212  
688 001630 022712 000012  
689 001634 001014  
690 001636 012737 100000 000000  
691 001644 012700 177757  
692 001650 000241  
693 001652 005200  
694 001654 001404

\*\*\*\*\*  
:TEST 10 LEFT SHIFT FROM BIT 0 TO C-BIT  
\*\*\*\*\*

TS10: INC (R2) ;UPDATE TEST NUMBER  
CMP #10,(R2) ;SEQUENCE ERROR?  
BNE TS11-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #1,a#0 ;SET BIT 0  
MOV #-21,R0 ;SET BIT COUNTER  
CLC ;CLEAR C-BIT  
SHL: INC R0 ;INCREMENT BIT COUNTER  
BEQ SHLE ;BR TO ERROR HALT IF BIT IS LOST  
ROL a#0 ;SHIFT LEFT ONE POSITION  
BCC SHL ;BRANCH IF C-BIT NOT SET  
BEQ TS11

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 763 <====

SHLE: MOV #14,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 14 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;LEFT SHIFTING LOGIC FAILED  
; OR SEQUENCE ERROR

\*\*\*\*\*  
:TEST 11 SHIFT BIT 15 TO BIT 14  
\*\*\*\*\*

TS11: INC (R2) ;UPDATE TEST NUMBER  
CMP #11,(R2) ;SEQUENCE ERROR?  
BNE TS12-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #100000,a#0 ;SET BIT 15  
CLC ;CLEAR CARRY  
ROR a#0 ;SHIFT BIT 15 TO BIT 14  
CMP #40000,a#0 ;SUCCESSFUL  
BEQ TS12

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 765 <====

MOV #15,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 15 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;BIT 14 NOT SET  
; OR SEQUENCE ERROR

\*\*\*\*\*  
:TEST 12 RIGHT SHIFT FROM BIT 15 TO C-BIT  
\*\*\*\*\*

TS12: INC (R2) ;UPDATE TEST NUMBER  
CMP #12,(R2) ;SEQUENCE ERROR?  
BNE TS13-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #100000,a#0 ;SET BIT 15  
MOV #-21,R0 ;SET BIT COUNTER  
CLC ;CLEAR C-BIT  
SHR: INC R0 ;INCREMENT BIT COUNTER  
BEQ SHRE ;BR TO ERROR HALT IF BIT IS LOST



```
695 001656 006037 000000      ROR      @#0      ;ROTATE RIGHT ONE POSITION
696 001662 103373      BCC      SHR      ;BRANCH IF C-BIT CLEAR
697 001664 001404      BEQ      TS13
698                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
699                                     ;                               CONDITIONAL BRANCH INST. AND <====
700                                     ;                               REPLACE THE MOVE INSTRUCTION <====
701                                     ;                               WHICH FOLLOWS W/ 763 <====
702 001666                                     SHRE:
703 001666 012742 000016      MOV      #16,-(R2) ;MOVE TO MAILBOX # ***** 16 *****
704 001672 005242      INC      -(R2)    ;SET MSGTYP TO FATAL ERROR
705 001674 000000      HALT          ;RIGHT SHIFT LOGIC FAILED
706                                     ; OR SEQUENCE ERROR
707
708
709 :*****
710 :SBTTL  SCRATCH PAD TESTS
711 :
712 :      THE SCRATCH PAD TESTS ARE USED TO VERIFY THAT VARIOUS
713 :DATA PATTERNS CAN BE SUCCESSFULLY HELD IN THE SCRATCH PAD
714 :CIRCUITRY. MOVE AND COMPARE INSTRUCTIONS ARE USED TO TEST THAT
715 :R0 CAN HOLD VARIOUS DATA PATTERNS. EACH DATA PATTERN IS
716 :MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR SCOPING. THE
717 :SUCCESSFUL COMPLETION OF THESE TESTS SHOULD VERIFY THE CIRCUITRY EXTERNAL
718 :TO THE SCRATCH PAD ITSELF.
719 :      THE REMAINDER OF THE GENERAL REGISTERS ARE TESTED BY MOVING
720 :A BIT INTO BIT 0 OF THE REGISTER AND SHIFTING IT LEFT ONE
721 :BIT AT A TIME INTO THE CARRY BIT. THE RESULT IS THEN CHECKED TO INSURE THAT
722 :NO BITS WERE PICKED. THE PROCEDURE IS THEN REPEATED UNDER OPPOSITE
723 :CONDITIONS. THE GENERAL REGISTER AND THE CARRY BIT ARE SET TO
724 :ALL ONES, AND A ZERO IS SHIFTED LEFT FROM BIT 0 INTO THE CARRY BIT.
725 :THE RESULT IS THEN CHECKED TO INSURE THAT NO ZEROES WERE PICKED.
726 :      AT THIS POINT ALL OF THE GENERAL REGISTERS HAVE BEEN EXERCISED
727 :AS WELL AS REGISTER 11.
728 :
729 :*****
730 :TEST 13      TEST IF R0 CAN HOLD ALL ZEROES
731 :*****
732 TS13:  INC      (R2)      ;UPDATE TEST NUMBER
733      CMP      #13,(R2)   ;SEQUENCE ERROR?
734      BNE      TS14-10    ;BR TO ERROR HALT ON SEQ ERROR
735
736      MOV      #0,R0      ;MOVE ZEROES TO R0
737      TST      R0         ;SUCCESSFUL?
738      BEQ      TS14
739                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
740                                     ;                               CONDITIONAL BRANCH INST. AND <====
741                                     ;                               REPLACE THE MOVE INSTRUCTION <====
742                                     ;                               WHICH FOLLOWS W/ 773 <====
743 001716 012742 000017      MOV      #17,-(R2) ;MOVE TO MAILBOX # ***** 17 *****
744 001722 005242      INC      -(R2)    ;SET MSGTYP TO FATAL ERROR
745 001724 000000      HALT          ;R0 NOT 0
746                                     ; OR SEQUENCE ERROR
747
748 :*****
749 :TEST 14      TEST IF R0 CAN HOLD ONES AND ZEROES
750 :*****
751 TS14:  INC      (R2)      ;UPDATE TEST NUMBER
```

```
751 001730 022712 000014      CMP      #14,(R2)      ;SEQUENCE ERROR?  
752 001734 001005              BNE      TS15-10      ;BR TO ERROR HALT ON SEQ ERROR  
753 001736 012700 125252      MOV      #125252,R0   ;MOVE ALTERNATING ONES AND ZEROES TO R0  
754 001742 020027 125252      CMP      R0,#125252   ;SUCCESSFUL?  
755 001746 001404              BEQ      TS15  
756                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
757                          ; CONDITIONAL BRANCH INST. AND <====  
758                          ; REPLACE THE MOVE INSTRUCTION <====  
759                          ; WHICH FOLLOWS W/ 772 <====  
760 001750 012742 000020      MOV      #20,-(R2)    ;MOVE TO MAILBOX # ***** 20 *****  
761 001754 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR  
762 001756 000000              HALT                    ;RO NOT 125252  
763                          ; OR SEQUENCE ERROR
```

\*\*\*\*\*  
:TEST 15 TEST IF R0 CAN HOLD ZEROES AND ONES  
\*\*\*\*\*

```
768 001760 005212              TS15:  INC      (R2)      ;UPDATE TEST NUMBER  
769 001762 022712 000015      CMP      #15,(R2)    ;SEQUENCE ERROR?  
770 001766 001005              BNE      TS16-10     ;BR TO ERROR HALT ON SEQ ERROR  
771 001770 012700 052525      MOV      #052525,R0  ;MOVE ALTERNATING ZEROES AND ONES TO R0  
772 001774 020027 052525      CMP      R0,#052525  ;SUCCESSFUL?  
773 002000 001404              BEQ      TS16  
774                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
775                          ; CONDITIONAL BRANCH INST. AND <====  
776                          ; REPLACE THE MOVE INSTRUCTION <====  
777                          ; WHICH FOLLOWS W/ 772 <====  
778 002002 012742 000021      MOV      #21,-(R2)    ;MOVE TO MAILBOX # ***** 21 *****  
779 002006 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR  
780 002010 000000              HALT                    ;RO NOT 52525  
781                          ; OR SEQUENCE ERROR
```

\*\*\*\*\*  
:TEST 16 TEST IF R0 CAN HOLD ALL ONES  
\*\*\*\*\*

```
786 002012 005212              TS16:  INC      (R2)      ;UPDATE TEST NUMBER  
787 002014 022712 000016      CMP      #16,(R2)    ;SEQUENCE ERROR?  
788 002020 001005              BNE      TS17-10     ;BR TO ERROR HALT ON SEQ ERROR  
789 002022 012700 177777      MOV      #177777,R0  ;MOVE ALL ONES TO R0  
790 002026 020027 177777      CMP      R0,#177777  ;SUCCESSFUL?  
791 002032 001404              BEQ      TS17  
792                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
793                          ; CONDITIONAL BRANCH INST. AND <====  
794                          ; REPLACE THE MOVE INSTRUCTION <====  
795                          ; WHICH FOLLOWS W/ 772 <====  
796 002034 012742 000022      MOV      #22,-(R2)    ;MOVE TO MAILBOX # ***** 22 *****  
797 002040 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR  
798 002042 000000              HALT                    ;RO NOT 177777  
799                          ; OR SEQUENCE ERROR
```

\*\*\*\*\*  
:TEST 17 TEST IF R1 CAN HOLD A ONE IN ALL BITS  
\*\*\*\*\*

```
804 002044 005212              TS17:  INC      (R2)      ;UPDATE TEST NUMBER  
805 002046 022712 000017      CMP      #17,(R2)    ;SEQUENCE ERROR?  
806 002052 001012              BNE      TS20-10     ;BR TO ERROR HALT ON SEQ ERROR
```

807 002054 012701 000001  
808 002060 012700 177757  
809 002064 000241  
810 002066 005200  
811 002070 001403  
812 002072 006101  
813 002074 103374  
814 002076 001404

REG1: MOV #1,R1 ;SET BIT 0  
MOV #-21,R0 ;SET BIT COUNTER  
CLC ;CLEAR C-BIT  
INC R0 ;INCREMENT BIT COUNTER  
BEQ REG1E ;BR TO ERROR HALT IF BIT IS LOST  
ROL R1 ;ROTATE 1 POSITION  
BCC REG1 ;ALL DONE  
BEQ TS20

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 765 <====

819 002100  
820 002100 012742 000023  
821 002104 005242  
822 002106 000000

REG1E: MOV #23,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 23 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;FAILURE WITH R1  
; OR SEQUENCE ERROR

\*\*\*\*\*  
:TEST 20 TEST IF R1 CAN HOLD A ZERO IN ALL BITS  
\*\*\*\*\*

828 002110 005212  
829 002112 022712 000020  
830 002116 001014  
831 002120 012701 177776  
832 002124 012700 177757  
833 002130 000261  
834 002132 005200  
835 002134 001405  
836 002136 006101  
837 002140 103774  
838 002142 022701 177777  
839 002146 001404

TS20: INC (R2) ;UPDATE TEST NUMBER  
CMP #20,(R2) ;SEQUENCE ERROR?  
BNE TS21-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #-2,R1 ;SET ALL ONES IN R1 EXCEPT FOR BIT 0  
MOV #-21,R0 ;SET BIT COUNTER  
SEC ;SET C-BIT  
REG1A: INC R0 ;INCREMENT COUNTER  
BEQ R1ERR ;BR TO ERROR HALT IF COUNTER=0  
ROL R1 ;ROTATE 1 POSITION  
BCS REG1A ;CONTINUE UNTIL C-BIT IS CLEAR  
CMP #-1,R1 ;CHECK DATA IN R1  
BEQ TS21

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 763 <====

844 002150  
845 002150 012742 000024  
846 002154 005242  
847 002156 000000

R1ERR: MOV #24,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 24 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;FAILURE WITH R1  
; OR SEQUENCE ERROR

\*\*\*\*\*  
:TEST 21 TEST IF R2 CAN HOLD A ONE IN ALL BITS  
\*\*\*\*\*

852 002160 005212  
853 002162 022712 000021  
854 002166 001012  
855 002170 012702 000001  
856 002174 012700 177757  
857 002200 000241  
858 002202 005200  
859 002204 001403  
860 002206 006102  
861 002210 103374  
862 002212 001406

TS21: INC (R2) ;UPDATE TEST NUMBER  
CMP #21,(R2) ;SEQUENCE ERROR?  
BNE REG2A-14 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #1,R2 ;SET BIT 0  
MOV #-21,R0 ;SET BIT COUNTER  
CLC ;CLEAR C-BIT  
REG2: INC R0 ;INCREMENT BIT COUNTER  
BEQ REG2A-14 ;BR TO ERROR HALT IF BIT IS LOST  
ROL R2 ;ROTATE 1 POSITION  
BCC REG2 ;ALL DONE  
BEQ REG2A

```
863
864
865
866
867 002214 012702 000304      MOV    #STESTN,R2      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
868 002220 012742 000025      MOV    #25,-(R2)      ; BRANCH INSTRUCTION AND <====
869 002224 005242              INC    -(R2)          ; REPLACE THE MOVE INSTRUCTION <====
870 002226 000000              HALT                    ; FOLLOWING W/ 771 <====
871 002230 012702 000304      REG2A: MOV    #STESTN,R2      ;RESTORE POINTER
872
873
874 ;*****
875 ;TEST 22      TEST IF R2 CAN HOLD A ZERO IN ALL BITS
876 002234 005212              TS22:  INC    (R2)          ;UPDATE TEST NUMBER
877 002236 022712 000022      CMP    #22,(R2)      ;SEQUENCE ERROR?
878 002242 001020              BNE    TS23-10        ;BR TO ERROR HALT ON SEQ ERROR
879 002244 012702 177776      MOV    #-2,R2        ;SET ALL ONES IN R2 EXCEPT FOR BIT 0
880 002250 012700 177757      MOV    #-21,R0       ;SET BIT COUNTER
881 002254 000261              SEC                    ;SET C-BIT
882 002256 005200      REG2B:  INC    R0          ;INCREMENT BIT COUNTER
883 002260 001407              BEQ    R2ERR          ;BR TO ERROR HALT IF COUNTER=0
884 002262 006102              ROL    R2             ;ROTATE 1 POSITION
885 002264 103774              BCS    REG2B          ;CONTINUE UNTIL C-BIT IS CLEAR
886 002266 022702 177777      CMP    #-1,R2        ;CHECK DATA IN R2
887 002272 001406              BEQ    REG2C
888 002274 012702 000304      MOV    #STESTN,R2    ;RESTORE POINTER
889 002300      R2ERR:
890 002300 012742 000026      MOV    #26,-(R2)     ;MOVE TO MAILBOX # ***** 26 *****
891 002304 005242              INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
892 002306 000000              HALT                    ;FAILURE WITH R2
893 002310 012702 000304      REG2C: MOV    #STESTN,R2    ;RESTORE POINTER
894
895 ;*****
896 ;TEST 23      TEST IF R3 CAN HOLD A ONE IN ALL BITS
897 ;*****
898 002314 005212              TS23:  INC    (R2)          ;UPDATE TEST NUMBER
899 002316 022712 000023      CMP    #23,(R2)      ;SEQUENCE ERROR?
900 002322 001012              BNE    TS24-10        ;BR TO ERROR HALT ON SEQ ERROR
901 002324 012703 000001      MOV    #1,R3         ;SET BIT 0
902 002330 012700 177757      MOV    #-21,R0       ;SET BIT COUNTER
903 002334 000241              CLC                    ;CLEAR C-BIT
904 002336 005200      REG3:  INC    R0          ;INCREMENT BIT COUNTER
905 002340 001403              BEQ    REG3E          ;BR TO ERROR HALT IF BIT IS LOST
906 002342 006103              ROL    R3             ;ROTATE 1 POSITION
907 002344 103374              BCC    REG3           ;ALL DONE
908 002346 001404              BEQ    TS24
909
910 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
911 ;          CONDITIONAL BRANCH INST. AND <====
912 ;          REPLACE THE MOVE INSTRUCTION <====
913 ;          WHICH FOLLOWS W/ 765 <====
914 002350      REG3E:
915 002350 012742 000027      MOV    #27,-(R2)     ;MOVE TO MAILBOX # ***** 27 *****
916 002354 005242              INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
917 002356 000000              HALT                    ;FAILURE WITH R3
918 ; OR SEQUENCE ERROR
```

919  
920  
921  
922 002360 005212  
923 002362 022712 000024  
924 002366 001014  
925 002370 012703 177776  
926 002374 012700 177757  
927 002400 000261  
928 002402 005200  
929 002404 001405  
930 002406 006103  
931 002410 103774  
932 002412 022703 177777  
933 002416 001404  
934  
935  
936  
937  
938 002420  
939 002420 012742 000030  
940 002424 005242  
941 002426 000000  
942  
943  
944  
945  
946  
947 002430 005212  
948 002432 022712 000025  
949 002436 001012  
950 002440 012704 000001  
951 002444 012700 177757  
952 002450 000241  
953 002452 005200  
954 002454 001403  
955 002456 006104  
956 002460 103374  
957 002462 001404  
958  
959  
960  
961  
962 002464  
963 002464 012742 000031  
964 002470 005242  
965 002472 000000  
966  
967  
968  
969  
970  
971 002474 005212  
972 002476 022712 000026  
973 002502 001014  
974 002504 012704 177776

```
*****  
:TEST 24 TEST IF R3 CAN HOLD A ZERO IN ALL BITS  
*****  
TS24: INC (R2) ;UPDATE TEST NUMBER  
CMP #24,(R2) ;SEQUENCE ERROR?  
BNE TS25-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #-2,R3 ;SET ALL ONES IN R3 EXCEPT FOR BIT 0  
MOV #-21,R0 ;SET BIT COUNTER  
SEC ;SET C-BIT  
REG3A: INC R0 ;INCREMENT BIT COUNTER  
BEQ R3ERR ;BR TO ERROR HALT IF COUNTER=0  
ROL R3 ;ROTATE 1 POSITION  
BCS REG3A ;CONTINUE UNTIL C-BIT IS CLEAR  
CMP #-1,R3 ;CHECK DATA  
BEQ TS25  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 763 <====  
  
R3ERR: MOV #30,-(R2) ;MOVE TO MAILBOX # ***** 30 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;FAILURE WITH R3  
; OR SEQUENCE ERROR  
  
*****  
:TEST 25 TEST IF R4 CAN HOLD A ONE IN ALL BITS  
*****  
TS25: INC (R2) ;UPDATE TEST NUMBER  
CMP #25,(R2) ;SEQUENCE ERROR?  
BNE TS26-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #1,R4 ;SET BIT 0  
MOV #-21,R0 ;SET BIT COUNTER  
CLC ;CLEAR C-BIT  
REG4: INC R0 ;INCREMENT BIT COUNTER  
BEQ REG4E ;BR TO ERROR HALT IF BIT IS LOST  
ROL R4 ;ROTATE 1 POSITION  
BCC REG4 ;ALL DONE  
BEQ TS26  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 765 <====  
  
REG4E: MOV #31,-(R2) ;MOVE TO MAILBOX # ***** 31 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;FAILURE WITH R4  
; OR SEQUENCE ERROR  
  
*****  
:TEST 26 TEST IF R4 CAN HOLD A ZERO IN ALL BITS  
*****  
TS26: INC (R2) ;UPDATE TEST NUMBER  
CMP #26,(R2) ;SEQUENCE ERROR?  
BNE TS27-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #-2,R4 ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
```

975 002510 012700 177757  
976 002514 000261  
977 002516 005200  
978 002520 001405  
979 002522 006104  
980 002524 103774  
981 002526 022704 177777  
982 002532 001404

REG4A: MOV #-21,R0 ;SET BIT COUNTER  
SEC ;SET C-BIT  
INC R0 ;INCREMENT BIT COUNTER  
BEQ R4ERR ;BR TO ERROR HALT IF COUNTER=0  
ROL R4 ;ROTATE 1 POSITION  
BCS REG4A ;CONTINUE UNTIL C-BIT IS CLEAR  
CMP #-1,R4 ;CHECK DATA  
BEQ TS27

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 763 <====

987 002534  
988 002534 012742 000032  
989 002540 005242  
990 002542 000000  
991  
992  
993

R4ERR: MOV #32,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 32 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;FAILURE WITH R4  
; OR SEQUENCE ERROR

994  
995  
996

:\*\*\*\*\*  
:TEST 27 TEST IF R5 CAN HOLD A ONE IN ALL BITS  
:\*\*\*\*\*

997 002544 005212  
998 002546 022712 000027  
999 002552 001012  
1000 002554 012705 000001  
1001 002560 012700 177757  
1002 002564 000241  
1003 002566 005200  
1004 002570 001403  
1005 002572 006105  
1006 002574 103374  
1007 002576 001404

TS27: INC (R2) ;UPDATE TEST NUMBER  
CMP #27,(R2) ;SEQUENCE ERROR?  
BNE TS30-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #1,R5 ;SET BIT 0  
MOV #-21,R0 ;SET BIT COUNTER  
CLC ;CLEAR C-BIT  
REG5: INC R0 ;INCREMENT BIT COUNTER  
BEQ REG5E ;BR TO ERROR HALT IF BIT IS LOST  
ROL R5 ;ROTATE 1 POSITION  
BCC REG5 ;ALL DONE  
BEQ TS30

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 765 <====

1008  
1009  
1010  
1011  
1012 002600  
1013 002600 012742 000033  
1014 002604 005242  
1015 002606 000000  
1016  
1017

REG5E: MOV #33,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 33 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;FAILURE WITH R5  
; OR SEQUENCE ERROR

1018  
1019  
1020

:\*\*\*\*\*  
:TEST 30 TEST IF R5 CAN HOLD A ZERO IN ALL BITS  
:\*\*\*\*\*

1021 002610 005212  
1022 002612 022712 000030  
1023 002616 001014  
1024 002620 012705 177776  
1025 002624 012700 177757  
1026 002630 000261  
1027 002632 005200  
1028 002634 001405  
1029 002636 006105  
1030 002640 103774

TS30: INC (R2) ;UPDATE TEST NUMBER  
CMP #30,(R2) ;SEQUENCE ERROR?  
BNE TS31-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #-2,R5 ;SET ALL ONES IN R5 EXCEPT FOR BIT 0  
MOV #-21,R0 ;SET BIT COUNTER  
SEC ;SET C-BIT  
REG5A: INC R0 ;INCREMENT BIT COUNTER  
BEQ R5ERR ;BR TO ERROR HALT IF COUNTER=0  
ROL R5 ;ROTATE 1 POSITION  
BCS REG5A ;CONTINUE UNTIL C-BIT IS CLEAR

```
1031 002642 022705 177777          CMP    #-1,R5          ;CHECK DATA
1032 002646 001404          BEQ    TS31
1033                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1034                               ;          CONDITIONAL BRANCH INST. AND <====
1035                               ;          REPLACE THE MOVE INSTRUCTION <====
1036                               ;          WHICH FOLLOWS W/ 763 <====
1037 002650          RSERR:
1038 002650 012742 000034      MOV    #34,-(R2)      ;MOVE TO MAILBOX # ***** 34 *****
1039 002654 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
1040 002656 000000          HALT                ;FAILURE WITH R5
1041                               ; OR SEQUENCE ERROR
1042
1043                               ;*****
1044                               ;TEST 31          TEST IF R6 CAN HOLD A ONE IN ALL BITS
1045                               ;*****
1046 002660 005212          TS31:  INC    (R2)          ;UPDATE TEST NUMBER
1047 002662 022712 000031      CMP    #31,(R2)      ;SEQUENCE ERROR?
1048 002666 001012          BNE    TS32-10      ;BR TO ERROR HALT ON SEQ ERROR
1049 002670 012706 000001      MOV    #1,R6         ;SET BIT 0
1050 002674 012700 177757      MOV    #-21,R0      ;SET BIT COUNTER
1051 002700 000241          CLC                ;CLEAR C-BIT
1052 002702 005200          REG6: INC    R0         ;INCREMENT BIT COUNTER
1053 002704 001403          BEQ    REG6E        ;BR TO ERROR HALT IF BIT IS LOST
1054 002706 006106          ROL    R6           ;ROTATE 1 POSITION
1055 002710 103374          BCC    REG6         ;ALL DONE
1056 002712 001404          BEQ    TS32
1057                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1058                               ;          CONDITIONAL BRANCH INST. AND <====
1059                               ;          REPLACE THE MOVE INSTRUCTION <====
1060                               ;          WHICH FOLLOWS W/ 765 <====
1061 002714          REG6E:
1062 002714 012742 000035      MOV    #35,-(R2)      ;MOVE TO MAILBOX # ***** 35 *****
1063 002720 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
1064 002722 000000          HALT                ;FAILURE WITH R6
1065                               ; OR SEQUENCE ERROR
1066
1067                               ;*****
1068                               ;TEST 32          TEST IF R6 CAN HOLD A ZERO IN ALL BITS
1069                               ;*****
1070 002724 005212          TS32:  INC    (R2)          ;UPDATE TEST NUMBER
1071 002726 022712 000032      CMP    #32,(R2)      ;SEQUENCE ERROR?
1072 002732 001014          BNE    TS33-10      ;BR TO ERROR HALT ON SEQ ERROR
1073 002734 012706 177776      MOV    #-2,R6        ;SET ALL ONES IN R6 EXCEPT FOR BIT 0
1074 002740 012700 177757      MOV    #-21,R0      ;SET BIT COUNTER
1075 002744 000261          SEC                ;SET C-BIT
1076 002746 005200          REG6A: INC    R0         ;INCREMENT BIT COUNT
1077 002750 001405          BEQ    R6ERR        ;BR TO ERROR HALT IF COUNTER=0
1078 002752 006106          ROL    R6           ;ROTATE 1 POSITION
1079 002754 103774          BCS    REG6A        ;CONTINUE UNTIL C-BIT IS CLEAR
1080 002756 022706 177777      CMP    #-1,R6        ;CHECK DATA
1081 002762 001404          BEQ    TS33
1082                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1083                               ;          CONDITIONAL BRANCH INST. AND <====
1084                               ;          REPLACE THE MOVE INSTRUCTION <====
1085                               ;          WHICH FOLLOWS W/ 763 <====
1086 002764          R6ERR:
```

1087 002764 012742 000036  
1088 002770 005242  
1089 002772 000000

MOV #36,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 36 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;FAILURE WITH R6  
; OR SEQUENCE ERROR

1090  
1091

1092  
1093

\*\*\*\*\*  
:SBTTL PSW TESTS

1094

1095

1096

1097

1098

1099

1100

1101

1102

1103

1104

1105

1106

1107

1108

THE PSW TESTS ARE USED TO VERIFY THAT VARIOUS DATA  
PATTERNS CAN BE SUCCESSFULLY HELD IN THE PSW AND THAT THE  
PSW ADDRESSING LOGIC IS FUNCTIONING. MOVE AND COMPARE INSTRUCTIONS  
ARE USED TO TEST THAT THE PSW CAN HOLD VARIOUS DATA PATTERNS.  
EACH DATA PATTERN IS MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR  
SCOPING.  
THE PSW REGISTER IS TESTED, THE CC INPUTS ARE TESTED  
LATER IN THE MICROCODE TESTS. SETTING OF THE T-BIT BY THE  
TEST PATTERNS IS PURPOSELY AVOIDED. TESTING OF THE  
T-BIT TRAP CIRCUITRY IS LEFT FOR THE TRAP TEST.

1109

1110

1111

1112

1113

1114

1115

1116

1117

1118

1119

1120

1121

1122

1123

1124

1125

\*\*\*\*\*  
:TEST 33 TEST IF PSW WILL HOLD ZEROES  
\*\*\*\*\*

1126

1127

1128

1129

1130

1131

1132

1133

1134

1135

1136

1137

1138

1139

1140

1141

1142

TS33: INC (R2) ;UPDATE TEST NUMBER  
CMP #33,(R2) ;SEQUENCE ERROR?  
BNE TS34-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #STBOT,R6  
MOV #0,@#PS ;SET PSW TO ZERO  
TST @#PS ;SUCCESSFUL  
BEQ TS34

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 767 <====

MOV #37,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 37 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;PSW NOT 0  
; OR SEQUENCE ERROR

\*\*\*\*\*  
:TEST 34 TEST IF PSW WILL HOLD ONES AND ZEROES  
\*\*\*\*\*

TS34: INC (R2) ;UPDATE TEST NUMBER  
CMP #34,(R2) ;SEQUENCE ERROR?  
BNE TS35-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #252,@#PS ;MOVE ALT. ONES AND ZEROES TO PSW  
CMP @#PS,#252 ;SUCCESSFUL?  
BEQ TS35

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 770 <====

MOV #40,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 40 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;PSW NOT 252  
; OR SEQUENCE ERROR



1143  
1144  
1145  
1146 003072 005212  
1147 003074 022712 000035  
1148 003100 001007  
1149 003102 012737 000105 177776  
1150 003110 023727 177776 000105  
1151 003116 001404  
1152  
1153  
1154  
1155  
1156 003120 012742 000041  
1157 003124 005242  
1158 003126 000000  
1159  
1160  
1161  
1162  
1163  
1164 003130 005212  
1165 003132 022712 000036  
1166 003136 001007  
1167 003140 012737 000357 177776  
1168 003146 023727 177776 000357  
1169 003154 001404  
1170  
1171  
1172  
1173  
1174 003156 012742 000042  
1175 003162 005242  
1176 003164 000000  
1177  
1178  
1179  
1180  
1181  
1182  
1183  
1184  
1185  
1186  
1187  
1188  
1189  
1190  
1191  
1192  
1193  
1194  
1195  
1196 003166 005212  
1197 003170 022712 000037  
1198 003174 001014

```
*****  
:TEST 35 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES  
*****  
TS35: INC (R2) ;UPDATE TEST NUMBER  
CMP #35,(R2) ;SEQUENCE ERROR?  
BNE TS36-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #105,@#PS ;MOVE ALT. ONES AND ZEROES TO PSW  
CMP @#PS,#105 ;SUCCESSFUL?  
BEQ TS36  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 770 <====  
MOV #41,-(R2) ;MOVE TO MAILBOX # ***** 41 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;PSW NOT 105  
; OR SEQUENCE ERROR  
  
*****  
:TEST 36 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES  
*****  
TS36: INC (R2) ;UPDATE TEST NUMBER  
CMP #36,(R2) ;SEQUENCE ERROR?  
BNE TS37-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #357,@#PS ;MOVE ONES TO PSW  
CMP @#PS,#357 ;SUCCESSFUL  
BEQ TS37  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 770 <====  
MOV #42,-(R2) ;MOVE TO MAILBOX # ***** 42 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;PSW NOT 357  
; OR SEQUENCE ERROR  
  
.SBTTL CONDITION CODE TEST  
  
*****  
: THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE Z-BIT.  
: THE Z-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS  
: BEQ AND BNE ARE TESTED FOR PROPER EXECUTION. THEN THE Z-BIT IS  
: SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED  
: AGAIN FOR PROPER OPERATION.  
: THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION  
: CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL  
: BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR  
: LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY  
: USED IN THE TEST ARE VERIFIED HERE.  
*****  
:TEST 37 TEST BRANCHES AROUND Z-BIT  
*****  
TS37: INC (R2) ;UPDATE TEST NUMBER  
CMP #37,(R2) ;SEQUENCE ERROR?  
BNE TS40-10 ;BR TO ERROR HALT ON SEQ ERROR
```

```
1199                                     ;FIRST WITH Z-BIT ON
1200 003176 000257                       CCC           ;CC=0100: JUST Z-BIT
1201 003200 000264                       SEZ
1202 003202 001001                       BNE BRZ1       ;CHECK OPPOSITE CONDITION
1203 003204 001404                       BEQ BRZ2
1204                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1205                                     ;                               CONDITIONAL BRANCH INST. AND <====
1206                                     ;                               REPLACE THE MOVE INSTRUCTION <====
1207                                     ;                               WHICH FOLLOWS W/ 773 <====
1208 003206                                BRZ1:
1209 003206 012742 000043                 MOV #43,-(R2)   ;MOVE TO MAILBOX # ***** 43 *****
1210 003212 005242                         INC -(R2)       ;SET MSGTYP TO FATAL ERROR
1211 003214 000000                         HALT           ;IMPROPER BR W/ Z=1
1212                                     ;CHECK WITH Z-BIT OFF
1213 003216 000277                       SCC           ;CC=1011: ALL BUT Z-BIT
1214 003220 000244                       CLZ
1215 003222 001401                       BEQ BRZ3
1216 003224 001004                       BNE TS40
1217                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1218                                     ;                               CONDITIONAL BRANCH INST. AND <====
1219                                     ;                               REPLACE THE MOVE INSTRUCTION <====
1220                                     ;                               WHICH FOLLOWS W/ 763 <====
1221 003226                                BRZ3:
1222 003226 012742 000044                 MOV #44,-(R2)   ;MOVE TO MAILBOX # ***** 44 *****
1223 003232 005242                         INC -(R2)       ;SET MSGTYP TO FATAL ERROR
1224 003234 000000                         HALT           ;IMPROPER BR W/ Z=0
1225                                     ; OR SEQUENCE ERROR
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243 003236 005212 000040                 TS40: INC (R2)   ;UPDATE TEST NUMBER
1244 003240 022712                         CMP #40,(R2)   ;SEQUENCE ERROR?
1245 003244 001014                         BNE TS41-10    ;BR TO ERROR HALT ON SEQ ERROR
1246                                     ;FIRST WITH N-BIT ON
1247 003246 000257                       CCC           ;CC=1000: JUST N-BIT
1248 003250 000270                       SEN
1249 003252 100001                       BPL BRN1       ;CHECK OPPOSITE CONDITION
1250 003254 100404                       BMI BRN2
1251                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1252                                     ;                               CONDITIONAL BRANCH INST. AND <====
1253                                     ;                               REPLACE THE MOVE INSTRUCTION <====
1254                                     ;                               WHICH FOLLOWS W/ 773 <====
```

```
.....
:
: THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE N-BIT.
: THE N-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
: BMI AND BPL ARE TESTED FOR PROPER EXECUTION. THEN THE N-BIT IS
: SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
: AGAIN FOR PROPER OPERATION.
: THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
: CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
: BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
: LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY
: USED IN THE TEST ARE VERIFIED HERE.
:
:.....
```

```
.....
: TEST 40 TEST BRANCHES AROUND N-BIT
:.....
```

1255 003256  
1256 003256 012742 000045  
1257 003262 005242  
1258 003264 000000  
1259  
1260 003266 000277  
1261 003270 000250  
1262 003272 100401  
1263 003274 100004

BRN1: MOV #45, -(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 45 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;IMPROPER BR W/ N=1  
;CHECK WITH N-BIT OFF  
BRN2: SCC ;CC=0111  
CLN  
BMI BRN3 ;CHECK OPPOSITE CONDITION  
BPL TS41

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 763 <====

1268 003276  
1269 003276 012742 000046  
1270 003302 005242  
1271 003304 000000  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284

BRN3: MOV #46, -(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 46 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;IMPROPER BR W/ N=0  
; OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE V-BIT.  
: THE V-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS  
: BVS AND BVC ARE TESTED FOR PROPER EXECUTION. THEN THE V-BIT IS  
: SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED  
: AGAIN FOR PROPER OPERATION.  
\*\*\*\*\*

TEST 41 TEST BRANCHES AROUND V-BIT

1285 003306 005212  
1286 003310 022712 000041  
1287 003314 001014  
1288  
1289 003316 000257  
1290 003320 000262  
1291 003322 102001  
1292 003324 102404  
1293  
1294  
1295  
1296  
1297 003326  
1298 003326 012742 000047  
1299 003332 005242  
1300 003334 000000  
1301  
1302 003336 000277  
1303 003340 000242  
1304 003342 102401  
1305 003344 102004  
1306  
1307  
1308  
1309  
1310 003346

TS41: INC (R2) ;UPDATE TEST NUMBER  
CMP #41, (R2) ;SEQUENCE ERROR?  
BNE TS42-10 ;BR TO ERROR HALT ON SEQ ERROR  
;FIRST WITH V-BIT ON  
CCC ;CC=0010: JUST V-BIT  
SEV  
BVC BRV1 ;CHECK OPPOSITE CONDITION  
BVS BRV2

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 773 <====

BRV1: MOV #47, -(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 47 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;IMPROPER BR W/ V=1  
;CHECK WITH V-BIT OFF

BRV2: SCC ;CC=1101: ALL BVT V-BIT  
CLV  
BVS BRV3 ;CHECK OPPOSITE CONDITION  
BVC TS42

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 763 <====

BRV3:

1311 003346 012742 000050  
1312 003352 005242  
1313 003354 000000

MOV #50,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 50 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;IMPROPER BR W/ V=0  
; OR SEQUENCE ERROR

1314  
1315  
1316  
1317  
1318  
1319  
1320  
1321  
1322  
1323

.....  
: THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE C-BIT.  
: THE C-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS  
: BCS AND BCC ARE TESTED FOR PROPER EXECUTION. THEN THE C-BIT IS  
: SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED  
: AGAIN FOR PROPER OPERATION.  
: .....

1324  
1325  
1326

: TEST 42 TEST BRANCHES AROUND C-BIT  
: .....

1327 003356 005212  
1328 003360 022712 000042  
1329 003364 001014

TS42: INC (R2) ;UPDATE TEST NUMBER  
CMP #42,(R2) ;SEQUENCE ERROR?  
BNE TS43-10 ;BR TO ERROR HALT ON SEQ ERROR  
;FIRST WITH C-BIT ON

1330  
1331  
1332  
1333  
1334

CCC ;CC=0001: JUST C-BIT  
SEC  
BCC BRC1 ;CHECK OPPOSITE CONDITION  
BCS BRC2

1335  
1336  
1337  
1338

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 773 <====

1339 003376  
1340 003376 012742 000051  
1341 003402 005242  
1342 003404 000000

BRC1: MOV #51,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 51 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;IMPROPER BR W/ C=1  
;CHECK WITH C-BIT OFF

1343  
1344  
1345  
1346  
1347

BRC2: SCC ;CC=1110  
CLC  
BCS BRC3 ;CHECK OPPOSITE CONDITION  
BMI TS43

1348  
1349  
1350  
1351

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 763 <====

1352 003416  
1353 003416 012742 000052  
1354 003422 005242  
1355 003424 000000

BRC3: MOV #52,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 52 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;IMPROPER BR W/ C=0  
; OR SEQUENCE ERROR

1356  
1357  
1358  
1359

.....  
:SBTTL MICROCODE TESTS

1360  
1361  
1362  
1363  
1364  
1365  
1366

: THE TEST EXERCISES BRANCHES IN THE MICROCODE BY  
: TESTING AT LEAST ONE INSTRUCTION FROM EVERY CLASS OF INSTRUCTION IN  
: ALL POSSIBLE MODES. FOR EXAMPLE, TO TEST THE SINGLE OPERAND INSTRUCTIONS,  
: AT LEAST ONE SINGLE OPERAND INSTRUCTION IS VERIFIED IN ALL UNIQUE  
: ADDRESSING MODES. BYTE MODES ARE ALSO TESTED. AS EACH NEW  
: MODE IS INTRODUCED THE SAME INSTRUCTION IS TRIED AND TESTED IN

1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376  
1377  
1378  
1379  
1380  
1381  
1382  
1383  
1384  
1385  
1386  
1387  
1388  
1389  
1390  
1391  
1392  
1393  
1394  
1395  
1396  
1397  
1398  
1399  
1400  
1401  
1402  
1403  
1404  
1405  
1406  
1407  
1408  
1409  
1410  
1411  
1412  
1413  
1414  
1415  
1416  
1417  
1418  
1419  
1420  
1421  
1422

003426 005212  
003430 022712 000043  
003434 001020  
003436 005000  
003440 001404  
  
003442 012742 000053  
003446 005242  
003450 000000  
003452 005200  
003454 005100  
003456 005200  
003460 100404  
  
003462 012742 000054  
003466 005242  
003470 000000  
003472 005100  
003474 001404  
  
003476 012742 000055  
003502 005242

:A SMALL LOOP CONVENIENT FOR SCOPING. THE TEST IS SET UP USING  
:ONLY INSTRUCTIONS AND ADDRESSING MODES WHICH HAVE BEEN PREVIOUSLY  
:VERIFIED.  
: IF THESE TESTS FAIL, CHECK THE RESULTS FOR A CLUE TO THE  
:FAULT.  
:.....  
:.....  
: THE CLR INSTRUCTION IS USED TO INTRODUCE EACH ADDRESSING  
:MODE WITH THE SINGLE OPERAND INSTRUCTION. FOLLOWING THE SEQUENCE CHECK,  
:THE CLR INSTRUCTION IS EXECUTED AND A BRANCH TEST IS EXECUTED WHICH  
:CHECKS THAT THE Z-BIT WAS PROPERLY SET. THIS TEST CAN CHECK IR DECODE  
:AND MICROCODE FOR SOP INSTRUCTIONS WITH MODE 0. FOLLOWING THIS TEST  
:SEVERAL OTHER SOP INSTRUCTIONS ARE INTRODUCED WITH MODE 0. THESE  
:INSTRUCTIONS MAINPULATE DATA AND SERVE TO CHECK THE DATA RESULTS  
:OF THE SOP INSTRUCTIONS IN THIS TEST. THE DATA IN THIS TEST IS  
:OPERATED ON BY EACH INSTRUCTION WITHOUT REINITIALIZING.  
:.....  
:TEST 43 TEST MODE 0 USING SOP INST.  
:.....  
TS43: INC (R2) ;UPDATE TEST NUMBER  
CMP #43,(R2) ;SEQUENCE ERROR?  
BNE TS44-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;TRY THE CLEAR INST.  
BEQ SOPOA  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 775 <====  
MOV #53,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 53 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CLR DID NOT SET Z-BIT  
SOP0A: INC R0 ;TRY THE INCREMENT INST.  
COM R0 ;TRY COMPLEMENT  
INC R0  
BMI SOP0B  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 765 <====  
MOV #54,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 54 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;NEGATE DID NOT SET N-BIT  
SOP0B: COM R0 ;TRY COMPLEMENT INST.  
BEQ TS44  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 757 <====  
MOV #55,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 55 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR

1423 003504 000000

HALT

: CUMMULATIVE RESULT OF CLR, INC, NEG AND COM INSTS. FAILED  
: OR SEQUENCE ERROR

1424

1425

1426

1427

1428

1429

1430

1431

1432

1433

1434

1435

1436

1437

1438

1439 003506 005212

1440 003510 022712

000044

1441 003514 001021

1442 003516 005000

1443 003520 005300

1444 003522 100404

1445

1446

1447

1448

1449 003524 012742

000056

1450 003530 005242

1451 003532 000000

1452 003534 000261

1453 003536 005500

1454 003540 001007

1455 003542 000261

1456 003544 005600

1457 003546 100004

1458 003550 005100

1459 003552 005200

1460 003554 005300

1461 003556 001404

1462

1463

1464

1465

1466 003560

1467 003560 012742

000057

1468 003564 005242

1469 003566 000000

1470

1471

1472

1473

1474

1475

1476

1477

1478

\*\*\*\*\*

: THIS TEST INTRODUCES THE REMAINING SOP INSTRUCTIONS AND TESTS  
: THEM IN MODE 0. THE PURPOSE IS TO PROVIDE A BASELINE OF  
: INSTRUCTIONS FOR USE IN THE SUBSEQUENT TESTS. SINCE THE MICROCODE FOR  
: THESE INSTRUCTIONS IS IDENTICAL TO THAT ALREADY TESTED, ANY TROUBLE  
: SHOOTING EFFORTS SHOULD BE AIMED AT THE ACTUAL IR DECODE AND ALU  
: FUNCTIONING.

: TEST 44 TEST REMAINDER OF SOP INSTS IN MODE 0

TS44: INC (R2) : UPDATE TEST NUMBER

CMP #44, (R2) : SEQUENCE ERROR?

BNE TS45-10 : BR TO ERROR HALT ON SEQ ERROR

CLR R0 : INITIALIZE

DEC R0 : TRY DECREMENT INST.

BMI SOPOC

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 774 <====

MOV #56, -(R2) : MOVE TO MAILBOX # \*\*\*\*\* 56 \*\*\*\*\*

INC -(R2) : SET MSGTYP TO FATAL ERROR

SOPOC: HALT : N-BIT NOT SET ON DEC

SEC : INITIALIZE CARRY

ADC R0 : TRY ADD CARRY INST

BNE SOPOD

: INITIALIZE CARRY

: TRY SUBTRACT-CARRY INST

SEC : INITIALIZE CARRY

SBC R0 : TRY SUBTRACT-CARRY INST

BPL SOPOD

COM R0

INC R0

DEC R0

BEQ TS45

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 756 <====

SOPOD: MOV #57, -(R2) : MOVE TO MAILBOX # \*\*\*\*\* 57 \*\*\*\*\*

INC -(R2) : SET MSGTYP TO FATAL ERROR

HALT : CUMMULATIVE RESULT OF ADC, SBC, COM, INC AND DEC INSTS. F

: OR SEQUENCE ERROR

\*\*\*\*\*

: THIS TEST INTRODUCES THE BYTE CONTROL LOGIC OF THE PROCESSOR.  
: THE MODE 0 BYTE MICROCODE IS TESTED. THE METHOD AND SEQUENCE  
: OF TESTING IS THE SAME AS THAT USED IN THE SOP MODE 0 TESTS.

\*\*\*\*\*

```
1479 ;TEST 45 TEST MODE 0 EVEN BYTE USING SOP INST
1480 :*****
1481 003570 005212 TS45: INC (R2) ;UPDATE TEST NUMBER
1482 003572 022712 000045 CMP #45,(R2) ;SEQUENCE ERROR?
1483 003576 001012 BNE TS46-10 ;BR TO ERROR HALT ON SEQ ERROR
1484 003600 105000 CLR R0 ;TRY CLEARING EVEN BYTE OF REGISTER
1485 003602 001404 BEQ SOPBOA
1486 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1487 ; CONDITIONAL BRANCH INST. AND <====
1488 ; REPLACE THE MOVE INSTRUCTION <====
1489 ; WHICH FOLLOWS W/ 775 <====
1490 003604 012742 000060 MOV #60,-(R2) ;MOVE TO MAILBOX # ***** 60 *****
1491 003610 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1492 003612 000000 HALT ;CLRB DID NOT SET Z-BIT
1493 003614 105100 SOPBOA: COMB R0 ;TRY SETTING EVEN BYTE OF REGISTER
1494 003616 100002 BPL SOPBOB
1495 003620 105200 INCB R0 ;TRY INCREMENTING EVEN BYTE OF REGISTER>>
1496 003622 001404 BEQ TS46
1497 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1498 ; CONDITIONAL BRANCH INST. AND <====
1499 ; REPLACE THE MOVE INSTRUCTION <====
1500 ; WHICH FOLLOWS W/ 765 <====
1501 003624 SOPBOB:
1502 003624 012742 000061 MOV #61,-(R2) ;MOVE TO MAILBOX # ***** 61 *****
1503 003630 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1504 003632 000000 HALT ;TEST CUMULATIVE RESULT OF ABOVE BYTE INST.
1505 ; OR SEQUENCE ERROR
1506
1507 :*****
1508 ; THIS TEST USES THE CLR INSTRUCTION TO INTRODUCE AND TEST
1509 ; SINGLE OPERAND MODE 1 INSTRUCTIONS. AGAIN, THE CLR INSTRUCTION
1510 ; IS USED TO INTRODUCE THE MICROCODE AND TO TEST THAT THE PROPER
1511 ; CONDITION CODES ARE SET. OTHER SOP INSTRUCTIONS ARE USED TO MANIPULATE
1512 ; COMMON DATA TO VERIFY THAT THE CORRECT DATA IS PRODUCED.
1513 :*****
1514
1515 :TEST 46 TEST MODE 1 USING SOP INST.
1516 :*****
1517
1518 003634 005212 TS46: INC (R2) ;UPDATE TEST NUMBER
1519 003636 022712 000046 CMP #46,(R2) ;SEQUENCE ERROR?
1520 003642 001014 BNE TS47-10 ;BR TO ERROR HALT ON SEQ ERROR
1521 003644 005000 CLR R0 ;INITIALIZE R0
1522 003646 005010 CLR (R0) ;TRY CLEAR INST W/MODE 1
1523 003650 001404 BEQ SOP1A
1524 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1525 ; CONDITIONAL BRANCH INST. AND <====
1526 ; REPLACE THE MOVE INSTRUCTION <====
1527 ; WHICH FOLLOWS W/ 774 <====
1528 003652 012742 000062 MOV #62,-(R2) ;MOVE TO MAILBOX # ***** 62 *****
1529 003656 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1530 003660 000000 HALT ;CLRB DID NOT SET Z-BIT
1531 003662 005310 SOP1A: DEC (R0) ;TRY DECREMENT INST W/MODE 1
1532 003664 100003 BPL SOP1B
1533 003666 000261 SEC ;INITIALIZE CARRY
1534 003670 005510 ADC (R0) ;TRY ADD-CARRY W/MODE 1
```

1535 003672 001404  
1536  
1537  
1538  
1539  
1540 003674  
1541 003674 012742 000063  
1542 003700 005242  
1543 003702 000000  
1544  
1545  
1546  
1547  
1548  
1549  
1550  
1551  
1552  
1553  
1554  
1555  
1556 003704 005212  
1557 003706 022712 000047  
1558 003712 001020  
1559 003714 005000  
1560 003716 005010  
1561 003720 005110  
1562 003722 105010  
1563 003724 001404  
1564  
1565  
1566  
1567  
1568 003726 012742 000064  
1569 003732 005242  
1570 003734 000000  
1571 003736 005210  
1572 003740 100005  
1573 003742 105110  
1574 003744 105210  
1575 003746 100002  
1576 003750 105210  
1577 003752 001404  
1578  
1579  
1580  
1581  
1582 003754  
1583 003754 012742 000065  
1584 003760 005242  
1585 003762 000000  
1586  
1587  
1588  
1589  
1590

BEQ TS47  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 763 <====  
SOP1B:  
MOV #63,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 63 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INST  
; OR SEQUENCE ERROR  
:\*\*\*\*\*  
: THIS TEST VERIFIES THE BYTE INSTRUCTION MICROCODE FOR MODE 1  
: SINGLE OPERAND INSTRUCTIONS.  
: THIS IS THE FIRST PLACE THE SIGN EXTEND LOGIC IS EXERCISED  
: AND VERIFIED.  
:\*\*\*\*\*  
: TEST 47 TEST MODE 1 EVEN BYTE USING SOP INST  
:\*\*\*\*\*  
TS47: INC (R2) ;UPDATE TEST NUMBER  
CMP #47,(R2) ;SEQUENCE ERROR?  
BNE TS50-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;INITIALIZE R0  
CLR (R0) ;INITIALIZE LOC. 0  
COM (R0)  
CLRB (R0) ;TRY TO CLEAR BYTE 0  
BEQ SOPB1A  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 772 <====  
MOV #64,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 64 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CLRB DID NOT SET Z-BIT  
SOPB1A: INC (R0) ;INCREMENT TO TEST WORD  
BPL SOPB1B  
COMB (R0) ;COMPLEMENT: ODD BYTE = 376  
INCB (R0) ;INC: ODD BYTE = 377  
BPL SOPB1B  
INCB (R0) ;INCREMENT ODD BYTE=0  
BEQ TS50  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 757 <====  
SOPB1B:  
MOV #65,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 65 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CHECK CUMMULATIVE RESULT OF ABOVE INST  
; OR SEQUENCE ERROR  
:\*\*\*\*\*  
:



1591  
1592  
1593  
1594  
1595  
1596  
1597  
1598  
1599  
1600  
1601 003764 005212  
1602 003766 022712 000050  
1603 003772 001022  
1604 003774 005000  
1605 003776 005010  
1606 004000 005110  
1607 004002 005200  
1608 004004 105010  
1609 004006 001404  
1610  
1611  
1612  
1613  
1614 004010 012742 000066  
1615 004014 005242  
1616 004016 000000  
1617 004020 005300  
1618 004022 005210  
1619 004024 005200  
1620 004026 105110  
1621 004030 105210  
1622 004032 100002  
1623 004034 105210  
1624 004036 001404  
1625  
1626  
1627  
1628  
1629 004040  
1630 004040 012742 000067  
1631 004044 005242  
1632 004046 000000  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646

: THIS TEST VERIFIES THAT SINGLE OPERAND BYTE INSTRUCTIONS WILL  
: FUNCTION CORRECTLY FOR ODD BYTES.  
: THIS IS THE FIRST TIME THAT ADDRESS LINE 0 HAS BEEN  
: EXERCISED. CHECKS ARE MADE THAT THE PROPER BYTE IS MODIFIED AND  
: THE CONDITION CODES ARE CHECKED. IT IS ALSO VERIFIED THAT THE UNADDRESSED  
: BYTE IS NOT ALTERED BY THE INSTRUCTION.

\*\*\*\*\*  
: TEST 50 TEST MODE 1 ODD BYTE USING SOP INST  
\*\*\*\*\*

TS50: INC (R2) ;UPDATE TEST NUMBER  
CMP #50,(R2) ;SEQUENCE ERROR?  
BNE TS51-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;INITIALIZE R0  
CLR (R0) ;INITIALIZE LOC. 0  
COM (R0)  
INC R0 ;R0=ODD BYTE  
CLRB (R0) ;TRY TO CLEAR BYTE 1  
BEQ SOPB1C  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 771 <====

MOV #66,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 66 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CLRB DID NOT SET Z-BIT  
SOPB1C: DEC R0 ;R0=WORD ADDR.  
INC (R0) ;INCREMENT TO TEST WORD  
INC R0 ;R0=ODD BYTE  
COMB (R0) ;TRY TO COMPLEMENT BYTE 1  
INCB (R0)  
BPL SOPB1D  
INCB (R0) ;TRY TO INCREMENT BYTE 1  
BEQ TS51

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 755 <====

SOPB1D: MOV #67,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 67 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TEST CUMULATIVE RESULT OF ABOVE INST.  
: OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 2 SINGLE-OPERAND INSTRUCTIONS. PREVIOUSLY  
: TESTED INSTRUCTIONS ARE USED TO SET A POINTER IN R0 TO LOC. 400.  
: LOC. 400 IS INITIALIZED TO -1 BEFORE A CLR MODE 2 IS EXECUTED.  
: THEN R0 IS DECREMENTED BY TWO TO AGAIN POINT TO 400 BEFORE EACH  
: OF SEVERAL MODE 2 INSTRUCTIONS ARE USED TO VERIFY THE DATA RESULTS OF  
: THE TEST. THIS PROCEDURE ALSO VERIFIES THE PROPER INCREMENTING OF THE  
: REGISTER.

\*\*\*\*\*  
: TEST 51 TEST MODE 2 USING SOP INST.  
\*\*\*\*\*

1647  
1648 004050 005212  
1649 004052 022712 000051  
1650 004056 001023  
1651 004060 005000  
1652 004062 105100  
1653 004064 005200  
1654 004066 005010  
1655 004070 005110  
1656 004072 005020  
1657 004074 001404  
1658  
1659  
1660  
1661  
1662 004076 012742 000070  
1663 004102 005242  
1664 004104 000000  
1665 004106 005300  
1666 004110 005300  
1667 004112 005120  
1668 004114 100004  
1669 004116 005300  
1670 004120 005300  
1671 004122 005220  
1672 004124 001404  
1673  
1674  
1675  
1676  
1677 004126  
1678 004126 012742 000071  
1679 004132 005242  
1680 004134 000000  
1681  
1682  
1683  
1684  
1685  
1686  
1687  
1688  
1689  
1690  
1691  
1692  
1693  
1694  
1695  
1696 004136 005212  
1697 004140 022712 000052  
1698 004144 001023  
1699 004146 005000  
1700 004150 105100  
1701 004152 005200  
1702 004154 005010

```
.....  
T551:  INC      (R2)           ;UPDATE TEST NUMBER  
      CMP      #51,(R2)       ;SEQUENCE ERROR?  
      BNE     TS52-10         ;BR TO ERROR HALT ON SEQ ERROR  
      CLR     RO              ;SET RO=400  
      COMB   RO              ;  
      INC     RO              ;  
      CLR     (RO)           ;CLEAR 400  
      COM     (RO)           ;INITIALIZE: 400=-1  
      CLR     (RO)+          ;TRY CLEARING WITH MODE 2  
      BEQ    SOPZA           ;  
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
      ; CONDITIONAL BRANCH INST. AND <====  
      ; REPLACE THE MOVE INSTRUCTION <====  
      ; WHICH FOLLOWS W/ 770 <====  
      MOV     #70,-(R2)       ;MOVE TO MAILBOX # ***** 70 *****  
      INC     -(R2)          ;SET MSGTYP TO FATAL ERROR  
      HALT                    ;CLR INST DID NOT SET Z-BIT  
SOPZA: DEC     RO            ;RESET RO  
      DEC     RO            ;  
      COM     (RO)+          ;TRY COMPLEMENTING WITH MODE 2  
      BPL     SOP2B         ;  
      DEC     RO            ;RESET RO  
      DEC     RO            ;  
      INC     (RO)+          ;TRY INCREMENTING WITH MODE 2  
      BEQ    TS52           ;  
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
      ; CONDITIONAL BRANCH INST. AND <====  
      ; REPLACE THE MOVE INSTRUCTION <====  
      ; WHICH FOLLOWS W/ 754 <====  
SOP2B: MOV     #71,-(R2)       ;MOVE TO MAILBOX # ***** 71 *****  
      INC     -(R2)          ;SET MSGTYP TO FATAL ERROR  
      HALT                    ;CHECK CUMMULATIVE RESULT OF ABOVE INST  
      ; OR SEQUENCE ERROR  
.....
```

```
.....  
: THIS TEST VERIFIES MODE 2 SINGLE OPERAND INSTRUCTIONS WHICH  
: ADDRESS EVEN BYTES. RO IS SET TO 400 AND USED TO INITIALIZE LOCATION  
: 400 TO -1. CLRB INSTRUCTION IS THEN EXECUTED ON BYTE 400 WITH  
: MODE 2.  
: RO IS THEN DECREMENTED BEFORE EACH OF SEVERAL MODE 2 INSTRUCTIONS  
: WHICH ARE USED TO VERIFY THE DATA RESULTS OF THE TEST. THIS PROCEDURE ALSO  
: VERIFIES THE PROPER INCREMENTING OF THE REGISTER.  
.....
```

TEST 52 TEST MODE 2 EVEN BYTE USING SOP INST.

```
.....  
T552:  INC      (R2)           ;UPDATE TEST NUMBER  
      CMP      #52,(R2)       ;SEQUENCE ERROR?  
      BNE     TS53-10         ;BR TO ERROR HALT ON SEQ ERROR  
      CLR     RO              ;SET RO=400  
      COMB   RO              ;  
      INC     RO              ;  
      CLR     (RO)           ;CLEAR 400  
.....
```

```
1703 004156 005110 COM (R0) ;INITIALIZE: 400=-1
1704 004160 105020 CLRB (R0)+ ;TRY TO CLEAT 400 W/MODE 2
1705 004162 001404 BEQ SOPB2A
1706 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1707 ; CONDITIONAL BRANCH INST. AND <====
1708 ; REPLACE THE MOVE INSTRUCTION <====
1709 ; WHICH FOLLOWS W/ 770 <====
1710 004164 012742 000072 MOV #72,-(R2) ;MOVE TO MAILBOX # ***** 72 *****
1711 004170 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1712 004172 000000 HALT ;CLR DID NOT SET Z-BIT
1713 004174 005300 SOPB2A: DEC R0 ;RESULT R0=400
1714 004176 005210 INC (R0) ;INC 400 TO TEST WORD
1715 004200 105110 COMB (R0)
1716 004202 105220 INCB (R0)+ ;TRY TO INC EVEN BYTE
1717 004204 100003 BPL SOPB2B
1718 004206 005300 DEC R0 ;RESET R0=400
1719 004210 105220 INCB (R0)+ ;TRY INCREMENT OF EVEN BYTE
1720 004212 001404 BEQ TS53
1721 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1722 ; CONDITIONAL BRANCH INST. AND <====
1723 ; REPLACE THE MOVE INSTRUCTION <====
1724 ; WHICH FOLLOWS W/ 754 <====
1725 004214 SOPB2B:
1726 004214 012742 000073 MOV #73,-(R2) ;MOVE TO MAILBOX # ***** 73 *****
1727 004220 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1728 004222 000000 HALT ;TEST CUMMULATIVE RESULT OF ABOVE INST.
1729 ; OR SEQUENCE ERROR
1730
1731 ;*****
1732 ; THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS
1733 ; TEST. HERE, THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.
1734 ;*****
1735 ;TEST 53 TEST MODE 2 ODD BYTE USING SOP INST.
1736 ;*****
1737 ;*****
1738 ;*****
1739 004224 005212 TS53: INC (R2) ;UPDATE TEST NUMBER
1740 004226 022712 000053 CMP #53,(R2) ;SEQUENCE ERROR?
1741 004232 001026 BNE TS54-10 ;BR TO ERROR HALT ON SEQ ERROR
1742 004234 005000 CLR R0 ;SET R0=400
1743 004236 105100 COMB R0
1744 004240 005200 INC R0
1745 004242 005010 CLR (R0) ;CLEAR LOC 400
1746 004244 005110 COM (R0) ;INITIALIZE: 400=-1
1747 004246 005200 INC R0 ;R0=ODD BYTE
1748 004250 105020 CLRB (R0)+ ;TRY TO CLEAR ODD BYTE
1749 004252 001404 BEQ SOPB2C
1750 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1751 ; CONDITIONAL BRANCH INST. AND <====
1752 ; REPLACE THE MOVE INSTRUCTION <====
1753 ; WHICH FOLLOWS W/ 767 <====
1754 004254 012742 000074 MOV #74,-(R2) ;MOVE TO MAILBOX # ***** 74 *****
1755 004260 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1756 004262 000000 HALT ;CLRB DID NOT SET Z-BIT
1757 004264 005300 SOPB2C: DEC R0 ;R0=WORD ADDR.
1758 004266 005300 DEC R0
```

|      |        |        |        |         |            |  |                                 |
|------|--------|--------|--------|---------|------------|--|---------------------------------|
| 1759 | 004270 | 005220 |        | INC     | (R0)+      | : INCREMENT WORD                         |                                 |
| 1760 | 004272 | 005300 |        | DEC     | R0         | : POINT TO ODD BYTE                      |                                 |
| 1761 | 004274 | 105110 |        | COMB    | (R0)       | : COMPLEMENT ODD BYTE                    |                                 |
| 1762 | 004276 | 105220 |        | INCB    | (R0)+      | : TRY TO INCREMENT ODD BYTE              |                                 |
| 1763 | 004300 | 100003 |        | BPL     | SOPB2D     |  |                                 |
| 1764 | 004302 | 005300 |        | DEC     | R0         | : RESET R0 TO ODD BYTE                   |                                 |
| 1765 | 004304 | 105220 |        | INCB    | (R0)+      | : TRY TO INCREMENT ODD BYTE              |                                 |
| 1766 | 004306 | 001404 |        | BEQ     | T554       |  |                                 |
| 1767 |        |        |        |         |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <====                           |
| 1768 |        |        |        |         |            | : CONDITIONAL BRANCH INST. AND           | <====                           |
| 1769 |        |        |        |         |            | : REPLACE THE MOVE INSTRUCTION           | <====                           |
| 1770 |        |        |        |         |            | : WHICH FOLLOWS W/ 751                   | <====                           |
| 1771 | 004310 |        |        | SOPB2D: |            |  |                                 |
| 1772 | 004310 | 012742 | 000075 | MOV     | #75, -(R2) | : MOVE TO MAILBOX # ***** 75 *****       |                                 |
| 1773 | 004314 | 005242 |        | INC     | -(R2)      | : SET MSGTYP TO FATAL ERROR              |                                 |
| 1774 | 004316 | 000000 |        | HALT    |            | : TEST CUMMULATIVE RESULT OF ABOVE INST. |                                 |
| 1775 |        |        |        |         |            | : OR SEQUENCE ERROR                      |                                 |
| 1776 |        |        |        |         |            |  |                                 |
| 1777 |        |        |        |         |            |  |                                 |
| 1778 |        |        |        |         |            |  |                                 |
| 1779 |        |        |        |         |            |  |                                 |
| 1780 |        |        |        |         |            |  |                                 |
| 1781 |        |        |        |         |            |  |                                 |
| 1782 |        |        |        |         |            |  |                                 |
| 1783 |        |        |        |         |            |  |                                 |
| 1784 |        |        |        |         |            |  |                                 |
| 1785 | 004320 | 005212 |        | T554:   | INC        | (R2)                                     | : UPDATE TEST NUMBER            |
| 1786 | 004322 | 022712 | 000054 |         | CMP        | #54, (R2)                                | : SEQUENCE ERROR?               |
| 1787 | 004326 | 001035 |        |         | BNE        | T555-10                                  | : BR TO ERROR HALT ON SEQ ERROR |
| 1788 | 004330 | 005000 |        |         | CLR        | R0                                       | : SET R0=0                      |
| 1789 | 004332 | 005200 |        |         | INC        | R0                                       | : R0=1                          |
| 1790 | 004334 | 005400 |        |         | NEG        | R0                                       | : TRY NEGATE MODE 0: R0=-1      |
| 1791 | 004336 | 100003 |        |         | BPL        | NEG00                                    | : CC=1001?                      |
| 1792 | 004340 | 001402 |        |         | BEQ        | NEG00                                    |                                 |
| 1793 | 004342 | 102401 |        |         | BVS        | NEG00                                    |                                 |
| 1794 | 004344 | 103404 |        |         | BCS        | NEG01                                    |                                 |
| 1795 |        |        |        |         |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <====                           |
| 1796 |        |        |        |         |            | : CONDITIONAL BRANCH INST. AND           | <====                           |
| 1797 |        |        |        |         |            | : REPLACE THE MOVE INSTRUCTION           | <====                           |
| 1798 |        |        |        |         |            | : WHICH FOLLOWS W/ 770                   | <====                           |
| 1799 | 004346 |        |        | NEG00:  |            |  |                                 |
| 1800 | 004346 | 012742 | 000076 | MOV     | #76, -(R2) | : MOVE TO MAILBOX # ***** 76 *****       |                                 |
| 1801 | 004352 | 005242 |        | INC     | -(R2)      | : SET MSGTYP TO FATAL ERROR              |                                 |
| 1802 | 004354 | 000000 |        | HALT    |            | : NEGATE DID NOT SET CC'S CORRECTLY      |                                 |
| 1803 |        |        |        |         |            |  |                                 |
| 1804 | 004356 | 005200 |        | NEG01:  | INC        | R0                                       | : TEST DATA RESULT              |
| 1805 | 004360 | 001404 |        |         | BEQ        | NEG02                                    |                                 |
| 1806 |        |        |        |         |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <====                           |
| 1807 |        |        |        |         |            | : CONDITIONAL BRANCH INST. AND           | <====                           |
| 1808 |        |        |        |         |            | : REPLACE THE MOVE INSTRUCTION           | <====                           |
| 1809 |        |        |        |         |            | : WHICH FOLLOWS W/ 762                   | <====                           |
| 1810 | 004362 | 012742 | 000077 | MOV     | #77, -(R2) | : MOVE TO MAILBOX # ***** 77 *****       |                                 |
| 1811 | 004366 | 005242 |        | INC     | -(R2)      | : SET MSGTYP TO FATAL ERROR              |                                 |
| 1812 | 004370 | 000000 |        | HALT    |            | : DATA RESULT OF NEGATE INCORRECT        |                                 |
| 1813 |        |        |        |         |            |  |                                 |
| 1814 | 004372 | 105100 |        | NEG02:  | COMB       | R0                                       | : R0=377                        |

```
1815 004374 105400      NEGB  R0          :R0=1
1816 004376 100403      BMI  NEG03       :CC=0001?
1817 004400 001402      BEQ  NEG03
1818 004402 102401      BVS  NEG03
1819 004404 103404      BCS  NEG04
1820                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1821                                     ;          CONDITIONAL BRANCH INST. AND <====
1822                                     ;          REPLACE THE MOVE INSTRUCTION <====
1823                                     ;          WHICH FOLLOWS W/ 750 <====
1824 004406      NEG03:
1825 004406 012742 000100      MOV  #100,-(R2)  ;MOVE TO MAILBOX # ***** 100 *****
1826 004412 005242      INC  -(R2)       ;SET MSGTYP TO FATAL ERROR
1827 004414 000000      HALT            ;NEGB DID NOT SET CC'S CORRECTLY
1828 004416 005300      NEG04: DEC  R0    ;TEST DATA RESULT
1829 004420 001404      BEQ  T555
1830                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1831                                     ;          CONDITIONAL BRANCH INST. AND <====
1832                                     ;          REPLACE THE MOVE INSTRUCTION <====
1833                                     ;          WHICH FOLLOWS W/ 742 <====
1834 004422 012742 000101      MOV  #101,-(R2)  ;MOVE TO MAILBOX # ***** 101 *****
1835 004426 005242      INC  -(R2)       ;SET MSGTYP TO FATAL ERROR
1836 004430 000000      HALT            ;DATA RESULT OF NEGB INCORRECT
1837                                     ; OR SEQUENCE ERROR
1838 ;*****
1839 ;TEST 55 TEST MODE 1 USING NEGATE INST.
1840 ;*****
1841 004432 005212      T555: INC  (R2)     ;UPDATE TEST NUMBER
1842 004434 022712 000055      CMP  #55,(R2)   ;SEQUENCE ERROR?
1843 004440 001040      BNE  T556-10    ;BR TO ERROR HALT ON SEQ ERROR
1844 004442 005000      CLR  R0         ;POINT TO LOC. 0
1845 004444 005010      CLR  (R0)       ;CLEAR LOC. 0
1846 004446 005210      INC  (R0)       ;LOC. 0=1
1847 004450 005410      NEG  (R0)       ;TRY NEG. LOC. 0=-1
1848 004452 100003      BPL  NEG10      ;CC=1001
1849 004454 001402      BEQ  NEG10
1850 004456 102401      BVS  NEG10
1851 004460 103404      BCS  NEG11
1852                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1853                                     ;          CONDITIONAL BRANCH INST. AND <====
1854                                     ;          REPLACE THE MOVE INSTRUCTION <====
1855                                     ;          WHICH FOLLOWS W/ 767 <====
1856 004462      NEG10:
1857 004462 012742 000102      MOV  #102,-(R2)  ;MOVE TO MAILBOX # ***** 102 *****
1858 004466 005242      INC  -(R2)       ;SET MSGTYP TO FATAL ERROR
1859 004470 000000      HALT            ;NEGATE DID NOT SET CC'S CORRECTLY
1860
1861 004472 005237 000000      NEG11: INC  @#0   ;TEST DATA RESULT
1862 004476 001404      BEQ  NEG12
1863                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1864                                     ;          CONDITIONAL BRANCH INST. AND <====
1865                                     ;          REPLACE THE MOVE INSTRUCTION <====
1866                                     ;          WHICH FOLLOWS W/ 760 <====
1867 004500 012742 000103      MOV  #103,-(R2)  ;MOVE TO MAILBOX # ***** 103 *****
1868 004504 005242      INC  -(R2)       ;SET MSGTYP TO FATAL ERROR
1869 004506 000000      HALT            ;DATA RESULT OF NEGATE INCORRECT
1870 004510 105110      NEG12: COMB  (R0) ;LOC. 0=377
```

```
1871 004512 105410      NEGB    (R0)      ;TRY NEGB LOC. 0=1
1872 004514 100403      BMI     NEG13     ;CC=0001?
1873 004516 001402      BEQ     NEG13
1874 004520 102401      BVS     NEG13
1875 004522 103404      BCS     NEG14
1876                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1877                                     ;          CONDITIONAL BRANCH INST. AND <====
1878                                     ;          REPLACE THE MOVE INSTRUCTION <====
1879                                     ;          WHICH FOLLOWS W/ 746 <====
1880 004524                                     NEG13:
1881 004524 012742 000104      MOV     #104,-(R2) ;MOVE TO MAILBOX # ***** 104 *****
1882 004530 005242                                     INC     -(R2)      ;SET MSGTYP TO FATAL ERROR
1883 004532 000000                                     HALT                                         ;NEGB DID NOT SET CC'S CORRECTLY
1884 004534 005337 000000      NEG14: DEC     @#0      ;TEST DATA RESULT
1885 004540 001404      BEQ     T556
1886                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1887                                     ;          CONDITIONAL BRANCH INST. AND <====
1888                                     ;          REPLACE THE MOVE INSTRUCTION <====
1889                                     ;          WHICH FOLLOWS W/ 737 <====
1890 004542 012742 000105      MOV     #105,-(R2) ;MOVE TO MAILBOX # ***** 105 *****
1891 004546 005242                                     INC     -(R2)      ;SET MSGTYP TO FATAL ERROR
1892 004550 000000                                     HALT                                         ;DATA RESULT OF NEGB INCORRECT
1893                                     ; OR SEQUENCE ERROR
1894 :*****
1895 ;TEST 56 TEST MODE 2 USING NEGATE INSTRUCTION
1896 :*****
1897 004552 005212 000056      T556:  INC     (R2)      ;UPDATE TEST NUMBER
1898 004554 022712                                     CMP     #56,(R2)   ;SEQUENCE ERROR?
1899 004560 001032                                     BNE     T557-10    ;BR TO ERROR HALT ON SEQ ERROR
1900 004562 005000                                     CLR     R0         ;POINT TO LOC. 0
1901 004564 005010                                     CLR     (R0)       ;CLEAR LOC. 0
1902 004566 005210                                     INC     (R0)       ;LOC. 0=1
1903 004570 005420                                     NEG     (R0)+      ;TRY NEG.: LOC. 0=-1
1904 004572 100003      BPL     NEG20     ;CC=1001?
1905 004574 001402      BEQ     NEG20
1906 004576 102401      BVS     NEG20
1907 004600 103404      BCS     NEG21
1908                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1909                                     ;          CONDITIONAL BRANCH INST. AND <====
1910                                     ;          REPLACE THE MOVE INSTRUCTION <====
1911                                     ;          WHICH FOLLOWS W/ 767 <====
1912 004602                                     NEG20:
1913 004602 012742 000106      MOV     #106,-(R2) ;MOVE TO MAILBOX # ***** 106 *****
1914 004606 005242                                     INC     -(R2)      ;SET MSGTYP TO FATAL ERROR
1915 004610 000000                                     HALT                                         ;NEGATE DID NOT SET CC'S CORRECTLY
1916 004612 105300      NEG21: DECB    R0        ;R0=LOC. 0
1917 004614 105300      DECB    R0
1918 004616 105420      NEGB   (R0)+      ;BYTE 0=1 R0=1
1919 004620 105420      NEGB   (R0)+      ;BYTE 1=1 R0=2
1920 004622 105340      DECB   -(R0)     ;R0=1 LOC. 0=01
1921 004624 005300      DEC     R0        ;R0=0
1922 004626 001404      BEQ     NEG22
1923                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1924                                     ;          CONDITIONAL BRANCH INST. AND <====
1925                                     ;          REPLACE THE MOVE INSTRUCTION <====
1926                                     ;          WHICH FOLLOWS W/ 754 <====
```

1927 004630 012742 000107  
1928 004634 005242  
1929 004636 000000  
1930 004640 005337 000000  
1931 004644 001404

MOV #107,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 107 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
NEG22: HALT ;REGISTER NOT INCREMENTED CORRECTLY  
DEC @#0 ;LOC. 0=0  
BEQ TS57

1932  
1933  
1934  
1935  
1936 004646 012742 000110  
1937 004652 005242  
1938 004654 000000  
1939  
1940

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
CONDITIONAL BRANCH INST. AND <====  
REPLACE THE MOVE INSTRUCTION <====  
WHICH FOLLOWS W/ 745 <====  
MOV #110,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 110 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;NEG BYTE INSTRUCTIONS FAILED  
OR SEQUENCE ERROR

1941  
1942  
1943  
1944  
1945  
1946  
1947  
1948  
1949  
1950  
1951  
1952  
1953  
1954  
1955

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 3 SINGLE OPERAND INSTRUCTIONS. IT  
: USES LOCATION 0 AS ITS TARGET DATA. A TABLE LOCATED AT LOC. 400  
: THRU 402 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE  
: INSTRUCTIONS UNDER TEST.  
: R0 IS SET TO 400, THE START OF THE ADDRESS TABLE, AND A CLR  
: INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR LOC. 0. THEN R0  
: IS DECREMENTED BY TWO AND TWO OTHER MODE 3 INSTRUCTIONS OPERATE ON  
: LOC. 0 TO VERIFY THE DATA RESULTS OF THE TEST. THE PROPER INCREMENTING  
: OF THE REGISTER IS ALSO VERIFIED IN THIS MANNER.  
: IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE  
: (LOC. 400-402) HAS THE PROPER VALUES (0).  
\*\*\*\*\*

1956  
1957  
1958 004656 005212  
1959 004660 022712 000057  
1960 004664 001020  
1961 004666 005000  
1962 004670 105100  
1963 004672 005200  
1964 004674 005010  
1965 004676 005030  
1966 004700 001404

TEST 57 TEST MODE 3 USING SOP INST.  
\*\*\*\*\*  
TS57: INC (R2) ;UPDATE TEST NUMBER  
CMP #57,(R2) ;SEQUENCE ERROR?  
BNE TS60-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;SET R0=400  
COMB R0  
INC R0  
CLR (R0) ;CLEAR LOC 400  
CLR @(R0)+ ;TRY TO CLEAR LOC 0 USING MODE 3 ;R0=402  
BEQ SOP3A

1967  
1968  
1969  
1970

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
CONDITIONAL BRANCH INST. AND <====  
REPLACE THE MOVE INSTRUCTION <====  
WHICH FOLLOWS W/ 771 <====

1971 004702 012742 000111  
1972 004706 005242  
1973 004710 000000  
1974 004712 005300  
1975 004714 005300  
1976 004716 005130  
1977 004720 100002  
1978 004722 005230  
1979 004724 001404

MOV #111,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 111 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
SOP3A: HALT ;CLR DID NOT SET Z-BIT  
DEC R0 ;RESET R0=400  
DEC R0  
COM @(R0)+ ;TRY TO COMPLEMENT LOC 0 OF MODE 3 ;R0=402  
BPL SOP3B  
INC @(R0)+ ;TRY TO INCREMENT LOC 0 W/MODE 3 ;R0=404  
BEQ TS60

1980  
1981  
1982

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
CONDITIONAL BRANCH INST. AND <====  
REPLACE THE MOVE INSTRUCTION <====

1983  
1984 004726  
1985 004726 012742 000112  
1986 004732 005242  
1987 004734 000000  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005 004736 005212  
2006 004740 022712 000060  
2007 004744 001026  
2008 004746 005004  
2009 004750 105104  
2010 004752 005204  
2011 004754 005000  
2012 004756 005010  
2013 004760 005110  
2014 004762 105034  
2015 004764 001404  
2016  
2017  
2018  
2019  
2020 004766 012742 000113  
2021 004772 005242  
2022 004774 000000  
2023 004776 005304  
2024 005000 005304  
2025 005002 005234  
2026 005004 100006  
2027 005006 105434  
2028 005010 100004  
2029 005012 005304  
2030 005014 005304  
2031 005016 105234  
2032 005020 001404  
2033  
2034  
2035  
2036  
2037 005022  
2038 005022 012742 000114

SOP3B: ; WHICH FOLLOWS W/ 757 <====  
MOV #112,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 112 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CUMMULATIVE RESULT OF ABOVE INST FAILED  
; OR SEQUENCE ERROR

.....  
: THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS  
: WHICH ADDRESS EVEN BYTES. AGAIN, THE TARGET LOCATION 0 IS USED  
: AND THE SAME TABLE AT 400 IS EMPLOYED.  
: AFTER POINTING R4 TO THE TABLE (400) AND SETTING LOCATION  
: 0 TO -1, A CLRB INSTRUCTION IS USED TO CLEAR BYTE 0.  
: SEVERAL OTHER MODE 3 INSTRUCTIONS ARE THEN USED WITH THE TABLE  
: TO VERIFY THE DATA RESULTS AND THE PROPER INCREMENTING OF THE REGISTER.  
: IF A FAILURE IS DETECTED, BE SURE THAT THE TABLE (LOCATION 400-402) HAS  
: THE PROPER VALUES (0).  
:.....

:TEST 60 TEST MODE 3 EVEN BYTE USING SOP INST.  
:.....

TS60: INC (R2) ;UPDATE TEST NUMBER  
CMP #60,(R2) ;SEQUENCE ERROR?  
BNE TS61-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R4 ;SET R4=400  
COMB R4  
INC R4  
CLR R0 ;INITIALIZE LOC. 0=-1  
CLR (R0)  
COM (R0) ;LOC. 0=-1  
CLRB @(R4)+ ;TRY TO CLEAR EVEN BYTE ;LOC. 0=177400 R4=402  
BEQ SOPB3A  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 767 <====

SOPB3A: MOV #113,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 113 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CLRB DID NOT SET Z-BIT  
DEC R4 ;RESET POINTER R4=400  
DEC R4  
INC @(R4)+ ;TRY INCREMENTING WORD LOC.0=177401 R4=402  
BPL SOPB3B  
NEGB @(R4)+ ;TRY TO NEGATE EVEN BYTE ;LOC.0=-1 R4=404  
BPL SOPB3B  
DEC R4 ;R4=402  
DEC R4  
INCB @(R4)+ ;TRY TO INCREMENT EVEN BYTE ;LOC. 0=17400  
BEQ TS61  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 751 <====

SOPB3B: MOV #114,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 114 \*\*\*\*\*



2039 005026 005242  
2040 005030 000000  
2041  
2042  
2043  
2044  
2045  
2046  
2047  
2048  
2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059 005032 005212  
2060 005034 022712 000061  
2061 005040 001024  
2062 005042 005000  
2063 005044 105100  
2064 005046 005200  
2065 005050 005030  
2066 005052 005130  
2067 005054 105030  
2068 005056 001404  
2069  
2070  
2071  
2072  
2073 005060 012742 000115  
2074 005064 005242  
2075 005066 000000  
2076 005070 005300  
2077 005072 005300  
2078 005074 005300  
2079 005076 005300  
2080 005100 005230  
2081 005102 105430  
2082 005104 100002  
2083 005106 105230  
2084 005110 001404  
2085  
2086  
2087  
2088  
2089 005112  
2090 005112 012742 000116  
2091 005116 005242  
2092 005120 000000  
2093  
2094

INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CUMULATIVE RESULT OF ABOVE INST FAILED  
; OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS  
: WHICH ADDRESS ODD BYTES. THE TARGET IS BYTE 1. A TABLE AT  
: LOC. 400-406 IS USED. R0 SERVES AS THE TABLE POINTER.  
: R0 IS INITIALIZED TO 400. LOC. 0 IS SET TO -1 USING THE  
: FIRST TWO TABLE ENTRIES. A CLRB MODE 3 IS EXECUTED ON BYTE 1 USING  
: TABLE ADDRESS AT 404. R0 IS DECREMENTED TO 402 AND SEVERAL SOP  
: MODE 3 INSTRUCTIONS ARE USED TO VERIFY DATA RESULTS AND PROPER  
: REGISTER INCREMENTING.  
: THE TABLE (400-406) SHOULD CONTAIN 0,0,1,1 BEFORE AND  
: AFTER THE TEST IS RUN.  
\*\*\*\*\*

\*\*\*\*\*  
: TEST 61 TEST MODE 3 ODD BYTE USING SOP INST.  
\*\*\*\*\*

TS61: INC (R2) ;UPDATE TEST NUMBER  
CMP #61,(R2) ;SEQUENCE ERROR?  
BNE TS62-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;SET R0=400  
COMB R0  
INC R0  
CLR @(R0)+ ;INITIALIZE  
COM @(R0)+ ;LOC 0=-1 R0=404  
CLRB @(R0)+ ;TRY TO CLEAR ODD BYTE LOC. 0=377 R0=406  
BEQ SOPB3C

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 770 <====

MOV #115,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 115 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
SOPB3C: HALT ;CLR DID NOT SET Z-BIT  
DEC R0 ;RESET R0=402  
DEC R0  
DEC R0 ;POINT TO EVEN BYTE ADDR.  
DEC R0  
INC @(R0)+ ;INCREMENT WORD LOC. 0=400 R0=404  
NEGB @(R0)+ ;TRY TO NEGATE ODD BYTE LOC. 0=177400 R0=406  
BPL SOPB3D  
INCB @(R0)+ ;TRY TO INCREMENT ODD BYTE LOC.0=0 R0=410  
BEQ TS62

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 753 <====

SOPB3D: MOV #116,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 116 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CUMULATIVE RESULT OF ABOVE INSTS FAILED  
; OR SEQUENCE ERROR

\*\*\*\*\*

```
2095 ;TEST 62 TEST MODE 3 USING NEGATE INSTRUCTION
2096 :*****
2097 005122 005212 TS62: INC (R2) ;UPDATE TEST NUMBER
2098 005124 022712 000062 CMP #62,(R2) ;SEQUENCE ERROR?
2099 005130 001054 BNE TS63-10 ;BR TO ERROR HALT ON SEQ ERROR
2100 005132 005000 CLR R0 ;R0=400
2101 005134 105100 COMB R0
2102 005136 005270 INC R0
2103 005140 005000 CLR (R0) ;LOC. 400=0
2104 005142 005004 CLR R4 ;R4=0
2105 005144 005014 CLR (R4) ;LOC. 0=0
2106 005146 005214 INC (R4) ;LOC. 0=1
2107 005150 005430 NEG @ (R0)+ ;TRY NEGATE LOC. 0=-1 R0=402
2108 005152 100003 BPL NEG30 ;CC=1001?
2109 005154 001402 BEQ NEG30
2110 005156 102401 BVS NEG30
2111 005160 103404 BCS NEG31
2112 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2113 ; CONDITIONAL BRANCH INST. AND <====
2114 ; REPLACE THE MOVE INSTRUCTION <====
2115 ; WHICH FOLLOWS W/ 763 <====
2116 005162 NEG30:
2117 005162 012742 000117 MOV #117,-(R2) ;MOVE TO MAILBOX # ***** 117 *****
2118 005166 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2119 005170 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
2120 005172 005214 NEG31: INC (R4) ;LOC. 0=0
2121 005174 001404 BEQ NEG32
2122 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2123 ; CONDITIONAL BRANCH INST. AND <====
2124 ; REPLACE THE MOVE INSTRUCTION <====
2125 ; WHICH FOLLOWS W/ 755 <====
2126 005176 012742 000120 MOV #120,-(R2) ;MOVE TO MAILBOX # ***** 120 *****
2127 005202 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2128 005204 000000 HALT ;DATA RESULT OF NEG INCORRECT
2129 005206 105137 000001 NEG32: COMB @#1 ;LOC 0=177400
2130 005212 005237 000000 INC @#0 ;LOC. 0=177401
2131 005216 105430 NEGB @ (R0)+ ;TRY NEGB LOC. 0=177777 R0=404
2132 005220 100404 BMI NEG33
2133 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2134 ; CONDITIONAL BRANCH INST. AND <====
2135 ; REPLACE THE MOVE INSTRUCTION <====
2136 ; WHICH FOLLOWS W/ 743 <====
2137 005222 012742 000121 MOV #121,-(R2) ;MOVE TO MAILBOX # ***** 121 *****
2138 005226 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2139 005230 000000 HALT ;NEGB FAILED WITH EVEN BYTE
2140 005232 105430 NEG33: NEGB @ (R0)+ ;TRY NEGB LOC.0=777 R0=406
2141 005234 100004 BPL NEG34
2142 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2143 ; CONDITIONAL BRANCH INST. AND <====
2144 ; REPLACE THE MOVE INSTRUCTION <====
2145 ; WHICH FOLLOWS W/ 735 <====
2146 005236 012742 000122 MOV #122,-(R2) ;MOVE TO MAILBOX # ***** 122 *****
2147 005242 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2148 005244 000000 HALT ;NEGB FAILED WITH ODD BYTE
2149 005246 105137 000001 NEG34: COMB @#1 ;LOC. 0=177377
2150 005252 105237 000001 INCB @#1 ;LOC. 0=177777
```

```
2151 005256 005214      INC      (R4)      ;LOC. 0=0
2152 005260 001404      BEQ      TS63
2153                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
2154                      ;          CONDITIONAL BRANCH INST. AND <===
2155                      ;          REPLACE THE MOVE INSTRUCTION <===
2156                      ;          WHICH FOLLOWS W/ 723 <===
2157 005262 012742 000123  MOV      #123,-(R2) ;MOVE TO MAILBOX # ***** 123 *****
2158 005266 005242      INC      -(R2)     ;SET MSGTYP TO FATAL ERROR
2159 005270 000000      HALT           ;DATA RESULT OF NEGB'S INCORRECT
2160                      ; OR SEQUENCE ERROR
```

```
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
```

```
*****
: THIS TEST VERIFIES MODE 4 SINGLE OPERAND INSTRUCTIONS.
: RO IS SET TO 400. A CLR INSTRUCTION IS EXECUTED IN MODE 4 TO CLEAR
: LOC. 376. RO IS RESET TO 400 AND A COM INSTRUCTION USING MODE 4
: COMPLEMENTS LOC.376.
: TWO INC INSTRUCTIONS AND A MODE 4 INSTRUCTION ARE EXECUTED
: TO COMPLETE THE TEST.
*****
```

TEST 63 TEST MODE 4 USING SOP INSTS

```
TS63: 2173 005272 005212      INC      (R2)      ;UPDATE TEST NUMBER
2174 005274 022712 000063  CMP      #63,(R2)  ;SEQUENCE ERROR?
2175 005300 001021      BNE      TS64-10   ;BR TO ERROR HALT ON SEQ ERROR
2176 005302 005000      CLR      RO        ;SET RO=400
2177 005304 105100      COMB     RO
2178 005306 005200      INC      RO
2179 005310 005040      CLR      -(RO)    ;TRY TO CLEAR USING MODE 4
2180 005312 001404      BEQ      SOP4A
```

```
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
```

```
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
;          CONDITIONAL BRANCH INST. AND <===
;          REPLACE THE MOVE INSTRUCTION <===
;          WHICH FOLLOWS W/ 772 <===
```

```
2185 005314 012742 000124  MOV      #124,-(R2) ;MOVE TO MAILBOX # ***** 124 *****
2186 005320 005242      INC      -(R2)     ;SET MSGTYP TO FATAL ERROR
2187 005322 000000      HALT           ;CLR DID NOT SET Z-BIT
```

```
SOP4A: 2188 005324 005200      INC      RO        ;RESET RO
2189 005326 005200      INC      RO
2190 005330 005140      COM      -(RO)    ;TRY TO COMPLEMENT USING MODE 4
```

```
2191 005332 100004      BPL      SOP4B
2192 005334 005200      INC      RO        ;MOVE POINTER
2193 005336 005200      INC      RO
2194 005340 005240      INC      -(RO)
2195 005342 001404      BEQ      TS64
```

```
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
```

```
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
;          CONDITIONAL BRANCH INST. AND <===
;          REPLACE THE MOVE INSTRUCTION <===
;          WHICH FOLLOWS W/ 756 <===
```

```
SOP4B: 2201 005344 012742 000125  MOV      #125,-(R2) ;MOVE TO MAILBOX # ***** 125 *****
2202 005350 005242      INC      -(R2)     ;SET MSGTYP TO FATAL ERROR
2203 005352 000000      HALT           ;CHECK CUMMULATIVE RESULT OF ABOVE INST.
2204                      ; OR SEQUENCE ERROR
```

```
*****
```

2207  
2208  
2209  
2210  
2211  
2212  
2213  
2214  
2215  
2216  
2217  
2218  
2219  
2220  
2221  
2222  
2223  
2224  
2225  
2226  
2227  
2228  
2229  
2230  
2231  
2232  
2233  
2234  
2235  
2236  
2237  
2238  
2239  
2240  
2241  
2242  
2243  
2244  
2245  
2246  
2247  
2248  
2249  
2250  
2251  
2252  
2253  
2254  
2255  
2256  
2257  
2258  
2259  
2260  
2261  
2262

THIS TEST VERIFIES MODE 5 SINGLE OPERAND INSTRUCTIONS. IT  
USES LOCATION 0 AS ITS TARGET DATA. A TABLE LOCATED AT LOC. 372  
THRU 374 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE  
INSTRUCTIONS UNDER TEST.  
R0 IS SET TO 376, (THE START OF THE ADDRESS TABLE) +2,  
AND A CLR INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR  
LOC. 0. THEN R0 IS INCREMENTED BY TWO AND TWO OTHER MODE 3  
INSTRUCTIONS OPERATE ON LOC. 0 TO VERIFY THE DATA RESULTS OF  
THE TEST. THE PROPER DECREMENTING OF THE REGISTER IS ALSO  
VERIFIED IN THIS MANNER.  
IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE  
(LOC. 372 THRU 374) HAS THE PROPER VALUES (0).

\*\*\*\*\*  
TEST 64 TEST MODE 5 USING SOP INSTS  
\*\*\*\*\*

TS64: INC (R2) ;UPDATE TEST NUMBER  
CMP #64,(R2) ;SEQUENCE ERROR?  
BNE TS65-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #370,R0 ;CLEAR LOCATION 370-376  
CLR (R0)+ ;370  
CLR (R0)+ ;372  
CLR (R0)+ ;374  
CLR (R0) ;376  
CLR R0 ;SET R0=376 (LOW BYTE)  
CLR (R0)+  
NEGB R0  
CLR @-(R0) ;TRY TO CLEAR LOC 0 W/MODE 5  
BEQ SOP5A  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 764 <====  
MOV #126,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 126 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CLR DID NOT SET Z-BIT  
SOP5A: INC R0 ;RESET R0  
INC R0  
COM @-(R0) ;TRY TO COMPLEMENT LOC. 0 W/MODE 5  
BPL SOP5B  
INC @-(R0) ;TRY TO INCREMENT LOC. 0 W/MODE 5  
BEQ TS65  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 752 <====  
SOP5B: MOV #127,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 127 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INSTS  
; OR SEQUENCE ERROR

\*\*\*\*\*  
THIS TEST VERIFIES MODE 6 SINGLE OPERAND INSTRUCTIONS. IT

2263  
2264  
2265  
2266  
2267  
2268  
2269  
2270  
2271 005446 005212  
2272 005450 022712 000065  
2273 005454 001020  
2274 005456 005000  
2275 005460 105100  
2276 005462 005200  
2277 005464 005060 177400  
2278 005470 001404  
2279  
2280  
2281  
2282  
2283 005472 012742 000130  
2284 005476 005242  
2285 005500 000000  
2286 005502 005160 177400  
2287 005506 100003  
2288 005510 005260 177400  
2289 005514 001404  
2290  
2291  
2292  
2293  
2294 005516  
2295 005516 012742 000131  
2296 005522 005242  
2297 005524 000000  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312 005526 005212  
2313 005530 022712 000066  
2314 005534 001021  
2315 005536 005000  
2316 005540 105100  
2317 005542 005200  
2318 005544 005210

:USES LOCATION 0 AS ITS TARGET DATA. R0 IS SET TO 400 USING  
:PREVIOUSLY TESTED INSTRUCTIONS AND A MODE 6 CLR INSTRUCTION IS  
:EXECUTED ON LOC. 0 USING R0 AND A -400 OFFSET. COM AND INC  
:INSTRUCTIONS ARE THEN USED TO VERIFY THE DATA.  
:.....  
:TEST 65 TEST MODE 6 USING SOP INSTS  
:.....  
TS65: INC (R2) :UPDATE TEST NUMBER  
CMP #65,(R2) :SEQUENCE ERROR?  
BNE TS66-10 :BR TO ERROR HALT ON SEQ ERROR  
CLR R0 :SET R0=400  
COMB R0  
INC R0  
CLR -400(R0) :TRY TO CLEAR LOCATION 0 W/MODE 6  
BEQ SOP6A  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 771 <====  
MOV #130,-(R2) :MOVE TO MAILBOX # \*\*\*\*\* 130 \*\*\*\*\*  
INC -(R2) :SET MSGTYP TO FATAL ERROR  
HALT :CLR DID NOT SET Z-BIT  
SOP6A: COM -400(R0) :TRY TO COMPLEMENT LOCATION 0 W/MODE 6  
BPL SOP6B  
INC -400(R0) :TRY TO INCREMENT LOCATION 0 W/MODE 6  
BEQ TS66  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 757 <====  
SOP6B: MOV #131,-(R2) :MOVE TO MAILBOX # \*\*\*\*\* 131 \*\*\*\*\*  
INC -(R2) :SET MSGTYP TO FATAL ERROR  
HALT :TEST CUMMULATIVE RESULT OF ABOVE INSTS  
: OR SEQUENCE ERROR  
:.....  
: THIS TEST VERIFIES MODE 7 SINGLE OPERAND INSTRUCTIONS. IT USES  
: THE POINTER TO LOC. 0 WHICH IS STORED AT LOC. 402.  
: R0 IS SET TO 400 AND A MODE 7 CLR INSTRUCTION IS  
: EXECUTED WITH A +2 OFFSET TO CLEAR LOC. 0.  
: SEVERAL OTHER MODE 7 INSTRUCTIONS ARE THEN USED ON THE COMMON  
: LOCATION TO VERIFY THE DATA RESULTS.  
:.....  
:TEST 66 TEST MODE 7 USING SOP INST.  
:.....  
TS66: INC (R2) :UPDATE TEST NUMBER  
CMP #66,(R2) :SEQUENCE ERROR?  
BNE TS67-10 :BR TO ERROR HALT ON SEQ ERROR  
CLR R0 :SET R0=400  
COMB R0  
INC R0  
INC (R0) :R0=1

```
2319 005546 005070 000002      CLR    @2(R0)      ;TRY TO CLEAR LOC. 0 W/MODE 7
2320 005552 001404              BEQ    SOP7A
2321                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2322                          ;          CONDITIONAL BRANCH INST. AND <====
2323                          ;          REPLACE THE MOVE INSTRUCTION <====
2324                          ;          WHICH FOLLOWS W/ 770 <====
2325 005554 012742 000132      MOV    #132,-(R2)  ;MOVE TO MAILBOX # ***** 132 *****
2326 005560 005242              INC    -(R2)      ;SET MSGTYP TO FATAL ERROR
2327 005562 000000              HALT                   ;CLR DID NOT SET Z-BIT
2328 005564 005170 000002      SOP7A: COM    @2(R0)  ;TRY TO COMPLEMENT LOC. 0 W/MODE 7
2329 005570 100003              BPL    SOP7B
2330 005572 005270 000002      INC    @2(R0)      ;TRY TO INCREMENT LOC. 0 W/MODE 7
2331 005576 001404              BEQ    TS67
2332                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2333                          ;          CONDITIONAL BRANCH INST. AND <====
2334                          ;          REPLACE THE MOVE INSTRUCTION <====
2335                          ;          WHICH FOLLOWS W/ 756 <====
2336 005600                      SOP7B:
2337 005600 012742 000133      MOV    #133,-(R2)  ;MOVE TO MAILBOX # ***** 133 *****
2338 005604 005242              INC    -(R2)      ;SET MSGTYP TO FATAL ERROR
2339 005606 000000              HALT                   ;TEST CUMMULATIVE RESULT OF ABOVE INSTS.
2340                          ; OR SEQUENCE ERROR
2341
2342 ;*****
2343 ;TEST 67          TEST MODE 4 WITH NEGATE INSTRUCTION
2344 ;*****
2345 005610 005212 000067      TS67: INC    (R2)      ;UPDATE TEST NUMBER
2346 005612 022712              CMP    #67,(R2)    ;SEQUENCE ERROR?
2347 005616 001024              BNE    TS70-10     ;BR TO ERROR HALT ON SEQ ERROR
2348 005620 005000              CLR    R0
2349 005622 005010              CLR    (R0)
2350 005624 005120              COM    (R0)+      ;LOC. 0=177777, R0=2
2351 005626 005440              NEG    -(R0)      ;TRY NEGATE, LOC. 0=1
2352 005630 100403              BMI    NEG40      ;CC=0001?
2353 005632 001402              BEQ    NEG40
2354 005634 102401              BVS    NEG40
2355 005636 103404              BCS    NEG41
2356                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2357                          ;          CONDITIONAL BRANCH INST. AND <====
2358                          ;          REPLACE THE MOVE INSTRUCTION <====
2359                          ;          WHICH FOLLOWS W/ 767 <====
2360 005640                      NEG40:
2361 005640 012742 000134      MOV    #134,-(R2)  ;MOVE TO MAILBOX # ***** 134 *****
2362 005644 005242              INC    -(R2)      ;SET MSGTYP TO FATAL ERROR
2363 005646 000000              HALT                   ;NEG DID NOT SET CC'S CORRECTLY
2364 005650 005400              NEG41: NEG    R0      ;TST R0 WITH A NEG.
2365 005652 001404              BEQ    NEG42
2366                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2367                          ;          CONDITIONAL BRANCH INST. AND <====
2368                          ;          REPLACE THE MOVE INSTRUCTION <====
2369                          ;          WHICH FOLLOWS W/ 761 <====
2370 005654 012742 000135      MOV    #135,-(R2)  ;MOVE TO MAILBOX # ***** 135 *****
2371 005660 005242              INC    -(R2)      ;SET MSGTYP TO FATAL ERROR
2372 005662 000000              HALT                   ;R0 NOT DECREMENTED PROPERLY
2373 005664 005310              NEG42: DEC    (R0)    ;TEST DTA RESULT OF NEG
2374 005666 001404              BEQ    TS70
```

```
2375 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2376 ; CONDITIONAL BRANCH INST. AND <====
2377 ; REPLACE THE MOVE INSTRUCTION <====
2378 ; WHICH FOLLOWS W/ 753 <====
2379 005670 012742 000136 MOV #136,-(R2) ;MOVE TO MAILBOX # ***** 136 *****
2380 005674 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2381 005676 000000 HALT ;DATA RESULT OF NEG INCORRECT
2382 ; OR SEQUENCE ERROR
2383 ;*****
2384 ;TEST 70 TEST MODE 5 WITH NEGATE INSTRUCTION
2385 ;*****
2386 005700 005212 TS70: INC (R2) ;UPDATE TEST NUMBER
2387 005702 022712 000070 CMP #70,(R2) ;SEQUENCE ERROR?
2388 005706 001031 BNE TS71-10 ;BR TO ERROR HALT ON SEQ ERROR
2389 005710 005000 CLR R0 ;R0=0
2390 005712 005010 CLR (R0) ;LOC. 0=0
2391 005714 105100 COMB R0 ;R0=377
2392 005716 005200 INC R0 ;R0=400
2393 005720 005010 CLR (R0) ;SET 400 = 0
2394 005722 005004 CLR R4 ;R4=0
2395 005724 005314 DEC (R4) ;LOC. 0=177777
2396 005726 005450 NEG @-(R0) ;TRY NEGATE: LOC. 0=1
2397 005730 100403 BMI NEG50 ;CC=0001?
2398 005732 001402 BEQ NEG50
2399 005734 102401 BVS NEG50
2400 005736 103404 BCS NEG51
2401 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2402 ; CONDITIONAL BRANCH INST. AND <====
2403 ; REPLACE THE MOVE INSTRUCTION <====
2404 ; WHICH FOLLOWS W/ 763 <====
2405 005740 NEG50:
2406 005740 012742 000137 MOV #137,-(R2) ;MOVE TO MAILBOX # ***** 137 *****
2407 005744 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2408 005746 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
2409 005750 005314 NEG51: DEC (R4)
2410 005752 001404 BEQ NEG52
2411 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2412 ; CONDITIONAL BRANCH INST. AND <====
2413 ; REPLACE THE MOVE INSTRUCTION <====
2414 ; WHICH FOLLOWS W/ 755 <====
2415 005754 012742 000140 MOV #140,-(R2) ;MOVE TO MAILBOX # ***** 140 *****
2416 005760 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2417 005762 000000 HALT ;DATA RESULT OF NEG INCORRECT
2418 005764 105100 NEG52: COMB R0
2419 005766 005300 DEC R0
2420 005770 001404 BEQ TS71
2421 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2422 ; CONDITIONAL BRANCH INST. AND <====
2423 ; REPLACE THE MOVE INSTRUCTION <====
2424 ; WHICH FOLLOWS W/ 746 <====
2425 005772 012742 000141 MOV #141,-(R2) ;MOVE TO MAILBOX # ***** 141 *****
2426 005776 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2427 006000 000000 HALT ;REGISTER NOT DECREMENTED PROPERLY
2428 ; OR SEQUENCE ERROR
2429 ;*****
2430 ;TEST 71 TEST MODE 6 WITH NEGATE
```

```
2431 ;*****
2432 006002 005212 TS71: INC (R2) ;UPDATE TEST NUMBER
2433 006004 022712 000071 CMP #71,(R2) ;SEQUENCE ERROR?
2434 006010 001022 BNE TS72-10 ;BR TO ERROR HALT ON SEQ ERROR
2435 006012 005000 CLR R0 ;R0=0
2436 006014 005004 CLR R4 ;R4=0
2437 006016 105100 COMB R0 ;R0=377
2438 006020 005014 CLR (R4) ;LOC. 0=0
2439 006022 105024 CLRB (R4)+ ;LOC. 0=177777, R4=1
2440 006024 105114 COMB (R4) ;LOC. 0=177400
2441 006026 005460 177401 NEG -377(R0) ;LOC. 0=400
2442 006032 100403 BMI NEG60 ;CC=0001
2443 006034 001402 BEQ NEG60
2444 006036 102401 BVS NEG60
2445 006040 103404 BCS NEG61
2446 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2447 ; CONDITIONAL BRANCH INST. AND <====
2448 ; REPLACE THE MOVE INSTRUCTION <====
2449 ; WHICH FOLLOWS W/ 763 <====
2450 006042 NEG60:
2451 006042 012742 000142 MOV #142,-(R2) ;MOVE TO MAILBOX # ***** 142 *****
2452 006046 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2453 006050 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
2454 006052 105314 NEG61: DECB (R4)
2455 006054 001404 BEQ TS72
2456 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2457 ; CONDITIONAL BRANCH INST. AND <====
2458 ; REPLACE THE MOVE INSTRUCTION <====
2459 ; WHICH FOLLOWS W/ 755 <====
2460 006056 012742 000143 MOV #143,-(R2) ;MOVE TO MAILBOX # ***** 143 *****
2461 006062 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2462 006064 000000 HALT ;DATA RESULT OF NEG INCORRECT
2463 ; OR SEQUENCE ERROR
2464 ;*****
2465 ;TEST 72 TEST MODE 7 W/ NEGATE
2466 ;*****
2467 006066 005212 TS72: INC (R2) ;UPDATE TEST NUMBER
2468 006070 022712 000072 CMP #72,(R2) ;SEQUENCE ERROR?
2469 006074 001024 BNE TS73-10 ;BR TO ERROR HALT ON SEQ ERROR
2470 006076 005000 CLR R0 ;R0=0
2471 006100 005010 CLR (R0) ;LOC. 0=0
2472 006102 005110 COM (R0) ;LOC. 0=177777
2473 006104 105100 COMB R0 ;R0=377
2474 006106 105470 000005 NEGB @5(R0) ;R0+5=404, 404=1, LOC. 0=777
2475 006112 100403 BMI NEG70 ;CC=0001?
2476 006114 001402 BEQ NEG70
2477 006116 102401 BVS NEG70
2478 006120 103404 BCS NEG71
2479 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2480 ; CONDITIONAL BRANCH INST. AND <====
2481 ; REPLACE THE MOVE INSTRUCTION <====
2482 ; WHICH FOLLOWS W/ 765 <====
2483 006122 NEG70:
2484 006122 012742 000144 MOV #144,-(R2) ;MOVE TO MAILBOX # ***** 144 *****
2485 006126 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2486 006130 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
```



2487 006132 105100  
2488 006134 105120  
2489 006136 105310  
2490 006140 005467 171634  
2491 006144 001404  
2492  
2493  
2494  
2495  
2496 006146 012742 000145  
2497 006152 005242  
2498 006154 000000  
2499  
2500  
2501  
2502  
2503  
2504  
2505  
2506  
2507  
2508  
2509  
2510  
2511  
2512 006156 005212  
2513 006160 022712 000073  
2514 006164 001017  
2515 006166 005027  
2516 006170 177777  
2517 006172 001404  
2518  
2519  
2520  
2521  
2522 006174 012742 000146  
2523 006200 005242  
2524 006202 000000  
2525 006204 005237 006170  
2526 006210 005467 177754  
2527 006214 100003  
2528 006216 005277 000012  
2529 006222 001405  
2530  
2531  
2532  
2533  
2534 006224  
2535 006224 012742 000147  
2536 006230 005242  
2537 006232 000000  
2538  
2539 006234 006170  
2540  
2541  
2542

NEG71: COMB R0 ;R0=0  
COMB (R0)+ ;LOC. 0=400, R0=1  
DECB (R0) ;LOC. 0=0  
NEG 0 ;USE NEG MODE 67 TO TST FOR ZERO  
BEQ TS73  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 753 <====  
MOV #145,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 145 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DATA RESULT OF NEG WAS INCORRECT  
; OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES PROGRAM COUNTER ADDRESSING WITH SOP  
: INSTRUCTIONS. CLR MODE 77 IS USED TO CLEAR THE LOCATION FOLLOWING THE  
: INSTRUCTION (SOPX). THEN SINGLE OPERAND INSTRUCTIONS WITH MODES 37, 67, AND  
: 77, USING INDIRECT POINTER SOPXAD ARE USED TO VERIFY THE DATA RESULTS  
: OF THESE INSTRUCTIONS.  
\*\*\*\*\*

TEST 73 TEST SOP INSTRUCTIONS MODES 2,3,6,7 WITH REGISTER 7  
\*\*\*\*\*

TS73: INC (R2) ;UPDATE TEST NUMBER  
CMP #73,(R2) ;SEQUENCE ERROR?  
BNE SOPB ;BR TO ERROR HALT ON SEQ ERROR  
CLR (R7)+ ;CLEAR NEXT LOCATION: (SOPX)  
SOPX: -1 ;USE MODE 27  
BEQ SOPA

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 774 <====  
MOV #146,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 146 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CLR DID NOT SET Z-BIT  
SOPA: INC @#SOPX ;INC SOPX W/MODE 37  
NEG SOPX ;NEGATE SOPX W/MODE 67  
BPL SOPB  
INC @SOPXAD ;INC SOPX W/MODE 77  
BEQ TS74

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 760 <====

SOPB: MOV #147,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 147 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;INC DID NOT SET Z-BIT  
; OR SEQUENCE ERROR  
SOPXAD: SOPX ;INDIRECT ADDRESS OF SOPX

\*\*\*\*\*  
:

2543  
2544  
2545  
2546  
2547  
2548  
2549  
2550  
2551  
2552 006236 005212  
2553 006240 022712 000074  
2554 006244 001010  
2555 006246 005000  
2556 006250 000277  
2557 006252 000244  
2558 006254 005700  
2559 006256 102403  
2560 006260 100402  
2561 006262 103401  
2562 006264 001404  
2563  
2564  
2565  
2566  
2567 006266  
2568 006266 012742 000150  
2569 006272 005242  
2570 006274 000000  
2571  
2572  
2573  
2574  
2575  
2576  
2577  
2578  
2579  
2580  
2581  
2582  
2583  
2584 006276 005212  
2585 006300 022712 000075  
2586 006304 001010  
2587 006306 005000  
2588 006310 105100  
2589 006312 000277  
2590 006314 000250  
2591 006316 105700  
2592 006320 102402  
2593 006322 101401  
2594 006324 100404  
2595  
2596  
2597  
2598

: THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING INSTRUCTIONS  
: USING MODE 0. R0 IS SET TO ZERO AND THE CONDITION CODES ARE SET  
: TO THE COMPLEMENT OF THAT EXPECTED BY THE INSTRUCTION. A TST INSTRUCTION  
: IS EXECUTED AND CONDITIONAL BRANCHES ARE USED TO TEST THE CONDITION  
: CODES.

\*\*\*\*\*  
: TEST 74 TEST MODE 0 SOP NON-MODIFYING  
\*\*\*\*\*

TS74: INC (R2) ;UPDATE TEST NUMBER  
CMP #74,(R2) ;SEQUENCE ERROR?  
BNE TS75-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;INITIALIZE R0=0  
SCC ;SET CC=1011  
CLZ  
TST R0 ;TRY TST W/ MODE 0  
BVS SNMOA ;CHECK THAT CC=0100  
BMI SNMOA  
BCS SNMOA  
BEQ TS75

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 767 <====

SNMOA: MOV #150,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 150 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CONDITION CODES NOT SET PROPERLY  
; OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING BYTE INSTRUCTIONS WITH MODE 0.  
: R0 IS SET TO 377 AND COMPLEMENT OF THE EXPECTED CONDITION CODES  
: IS LOADED IN PSW. A TSTB INSTRUCTION IS EXECUTED AND THE RESULTS  
: ARE CHECKED WITH SEVERAL CONDITIONAL BRANCH INSTRUCTIONS.  
: THIS VERIFIES THAT THE PROPER BYTE WAS TESTED.

\*\*\*\*\*  
: TEST 75 TEST MODE 0 EVEN BYTE W/ SOP NON-MODIFYING  
\*\*\*\*\*

TS75: INC (R2) ;UPDATE TEST NUMBER  
CMP #75,(R2) ;SEQUENCE ERROR?  
BNE TS76-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;INITIALIZE  
COMB R0 ;R0=377  
SCC ;SET CC=0111  
CLN  
TSTB R0 ;TRY TST EVEN BYTE  
BVS SNMBOA ;CHECK CC=1000  
BLOS SNMBOA  
BMI TS76

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 767 <====

2599 006326  
2600 006326 012742 000151  
2601 006332 005242  
2602 006334 000000

SNMBOA:  
MOV #151,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 151 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CONDITION CODES NOT SET PROPERLY  
; OR SEQUENCE ERROR

2603  
2604  
2605  
2606  
2607  
2608  
2609  
2610  
2611  
2612  
2613

.....  
: THIS TEST VERIFIES SINGLE OPERAND INSTRUCTIONS WITH MODE 1.  
: RO IS USED TO POINT TO AND CLEAR LOC. 0. THE COMPLEMENT OF THE  
: EXPECTED CONDITION CODES ARE LOADED IN THE PSW. A TST INSTRUCTION  
: IS THEN EXECUTED ON LOC. 0 USING RO AND CONDITIONAL BRANCHES TEST  
: THE RESULTS.  
: .....

2614  
2615

: TEST 76 TEST MODE 1 SOP NON-MODIFYING  
: .....

2616 006336 005212  
2617 006340 022712 000076  
2618 006344 001011  
2619 006346 005000  
2620 006350 005010  
2621 006352 000277  
2622 006354 000244  
2623 006356 005710  
2624 006360 102403  
2625 006362 103402  
2626 006364 100401  
2627 006366 001404

TS76: INC (R2) ;UPDATE TEST NUMBER  
CMP #76,(R2) ;SEQUENCE ERROR?  
BNE TS77-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR RO ;POINT TO LOC 0  
CLR (RO) ;CLEAR LOC 0  
SCC ;INITIALIZE  
CLZ ;CC=1011  
TST (RO) ;TRY TST W/ MODE 1  
BVS SNM1A ;CHECK CC=0100  
BCS SNM1A  
BMI SNM1A  
BEQ TS77

2628  
2629  
2630  
2631

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 766 <====

2632 006370  
2633 006370 012742 000152  
2634 006374 005242  
2635 006376 000000

SNM1A:  
MOV #152,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 152 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S NOT SET PROPERLY  
; OR SEQUENCE ERROR

2636  
2637  
2638  
2639  
2640  
2641  
2642  
2643  
2644  
2645

.....  
: THIS TEST SETS LOCATION 0 TO 377 AND THEN USES RO TO TEST  
: THE EVEN BYTE AND THE ODD BYTE USING SOP BYTE INSTRUCTIONS WITH MODE 1.  
: AGAIN, CONDITIONAL BRANCHES ARE USED TO VERIFY THE SETTING OF THE  
: PROPER CONDITION CODE BITS.  
: .....

2646  
2647

: TEST 77 TEST MODE 1 BYTE INST. NON-MODIFYING  
: .....

2648 006400 005212  
2649 006402 022712 000077  
2650 006406 001026  
2651 006410 005000  
2652 006412 005010  
2653 006414 105110  
2654 006416 000277

TS77: INC (R2) ;UPDATE TEST NUMBER  
CMP #77,(R2) ;SEQUENCE ERROR?  
BNE TS100-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR RO ;POINT TO LOC 0  
CLR (RO) ;CLEAR LOC 0  
COMB (RO) ;COMPLEMENT BYTE 0  
SCC ;SET CC=0111

```
2655 006420 000250      CLN
2656 006422 105710      TSTB      (R0)      ;TRY TST ON EVEN BYTE
2657 006424 102402      BVS      SNMB1A
2658 006426 101401      BLOS     SNMB1A
2659 006430 100404      BMI      SNMB1B
2660
2661
2662
2663
2664 006432
2665 006432 012742 000153      SNMB1A: MOV      #153,-(R2)      ;MOVE TO MAILBOX # ***** 153 *****
2666 006436 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
2667 006440 000000      HALT
2668 006442 005000      SNMB1B: CLR      R0      ;CC'S NOT CORRECT
2669 006444 005200      INC      R0
2670 006446 000277      SCC
2671 006450 000244      CLZ
2672 006452 105710      TSTB      (R0)      ;TRY TO TST AN ODD BYTE
2673 006454 102403      BVS      SNMB1C      ;CHECK CC=0100
2674 006456 103402      BCS      SNMB1C
2675 006460 100401      BMI      SNMB1C
2676 006462 001404      BEQ      TS100
2677
2678
2679
2680
2681 006464
2682 006464 012742 000154      SNMB1C: MOV      #154,-(R2)      ;MOVE TO MAILBOX # ***** 154 *****
2683 006470 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
2684 006472 000000      HALT
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697 006474 005212 000100      TS100: INC      (R2)      ;UPDATE TEST NUMBER
2698 006476 022712      CMP      #100,(R2)      ;SEQUENCE ERROR?
2699 006502 001020      BNE      TS101-10      ;BR TO ERROR HALT ON SEQ ERROR
2700 006504 005000      CLR      R0      ;INITIALIZE R0=0
2701 006506 005010      CLR      (R0)      ;CLEAR LOC 0
2702 006510 000277      SCC
2703 006512 000244      CLZ
2704 006514 005720      TST      (R0)+      ;TRY TST W/ MODE 2
2705 006516 102403      BVS      SNM2A      ;CHECK CC=0100
2706 006520 103402      BCS      SNM2A
2707 006522 100401      BMI      SNM2A
2708 006524 001404      BEQ      SNM2B
2709
2710
```

```
*****
:
: THIS TEST VERIFIES THE SINGLE-OPERAND NON-MODIFYING INSTRUCTIONS
: USING MODE 2. IT USES THE IDENTICAL PROCEDURE EMPLOYED IN THE
: MODE 1 TESTS. ADDITIONALLY, THE REGISTER IS CHECKED TO ASSURE THAT
: IT IS INCREMENTED PROPERLY.
:
: *****
```

```
*****
: TEST 100 TEST MODE 2 WITH SOP NON-MODIFYING
: *****
```

```
2711                                     :           REPLACE THE MOVE INSTRUCTION <====
2712                                     :           WHICH FOLLOWS W/ 766           <====
2713 006526                               SNM2A:                                     :
2714 006526 012742 000155                MOV    #155,-(R2)           ;MOVE TO MAILBOX # ***** 155 *****
2715 006532 005242                        INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
2716 006534 000000                        HALT                                     ;CC'S NOT CORRECT
2717 006536 005300                        SNM2B: DEC    R0         ;RESET R0
2718 006540 005300                        DEC    R0
2719 006542 001404                        BEQ    TS101
2720                                     :
2721                                     :           TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2722                                     :           CONDITIONAL BRANCH INST. AND <====
2723                                     :           REPLACE THE MOVE INSTRUCTION <====
2724 006544 012742 000156                MOV    #156,-(R2)           ;MOVE TO MAILBOX # ***** 156 *****
2725 006550 005242                        INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
2726 006552 000000                        HALT                                     ;MODE 2 DID NOT INC REQ CORRECTLY
2727                                     :           OR SEQUENCE ERROR
2728
2729
2730
2731                                     :*****
2732                                     :           THIS TEST VERIFIES MODE 2 SINGLE OPERAND NON-MODIFYING BYTE
2733 :INSTRUCTIONS IT USES R0 TO POINT TO LOC. 0. WITH LOCATION 0
2734 :SET TO 377, THE EVEN AND ODD BYTE IS TESTED WITH TSTB INSTRUCTIONS
2735 :TO VERIFY THE CORRECT CC ARE SET. THE REGISTER IS CHECKED FOR
2736 :PROPER INCREMENTING.
2737                                     :*****
2738 :TEST 101 TEST MODE 2 - BYTE W/ SOP NON-MODIFYING
2739 :*****
2740 006554 005212                               TS101: INC    (R2)           ;UPDATE TEST NUMBER
2741 006556 022712 000101                CMP    #101,(R2)          ;SEQUENCE ERROR?
2742 006562 001042                        BNE    TS102-10          ;BR TO ERROR HALT ON SEQ ERROR
2743 006564 005000                        CLR    R0                ;CLEAR R0
2744 006566 005010                        CLR    (R0)              ;CLEAR LOC 0
2745 006570 105110                        COMB   (R0)              ;SET LOC 0=377
2746 006572 000277                        SCC                                     ;SEI CC=0111
2747 006574 000250                        CLN
2748 006576 105720                        TSTB   (R0)+             ;TRY 1ST OF EVEN BYTE
2749 006600 102402                        BVS    SNMB2A
2750 006602 101401                        BLOS   SNMB2A
2751 006604 100404                        BMI    SNMB2B
2752                                     :
2753                                     :           TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2754                                     :           CONDITIONAL BRANCH INST. AND <====
2755                                     :           REPLACE THE MOVE INSTRUCTION <====
2756                                     :           WHICH FOLLOWS W/ 766           <====
2757 006606                               SNMB2A:                                     :
2758 006606 012742 000157                MOV    #157,-(R2)           ;MOVE TO MAILBOX # ***** 157 *****
2759 006612 005242                        INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
2760 006614 000000                        HALT                                     ;CC'S NOT SET CORRECTLY
2761 006616 005300                        SNMB2B: DEC    R0         ;DECREMENT R0
2762 006620 001404                        BEQ    SNMB2C
2763                                     :
2764                                     :           TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2765                                     :           CONDITIONAL BRANCH INST. AND <====
2766 006622 012742 000160                MOV    #160,-(R2)           ;MOVE TO MAILBOX # ***** 160 *****
```

```
2767 006626 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2768 006630 000000          HALT                    ;MODE 2 DID NOT INC REG CORRECTLY
2769 006632 005200          SNMB2C: INC      R0          ;POINT TO ODD BYTE
2770 006634 000277          SCC                    ;SET CC=1011
2771 006636 000244          CLZ                    ;
2772 006640 105720          TSTB     (R0)+          ;TRY TST OF ODD BYTE
2773 006642 102403          BVS     SNMB2D          ;CHECK CC'S=0100
2774 006644 103402          BCS     SNMB2D
2775 006646 100401          BMI     SNMB2D
2776 006650 001404          BEQ     SNMB2E
2777
2778
2779
2780
2781 006652 012742 000161          SNMB2D: MOV     #161,-(R2)      ;MOVE TO MAILBOX # ***** 161 *****
2782 006652 012742 000161          INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
2783 006656 005242          HALT                    ;CC'S NOT CORRECT
2784 006660 000000
2785 006662 005300          SNMB2E: DEC     R0
2786 006664 005300          DEC     R0
2787 006666 001404          BEQ     TS102
2788
2789
2790
2791
2792 006670 012742 000162          MOV     #162,-(R2)      ;MOVE TO MAILBOX # ***** 162 *****
2793 006674 005242          INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
2794 006676 000000          HALT                    ;R0 DID NOT INCREMENT PROPERLY
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807 006700 005212 000102          TS102: INC     (R2)          ;UPDATE TEST NUMBER
2808 006702 022712 000102          CMP     #102,(R2)      ;SEQUENCE ERROR?
2809 006706 001022 000102          BNE     TS103-10        ;BR TO ERROR HALT ON SEQ ERROR
2810 006710 005000          CLR     R0              ;R0=0
2811 006712 005010          CLR     (R0)            ;CLEAR LOC 0
2812 006714 105100          COMB    R0              ;R0=376
2813 006716 005300          DEC     R0
2814 006720 000277          SCC                    ;SET CC=1011
2815 006722 000244          CLZ                    ;
2816 006724 005730          TST     @ (R0)+         ;TRY TST W/ MODE 3
2817 006726 102403          BVS     SNM3A          ;CHECK CC=0100
2818 006730 103402          BCS     SNM3A
2819 006732 100401          BMI     SNM3A
2820 006734 001404          BEQ     SNM3B
2821
2822
```

```
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 744 <====
```

```
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 735 <====
```

```
.....
: THIS TEST VERIFIES MODE 3 SINGLE OPERAND NON-MODIFYING INSTRUCTIONS.
: A POINTER IN A TABLE AT LOC. 376 IS USED TO TEST LOCATION 0.
: THE CC'S AND THE REGISTER ARE CHECKED FOLLOWING THE
: TST MODE 3 INSTRUCTION.
: .....
```

```
.....
: TEST 102 TEST MODE 3 W/ SOP NON-MODIFYING INSTS
: .....
```

```
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
```

```
2823                                     :           REPLACE THE MOVE INSTRUCTION <====
2824                                     :           WHICH FOLLOWS W/ 764           <====
2825 006736                               SNM3A:
2826 006736 012742 000163                MOV    #163,-(R2)      ;MOVE TO MAILBOX # ***** 163 *****
2827 006742 005242                       INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
2828 006744 000000                       HALT                               ;CC'S NOT CORRECT
2829 006746 005300                       SNM3B: DEC    R0      ;R0=377
2830 006750 105100                       COMB   R0            ;R0=0
2831 006752 001404                       BEQ    TS103
2832                                     :           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2833                                     :           ;           CONDITIONAL BRANCH INST. AND <====
2834                                     :           ;           REPLACE THE MOVE INSTRUCTION <====
2835                                     :           ;           WHICH FOLLOWS W/ 755           <====
2836 006754 012742 000164                MOV    #164,-(R2)      ;MOVE TO MAILBOX # ***** 164 *****
2837 006760 005242                       INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
2838 006762 000000                       HALT                               ;MODE 3 DID NOT INC REG CORRECTLY
2839                                     :           ; OR SEQUENCE ERROR
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
```

\*\*\*\*\*  
: THIS TEST VERIFIES SOP NON-MODIFYING BYTE INSTRUCTIONS MODE 3  
: LOC. 0 IS SET TO 377. TABLE AT LOC. 402-404 IS USED TO TEST  
: BYTE 0 AND BYTE 1. THE REGISTER IS CHECKED FOR PROPER INCREMENTING AND  
: THE CC'S ARE VERIFIED.  
: THE TABLE AT LOC. 402-404 SHOULD CONTAIN 0 AND 1 BEFORE AND  
: AFTER THE TEST IS RUN.  
\*\*\*\*\*

TEST 103 TEST MODE 3 - BYTES W/ SOP NON-MODIFYING INSTS.  
\*\*\*\*\*

```
TS103: INC    (R2)          ;UPDATE TEST NUMBER
        CMP    #103,(R2) ;SEQUENCE ERROR?
        BNE   TS104-10  ;BR TO ERROR HALT ON SEQ ERROR
        CLR   R0        ;R0=0
        CLR   (R0)      ;CLEAR LOC 0
        COMB  (R0)      ;LOC. 0 =377
        COMB  R0
        INC   R0
        TST  (R0)+      ;R0=402
        SCC   ;CC=0111
        CLN
        TSTB @ (R0)+    ;TRY TST OF EVEN BYTE
        BVS  SNMB3A     ;CHECK CC=1000
        BLOS SNMB3A
        BMI  SNMB3B
2868                                     :           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2869                                     :           ;           CONDITIONAL BRANCH INST. AND <====
2870                                     :           ;           REPLACE THE MOVE INSTRUCTION <====
2871                                     :           ;           WHICH FOLLOWS W/ 763           <====
2872 007024                               SNMB3A:
2873 007024 012742 000165                MOV    #165,-(R2)      ;MOVE TO MAILBOX # ***** 165 *****
2874 007030 005242                       INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
2875 007032 000000                       HALT                               ;CC'S NOT CORRECT
2876 007034 000277                       SNMB3B: SCC        ;SET CC=1011
2877 007036 000244                       CLZ
2878 007040 105730                       TSTB  @ (R0)+      ;TRY TST OF ODD BYTE
```

```
2879 007042 102403          BVS      SNMB3C          ;CHECK CC=0100
2880 007044 103402          BCS      SNMB3C
2881 007046 100401          BMI      SNMB3C
2882 007050 001404          BEQ      SNMB3D
2883                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2884                                     ;          CONDITIONAL BRANCH INST. AND <====
2885                                     ;          REPLACE THE MOVE INSTRUCTION <====
2886                                     ;          WHICH FOLLOWS W/ 750 <====
2887 007052          SNMB3C:
2888 007052 012742 000166      MOV      #166,-(R2)      ;MOVE TO MAILBOX # ***** 166 *****
2889 007056 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2890 007060 000000          HALT
2891 007062 005720          SNMB3D: TST      (R0)+      ;CC'S NOT CORRECT
2892 007064 005710          TST      (R0)          ;RO=410
2893 007066 100404          BMI      TS104
2894                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2895                                     ;          CONDITIONAL BRANCH INST. AND <====
2896                                     ;          REPLACE THE MOVE INSTRUCTION <====
2897                                     ;          WHICH FOLLOWS W/ 741 <====
2898 007070 012742 000167      MOV      #167,-(R2)      ;MOVE TO MAILBOX # ***** 167 *****
2899 007074 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2900 007076 000000          HALT          ;TSTB DID NOT INCREMENT RO CORRECTLY
2901                                     ; OR SEQUENCE ERROR
2902
2903 .....
2904
2905          THIS TEST VERIFIES MODE 4 SOP NON-MODIFYING INSTRUCTIONS.
2906          :LOC. 0 IS SET TO -1 AND THE CC'S ARE SET TO THE COMPLEMENT OF THE
2907          :EXPECTED RESULTS. RO AND SET TO 2 AND A TST MODE 4 IS EXECUTED.
2908          :THE CC'S ARE CHECKED WITH CONDITIONAL BRANCH INSTRUCTIONS AND THE REGISTER
2909          :IS CHECKED FOR PROPER DECREMENTING.
2910
2911 .....
2912          TEST 104          TEST MODE 4 W/ SOP NON-MODIFYING INSTS
2913 .....
2914 007100 005212 000104      TS104:  INC      (R2)          ;UPDATE TEST NUMBER
2915 007102 022712          CMP      #104,(R2)      ;SEQUENCE ERROR?
2916 007106 001017          BNE      TS105-10      ;BR TO ERROR HALT ON SEQ ERROR
2917 007110 005000          CLR      RO            ;RO=0
2918 007112 005010          CLR      (R0)          ;LOC 0=0
2919 007114 005120          COM      (R0)+        ;LOC 0=-1
2920 007116 000277          SCC
2921 007120 000244          CLZ
2922 007122 005740          TST      -(R0)          ;TRY TST W/ MODE 4
2923 007124 102402          BVS      SNM4A          ;CHECK CC=0100
2924 007126 101401          BLOS     SNM4A
2925 007130 100404          BMI      SNM4B
2926                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2927                                     ;          CONDITIONAL BRANCH INST. AND <====
2928                                     ;          REPLACE THE MOVE INSTRUCTION <====
2929                                     ;          WHICH FOLLOWS W/ 766 <====
2930 007132 012742 000170      SNM4A:  MOV      #170,-(R2)      ;MOVE TO MAILBOX # ***** 170 *****
2931 007136 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2932 007140 000000          HALT          ;CC'S NOT CORRECT
2933 007142 005700          SNM4B:  TST      RO
2934 007144 001404          BEQ      TS105
```



```
2935 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2936 ; CONDITIONAL BRANCH INST. AND <====  
2937 ; REPLACE THE MOVE INSTRUCTION <====  
2938 ; WHICH FOLLOWS W/ 760 <====  
2939 007146 012742 000171 MOV #171,-(R2) ;MOVE TO MAILBOX # ***** 171 *****  
2940 007152 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
2941 007154 000000 HALT ;TST MODE 4 DID NOT DEC R0 CORRECTLY  
2942 ; OR SEQUENCE ERROR  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953
```

```
.....  
: THIS TEST VERIFIES MODE 5 SOP NON-MODIFYING INSTRUCTIONS.  
: IT USES A POINTER AT LOC. 376 TO TEST LOC. 0. R0 IS SET  
: TO 400, A TST MODE 5 INSTRUCTION IS EXECUTED AND THE CC'S CHECKED.  
: R0 IS CHECKED TO INSURE PROPER DECREMENTING.  
.....
```

```
2952 :TEST 105 TEST MODE 5 W/ SOP NON-MODIFYING INSTS  
2953 :.....  
2954 007156 005212 TS105: INC (R2) ;UPDATE TEST NUMBER  
2955 007160 022712 000105 CMP #105,(R2) ;SEQUENCE ERROR?  
2956 007164 001022 BNE TS106-10 ;BR TO ERROR HALT ON SEQ ERROR  
2957 007166 005000 CLR R0 ;R0=0  
2958 007170 005010 CLR (R0) ;LOC 0=0  
2959 007172 005110 COM (R0) ;LOC 0=-1  
2960 007174 105100 COMB R0 ;R0=377  
2961 007176 005200 INC R0 ;R0=400  
2962 007200 000277 SCC ;SET CC=0111  
2963 007202 000250 CLN  
2964 007204 005750 TST @-(R0) ;TRY TST W/ MODE 5  
2965 007206 102402 BVS SNM5A ;CHECK CC=1000  
2966 007210 101401 BLOS SNM5A  
2967 007212 100404 BMI SNM5B  
2968  
2969  
2970  
2971
```

```
2971 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2972 ; CONDITIONAL BRANCH INST. AND <====  
2973 ; REPLACE THE MOVE INSTRUCTION <====  
2974 ; WHICH FOLLOWS W/ 764 <====
```

```
2972 007214 SNM5A: MOV #172,-(R2) ;MOVE TO MAILBOX # ***** 172 *****  
2973 007214 012742 000172 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
2974 007220 005242 HALT ;CC'S NOT SET PROPERLY  
2975 007222 000000 SNM5B: INC R0 ;R0=377  
2976 007224 005200 COMB R0 ;R0=0  
2977 007226 105100 BEQ TS106  
2978 007230 001404  
2979
```

```
2979 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2980 ; CONDITIONAL BRANCH INST. AND <====  
2981 ; REPLACE THE MOVE INSTRUCTION <====  
2982 ; WHICH FOLLOWS W/ 755 <====
```

```
2983 007232 012742 000173 MOV #173,-(R2) ;MOVE TO MAILBOX # ***** 173 *****  
2984 007236 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
2985 007240 000000 HALT ;MODE 5 DID NOT DEC R0 CORRECTLY  
2986 ; OR SEQUENCE ERROR  
2987  
2988  
2989  
2990
```

```
.....  
: THIS TEST VERIFIES MODE 6 SOP NON-MODIFYING INSTRUCTIONS.  
.....
```

2991  
2992  
2993  
2994  
2995  
2996  
2997  
2998 007242 005212  
2999 007244 022712 000106  
3000 007250 001021  
3001 007252 005000  
3002 007254 005010  
3003 007256 005110  
3004 007260 105100  
3005 007262 000277  
3006 007264 000250  
3007 007266 005760 177401  
3008 007272 102402  
3009 007274 101401  
3010 007276 100404  
3011  
3012  
3013  
3014  
3015 007300  
3016 007300 012742 000174  
3017 007304 005242  
3018 007306 000000  
3019 007310 105100  
3020 007312 001404  
3021  
3022  
3023  
3024  
3025 007314 012742 000175  
3026 007320 005242  
3027 007322 000000  
3028

:RO IS SET TO 377 AND A MODE 6 TST INSTRUCTION IS EXECUTED  
:USING RO AND AN OFFSET OF -377. THE CC'S ARE CHECKED AS WELL  
:AS RO TO INSURE IT WAS NOT ALTERED.

\*\*\*\*\*  
:TEST 106 TEST MODE 6 W/ SOP NON-MODIFYING INSTS  
\*\*\*\*\*

TS106: INC (R2) ;UPDATE TEST NUMBER  
CMP #106,(R2) ;SEQUENCE ERROR?  
BNE TS107-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR RO ;RO=0  
CLR (RO) ;LOC 0=0  
COM (RO) ;LOC 0=-1  
COMB RO ;RO=377  
SCC ;SET CC=0111  
CLN  
TST -377(RO) ;TRY TST W/ MODE 6  
BVS SNM6A ;CHECK CC=1000  
BLOS SNM6A  
BMI SNM6B

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 764 <====

SNM6A: MOV #174,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 174 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S INCORRECT  
SNM6B: COMB RO ;RO=0  
BEQ TS107

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 756 <====

MOV #175,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 175 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TST MODE 6 INCORRECTLY CHANGED RO  
: OR SEQUENCE ERROR

3029  
3030  
3031  
3032  
3033  
3034  
3035  
3036  
3037  
3038  
3039  
3040  
3041  
3042  
3043  
3044  
3045  
3046  
3047  
3048  
3049  
3050  
3051  
3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070

007324 005212  
007326 022712 000107  
007332 001021  
007334 005000  
007336 005010  
007340 005110  
007342 105100  
007344 000277  
007346 000250  
007350 005770 000001  
007354 102402  
007356 101401  
007360 100404  
  
007362  
012742 000176  
007366 005242  
007370 000000  
007372 105100  
007374 001404  
  
012742 000177  
007402 005242  
007404 000000

```
*****
:
:   THIS TEST VERIFIES MODE 7 SOP NON-MODIFYING INSTRUCTIONS.
: IT USES A POINTER TO LOC. 0 STORED AT LOC. 400 TO TST LOC. 0.
: RO IS SET TO 377 AND LOC. 0 IS TESTED THRU THE POINTER AT 400 USING
: RO AND AN OFFSET OF 1.
:
:*****
: TEST 107      TEST MODE 7 W/ SOP NON-MODIFYING INSTS.
:*****
TS107:  INC      (R2)           ;UPDATE TEST NUMBER
        CMP      #107,(R2)    ;SEQUENCE ERROR?
        BNE     TS110-10     ;BR TO ERROR HALT ON SEQ ERROR
        CLR     RO           ;RO=0
        CLR     (RO)         ;LOC 0=0
        COM     (RO)         ;LOC 0=-1
        COMB    RO           ;RO=377
        SCC     ;            ;CC=0111
        CLN
        TST     @1(R0)       ;TRY TST W/ MODE 7
        BVS     SNM7A        ;CHECK CC=1000
        BLOS    SNM7A
        BMI     SNM7B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 764 <====

SNM7A:  MOV      #176,-(R2)   ;MOVE TO MAILBOX # ***** 176 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT
SNM7B:  COMB    RO           ;CC'S NOT CORRECT
        BEQ     TS110       ;RO=0

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 756 <====

        MOV     #177,-(R2)   ;MOVE TO MAILBOX # ***** 177 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                ;TST MODE 7 INCORRECTLY CHANGED RO
; OR SEQUENCE ERROR
```

3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082  
3083  
3084  
3085  
3086  
3087  
3088  
3089  
3090  
3091  
3092  
3093  
3094  
3095  
3096  
3097  
3098  
3099  
3100  
3101  
3102  
3103  
3104  
3105  
3106  
3107  
3108  
3109  
3110  
3111  
3112  
3113  
3114  
3115  
3116  
3117  
3118  
3119  
3120  
3121  
3122  
3123  
3124  
3125  
3126

007406 005212  
007410 022712 000110  
007414 001006  
007416 005000  
007420 005100  
007422 005004  
007424 060004  
007426 005204  
007430 001404  
  
007432 012742 000200  
007436 005242  
007440 000000  
  
007442 005212  
007444 022712 000111  
007450 001006  
007452 005000  
007454 005004  
007456 005100  
007460 010004  
007462 005204  
007464 001404  
  
007466 012742 000201  
007472 005242  
007474 000000

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 0 DOUBLE OPERAND INSTRUCTIONS. IT SETS  
: DATA IN R0 AND R4 AND USES THE ADD INSTRUCTION TO TEST THE DOP  
: MICROCODE.  
\*\*\*\*\*

: TEST 110 TEST MODE 0 DOUBLE-OPERAND (DOP) INSTS.  
\*\*\*\*\*

TS110: INC (R2) ;UPDATE TEST NUMBER  
CMP #110,(R2) ;SEQUENCE ERROR?  
BNE TS111-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
COM R0 ;R0=-1  
CLR R4 ;R4=0  
ADD R0,R4 ;TRY ADD: R4=-1  
INC R4 ;R4=0  
BEQ TS111  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 771 <====  
MOV #200,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 200 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ADD INST. FAILED W/ MODE 0  
: OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES THE MOVE INSTRUCTION WITH MODE 0 TO MODE 0.  
\*\*\*\*\*

: TEST 111 MOV MODE 0 TO MODE 0  
\*\*\*\*\*

TS111: INC (R2) ;UPDATE TEST NUMBER  
CMP #111,(R2) ;SEQUENCE ERROR?  
BNE TS112-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR R4 ;R4=0  
COM R0 ;R0=-1  
MOV R0,R4 ;TRY MOVE -1 TO R4  
INC R4 ;INC R4  
BEQ TS112  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 771 <====  
MOV #201,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 201 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;MOVE FAILED MODE 0 TO MODE 0  
: OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES THE SUBTRACT INSTRUCTION WITH MODE 0,0.  
\*\*\*\*\*

```
3127  
3128  
3129  
3130  
3131 007476 005212  
3132 007500 022712 000112  
3133 007504 001016  
3134 007506 005000  
3135 007510 005004  
3136 007512 005204  
3137 007514 160400  
3138 007516 100003  
3139 007520 001402  
3140 007522 102401  
3141 007524 103404  
3142  
3143  
3144  
3145  
3146 007526  
3147 007526 012742 000202  
3148 007532 005242  
3149 007534 000000  
3150 007536 005200  
3151 007540 001404  
3152  
3153  
3154  
3155  
3156 007542 012742 000203  
3157 007546 005242  
3158 007550 000000  
3159
```

```
;  
:*****  
:TEST 112 TEST SUB MODE 0,0  
:*****  
TS112: INC (R2) ;UPDATE TEST NUMBER  
CMP #112,(R2) ;SEQUENCE ERROR?  
BNE TS113-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR R4 ;R4=0  
INC R4 ;R4=1  
SUB R4,R0 ;TRY SUB 0,0 R0=-1  
BPL SUB0 ;CC=1001  
BEQ SUB0  
BVS SUB0  
BCS SUB0A  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 767 <====  
  
SUB0: MOV #202,-(R2) ;MOVE TO MAILBOX # ***** 202 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CONDITION CODE FAILED ON SUB  
  
SUB0A: INC R0  
BEQ TS113  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 761 <====  
  
MOV #203,-(R2) ;MOVE TO MAILBOX # ***** 203 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DATA RESULT OF SUB FAILED  
; OR SEQUENCE ERROR
```

3160  
3161  
3162  
3163  
3164  
3165  
3166  
3167  
3168  
3169  
3170  
3171  
3172  
3173  
3174  
3175  
3176  
3177  
3178  
3179  
3180  
3181  
3182  
3183  
3184  
3185  
3186  
3187  
3188  
3189  
3190  
3191  
3192  
3193  
3194  
3195  
3196  
3197  
3198  
3199  
3200  
3201  
3202  
3203  
3204  
3205  
3206  
3207  
3208  
3209  
3210  
3211  
3212  
3213  
3214  
3215

007552 005212  
007554 022712 000113  
007560 001051  
007562 005000  
007564 010004  
007566 001404  
  
007570 012742 000204  
007574 005242  
007576 000000  
007600 005200  
007602 005100  
007604 005104  
007606 040004  
007610 005304  
007612 001404  
  
007614 012742 000205  
007620 005242  
007622 000000  
007624 050004  
007626 005204  
007630 005204  
007632 001404  
  
007634 012742 000206  
007640 005242  
007642 000000  
007644 005000  
007646 105100  
007650 005004  
007652 005104  
007654 040004  
007656 060004  
007660 005204

```
*****
:
:   THIS TEST QUICKLY VERIFIES THE REMAINING DOP MODIFYING INSTRUCTIONS
: WITH MODE 0,0 TO PROVIDE A BASELINE FOR SUBSEQUENT TESTS.
: SINGLE OPERAND INSTRUCTIONS ARE USED TO SET UP DATA IN R0 AND R4
: BEFORE EACH OF THE SEVERAL DOP MODIFYING INSTRUCTIONS ARE USED AND
: VERIFIED.
:
:*****
: TEST 113      TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0,0
:*****
TS113:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #113,(R2)   ;SEQUENCE ERROR?
        BNE     TS114-10    ;BR TO ERROR HALT ON SEQ ERROR
        CLR     R0          ;R0=0
        MOV     R0,R4       ;TRY MOVE MODE 0,0
        BEQ     DOPOA
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
:   CONDITIONAL BRANCH INST. AND <====
:   REPLACE THE MOVE INSTRUCTION <====
:   WHICH FOLLOWS W/ 774 <====
:
:   MOVE TO MAILBOX # ***** 204 *****
:   SET MSGTYP TO FATAL ERROR
:   Z-BIT NOT SET
DOPOA:  INC      R0          ;R0=1
        COM     R0          ;R0=177776
        COM     R4          ;R4=177777
        BIC     R0,R4       ;TRY BIC: R4=1
        DEC     R4          ;R4=0
        BEQ     DOPOB
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
:   CONDITIONAL BRANCH INST. AND <====
:   REPLACE THE MOVE INSTRUCTION <====
:   WHICH FOLLOWS W/ 762 <====
:
:   MOVE TO MAILBOX # ***** 205 *****
:   SET MSGTYP TO FATAL ERROR
:   BIC CLEAR RESULT INCORRECT
DOPOB:  BIS     R0,R4       ;TRY BIS: R4=177777
        INC     R4
        INC     R4          ;R4=0
        BEQ     DOPOC
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
:   CONDITIONAL BRANCH INST. AND <====
:   REPLACE THE MOVE INSTRUCTION <====
:   WHICH FOLLOWS W/ 752 <====
:
:   MOVE TO MAILBOX # ***** 206 *****
:   SET MSGTYP TO FATAL ERROR
:   RESULT OF BIS INCORRECT
DOPOC:  CLR     R0          ;R0=0
        COMB   R0          ;R0=377
        CLR     R4          ;R4=0
        COM     R4          ;R4=177777
        BIC     R0,R4       ;R4=177400
        ADD     R0,R4       ;TRY ADD: R4=177777
        INC     R4          ;R4=0
```



3237  
3238  
3239  
3240  
3241  
3242  
3243  
3244  
3245 007714 005212  
3246 007716 022712 000114  
3247 007722 001024  
3248 007724 005000  
3249 007726 005010  
3250 007730 105110  
3251 007732 005220  
3252 007734 005400  
3253 007736 060037 000000  
3254 007742 100403  
3255 007744 001402  
3256 007746 102401  
3257 007750 103404  
3258  
3259  
3260  
3261  
3262 007752  
3263 007752 012742 000211  
3264 007756 005242  
3265 007760 000000  
3266 007762 105137 000000  
3267 007766 005337 000000  
3268 007772 001404  
3269  
3270  
3271  
3272  
3273 007774 012742 000212  
3274 010000 005242  
3275 010002 000000  
3276

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND INSTRUCTIONS. IT SETS  
: DATA IN R0 AND LOCATION 0 AND OPERATES UPON IT USING DOP INSTRUCTIONS.  
:\*\*\*\*\*

:TEST 114 TEST MODE 0,X DOUBLE-OPERAND INSTRUCTIONS  
:\*\*\*\*\*

TS114: INC (R2) ;UPDATE TEST NUMBER  
CMP #114,(R2) ;SEQUENCE ERROR?  
BNE TS115-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC. 0=0  
COMB (R0) ;LOC. 0=377  
INC (R0)+ ;LOC. 0=400 R0=2  
NEG R0 ;R0=-2  
ADD R0,@#0 ;TRY ADD 0,3; LOC. 0=376  
BMI DOP03A ;CC=0001?  
BEQ DOP03A  
BVS DOP03A  
BCS DOP03B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 764 <====

DOP03A: MOV #211,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 211 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S NOT SET CORRECTLY  
DOP03B: COMB @#0 ;LOC. 0=1  
DEC @#0 ;LOC. 0=0  
BEQ TS115

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 753 <====

MOV #212,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 212 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DATA RESULT INCORRECT  
; OR SEQUENCE ERROR



3277  
3278  
3279  
3280  
3281  
3282  
3283  
3284  
3285  
3286 010004 005212  
3287 010006 022712 000115  
3288 010012 001042  
3289 010014 005000  
3290 010016 005004  
3291 010020 005204  
3292 010022 020400  
3293 010024 003004  
3294  
3295  
3296  
3297  
3298 010026 012742 000213  
3299 010032 005242  
3300 010034 000000  
3301 010036 020004  
3302 010040 002404  
3303  
3304  
3305  
3306  
3307 010042 012742 000214  
3308 010046 005242  
3309 010050 000000  
3310 010052 005200  
3311 010054 020400  
3312 010056 001404  
3313  
3314  
3315  
3316  
3317 010060 012742 000215  
3318 010064 005242  
3319 010066 000000  
3320 010070 005000  
3321 010072 005100  
3322 010074 005004  
3323 010076 030004  
3324 010100 001404  
3325  
3326  
3327  
3328  
3329 010102 012742 000216  
3330 010106 005242  
3331 010110 000000  
3332 010112 005304

```

*****
THIS TEST VERIFIES MODE 0,0 DOP NON-MODIFYING INSTRUCTIONS.
R0 AND R4 ARE PRESET TO 0 AND 1 RESPECTIVELY. COMPARE INSTRUCTIONS ARE
THEN EXECUTED AND CHECKED. FIRST R4 IS COMPARED TO R0 THEN R0 TO R4.
*****
TEST 115 TEST DOP NON-MODIFYING INST. W/ SOURCE MODE 0,0
*****
TS115: INC (R2) ;UPDATE TEST NUMBER
        CMP #115,(R2) ;SEQUENCE ERROR?
        BNE TS116-10 ;BR TO ERROR HALT ON SEQ ERROR
        CLR R0 ;R0=0
        CLR R4 ;R4=0
        INC R4 ;R4=1
        CMP R4,R0 ;TRY COMPARE R4 TO R0
        BGT DNM1
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 772 <====
        MOV #213,-(R2) ;MOVE TO MAILBOX # ***** 213 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;CC'S NOT CORRECT FOR CMP
DNM1: CMP R0,R4 ;TRY COMPARE R0 TO R4
        BLT DNM2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 764 <====
        MOV #214,-(R2) ;MOVE TO MAILBOX # ***** 214 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;CC'S NOT CORRECT FOR CMP
DNM2: INC R0 ;R0=1
        CMP R4,R0 ;TRY COMPARE R4=1 TO R0=1
        BEQ DNM3
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 755 <====
        MOV #215,-(R2) ;MOVE TO MAILBOX # ***** 215 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;CC'S NOT CORRECT (Z=1) FOR CMP
DNM3: CLR R0 ;R0=0
        COM R0 ;R0=177777
        CLR R4 ;R4=0
        BIT R0,R4 ;TRY BIT R0 TO R4
        BEQ DNM4
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 744 <====
        MOV #216,-(R2) ;MOVE TO MAILBOX # ***** 216 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;CC'S NOT CORRECT FOR BIT
DNM4: DEC R4 ;R4=177777

```

```
3333 010114 030004 BIT R0,R4 ;TRY BIT AGAIN
3334 010116 100404 BMI TS116
3335 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3336 ; CONDITIONAL BRANCH INST. AND <====
3337 ; REPLACE THE MOVE INSTRUCTION <====
3338 ; WHICH FOLLOWS W/ 735 <====
3339 010120 012742 000217 MOV #217,-(R2) ;MOVE TO MAILBOX # ***** 217 *****
3340 010124 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3341 010126 000000 HALT ;CC'S NOT CORRECT FOR BIT
3342 ; OR SEQUENCE ERROR
3343
3344
3345
3346
3347
3348
3349
3350
```

\*\*\*\*\*  
THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND NON-MODIFYING INSTRUCTIONS.  
IT SETS DATA IN R0 AND LOCATION 0 AND COMPARES THEM USING DOPNM INSTRUCTIONS.  
\*\*\*\*\*

TEST 116 TEST MODE 0,X DOUBLE-OPERAND NON-MODIFYING INSTS.  
\*\*\*\*\*

```
3351 010130 005212 TS116: INC (R2) ;UPDATE TEST NUMBER
3352 010132 022712 000116 CMP #116,(R2) ;SEQUENCE ERROR?
3353 010136 001022 BNE TS117-10 ;BR TO ERROR HALT ON SEQ ERROR
3354 010140 005000 CLR R0 ;R0=0
3355 010142 005010 CLR (R0) ;LOC. 0=0
3356 010144 005110 COM (R0) ;LOC. 0=177777
3357 010146 005200 INC R0 ;R0=1
3358 010150 020037 000000 CMP R0,#0 ;TRY CMP MODE 0,3
3359 010154 100403 BMI DNM03A ;CC=0001
3360 010156 001402 BEQ DNM03A
3361 010160 102401 BVS DNM03A
3362 010162 103404 BCS DNM03B
```

```
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 765 <====
```

```
3367 010164 DNM03A: MOV #220,-(R2) ;MOVE TO MAILBOX # ***** 220 *****
3368 010164 012742 000220 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3369 010170 005242 HALT ;CC'S NOT SET CORRECTLY
3370 010172 000000
3371 010174 005300 DNM03B: DEC R0
3372 010176 001002 BNE DNM03C
3373 010200 005210 INC (R0)
3374 010202 001404 BEQ TS117
```

```
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 755 <====
```

```
3379 010204 DNM03C: MOV #221,-(R2) ;MOVE TO MAILBOX # ***** 221 *****
3380 010204 012742 000221 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3381 010210 005242 HALT ;DATA INCORRECTLY MODIFIED BY CMP
3382 010212 000000 ; OR SEQUENCE ERROR
3383
```

3384  
3385  
3386  
3387  
3388  
3389  
3390  
3391  
3392  
3393  
3394 010214 005212  
3395 010216 022712 000117  
3396 010222 001007  
3397 010224 005000  
3398 010226 005100  
3399 010230 005004  
3400 010232 005014  
3401 010234 005214  
3402 010236 061400  
3403 010240 001404  
3404  
3405  
3406  
3407  
3408 010242 012742 000222  
3409 010246 005242  
3410 010250 000000  
3411

```
.....  
: THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS. R0 IS SET TO -1  
: AND LOC 0 TO 1. R4 IS THEN CLEARED AND USED TO POINT TO LOC 0.  
: IN THE ADD MODE 1 INSTRUCTION, LOC 0 IS ADDED TO R0 AND THE  
: RESULTS VERIFIED.  
:.....  
: TEST 117 TEST MODE 1 W/ DOP INST.  
:.....  
TS117: INC (R2) ;UPDATE TEST NUMBER  
CMP #117,(R2) ;SEQUENCE ERROR?  
BNE TS120-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
COM R0 ;R0=177777  
CLR R4 ;R4=0  
CLR (R4) ;LOC 0=0  
INC (R4) ;LOC 0=1  
ADD (R4),R0 ;TRY ADD SOURCE MODE 1  
BEQ TS120  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 770 <====  
MOV #222,-(R2) ;MOVE TO MAILBOX # ***** 222 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF ADD INCORRECT  
: OR SEQUENCE ERROR
```

3412  
3413  
3414  
3415  
3416  
3417  
3418  
3419  
3420  
3421  
3422  
3423  
3424  
3425  
3426  
3427  
3428  
3429  
3430  
3431  
3432  
3433  
3434  
3435  
3436  
3437  
3438  
3439

010252 005212  
010254 022712 000120  
010260 001007  
010262 005000  
010264 005010  
010266 005110  
010270 005004  
010272 151004  
010274 105104  
010276 001404  
  
  
  
  
010300 012742 000223  
010304 005242  
010306 000000

```
.....  
: THIS TEST VERIFIES MODE 1 DOP BYTE INSTRUCTIONS WHICH ADDRESS  
: EVEN BYTES. LOC. 0 IS SET TO -1 AND R4 IS CLEARED. THEN R4 IS  
: SET TO -1 USING A BISB THRU R0 WITH MODE 1.  
:.....  
: TEST 120 TEST MODE 1 - EVEN BYTE W/ DOP INSTS.  
:.....  
TS120: INC (R2) ;UPDATE TEST NUMBER  
CMP #120,(R2) ;SEQUENCE ERROR?  
BNE TS121-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC. 0=0  
COM (R0) ;LOC. 0=177777  
CLR R4 ;R4=0  
BISB (R0),R4 ;TRY MODE 1- EVEN BYTE W/ DOP  
COMB R4 ;R4=0  
BEQ TS121  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 770 <====  
MOV #223,-(R2) ;MOVE TO MAILBOX # ***** 223 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF BISB IS INCORRECT  
: OR SEQUENCE ERROR
```

3440  
3441  
3442  
3443  
3444  
3445  
3446  
3447  
3448  
3449  
3450  
3451  
3452  
3453  
3454  
3455  
3456  
3457  
3458  
3459  
3460  
3461  
3462  
3463  
3464  
3465  
3466  
3467  
3468

010310 005212  
010312 022712 000121  
010316 001007  
010320 005000  
010322 005010  
010324 005110  
010326 005004  
010330 105104  
010332 121004  
010334 001404  
  
  
  
  
010336 012742 000224  
010342 005242  
010344 000000

```
.....  
: THIS TEST VERIFIES MODE 1 DOP NON-MODIFYING INSTRUCTIONS  
: WHICH ADDRESS EVEN BYTES. LOC. 0 IS SET TO -1 AND R0 IS CLEARED  
: AND USED AS THE ADDRESSING REGISTER. R4 IS SET TO 377 AND A  
: MODE 1,0 CMPB INSTRUCTION IS USED THE RESULTS VERIFIED.  
:.....  
: TEST 121 TEST MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING INST.  
:.....  
TS121: INC (R2) ;UPDATE TEST NUMBER  
CMP #121,(R2) ;SEQUENCE ERROR?  
BNE TS122-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC 0=0  
COM (R0) ;LOC 0=177777  
CLR R4 ;R4=0  
COMB R4 ;R4=377  
CMPB (R0),R4 ;TRY MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING  
BEQ TS122  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 770 <====  
: MOVE TO MAILBOX # ***** 224 *****  
: SET MSGTYP TO FATAL ERROR  
: RESULT OF CMPB INCORRECT  
: OR SEQUENCE ERROR
```

3469  
3470  
3471  
3472  
3473  
3474  
3475  
3476  
3477  
3478  
3479  
3480  
3481  
3482  
3483  
3484  
3485  
3486  
3487  
3488  
3489  
3490  
3491  
3492  
3493  
3494  
3495  
3496  
3497  
3498  
3499  
3500  
3501  
3502  
3503  
3504  
3505  
3506  
3507  
3508  
3509  
3510  
3511  
3512  
3513

010346 005212  
010350 022712 000122  
010354 001020  
010356 005000  
010360 005010  
010362 105110  
010364 005110  
010366 005004  
010370 005104  
010372 111004  
010374 005704  
010376 001404  
  
010400 012742 000225  
010404 005242  
010406 000000  
010410 005110  
010412 111004  
010414 100404  
  
010416 012742 000226  
010422 005242  
010424 000000

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 1,0 MOV B INSTRUCTIONS  
: WHICH ADDRESS EVEN BYTES. LOC. 0 IS SET TO 177400, R0 IS CLEARED AND  
: R4 IS SET TO -1. MOV B ARE USED TO MOVE BYTE 0 TO R4. THIS  
: VERIFIES THAT THE PROPER BYTE WAS SELECTED AND THAT THE SIGN-X-TEND  
: FUNCTION WITH MODE 0.  
: THEN LOC. 0 IS COMPLEMENTED AND THE SAME PROCEDURE EXERCISES  
: THE LOGIC FOR COMPLEMENTARY DATA.  
: THIS TEST EXERCISES UNIQUE MICROCODE.  
\*\*\*\*\*  
: TEST 122 TEST MOV INSTRUCTION MODE 1,0 EVEN BYTE  
\*\*\*\*\*  
TS122: INC (R2) ;UPDATE TEST NUMBER  
CMP #122,(R2) ;SEQUENCE ERROR?  
BNE TS123-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC 0=0  
COMB (R0) ;LOC 0=177400  
COM (R0)  
CLR R4 ;R4=0  
COM R4 ;R4=177777  
MOVB (R0),R4 ;R4=0  
TST R4 ;CHECK SIGN OF WORD  
BEQ DOP1  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 766 <====  
MOV #225,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 225 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;MOV B SHOULD SIGN X-TEND  
DOP1: COM (R0) ;LOC 0=177777  
MOVB (R0),R4 ;DO MOV B W/ EVEN BYTE  
BMI TS123  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 757 <====  
MOV #226,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 226 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;MOV B SHOULD SIGN X-TEND  
: OR SEQUENCE ERROR

3514  
3515  
3516  
3517  
3518  
3519  
3520  
3521  
3522  
3523  
3524  
3525  
3526  
3527  
3528  
3529  
3530  
3531  
3532  
3533  
3534  
3535  
3536  
3537  
3538  
3539  
3540  
3541  
3542  
3543

010426 005212  
010430 022712 000123  
010434 001010  
010436 005000  
010440 005010  
010442 005004  
010444 005204  
010446 105114  
010450 151410  
010452 005210  
010454 001404  
  
  
  
  
010456 012742 000227  
010462 005242  
010464 000000

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS WHICH REFERENCE  
: ODD BYTES. LOC. 0 IS SET TO 177400. R0 IS SET TO 0 AND R4 IS  
: SET TO 1. THE BISB INSTRUCTION USES THE DATA IN BYTE 1 TO SET BYTE 0.  
: THE RESULT IS CHECKED BY INCREMENTING THE WORD (LOC. 0) TO ZERO.  
\*\*\*\*\*

: TEST 123 TEST MODE 1-ODD BYTE W/ DOP INSTS.  
\*\*\*\*\*

TS123: INC (R2) ;UPDATE TEST NUMBER  
CMP #123,(R2) ;SEQUENCE ERROR?  
BNE TS124-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC. 0=0  
CLR R4 ;R4=0  
INC R4 ;R4=1  
COMB (R4) ;LOC. 0=177400  
BISB (R4),(R0) ;TRY TO BIS LOW ORDER BITS W/ MODE 1  
INC (R0) ;CHECK RESULT  
BEQ TS124  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 767 <====  
: MOVE TO MAILBOX # \*\*\*\*\* 227 \*\*\*\*\*  
: SET MSGTYP TO FATAL ERROR  
: RESULT OF BISB INCORRECT  
: OR SEQUENCE ERROR

3544  
3545  
3546  
3547  
3548  
3549  
3550  
3551  
3552  
3553  
3554  
3555  
3556  
3557  
3558  
3559  
3560  
3561  
3562  
3563  
3564  
3565  
3566  
3567  
3568  
3569  
3570  
3571  
3572  
3573  
3574  
3575  
3576  
3577  
3578  
3579  
3580  
3581

010466 005212  
010470 022712 000124  
010474 001015  
010476 005000  
010500 005010  
010502 005110  
010504 012004  
010506 005204  
010510 001404  
  
010512 012742 000230  
010516 005242  
010520 000000  
010522 005300  
010524 005300  
010526 001404  
  
010530 012742 000231  
010534 005242  
010536 000000

```
.....  
: THIS TEST VERIFIES MODE 2 DOP INSTRUCTIONS. LOC. 0 IS SET TO -1.  
: R0 IS CLEARED AND USED AS THE MODE 2 ADDRESSING REGISTER TO MOVE LOC. 0  
: TO R7. THE DATA RESULTS ARE VERIFIED AND THE INCREMENTING OF THE REGISTER  
: IS CHECKED.  
:.....  
: TEST 124 TEST MODE 2 W/ DOP INSTS.  
:.....  
TS124: INC (R2) ;UPDATE TEST NUMBER  
CMP #124,(R2) ;SEQUENCE ERROR?  
BNE TS125-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC. 0=0  
COM (R0) ;LOC. 0=177777  
MOV (R0)+,R4 ;TRY MOVE MODE 2,0  
INC R4 ;CHECK R4  
BEQ DOP2  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 771 <====  
MOV #230,-(R2) ;MOVE TO MAILBOX # ***** 230 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
DOP2: HALT ;RESULT OF MOV INST INCORRECT  
DEC R0 ;TEST R0 AFTER MODE 2  
DEC R0  
BEQ TS125  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 762 <====  
MOV #231,-(R2) ;MOVE TO MAILBOX # ***** 231 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;REGISTER NOT INCREMENTED IN MODE 2  
; OR SEQUENCE ERROR
```



3582  
3583  
3584  
3585  
3586  
3587  
3588  
3589  
3590  
3591  
3592  
3593  
3594  
3595  
3596  
3597  
3598  
3599  
3600  
3601  
3602  
3603  
3604  
3605  
3606  
3607  
3608  
3609  
3610  
3611  
3612  
3613  
3614  
3615  
3616  
3617  
3618  
3619  
3620

010540 005212  
010542 022712 000125  
010546 001016  
010550 005000  
010552 010010  
010554 005110  
010556 142010  
010560 105737 000001  
010564 001404  
  
010566 012742 000232  
010572 005242  
010574 000000  
010576 105137 000000  
010602 001404  
  
010604 012742 000233  
010610 005242  
010612 000000

```
.....  
: THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH ADDRESS  
: EVEN BYTES. LOC. 0 IS SET TO -1. R0 IS CLEARED AND USED AS THE  
: ADDRESSING REGISTER IN A TEST WHICH TRIES TO CLEAR BYTE 1 USING  
: BYTE 0 DATA AND A BICB. UNIQUE IN THIS TEST IS USE OF THE  
: SAME ADDRESSING REGISTER FOR BOTH SOURCE AND DESTINATION. THE SOURCE AND  
: DESTINATION IS CHECKED TO INSURE PROPER FUNCTIONING.  
:.....  
: TEST 125 TEST MODE 2 - EVEN BYTE W/ DOP INST.  
:.....  
TS125: INC (R2) ;UPDATE TEST NUMBER  
CMP #125,(R2) ;SEQUENCE ERROR?  
BNE TS126-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
MOV R0,(R0) ;LOC. 0=0  
COM (R0) ;LOC. 0=177777  
BICB (R0)+,(R0) ;TRY TO CLEAR BYTE 1 FROM BYTE 0 W/ BICB  
TSTB @#1 ;CHECK RESULT  
BEQ DOPB2A  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 770 <====  
MOV #232,-(R2) ;MOVE TO MAILBOX # ***** 232 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
DOPB2A: CUMB @#0 ;BICB DESTINATION INCORRECT  
BEQ TS126 ;CHECK BICB SOURCE  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 761 <====  
MOV #233,-(R2) ;MOVE TO MAILBOX # ***** 233 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;BICB SOURCE INCORRECTLY CHANGED  
; OR SEQUENCE ERROR
```

3621  
3622  
3623  
3624  
3625  
3626  
3627  
3628  
3629  
3630  
3631  
3632  
3633  
3634  
3635  
3636  
3637  
3638  
3639  
3640  
3641  
3642  
3643  
3644  
3645  
3646  
3647  
3648  
3649  
3650  
3651  
3652  
3653  
3654  
3655  
3656  
3657  
3658  
3659

010614 005212  
010616 022712 000126  
010622 001017  
010624 005000  
010626 005004  
010630 005010  
010632 005110  
010634 105120  
010636 112004  
010640 005204  
010642 001404  
  
010644 012742 000234  
010650 005242  
010652 000000  
010654 005740  
010656 005700  
010660 001404  
  
010662 012742 000235  
010666 005242  
010670 000000

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH REFERENCE  
: ODD BYTES. R0 IS SET TO 1, LOC. 0 IS SET TO 177400, AND R4 IS CLEARED.  
: A MODE 2 MOV B USES R0 TO MOVE BYTE 1 TO R4. AN INCREMENT  
: IS USED TO CHECK THAT THE PROPER BYTE WAS MOVED AND SIGN X-TENDED.  
\*\*\*\*\*  
: TEST 126 TEST MODE 2 - ODD BYTE W/ DOP INST.  
\*\*\*\*\*  
TS126: INC (R2) ;UPDATE TEST NUMBER  
CMP #126,(R2) ;SEQUENCE ERROR?  
BNE TS127-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR R4 ;R4=0  
CLR (R0) ;LOC. 0=0  
COM (R0) ;LOC. 0=177777  
COMB (R0)+ ;LOC 0=177400; R0=1  
MOVB (R0)+,R4 ;TRY DOP MODE 2 W/ ODD BYTE  
INC R4 ;CHECK RESULT OF MOV B  
BEQ DOPB2B  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 767 <====  
MOV #234,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 234 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF MOV B INCORRECT  
DOPB2B: TST -(R0) ;BUMP R0 DOWN BY 2  
TST R0 ;CHECK R0  
BEQ TS127  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 760 <====  
MOV #235,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 235 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;MODE 2 BYTE DID NOT INCREMENT REG. CORRECTLY  
: OR SEQUENCE ERROR

3660  
3661  
3662  
3663  
3664  
3665  
3666  
3667  
3668  
3669  
3670  
3671 010672 005212  
3672 010674 022712 000127  
3673 010700 001011  
3674 010702 012737 052525 000000  
3675 010710 012700 125252  
3676 010714 053700 000000  
3677 010720 005200  
3678 010722 001404  
3679  
3680  
3681  
3682  
3683 010724 012742 000236  
3684 010730 005242  
3685 010732 000000  
3686  
3687  
3688  
3689  
3690  
3691  
3692  
3693  
3694  
3695  
3696  
3697 010734 005212  
3698 010736 022712 000130  
3699 010742 001011  
3700 010744 012737 052652 000000  
3701 010752 005000  
3702 010754 153700 000000  
3703 010760 022700 000252  
3704 010764 001404  
3705  
3706  
3707  
3708  
3709 010766 012742 000237  
3710 010772 005242  
3711 010774 000000  
3712

```
*****
:
:   THIS TEST VERIFIES MODE 3 DOUBLE-OPERAND INSTRUCTIONS.
: LOC. 0 IS LOADED WITH ALTERNATING ZEROES AND ONES; AND R0 IS LOADED
: WITH ALTERNATING ONES AND ZEROES. A MODE 3 BIS IS USED TO SET R0
: TO -1 BY USING LOC. 0 AS THE SOURCE TO BIS THE ZEROES IN R0. THE
: RESULT IS TESTED BY INCREMENTING R0 AND CHECKING FOR ZERC.
:
:*****
: TEST 127      TEST MODE 3 W/ DOP INSTS.
:*****
TS127:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #127,(R2)    ;SEQUENCE ERROR?
        BNE     TS130-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #052525,@#0   ;MOVE 52525 TO LOC. 0
        MOV     #125252,R0    ;SET ALT. ONE AND ZERO IN R0
        BIS     @#0,R0        ;TRY TO SET ALL OTHER BITS W/ MODE 3
        INC     R0            ;TEST RESULT
        BEQ     TS130
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
:               CONDITIONAL BRANCH INST. AND <====
:               REPLACE THE MOVE INSTRUCTION <====
:               WHICH FOLLOWS W/ 766 <====
        MOV     #236,-(R2)    ;MOVE TO MAILBOX # ***** 236 *****
        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
        HALT    ;BIS W/ MODE 3 INCORRECT RESULT
:               OR SEQUENCE ERROR
:*****
:   THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS WHICH
: ADDRESS EVEN BYTES. BYTE 0 IS SET TO ALTERNATING 1'S AND 0'S; BYTE 1,
: ALTERNATING 0'S AND 1'S. R0 IS CLEARED AND A BISB IS USED TO
: SET THE LOW BYTE OF R0 TO 252.
:*****
: TEST 130      TEST MODE 3 - EVEN BYTE W/ DOP INSTS.
:*****
TS130:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #130,(R2)    ;SEQUENCE ERROR?
        BNE     TS131-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #52652,@#0   ;MOVE 1'S AND 0' PATTERN TO LOC. 0
        CLR     R0            ;R0=0
        BISB    @#0,R0        ;TRY R0=252 W/ MODE 3 - EVEN BYTE
        CMP     #252,R0       ;BISB W/ EVEN BYTE SUCCESSFUL?
        BEQ     TS131
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
:               CONDITIONAL BRANCH INST. AND <====
:               REPLACE THE MOVE INSTRUCTION <====
:               WHICH FOLLOWS W/ 766 <====
        MOV     #237,-(R2)    ;MOVE TO MAILBOX # ***** 237 *****
        INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
        HALT    ;BISB W/ MODE 3 - EVEN BYTE FAILED
:               OR SEQUENCE ERROR
```

3713  
3714  
3715  
3716  
3717  
3718  
3719  
3720  
3721  
3722  
3723 010776 005212  
3724 011000 022712 000131  
3725 011004 001011  
3726 011006 012737 052652 000000  
3727 011014 005000  
3728 011016 153700 000001  
3729 011022 022700 000125  
3730 011026 001404  
3731  
3732  
3733  
3734  
3735 011030 012742 000240  
3736 011034 005242  
3737 011036 000000  
3738  
3739  
3740  
3741  
3742  
3743 011040 005212  
3744 011042 022712 000132  
3745 011046 001017  
3746 011050 005000  
3747 011052 105100  
3748 011054 000263  
3749 011056 132700 000200  
3750 011062 001403  
3751 011064 102402  
3752 011066 103001  
3753 011070 100404  
3754  
3755  
3756  
3757  
3758 011072  
3759 011072 012742 000241  
3760 011076 005242  
3761 011100 000000  
3762 011102 105100  
3763 011104 001404  
3764  
3765  
3766  
3767  
3768 011106 012742 000242

```
.....  
: THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS  
: WHICH ADDRESS ODD BYTES. THE SAME PROCEDURE USED IN PREVIOUS  
: TEST IS USED HERE. THIS TIME BYTE 1 IS USED AS THE SOURCE BYTE.  
: THE EXPECTED RESULT IS: R0 = 125.  
:.....  
: TEST 131 TEST MODE 3 - ODD BYTE W/ DOP INSTS.  
:.....  
TS131: INC (R2) ;UPDATE TEST NUMBER  
CMP #131,(R2) ;SEQUENCE ERROR?  
BNE TS132-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #52652,@#0 ;MOVE 1'S AND 0'S PATTERN TO LOC 0  
CLR R0 ;R0=0  
BISB @#1,R0 ;TRY R0=152 W/ MODE 3 - ODD BYTE  
CMP #125,R0 ;R0=125?  
BEQ TS132  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 766 <====  
MOV #240,-(R2) ;MOVE TO MAILBOX # ***** 240 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;BISB W/ MODE 3 - ODD BYTE FAILED  
; OR SEQUENCE ERROR  
:.....  
: TEST 132 TEST DEST. MODE 0-BYTE W/ DOP NON-MODIFYING MST  
:.....  
TS132: INC (R2) ;UPDATE TEST NUMBER  
CMP #132,(R2) ;SEQUENCE ERROR?  
BNE TS133-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
COMB R0 ;R0=377  
+SEC!SEV ;SET C AND V BITS  
BITB #200,R0 ;TRY DOPNM DEST. MODE 0-BYTE  
BEQ DNMB0A ;BR TO ERROR IF Z BIT SET  
BVS DNMB0A ;BR TO ERROR IF V BIT SET  
BCC DNMB0A ;BR TO ERROR IF C BIT CLEAR.  
BMI DNMB0B  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 766 <====  
DNMB0A: MOV #241,-(R2) ;MOVE TO MAILBOX # ***** 241 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S INCORRECT  
DNMB0B: COMB R0 ;CHECK DESTINATION DATA  
BEQ TS133  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 766 <====  
MOV #242,-(R2) ;MOVE TO MAILBOX # ***** 242 *****
```

```
3769 011112 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
3770 011114 000000          HALT                ;DEST. DATA MODIFIED
3771                                     ; OR SEQUENCE ERROR
3772
3773
3774
3775
3776 011116 005212          TS133: INC      (R2)          ;UPDATE TEST NUMBER
3777 011120 022712 000133    CMP      #133,(R2)     ;SEQUENCE ERROR?
3778 011124 001017          BNE     TS134-10      ;BR TO ERROR HALT ON SEQ ERROR
3779 011126 005000          CLR     R0           ;R0=0
3780 011130 005010          CLR     (R0)         ;LOC. 0=0
3781 011132 000241          CLC                    ;CLEAR C BIT
3782 011134 032710 177777    BIT     #177777,(R0)  ;TRY DOPNM DEST. MODE 1
3783 011140 100403          BMI     DNM1A        ;BR TO ERROR IF N BIT SET
3784 011142 102402          BVS     DNM1A        ;BR TO ERROR IF V BIT SET
3785 011144 103401          BCS     DNM1A        ;BR TO ERROR IF C BIT SET
3786 011146 001404          BEQ     DNM1B
3787
3788                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3789                                     ;          CONDITIONAL BRANCH INST. AND <====
3790                                     ;          REPLACE THE MOVE INSTRUCTION <====
3791                                     ;          WHICH FOLLOWS W/ 766 <====
3792 011150 012742 000243    DNM1A: MOV     #243,-(R2)   ;MOVE TO MAILBOX # ***** 243 *****
3793 011154 005242          INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
3794 011156 000000          HALT                ;COND. CODES INCORRECT
3795 011160 005710          DNM1B: TST     (R0)   ;CHECK TEST DATA
3796 011162 001404          BEQ     TS134
3797
3798                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3799                                     ;          CONDITIONAL BRANCH INST. AND <====
3800                                     ;          REPLACE THE MOVE INSTRUCTION <====
3801                                     ;          WHICH FOLLOWS W/ 760 <====
3802 011164 012742 000244    MOV     #244,-(R2)   ;MOVE TO MAILBOX # ***** 244 *****
3803 011170 005242          INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
3804 011172 000000          HALT                ;DESTINATION DATA MODIFIED
3805                                     ; OR SEQUENCE ERROR
3806
3807
3808
3809 011174 005212          ;*****
3810 011176 022712 000134    ;TEST 134 TEST DEST, MODE 2 W/ DOP NON-MODIFYING INST.
3811 011202 001027          ;*****
3812 011204 005000          TS134: INC     (R2)   ;UPDATE TEST NUMBER
3813 011206 005010          CMP     #134,(R2)     ;SEQUENCE ERROR?
3814 011210 052710 125252    BNE     TS135-10      ;BR TO ERROR HALT ON SEQ ERROR
3815 011214 032720 077777    CLR     R0           ;R0=0
3816 011220 102402          CLR     (R0)         ;LOC. 0=125252
3817 011222 001401          BIS     #125252,(R0)  ;TRY DOPNM INST W/ MODE 2
3818 011224 100004          BIT     #77777,(R0)+ ;BR TO ERROR IF V BIT SET
3819                                     ;BR TO ERROR IF Z-BIT SET
3820                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3821                                     ;          CONDITIONAL BRANCH INST. AND <====
3822                                     ;          REPLACE THE MOVE INSTRUCTION <====
3823                                     ;          WHICH FOLLOWS W/ 766 <====
3824 011226 012742 000245    DNM2A: MOV     #245,-(R2) ;MOVE TO MAILBOX # ***** 245 *****
```

|      |        |        |        |             |              |  |       |
|------|--------|--------|--------|-------------|--------------|--|-------|
| 3825 | 011232 | 005242 |        | INC         | -(R2)        | :SET MSGTYP TO FATAL ERROR               |       |
| 3826 | 011234 | 000000 |        | HALT        |              | :COND. CODES INCORRECT                   |       |
| 3827 | 011236 | 005300 |        | DNM2B: DEC  | RO           | :DECREMENT RO TO CHECK IT.               |       |
| 3828 | 011240 | 005300 |        | DEC         | RO           |  |       |
| 3829 | 011242 | 001404 |        | BEQ         | DNM2D        |  |       |
| 3830 |        |        |        |             |              | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 3831 |        |        |        |             |              | : CONDITIONAL BRANCH INST. AND           | <==== |
| 3832 |        |        |        |             |              | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 3833 |        |        |        |             |              | : WHICH FOLLOWS W/ 757                   | <==== |
| 3834 | 011244 |        |        | DNM2C:      |              |  |       |
| 3835 | 011244 | 012742 | 000246 | MOV         | #246,-(R2)   | :MOVE TO MAILBOX # ***** 246 *****       |       |
| 3836 | 011250 | 005242 |        | INC         | -(R2)        | :SET MSGTYP TO FATAL ERROR               |       |
| 3837 | 011252 | 000000 |        | HALT        |              | :MODE 2 REGISTER NOT INCREMENTED BY 2    |       |
| 3838 | 011254 | 022710 | 125252 | DNM2D: CMP  | #125252,(R0) | :CHECK DEST. DATA                        |       |
| 3839 | 011260 | 001404 |        | BEQ         | TS135        |  |       |
| 3840 |        |        |        |             |              | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 3841 |        |        |        |             |              | : CONDITIONAL BRANCH INST. AND           | <==== |
| 3842 |        |        |        |             |              | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 3843 |        |        |        |             |              | : WHICH FOLLOWS W/ 750                   | <==== |
| 3844 | 011262 | 012742 | 000247 | MOV         | #247,-(R2)   | :MOVE TO MAILBOX # ***** 247 *****       |       |
| 3845 | 011266 | 005242 |        | INC         | -(R2)        | :SET MSGTYP TO FATAL ERROR               |       |
| 3846 | 011270 | 000000 |        | HALT        |              | :DEST. DATA MODIFIED                     |       |
| 3847 |        |        |        |             |              | : OR SEQUENCE ERROR                      |       |
| 3848 |        |        |        |             |              |  |       |
| 3849 |        |        |        |             |              |  |       |
| 3850 |        |        |        |             |              |  |       |
| 3851 |        |        |        |             |              |  |       |
| 3852 | 011272 | 005212 |        |             |              |  |       |
| 3853 | 011274 | 022712 | 000135 | TS135: INC  | (R2)         | :UPDATE TEST NUMBER                      |       |
| 3854 | 011300 | 001051 |        | CMP         | #135,(R2)    | :SEQUENCE ERROR?                         |       |
| 3855 | 011302 | 005000 |        | BNE         | TS136-10     | :BR TO ERROR HALT ON SEQ ERROR           |       |
| 3856 | 011304 | 005010 |        | CLR         | RO           | :RO=0                                    |       |
| 3857 | 011306 | 052710 | 052652 | CLR         | (R0)         | :LOC. 0=0                                |       |
| 3858 | 011312 | 000263 |        | BIS         | #52652,(R0)  | :LOC. 0=52652                            |       |
| 3859 | 011314 | 132720 | 000201 | +SEC!SEV    |              | :SET C AND V BITS                        |       |
| 3860 | 011320 | 001403 |        | BITB        | #201,(R0)+   | :TRY DOPNM INST. W/ MODE 2 EVEN BYTE     |       |
| 3861 | 011322 | 103002 |        | BEQ         | DNMB2A       | :BR TO ERROR IF Z-BIT SET                |       |
| 3862 | 011324 | 102401 |        | BCC         | DNMB2A       | :BR TO ERROR IF C-BIT CLEAR              |       |
| 3863 | 011326 | 100404 |        | BVS         | DNMB2A       | :BR TO ERROR IF V-BIT SET                |       |
| 3864 |        |        |        | BMI         | DNMB2B       |  |       |
| 3865 |        |        |        |             |              | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 3866 |        |        |        |             |              | : CONDITIONAL BRANCH INST. AND           | <==== |
| 3867 |        |        |        |             |              | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 3868 | 011330 |        |        | DNMB2A:     |              | : WHICH FOLLOWS W/ 764                   | <==== |
| 3869 | 011330 | 012742 | 000250 | MOV         | #250,-(R2)   | :MOVE TO MAILBOX # ***** 250 *****       |       |
| 3870 | 011334 | 005242 |        | INC         | -(R2)        | :SET MSGTYP TO FATAL ERROR               |       |
| 3871 | 011336 | 000000 |        | HALT        |              | :COND. CODES INCORRECT                   |       |
| 3872 | 011340 | 005300 |        | DNMB2B: DEC | RO           | :CHECK DEST. REGISTER.                   |       |
| 3873 | 011342 | 001404 |        | BEQ         | DNMB2C       |  |       |
| 3874 |        |        |        |             |              | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 3875 |        |        |        |             |              | : CONDITIONAL BRANCH INST. AND           | <==== |
| 3876 |        |        |        |             |              | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 3877 |        |        |        |             |              | : WHICH FOLLOWS W/ 756                   | <==== |
| 3878 | 011344 | 012742 | 000251 | MOV         | #251,-(R2)   | :MOVE TO MAILBOX # ***** 251 *****       |       |
| 3879 | 011350 | 005242 |        | INC         | -(R2)        | :SET MSGTYP TO FATAL ERROR               |       |
| 3880 | 011352 | 000000 |        | HALT        |              | :DEST. REGISTER NOT INCREMENTED BY 1     |       |

```
3881 011354 005200
3882 011356 132720 000201
3883 011362 001402
3884 011364 102401
3885 011366 100004
3886
3887
3888
3889
3890 011370
3891 011370 012742 000252
3892 011374 005242
3893 011376 000000
3894 011400 005300
3895 011402 005300
3896 011404 001404
3897
3898
3899
3900
3901 011406 012742 000253
3902 011412 005242
3903 011414 000000
3904 011416 022710 052652
3905 011422 001404
3906
3907
3908
3909
3910 011424 012742 000254
3911 011430 005242
3912 011432 000000
3913
3914
3915
3916
3917
3918
3919 011434 005212
3920 011436 022712 000136
3921 011442 001050
3922 011444 005000
3923 011446 005010
3924 011450 052710 125125
3925 011454 105100
3926 011456 005200
3927 011460 005010
3928 011462 000263
3929 011464 132730 000201
3930 011470 001403
3931 011472 102402
3932 011474 103001
3933 011476 100004
3934
3935
3936
```

```
DNMB2C: INC R0 ;RO=1
          BITB #201,(R0)+ ;TRY DOPNM INST. W/MODE 2-ODD BYTE
          BEQ DNMB2D ;BR TO ERROR IF Z-BIT SET
          BVS DNMB2D ;BR TO ERROR IF V-BIT SET
          BPL DNMB2E
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 744 <====

DNMB2D:
          MOV #252,-(R2) ;MOVE TO MAILBOX # ***** 252 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;COND. CODES INCORRECT
DNMB2E: DEC R0 ;DEC R0 TO CHECK IT.
          DEC R0
          BEQ DNMB2F
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 735 <====

          MOV #253,-(R2) ;MOVE TO MAILBOX # ***** 253 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;DEST. REGISTER NOT INCREMENTED BY 1
DNMB2F: CMP #52652,(R0) ;CHECK DEST. DATA IS UNMODIFIED
          BEQ TS136
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 726 <====

          MOV #254,-(R2) ;MOVE TO MAILBOX # ***** 254 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;DEST. DATA WAS MODIFIED.
          ; OR SEQUENCE ERROR

;*****
;TEST 136 TEST DEST. MODE 3-BYTES W/DOP NON-MODIFYING INST.
;*****
TS136: INC (R2) ;UPDATE TEST NUMBER
        CMP #136,(R2) ;SEQUENCE ERROR?
        BNE TS137-10 ;BR TO ERROR HALT ON SEQ ERROR
        CLR R0 ;RO=0
        CLR (R0) ;LOC. 0=0
        BIS #125125,(R0) ;LOC. 0=125125
        COMB R0 ;RO=377
        INC R0 ;RO=400
        CLR (R0) ;LOC. 400=0
        +SEC!SEV ;C-BIT=V-BIT=1
        BITB #201,@(R0)+ ;TRY DOPNM W/MODE 3-EVEN BYTE
        BEQ DNMB3A ;BR TO ERROR IF Z BIT SET
        BVS DNMB3A ;BR TO ERROR IF V BIT SET
        BCC DNMB3A ;BR TO ERROR IF C BIT CLEAR
        BPL DNMB3B
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
```

```
3937
3938 011500
3939 011500 012742 000255
3940 011504 005242
3941 011506 000000
3942 011510 022700 000402
3943 011514 001404
3944
3945
3946
3947
3948 011516 012742 000256
3949 011522 005242
3950 011524 000000
3951 011526 005200
3952 011530 005200
3953 011532 132730 000201
3954 011536 001402
3955 011540 102401
3956 011542 100404
3957
3958
3959
3960
3961 011544
3962 011544 012742 000257
3963 011550 005242
3964 011552 000000
3965 011554 005004
3966 011556 022714 125125
3967 011562 001404
3968
3969
3970
3971
3972 011564 012742 000260
3973 011570 005242
3974 011572 000000
3975
3976
3977
3978
3979
3980 011574 005212
3981 011576 022712 000137
3982 011602 001033
3983 011604 005000
3984 011606 005010
3985 011610 052710 125252
3986 011614 052700 000002
3987 011620 000277
3988 011622 032740 020000
3989 011626 100403
3990 011630 102402
3991 011632 103001
3992 011634 001004
```

DNMB3A: ; WHICH FOLLOWS W/ 761 <====  
MOV #255,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 255 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;COND. CODES INCORRECT  
DNMB3B: CMP #402,R0 ;CHECK DEST. REGISTER INC. BY 2 AND INC BY 2 AGAIN  
BEQ DNMB3C ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 752 <====  
MOV #256,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 256 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEST. REGISTER NOT INCREMENTED BY 2  
DNMB3C: INC R0 ;RO=404  
INC R0 ;TRY DOPNM DEST MODE 3-BYTE(ODD)  
BITB #201,@(R0)+ ;BR TO ERROR IF Z BIT SET  
BEQ DNMB3D ;BR TO ERROR IF V BIT SET  
BVS DNMB3D  
BMI DNMB3E ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 737 <====  
DNMB3D: MOV #257,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 257 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;COND. CODES INCORRECT  
DNMB3E: CLR R4 ;R4=0  
CMP #125125,(R4) ;CHECK DEST. DATA  
BEQ TS137 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 727 <====  
MOV #260,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 260 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEST. DATA MODIFIED  
; OR SEQUENCE ERROR

\*\*\*\*\*  
;TEST 137 TEST DEST. MODE 4 W/DOP NON-MODIFYING INST.  
\*\*\*\*\*  
TS137: INC (R2) ;UPDATE TEST NUMBER  
CMP #137,(R2) ;SEQUENCE ERROR?  
BNE TS140-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;RO=0  
CLR (R0) ;LOC. 0=0  
BIS #125252,(R0) ;LOC. 0=125125  
BIS #2,R0 ;RO=2  
SCC ;SET ALL COND. CODE BITS  
BIT #20000,-(R0) ;TRY DOPNM W/ MODE 4  
BMI DNM4A ;BR TO ERROR IF N-BIT SET  
BVS DNM4A ;BR TO ERROR IF V-BIT SET  
BCC DNM4A ;BR TO ERROR IF C-BIT CHAR  
BNE DNM4B



```
3993 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3994 ; CONDITIONAL BRANCH INST. AND <====
3995 ; REPLACE THE MOVE INSTRUCTION <====
3996 ; WHICH FOLLOWS W/ 762 <====
3997 011636 DNM4A: MOV #261,-(R2) ;MOVE TO MAILBOX # ***** 261 *****
3998 011636 012742 000261 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3999 011642 005242 HALT ;COND. CODES INCORRECT
4000 011644 000000 DNM4B: TST R0 ;CHECK DEST. REGISTER
4001 011646 005700 BEQ DNM4C
4002 011650 001404
4003 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4004 ; CONDITIONAL BRANCH INST. AND <====
4005 ; REPLACE THE MOVE INSTRUCTION <====
4006 ; WHICH FOLLOWS W/ 754 <====
4007 011652 012742 000262 MOV #262,-(R2) ;MOVE TO MAILBOX # ***** 262 *****
4008 011656 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4009 011660 000000 HALT ;DEST. REGISTER NOT DECREMENTED BY 2
4010 011662 022737 125252 000000 DNM4C: CMP #125252,@#0 ;CHECK DEST. DATA
4011 011670 001404 BEQ TS140
4012 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4013 ; CONDITIONAL BRANCH INST. AND <====
4014 ; REPLACE THE MOVE INSTRUCTION <====
4015 ; WHICH FOLLOWS W/ 744 <====
4016 011672 012742 000263 MOV #263,-(R2) ;MOVE TO MAILBOX # ***** 263 *****
4017 011676 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4018 011700 000000 HALT ;DEST. DATA MODIFIED
4019 ; OR SEQUENCE ERROR
4020
4021 ;*****
4022 ;TEST 140 TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
4023 ;*****
4024 011702 005212 TS140: INC (R2) ;UPDATE TEST NUMBER
4025 011704 022712 000140 CMP #140,(R2) ;SEQUENCE ERROR?
4026 011710 001051 BNE TS141-10 ;BR TO ERROR HALT ON SEQ ERROR
4027 011712 005000 CLR R0 ;R0=0
4028 011714 005010 CLR (R0) ;LOC. 0=0
4029 011716 052710 052652 BIS #52652,(R0) ;LOC. 0=52652
4030 011722 052700 000002 BIS #2,R0 ;R0=2
4031 011726 000257 CCC ;COND. CODES=0
4032 011730 132740 000201 BITB #201,-(R0) ;TRY DOPNM INST W/MODE 4 ODD BYTE
4033 011734 102403 BVS DNMB4A ;BR TO ERROR IF V BIT SET
4034 011736 001402 BEQ DNMB4A ;BR TO ERROR IF Z BIT SET
4035 011740 103401 BCS DNMB4A ;BR TO ERROR IF C BIT SET
4036 011742 001004 BNE DNMB4B
4037 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4038 ; CONDITIONAL BRANCH INST. AND <====
4039 ; REPLACE THE MOVE INSTRUCTION <====
4040 ; WHICH FOLLOWS W/ 762 <====
4041 011744 DNMB4A: MOV #264,-(R2) ;MOVE TO MAILBOX # ***** 264 *****
4042 011744 012742 000264 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4043 011750 005242 HALT ;COND. CODES INCORRECT
4044 011752 000000 DNM4B: CMP #1,R0 ;CHECK DEST. REGISTER
4045 011754 022700 000001 BEQ DNMB4C
4046 011760 001404
4047 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4048 ; CONDITIONAL BRANCH INST. AND <====
```

```
4049                                     :           REPLACE THE MOVE INSTRUCTION <====
4050                                     :           WHICH FOLLOWS W/ 753          <====
4051 011762 012742 000265                MOV    #265,-(R2)                       :MOVE TO MAILBOX # ***** 265 *****
4052 011766 005242                       INC    -(R2)                             :SET MSGTYP TO FATAL ERROR
4053 011770 000000                       HALT                                     :DEST REG. NOT DECREMENTED BY 1
4054 011772 132740 000201  DNMB4C: BITB  #201,-(R0)           :TRY DOPNM INST. W/MODE 4 EVEN BYTE
4055 011776 001401                       BEQ    DNMB4D                             :BR TO ERROR IF Z-BIT SET
4056 012000 100404                       BMI    DNMB4E
4057                                     :           : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4058                                     :           CONDITIONAL BRANCH INST. AND   <====
4059                                     :           REPLACE THE MOVE INSTRUCTION  <====
4060                                     :           WHICH FOLLOWS W/ 743          <====
4061 012002 012742 000266  DNMB4D: MOV    #266,-(R2)                       :MOVE TO MAILBOX # ***** 266 *****
4062 012002 005242                       INC    -(R2)                             :SET MSGTYP TO FATAL ERROR
4063 012006 000000                       HALT                                     :COND. CODES INCORRECT
4064 012010 005700  DNMB4E: TST    R0                               :CHECK DEST. REGISTER
4065 012012 001404                       BEQ    DNMB4F
4066                                     :           : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4067                                     :           CONDITIONAL BRANCH INST. AND   <====
4068                                     :           REPLACE THE MOVE INSTRUCTION  <====
4069                                     :           WHICH FOLLOWS W/ 735          <====
4070 012016 012742 000267                MOV    #267,-(R2)                       :MOVE TO MAILBOX # ***** 267 *****
4071 012022 005242                       INC    -(R2)                             :SET MSGTYP TO FATAL ERROR
4072 012024 000000                       HALT                                     :DEST. REG. NOT DECREMENTED BY 1
4073 012026 022710 052652  DNMB4F: CMP    #52652,(R0)                   :CHECK DESTINATION DATA
4074 012032 001404                       BEQ    TS141
4075                                     :           : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4076                                     :           CONDITIONAL BRANCH INST. AND   <====
4077                                     :           REPLACE THE MOVE INSTRUCTION  <====
4078                                     :           WHICH FOLLOWS W/ 726          <====
4079 012034 012742 000270                MOV    #270,-(R2)                       :MOVE TO MAILBOX # ***** 270 *****
4080 012040 005242                       INC    -(R2)                             :SET MSGTYP TO FATAL ERROR
4081 012042 000000                       HALT                                     :DEST. DATA MODIFIED
4082                                     :           OR SEQUENCE ERROR
4083
4084
4085 ;*****
4086 ;TEST 141 TEST DEST MODE 5 W/DOP NON-MODIFYING INST.
4087 ;*****
4088 012044 005212 000141  TS141: INC    (R2)                               :UPDATE TEST NUMBER
4089 012046 022712                CMP    #141,(R2)                       :SEQUENCE ERROR?
4090 012052 001034                BNE    TS142-10                         :BR TO ERROR HALT ON SEQ ERROR
4091 012054 005000                CLR    R0                               :R0=0
4092 012056 005010                CLR    (R0)                             :LOC 0=0
4093 012060 052710 100000        BIS    #100000,(R0)                   :LOC. 0=100000
4094 012064 052700 000402        BIS    #402,R0                       :R0=2
4095 012070 000277                SCC                                     :SET ALL COND. CODE BITS
4096 012072 032750 100000        BIT    #100000,a-(R0)                 :TRY DOPNM W/MODE 5
4097 012076 102403                BVS    DNMB5A                          :BR TO ERROR IF V-BIT SET
4098 012100 103002                BCC    DNMB5A                          :BR TO ERROR IF C-BIT CLEAR
4099 012102 001401                BEQ    DNMB5A                          :BR TO ERROR IF Z-BIT SET
4100 012104 100404                BMI    DNMB5B
4101                                     :           : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4102                                     :           CONDITIONAL BRANCH INST. AND   <====
4103                                     :           REPLACE THE MOVE INSTRUCTION  <====
4104                                     :           WHICH FOLLOWS W/ 762          <====
```

```

4105 012106
4106 012106 012742 000271
4107 012112 005242
4108 012114 000000
4109 012116 022700 000400
4110 012122 001404
4111
4112
4113
4114
4115 012124 012742 000272
4116 012130 005242
4117 012132 000000
4118 012134 022737 100000 000000
4119 012142 001404
4120
4121
4122
4123
4124 012144 012742 000273
4125 012150 005242
4126 012152 000000
4127
4128
4129
4130
4131
4132 012154 005212
4133 012156 022712 000142
4134 012162 001033
4135 012164 005000
4136 012166 005010
4137 012170 052710 000001
4138 012174 005100
4139 012176 032760 000001 000001
4140 012204 001403
4141 012206 102402
4142 012210 103001
4143 012212 100004
4144
4145
4146
4147
4148 012214
4149 012214 012742 000274
4150 012220 005242
4151 012222 000000
4152 012224 022700 177777
4153 012230 001404
4154
4155
4156
4157
4158 012232 012742 000275
4159 012236 005242
4160 012240 000000

```

```

DNMSA: MOV #271,-(R2) ;MOVE TO MAILBOX # ***** 271 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;COND, CODES INCORRECT
DNMSB: CMP #400,R0 ;CHECK DEST. REGISTER
BEQ DNM5C ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====
; CONDITIONAL BRANCH INST. AND <=====
; REPLACE THE MOVE INSTRUCTION <=====
; WHICH FOLLOWS W/ 753 <=====
DNMSB: MOV #272,-(R2) ;MOVE TO MAILBOX # ***** 272 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DEST. REGISTER NOT DECREMENTED BY 2
DNMSC: CMP #100000,#0 ;CHECK DESTINATION DATA
BEQ TS142 ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====
; CONDITIONAL BRANCH INST. AND <=====
; REPLACE THE MOVE INSTRUCTION <=====
; WHICH FOLLOWS W/ 743 <=====
DNMSA: MOV #273,-(R2) ;MOVE TO MAILBOX # ***** 273 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DEST. DATA INCORRECTLY MODIFIED
; OR SEQUENCE ERROR
;*****
;TEST 142 TEST DEST. MODE 6 W/DOP NON-MODIFYING INST.
;*****
TS142: INC (R2) ;UPDATE TEST NUMBER
CMP #142,(R2) ;SEQUENCE ERROR?
BNE TS143-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC> 0=0
BIS #1,(R0) ;LOC. 0=1
COM R0 ;R0=-1 C-BIT=1
BIT #1,1(R0) ;TRY DOPNM W/MODE 6
BEQ DNM6A ;BR TO ERROR IF Z-BIT SET
BVS DNM6A ;BR TO ERROR IF V-BIT SET
BCC DNM6A ;BR TO ERROR IF C-BIT CLEAR
BPL DNM6B ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====
; CONDITIONAL BRANCH INST. AND <=====
; REPLACE THE MOVE INSTRUCTION <=====
; WHICH FOLLOWS W/ 763 <=====
DNM6A: MOV #274,-(R2) ;MOVE TO MAILBOX # ***** 274 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;COND CODES INCORRECT
DNM6B: CMP #-1,R0 ;CHECK DEST. REGISTER
BEQ DNM6C ;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====
; CONDITIONAL BRANCH INST. AND <=====
; REPLACE THE MOVE INSTRUCTION <=====
; WHICH FOLLOWS W/ 754 <=====
DNMSA: MOV #275,-(R2) ;MOVE TO MAILBOX # ***** 275 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DEST. REGISTER MODIFIED

```

```
4161 012242 022737 000001 000000 DNM6C:  CMP      #1, @#0      ;CHECK DEST. DATA
4162 012250 001404                BEQ      TS143
4163                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4164                                ;         CONDITIONAL BRANCH INST. AND <====
4165                                ;         REPLACE THE MOVE INSTRUCTION <====
4166                                ;         WHICH FOLLOWS W/ 744 <====
4167 012252 012742 000276                MOV      #276, -(R2) ;MOVE TO MAILBOX # ***** 276 *****
4168 012256 005242                INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4169 012260 000000                HALT
4170                                ;DEST. DATA MODIFIED
4171                                ; OR SEQUENCE ERROR
```

```
4172 :*****
4173 :TEST 143 TEST DEST MODE 7 W/DOP NON-MODIFYING INST.
4174 :*****
```

```
4175 012262 005212                TS143:  INC      (R2)      ;UPDATE TEST NUMBER
4176 012264 022712 000143                CMP      #143, (R2)  ;SEQUENCE ERROR?
4177 012270 001034                BNE     TS144-10    ;BR TO ERROR HALT ON SEQ ERROR
4178 012272 005000                CLR      R0        ;R0=0
4179 012274 005010                CLR      (R0)      ;LOC. 0=0 C-BIT=0
4180 012276 052710 125125                BIS     #125125, (R0) ;LOC. 0=125125
4181 012302 052700 000001                BIS     #1, R0     ;R0=1
4182 012306 132770 000125 000403        BITB    #125, @403(R0) ;TRY DOPNM W/MODE 7
4183 012314 102403                BVS     DNM7A      ;BR TO ERROR IF V-BIT SET
4184 012316 100402                BMI     DNM7A      ;BR TO ERROR IF N-BIT SET
4185 012320 103401                BCS     DNM7A      ;BR TO ERROR IF C-BIT SET
4186 012322 001404                BEQ     DNM7B
```

```
4187                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4188                                ;         CONDITIONAL BRANCH INST. AND <====
4189                                ;         REPLACE THE MOVE INSTRUCTION <====
4190                                ;         WHICH FOLLOWS W/ 762 <====
```

```
4191 012324                DNM7A:  MOV      #277, -(R2) ;MOVE TO MAILBOX # ***** 277 *****
4192 012324 012742 000277                INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4193 012330 005242                HALT
4194 012332 000000                DNM7B:  CMP      #1, R0   ;COND. CODES INCORRECT
4195 012334 022700 000001                BEQ     DNM7C      ;CHECK DEST. REGISTER
4196 012340 001404
```

```
4197                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4198                                ;         CONDITIONAL BRANCH INST. AND <====
4199                                ;         REPLACE THE MOVE INSTRUCTION <====
4200                                ;         WHICH FOLLOWS W/ 753 <====
```

```
4201 012342 012742 000300                MOV      #300, -(R2) ;MOVE TO MAILBOX # ***** 300 *****
4202 012346 005242                INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4203 012350 000000                HALT
4204 012352 022737 125125 000000        DNM7C:  CMP      #125125, @#0 ;DESTINATION REGISTER MODIFIED
4205 012360 001404                BEQ     TS144      ;CHECK DEST. DATA
```

```
4206                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4207                                ;         CONDITIONAL BRANCH INST. AND <====
4208                                ;         REPLACE THE MOVE INSTRUCTION <====
4209                                ;         WHICH FOLLOWS W/ 743 <====
```

```
4210 012362 012742 000301                MOV      #301, -(R2) ;MOVE TO MAILBOX # ***** 301 *****
4211 012366 005242                INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4212 012370 000000                HALT
4213                                ;DEST. DATA INCORRECT
4214                                ; OR SEQUENCE ERROR
```

```
4215 :*****
4216 :
```

4217  
4218  
4219  
4220  
4221  
4222  
4223  
4224 012372 005212  
4225 012374 022712 000144  
4226 012400 001016  
4227 012402 005000  
4228 012404 005010  
4229 012406 005100  
4230 012410 005004  
4231 012412 010014  
4232 012414 102402  
4233 012416 001401  
4234 012420 100404  
4235  
4236  
4237  
4238  
4239 012422  
4240 012422 012742 000302  
4241 012426 005242  
4242 012430 000000  
4243 012432 005704  
4244 012434 001404  
4245  
4246  
4247  
4248  
4249 012436 012742 000303  
4250 012442 005242  
4251 012444 000000  
4252  
4253  
4254  
4255  
4256  
4257  
4258  
4259  
4260  
4261  
4262  
4263 012446 005212  
4264 012450 022712 000145  
4265 012454 001026  
4266 012456 005000  
4267 012460 005001  
4268 012462 005010  
4269 012464 005110  
4270 012466 010120  
4271 012470 100402  
4272 012472 102401

THIS TEST VERIFIES THE MOV DESTINATION MODE 1 INSTRUCTION.  
DATA IS SET IN R0 USING SOP INSTRUCTIONS AND THEN MOVED TO LOC. 0  
USING MOV SRC MODE 0, DEST. MODE 1.

TEST 144 TEST MOV DESTINATION MODE 1

TS144: INC (R2) ;UPDATE TEST NUMBER  
CMP #144,(R2) ;SEQUENCE ERROR?  
BNE TS145-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC. 0=0  
COM R0 ;R0=-1  
CLR R4 ;R4 POINTS TO LOC. 0  
MOV R0,(R4) ;TRY MOVE MODE 0,1  
BVS MDM1A ;BR TO ERROR IF V SET  
BEQ MDM1A ;BR TO ERROR IF Z SET  
BMI MDM1B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 767 <====

MDM1A: MOV #302,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 302 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CONDITION CODE NOT CORRECT

MDM1B: TST R4  
BEQ TS145

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 761 <====

MOV #303,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 303 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DESTINATION REGISTER INCORRECTLY ALTERED  
; OR SEQUENCE ERROR

THIS TEST VERIFIES THE MOV DESTINATION MODE 2 INSTRUCTION.  
DATA IS SET IN R0 USING SOP INSTRUCTIONS AND THEN MOVED  
TO LOCATION 0 USING MOV SRC MODE 0, DEST. MODE 1.

TEST 145 TEST MOV DESTINATION MODE 2

TS145: INC (R2) ;UPDATE TEST NUMBER  
CMP #145,(R2) ;SEQUENCE ERROR?  
BNE TS146-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR R1 ;R1=0  
CLR (R0) ;LOC.0=0  
COM (R0) ;LOC. 0= 1  
MOV R1,(R0)+ ;TRY MOVE MODE 0,2  
BMI MDM2A ;BR TO ERROR IF N SET  
BVS MDM2A ;BR TO ERROR IF V SET

```
4273 012474 001404          BEQ      MDM2B          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4274                                     ;          CONDITIONAL BRANCH INST. AND <====
4275                                     ;          REPLACE THE MOVE INSTRUCTION <====
4276                                     ;          WHICH FOLLOWS W/ 767 <====
4277
4278 012476          MDM2A:
4279 012476 012742 000304      MOV      #304,-(R2)      ;MOVE TO MAILBOX # ***** 304 *****
4280 012502 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4281 012504 000000          HALT
4282 012506 005300          MDM2B: DEC      R0          ;CC'S INCORRECT
4283 012510 005300          DEC      R0
4284 012512 001404          BEQ      MDM2D
4285                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4286                                     ;          CONDITIONAL BRANCH INST. AND <====
4287                                     ;          REPLACE THE MOVE INSTRUCTION <====
4288                                     ;          WHICH FOLLOWS W/ 760 <====
4289
4289 012514          MDM2C:
4290 012514 012742 000305      MOV      #305,-(R2)      ;MOVE TO MAILBOX # ***** 305 *****
4291 012520 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4292 012522 000000          HALT
4293 012524 005737 000000      MDM2D: TST      @#0
4294 012530 001404          BEQ      TS146          ;DESTINATION REGISTER NOT INCREMENTED PROPERLY
4295                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4296                                     ;          CONDITIONAL BRANCH INST. AND <====
4297                                     ;          REPLACE THE MOVE INSTRUCTION <====
4298                                     ;          WHICH FOLLOWS W/ 751 <====
4299 012532 012742 000306      MOV      #306,-(R2)      ;MOVE TO MAILBOX # ***** 306 *****
4300 012536 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4301 012540 000000          HALT
4302                                     ;DESTINATION DATA INCORRECT
4303                                     ; OR SEQUENCE ERROR
4304
4305
4306
4307
4308
4309
4310
4311
4312 012542 005212          TS146: INC      (R2)          ;UPDATE TEST NUMBER
4313 012544 022712 000146      CMP      #146,(R2)      ;SEQUENCE ERROR?
4314 012550 001046          BNE     TS147-10        ;BR TO ERROR HALT ON SEQ ERROR
4315 012552 005000          CLR      R0            ;R0=0
4316 012554 005010          CLR      (R0)          ;LOC. 0=0
4317 012556 112720 000125      MOVVB   #125,(R0)+      ;TRY DESTINATION MODE 2 W/EVEN BYTE
4318 012562 102402          BVS     MBDM2A          ;BR TO ERROR IF V SET
4319 012564 001401          BEQ     MBDM2A          ;BR TO ERROR IF Z SET
4320 012566 100004          BPL     MBDM2B
4321                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4322                                     ;          CONDITIONAL BRANCH INST. AND <====
4323                                     ;          REPLACE THE MOVE INSTRUCTION <====
4324                                     ;          WHICH FOLLOWS W/ 770 <====
4325
4325 012570          MBDM2A:
4326 012570 012742 000307      MOV      #307,-(R2)      ;MOVE TO MAILBOX # ***** 307 *****
4327 012574 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4328 012576 000000          HALT          ;CC'S INCORRECT
```

```
.....
:
: THIS TEST VERIFIES DESTINATION MODE 2 W/MOVB INSTS. TWO DIFFERENT MOVB
: INSTRUCTIONS ARE USED TO MOVE A TEST PATTERN FIRST TO BYTE 0 THEN TO BYTE 1.
:
:.....
:TEST 146 TEST MOV-BYTE DESTINATION MODE 2
:.....
```

|      |        |        |               |         |      |             |  |  |       |
|------|--------|--------|---------------|---------|------|-------------|--|--|-------|
| 4329 | 012600 | 022700 | 000001        |         |      |             |  |  |       |
| 4330 | 012604 | 001404 |               | MBDM2B: | CMP  | #1,R0       |  |  |       |
| 4331 |        |        |               |         | BEQ  | MBDM2C      |  |  |       |
| 4332 |        |        |               |         |      |             |  | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 4333 |        |        |               |         |      |             |  | CONDITIONAL BRANCH INST. AND             | <==== |
| 4334 |        |        |               |         |      |             |  | REPLACE THE MOVE INSTRUCTION             | <==== |
| 4335 | 012606 | 012742 | 000310        |         | MOV  | #310,-(R2)  |  | WHICH FOLLOWS W/ 761                     | <==== |
| 4336 | 012612 | 005242 |               |         | INC  | -(R2)       |  | :MOVE TO MAILBOX # ***** 310 *****       |       |
| 4337 | 012614 | 000000 |               |         | HALT |             |  | :SET MSGTYP TO FATAL ERROR               |       |
| 4338 | 012616 | 112720 | 000252        | MBDM2C: | MOVB | #252,(R0)+  |  | :REGISTER NOT INCREMENTED BY ONE         |       |
| 4339 | 012622 | 102402 |               |         | BVS  | MBDM2D      |  | :TRY DESTINATION MODE 2 W/ODD BYTE       |       |
| 4340 | 012624 | 001401 |               |         | BEQ  | MBDM2D      |  |  |       |
| 4341 | 012626 | 100404 |               |         | BMI  | MBDM2E      |  |  |       |
| 4342 |        |        |               |         |      |             |  | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 4343 |        |        |               |         |      |             |  | CONDITIONAL BRANCH INST. AND             | <==== |
| 4344 |        |        |               |         |      |             |  | REPLACE THE MOVE INSTRUCTION             | <==== |
| 4345 |        |        |               |         |      |             |  | WHICH FOLLOWS W/ 750                     | <==== |
| 4346 | 012630 |        |               | MBDM2D: |      |             |  |  |       |
| 4347 | 012630 | 012742 | 000311        |         | MOV  | #311,-(R2)  |  | :MOVE TO MAILBOX # ***** 311 *****       |       |
| 4348 | 012634 | 005242 |               |         | INC  | -(R2)       |  | :SET MSGTYP TO FATAL ERROR               |       |
| 4349 | 012636 | 000000 |               |         | HALT |             |  | :CC'S NOT SET CORRECT                    |       |
| 4350 | 012640 | 022700 | 000002        | MBDM2E: | CMP  | #2,R0       |  |  |       |
| 4351 | 012644 | 001404 |               |         | BEQ  | MBDM2F      |  |  |       |
| 4352 |        |        |               |         |      |             |  | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 4353 |        |        |               |         |      |             |  | CONDITIONAL BRANCH INST. AND             | <==== |
| 4354 |        |        |               |         |      |             |  | REPLACE THE MOVE INSTRUCTION             | <==== |
| 4355 |        |        |               |         |      |             |  | WHICH FOLLOWS W/ 741                     | <==== |
| 4356 | 012646 | 012742 | 000312        |         | MOV  | #312,-(R2)  |  | :MOVE TO MAILBOX # ***** 312 *****       |       |
| 4357 | 012652 | 005242 |               |         | INC  | -(R2)       |  | :SET MSGTYP TO FATAL ERROR               |       |
| 4358 | 012654 | 000000 |               |         | HALT |             |  | :REGISTER NOT INCREMENTED BY ONE         |       |
| 4359 | 012656 | 022737 | 125125 000000 | MBDM2F: | CMP  | #125125,@#0 |  | :CHECK DATA                              |       |
| 4360 | 012664 | 001404 |               |         | BEQ  | TS147       |  |  |       |
| 4361 |        |        |               |         |      |             |  | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 4362 |        |        |               |         |      |             |  | CONDITIONAL BRANCH INST. AND             | <==== |
| 4363 |        |        |               |         |      |             |  | REPLACE THE MOVE INSTRUCTION             | <==== |
| 4364 |        |        |               |         |      |             |  | WHICH FOLLOWS W/ 731                     | <==== |
| 4365 | 012666 | 012742 | 000313        |         | MOV  | #313,-(R2)  |  | :MOVE TO MAILBOX # ***** 313 *****       |       |
| 4366 | 012672 | 005242 |               |         | INC  | -(R2)       |  | :SET MSGTYP TO FATAL ERROR               |       |
| 4367 | 012674 | 000000 |               |         | HALT |             |  | :DESTINATION DATA INCORRECT              |       |
| 4368 |        |        |               |         |      |             |  | : OR SEQUENCE ERROR                      |       |

4370  
4371  
4372  
4373  
4374  
4375  
4376  
4377  
4378  
4379  
4380  
4381  
4382  
4383  
4384

```
*****  
: THIS TEST VERIFIES MOV DESTINATION MODE 3. R0 IS USED TO PICK UP  
: AN ADDRESS AT LOC. 400. LOC 400 POINTS TO LOC. 0 THE EFFECTIVE DEST. ADDR.. ALSO, MOVB  
: INST. ARE USED W/ EVEN AND ODD BYTES TO CHECK MOV BYTES INST AND MODE 37 DESTINATIONS.  
:*****  
:TEST 147 TEST MOV(B) DESTINATION MODE 3  
:*****  
TS147: INC (R2) ;UPDATE TEST NUMBER  
CMP #147,(R2) ;SEQUENCE ERROR?  
BNE TS150-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #400,R0 ;R0=400  
CLR (R0) ;LOC. 400 POINTS TO LOC. 0  
CLR @#0 ;LOC. 0=0  
MOV #125252,@(R0)+ ;TRY MOV DESTINATION MODE 2
```

```
4385 012724 102402      BVS      MDM3A      ;BR TO ERROR IF V SET
4386 012726 001401      BEQ      MDM3A      ;BR TO ERROR IF Z SET
4387 012730 100404      BMI      MDM3B
4388
4389                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4390                      ;          CONDITIONAL BRANCH INST. AND <====
4391                      ;          REPLACE THE MOVE INSTRUCTION <====
4392                      ;          WHICH FOLLOWS W/ 765 <====
4392 012732      MDM3A:
4393 012732 012742 000314      MOV      #314,-(R2)  ;MOVE TO MAILBOX # ***** 314 *****
4394 012736 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4395 012740 000000      HALT
4396 012742 022700 000402      MDM3B:  CMP      #402,R0  ;CC'S INCORRECT
4397 012746 001404      BEQ      MDM3C      ;CHECK DEST. MODE REGISTER
4398
4399                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4400                      ;          CONDITIONAL BRANCH INST. AND <====
4401                      ;          REPLACE THE MOVE INSTRUCTION <====
4402                      ;          WHICH FOLLOWS W/ 756 <====
4402 012750 012742 000315      MOV      #315,-(R2)  ;MOVE TO MAILBOX # ***** 315 *****
4403 012754 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4404 012756 000000      HALT
4405 012760 022737 125252 000000  MDM3C:  CMP      #125252,@#0 ;REGISTER NOT INCREMENTED BY 2
4406 012766 001404      BEQ      MDM3D      ;CHECK DESTINATION DATA
4407
4408                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4409                      ;          CONDITIONAL BRANCH INST. AND <====
4410                      ;          REPLACE THE MOVE INSTRUCTION <====
4411                      ;          WHICH FOLLOWS W/ 746 <====
4411 012770 012742 000316      MOV      #316,-(R2)  ;MOVE TO MAILBOX # ***** 316 *****
4412 012774 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4413 012776 000000      HALT
4414 013000 112737 000125 000000  MDM3D:  MOVB    #125,@#0    ;DESTINATION DATA INCORRECT
4415 013006 022737 125125 000000  CMP      #125125,@#0 ;TRY MOV(B) DESTINATION MODE Z EVEN BYTE
4416 013014 001404      BEQ      MDM3E      ;CHECK DATA
4417
4418                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4419                      ;          CONDITIONAL BRANCH INST. AND <====
4420                      ;          REPLACE THE MOVE INSTRUCTION <====
4421                      ;          WHICH FOLLOWS W/ 733 <====
4421 013016 012742 000317      MOV      #317,-(R2)  ;MOVE TO MAILBOX # ***** 317 *****
4422 013022 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4423 013024 000000      HALT
4424 013026 112737 000525 000001  MDM3E:  MOVB    #525,@#1    ;DESTINATION DATA INCORRECT
4425 013034 022737 052525 000000  CMP      #52525,@#0 ;TRY MOV(B) DESTINATION MODE 2 ODD BYTE
4426 013042 001404      BEQ      TS150      ;CHECK DATA
4427
4428                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4429                      ;          CONDITIONAL BRANCH INST. AND <====
4430                      ;          REPLACE THE MOVE INSTRUCTION <====
4431                      ;          WHICH FOLLOWS W/ 720 <====
4431 013044 012742 000320      MOV      #320,-(R2)  ;MOVE TO MAILBOX # ***** 320 *****
4432 013050 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4433 013052 000000      HALT
4434
4435                      ; *****
4436                      ;
4437                      ; THIS TEST VERIFIES THE MOV DESTINATION MODE 4 INSTRUCTION.
4438                      ; SOP INSTRUCTIONS ON R0 ARE USED TO CLEAR TARGET LOCATION 0.
4439                      ; R4 IS USED AS THE MODE 4 ADDRESSING REGISTER, AND
4440                      ; CONDITIONAL BRANCHES ARE USED TO VERIFY THE DATA.
```



```
4441
4442
4443
4444
4445 013054 005212
4446 013056 022712 000150
4447 013062 001026
4448 013064 005000
4449 013066 005010
4450 013070 012704 000002
4451 013074 012744 012345
4452 013100 102402
4453 013102 001401
4454 013104 100004
4455
4456
4457
4458
4459 013106
4460 013106 012742 000321
4461 013112 005242
4462 013114 000000
4463 013116 005704
4464 013120 001404
4465
4466
4467
4468
4469 013122 012742 000322
4470 013126 005242
4471 013130 000000
4472 013132 022710 012345
4473 013136 001404
4474
4475
4476
4477
4478 013140 012742 000323
4479 013144 005242
4480 013146 000000
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494 013150 005212
4495 013152 022712 000151
4496 013156 001046
```

.....  
:TEST 150 TEST MOV DESTINATION MODE 4  
:.....  
TS150: INC (R2) ;UPDATE TEST NUMBER  
CMP #150,(R2) ;SEQUENCE ERROR?  
BNE TS151-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC 0=0  
MOV #2,R4 ;R4=2  
MOV #12345,-(R4) ;TRY MOV DEST. MODE 4  
BVS MDM4A ;BR TO ERROR IF V-BIT SET  
BEQ MDM4A ;BR TO ERROR IF Z-BIT SET  
BPL MDM4B

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 766 <====

MDM4A: MOV #321,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 321 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S NOT CORRECT

MDM4B: TST R4 ;CHECK DECREMENTING OF MODE 4 REG.  
BEQ MDM4C

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 760 <====

MDM4C: MOV #322,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 322 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DESTINATION MODE REGISTER NOT DECREMENTED BY 2  
CMP #12345,(R0) ;CHECK DESTINATION DATA  
BEQ TS151

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 751 <====

MOV #323,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 323 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DESTINATION DATA INCORRECT  
; OR SEQUENCE ERROR

.....  
: THIS TEST VERIFIES THE MOV(B) DESTINATION MODE 4 INSTRUCTION  
: ON BOTH ODD AND EVEN BYTES. SOP INSTRUCTIONS ON R4 ARE  
: USED TO CLEAR TARGET LOCATION 0. R0 IS USED AS THE MODE 4  
: ADDRESSING REGISTER, AND CMP AND CONDITIONAL BRANCH  
: INSTRUCTIONS ARE USED TO VERIFY THE DATA.  
:.....

:TEST 151 TEST MOV(B) DESTINATION MODE 4  
:.....  
TS151: INC (R2) ;UPDATE TEST NUMBER  
CMP #151,(R2) ;SEQUENCE ERROR?  
BNE TS152-10 ;BR TO ERROR HALT ON SEQ ERROR

|      |        |        |        |              |               |  |       |
|------|--------|--------|--------|--------------|---------------|--|-------|
| 4497 | 013160 | 005004 |        | CLR          | R4            | :R4=0                                    |       |
| 4498 | 013162 | 005014 |        | CLR          | (R4)          | :LOC. 0=0                                |       |
| 4499 | 013164 | 012700 | 000002 | MOV          | #2,R0         | :R0 = 2                                  |       |
| 4500 | 013170 | 112740 | 125125 | MOVB         | #125125,-(R0) | :TRY MOV B DEST. MODE 4-ODD BYTE         |       |
| 4501 | 013174 | 020027 | 000001 | CMP          | R0,#1         | :CHECK THAT DEST. REG. WAS DECREMENTED   |       |
| 4502 | 013200 | 001404 |        | BEQ          | MBDM4A        |  |       |
| 4503 |        |        |        |              |               | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 4504 |        |        |        |              |               | : CONDITIONAL BRANCH INST. AND           | <==== |
| 4505 |        |        |        |              |               | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 4506 |        |        |        |              |               | : WHICH FOLLOWS W/ 766                   | <==== |
| 4507 | 013202 | 012742 | 000324 | MOV          | #324,-(R2)    | :MOVE TO MAILBOX # ***** 324 *****       |       |
| 4508 | 013206 | 005242 |        | INC          | -(R2)         | :SET MSGTYP TO FATAL ERROR               |       |
| 4509 | 013210 | 000000 |        | HALT         |               | :DESTINATION REG. NOT DECREMENTED BY 1   |       |
| 4510 | 013212 | 021427 | 052400 | MBDM4A: CMP  | (R4),#52400   | :CHECK DEST. DATA                        |       |
| 4511 | 013216 | 001404 |        | BEQ          | MBDM4B        |  |       |
| 4512 |        |        |        |              |               | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 4513 |        |        |        |              |               | : CONDITIONAL BRANCH INST. AND           | <==== |
| 4514 |        |        |        |              |               | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 4515 |        |        |        |              |               | : WHICH FOLLOWS W/ 757                   | <==== |
| 4516 | 013220 | 012742 | 000325 | MOV          | #325,-(R2)    | :MOVE TO MAILBOX # ***** 325 *****       |       |
| 4517 | 013224 | 005242 |        | INC          | -(R2)         | :SET MSGTYP TO FATAL ERROR               |       |
| 4518 | 013226 | 000000 |        | HALT         |               | :DEST. DATA NOT CORRECT                  |       |
| 4519 | 013230 | 112740 | 125125 | MBDM4B: MOVB | #125125,-(R0) | :TRY MOV B DEST. MODE 4--EVEN BYTE       |       |
| 4520 | 013234 | 102402 |        | BVS          | MBDM4C        | :BR. TO ERROR IF V-BIT SET               |       |
| 4521 | 013236 | 001401 |        | BEQ          | MBDM4C        | :BR TO ERROR IF Z-BIT SET                |       |
| 4522 | 013240 | 100004 |        | BPL          | MBDM4D        |  |       |
| 4523 |        |        |        |              |               | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 4524 |        |        |        |              |               | : CONDITIONAL BRANCH INST. AND           | <==== |
| 4525 |        |        |        |              |               | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 4526 |        |        |        |              |               | : WHICH FOLLOWS W/ 746                   | <==== |
| 4527 | 013242 |        |        | MBDM4C:      |               |  |       |
| 4528 | 013242 | 012742 | 000326 | MOV          | #326,-(R2)    | :MOVE TO MAILBOX # ***** 326 *****       |       |
| 4529 | 013246 | 005242 |        | INC          | -(R2)         | :SET MSGTYP TO FATAL ERROR               |       |
| 4530 | 013250 | 000000 |        | HALT         |               | :COND. CODES INCORRECT                   |       |
| 4531 | 013252 | 005700 |        | MBDM4D: TST  | R0            | :CHECK MODE 4 DEST. REGISTER             |       |
| 4532 | 013254 | 001404 |        | BEQ          | MBDM4E        |  |       |
| 4533 |        |        |        |              |               | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 4534 |        |        |        |              |               | : CONDITIONAL BRANCH INST. AND           | <==== |
| 4535 |        |        |        |              |               | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 4536 |        |        |        |              |               | : WHICH FOLLOWS W/ 740                   | <==== |
| 4537 | 013256 | 012742 | 000327 | MOV          | #327,-(R2)    | :MOVE TO MAILBOX # ***** 327 *****       |       |
| 4538 | 013262 | 005242 |        | INC          | -(R2)         | :SET MSGTYP TO FATAL ERROR               |       |
| 4539 | 013264 | 000000 |        | HALT         |               | :DESTINATION REG NOT DECREMENTED BY 1    |       |
| 4540 | 013266 | 021427 | 052525 | MBDM4E: CMP  | (R4),#52525   | :CHECK DEST. DATA                        |       |
| 4541 | 013272 | 001404 |        | BEQ          | TS152         |  |       |
| 4542 |        |        |        |              |               | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 4543 |        |        |        |              |               | : CONDITIONAL BRANCH INST. AND           | <==== |
| 4544 |        |        |        |              |               | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 4545 |        |        |        |              |               | : WHICH FOLLOWS W/ 731                   | <==== |
| 4546 | 013274 | 012742 | 000330 | MOV          | #330,-(R2)    | :MOVE TO MAILBOX # ***** 330 *****       |       |
| 4547 | 013300 | 005242 |        | INC          | -(R2)         | :SET MSGTYP TO FATAL ERROR               |       |
| 4548 | 013302 | 000000 |        | HALT         |               | :DESTINATION DATA INCORRECT              |       |
| 4549 |        |        |        |              |               | : OR SEQUENCE ERROR                      |       |
| 4550 |        |        |        |              |               |  |       |
| 4551 |        |        |        |              |               |  |       |
| 4552 |        |        |        |              |               |  |       |

4553  
4554  
4555  
4556  
4557  
4558  
4559  
4560  
4561  
4562  
4563 013304 005212  
4564 013306 022712 000152  
4565 013312 001051  
4566 013314 005004  
4567 013316 005014  
4568 013320 012700 000400  
4569 013324 012750 004321  
4570 013330 102402  
4571 013332 001401  
4572 013334 100004  
4573  
4574  
4575  
4576  
4577 013336  
4578 013336 012742 000331  
4579 013342 005242  
4580 013344 000000  
4581 013346 022700 000376  
4582 013352 001404  
4583  
4584  
4585  
4586  
4587 013354 012742 000332  
4588 013360 005242  
4589 013362 000000  
4590 013364 022714 004321  
4591 013370 001404  
4592  
4593  
4594  
4595  
4596 013372 012742 000333  
4597 013376 005242  
4598 013400 000000  
4599 013402 012700 000406  
4600 013406 112750 000377  
4601 013412 022700 000404  
4602 013416 001404  
4603  
4604  
4605  
4606  
4607 013420 012742 000334  
4608 013424 005242

THIS TEST VERIFIES THE MOV DESTINATION MODE 5 AND THE MOVB  
DESTINATION MODE 5 - EVEN BYTE INSTRUCTIONS. R4 IS A  
POINTER TO TARGET LOCATION 0 AND R0 IS SETUP TO  
POINT TO LOCATION 376 FOR THE MOV, AND LOCATION 404 FOR  
THE MOVB INSTRUCTIONS. CMP INSTRUCTIONS ARE USED TO VERIFY  
PROPER ADDRESSING AND DATA.

\*\*\*\*\*  
TEST 152 TEST MOV DESTINATION MODE 5  
\*\*\*\*\*

TS152: INC (R2) ;UPDATE TEST NUMBER  
CMP #152,(R2) ;SEQUENCE ERROR?  
BNE TS153-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R4 ;R4=0  
CLR (R4) ;LOC. 0 = 0  
MOV #400,R0 ;R0=400  
MOV #4321,@-(R0) ;TRY MOV DEST. MODE 5  
BVS MDM5A ;BR TO ERROR IF V-BIT SET  
BEQ MDM5A ;BR TO ERROR IF Z-BIT SET  
BPL MDM5B

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
CONDITIONAL BRANCH INST. AND <====  
REPLACE THE MOVE INSTRUCTION <====  
WHICH FOLLOWS W/ 766 <====

MDM5A: MOV #331,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 331 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;COND. CODES INCORRECT

MDM5B: CMP #376,R0 ;CHECK MODE 5 REG. WAS DECREMENTED  
BEQ MDM5C

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
CONDITIONAL BRANCH INST. AND <====  
REPLACE THE MOVE INSTRUCTION <====  
WHICH FOLLOWS W/ 757 <====

MDM5C: MOV #332,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 332 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;MODE 5 REGISTER NOT DECREMENTED BY 2  
CMP #4321,(R4) ;CHECK DEST. DATA  
BEQ MDM5D

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
CONDITIONAL BRANCH INST. AND <====  
REPLACE THE MOVE INSTRUCTION <====  
WHICH FOLLOWS W/ 750 <====

MDM5D: MOV #333,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 333 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEST. DATA INCORRECT  
MOV #406,R0 ;R0=406  
MOVB #377,@-(R0) ;TRY MOV DEST. MODE 5 --EVEN BYTE  
CMP #404,R0 ;CHECK MODE 5 REG.  
BEQ MDM5E

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
CONDITIONAL BRANCH INST. AND <====  
REPLACE THE MOVE INSTRUCTION <====  
WHICH FOLLOWS W/ 735 <====

MOV #334,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 334 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR

4609 013426 000000  
4610 013430 022714 177721  
4611 013434 001404  
4612  
4613  
4614  
4615  
4616 013436 012742 000335  
4617 013442 005242  
4618 013444 000000  
4619  
4620  
4621  
4622  
4623  
4624  
4625  
4626  
4627  
4628  
4629  
4630  
4631  
4632 013446 005212  
4633 013450 022712 000153  
4634 013454 001054  
4635 013456 005000  
4636 013460 005010  
4637 013462 005200  
4638 013464 012760 052525 177777  
4639 013472 102402  
4640 013474 001401  
4641 013476 100004  
4642  
4643  
4644  
4645  
4646 013500  
4647 013500 012742 000336  
4648 013504 005242  
4649 013506 000000  
4650 013510 022700 000001  
4651 013514 001404  
4652  
4653  
4654  
4655  
4656 013516 012742 000337  
4657 013522 005242  
4658 013524 000000  
4659 013526 022737 052525 000000  
4660 013534 001404  
4661  
4662  
4663  
4664

MDM5E: HALT ;MODE 5 REGISTER NOT DECREMENTED BY 2  
CMP #177721,(R4) ;CHECK DEST. DATA  
BEQ TS153  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 726 <====  
MOV #335,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 335 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEST. DATA INCORRECT  
; OR SEQUENCE ERROR

.....  
: THIS TEST VERIFIES THE MOV DESTINATION MODE 6 AND MOVB - EVEN BYTE  
: DESTINATION MODE 6 INSTRUCTIONS. R0 IS USED TO SETUP TARGET LOC.0  
: FOR BOTH TESTS. PATTERNS OF ONES AND ZEROES ARE MOVED INTO LOC.0  
: BY MODE 6 INSTRUCTIONS, AND CMP INSTRUCTIONS ARE USED TO VERIFY  
: PROPER ADDRESSING AND DATA.  
:.....

:TEST 153 TEST MOV DESTINATION MODE 6  
:.....

TS153: INC (R2) ;UPDATE TEST NUMBER  
CMP #153,(R2) ;SEQUENCE ERROR?  
BNE TS154-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC. 0=0  
INC R0 ;R0=1  
MOV #052525,-1(R0) ;TRY MOV DEST. MODE 6  
BVS MDM6A ;BR TO ERROR IF V-BIT SET  
BEQ MDM6A ;BR TO ERROR IF Z-BIT SET  
BPL MDM6B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 766 <====

MDM6A: MOV #336,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 336 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;COND. CODES INCORRECT  
MDM6B: CMP #1,R0 ;CHECK DEST. REGISTER UNALTERED  
BEQ MDM6C

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 757 <====

MDM6C: MOV #337,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 337 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEST. REGISTER INCORRECTLY ALTERED  
MDM6C: CMP #52525,#0 ;CHECK DEST. DATA  
BEQ MDM6D

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 747 <====

```
4665 013536 012742 000340      MOV      #340,-(R2)      ;MOVE TO MAILBOX # ***** 340 *****
4666 013542 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4667 013544 000000              HALT                    ;DEST. DATA INCORRECT
4668 013546 012700 000002      MDM6D: MOV      #2,R0      ;R0=2
4669 013552 112760 000377 177777  MOVB     #377,-1(R0)    ;TRY MOVB DEST. MODE 6
4670 013560 022700 000002      CMP      #2,R0         ;CHECK DEST. REGISTER UNALTERED
4671 013564 001404              BEQ      MDM6E          ;
4672                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4673                                ;         CONDITIONAL BRANCH INST. AND <====
4674                                ;         REPLACE THE MOVE INSTRUCTION <====
4675                                ;         WHICH FOLLOWS W/ 733 <====
4676 013566 012742 000341      MOV      #341,-(R2)    ;MOVE TO MAILBOX # ***** 341 *****
4677 013572 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4678 013574 000000              HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
4679 013576 022737 177525 000000  MDM6E: CMP      #177525,#0 ;CHECK DEST. DATA
4680 013604 001404              BEQ      TS154          ;
4681                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4682                                ;         CONDITIONAL BRANCH INST. AND <====
4683                                ;         REPLACE THE MOVE INSTRUCTION <====
4684                                ;         WHICH FOLLOWS W/ 723 <====
4685 013606 012742 000342      MOV      #342,-(R2)    ;MOVE TO MAILBOX # ***** 342 *****
4686 013612 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4687 013614 000000              HALT                    ;DEST. DATA INCORRECT
4688                                ; OR SEQUENCE ERROR
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700 013616 005212              ;*****
4701 013620 022712 000154      ;THIS TEST VERIFIES THE MOV DESTINATION MODE 7 AND MOVB - ODD BYTE
4702 013624 001053              ;DESTINATION MODE 7 INSTRUCTIONS. R4 POINTS TO TARGET LOC.0 AND R0
4703 013626 005004              ;IS USED AS THE MODE 7 ADDRESSING REGISTER. CMP INSTRUCTIONS ARE
4704 013630 005014              ;USED TO VERIFY PROPER ADDRESSING AND DATA.
4705 013632 012700 000403      ;*****
4706 013636 012770 070707 177777  ;TEST 154      TEST MOV DESTINATION MODE 7
4707 013644 102402              ;*****
4708 013646 001401              TS154: INC      (R2)      ;UPDATE TEST NUMBER
4709 013650 100004              CMP      #154,(R2)     ;SEQUENCE ERROR?
4710                                BNE     TS155-10      ;BR TO ERROR HALT ON SEQ ERROR
4711                                CLR      R4           ;R4=0
4712                                CLR      (R4)         ;LOC.0=0
4713                                MOV      #403,R0       ;R0=403
4714                                MOV      #70707,#-1(R0) ;TRY MOV W/DEST MODE 7
4715                                BVS     MDM7A         ;BR. TO ERROR IF V-BIT SET
4716                                BEQ     MDM7A         ;BR TO ERROR IF Z-BIT SET
4717                                BPL     MDM7B         ;
4718                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4719                                ;         CONDITIONAL BRANCH INST. AND <====
4720                                ;         REPLACE THE MOVE INSTRUCTION <====
4721                                ;         WHICH FOLLOWS W/ 765 <====
4722                                MDM7A: MOV      #343,-(R2)    ;MOVE TO MAILBOX # ***** 343 *****
4723                                INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4724                                HALT                    ;COND. CODES INCORRECT
4725                                MDM7B: CMP      #403,R0       ;CHECK DEST. REGISTER
4726                                BEQ     MDM7C         ;
4727                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
```

```
4721 :                               :                               :<====<br>4722 :                               :                               :<====<br>4723 :                               :                               :<====<br>4724 013670 012742 000344      MOV   #344,-(R2)      :MOVE TO MAILBOX # ***** 344 *****<br>4725 013674 005242              INC   -(R2)          :SET MSGTYP TO FATAL ERROR<br>4726 013676 000000              HALT                    :DEST. REGISTER INCORRECTLY ALTERED<br>4727 013700 022737 070707 000000 MDM7C: CMP   #70707,@#0      :CHECK DEST. DATA<br>4728 013706 001404              BEQ   MDM7D          :<br>4729 :                               : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====<br>4730 :                               : CONDITIONAL BRANCH INST. AND <====<br>4731 :                               : REPLACE THE MOVE INSTRUCTION <====<br>4732 :                               : WHICH FOLLOWS W/ 756 <====<br>4733 013710 012742 000345      MOV   #345,-(R2)      :MOVE TO MAILBOX # ***** 345 *****<br>4734 013714 005242              INC   -(R2)          :SET MSGTYP TO FATAL ERROR<br>4735 013716 000000              HALT                    :DEST. DATA INCORRECT<br>4736 013720 112770 107070 000001 MDM7D: MOVB  #107070,@1(R0)   :TRY MOVB W/DEST MODE 7--ODD BYTE<br>4737 013726 022700 000403      CMP   #403,R0        :CHECK MODE 7 DEST. REG.<br>4738 013732 001404              BEQ   MDM7E          :<br>4739 :                               : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS / <====<br>4740 :                               : CONDITIONAL BRANCH INST. AND <====<br>4741 :                               : REPLACE THE MOVE INSTRUCTION <====<br>4742 :                               : WHICH FOLLOWS W/ 746 <====<br>4743 013734 012742 000346      MOV   #346,-(R2)      :MOVE TO MAILBOX # ***** 346 *****<br>4744 013740 005242              INC   -(R2)          :SET MSGTYP TO FATAL ERROR<br>4745 013742 000000              HALT                    :DEST. DATA INCORRECT<br>4746 013744 022737 034307 000000 MDM7E: CMP   #34307,@#0      :CHECK DEST. DATA<br>4747 013752 001404              BEQ   TS155          :<br>4748 :                               : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====<br>4749 :                               : CONDITIONAL BRANCH INST. AND <====<br>4750 :                               : REPLACE THE MOVE INSTRUCTION <====<br>4751 :                               : WHICH FOLLOWS W/ 734 <====<br>4752 013754 012742 000347      MOV   #347,-(R2)      :MOVE TO MAILBOX # ***** 347 *****<br>4753 013760 005242              INC   -(R2)          :SET MSGTYP TO FATAL ERROR<br>4754 013762 000000              HALT                    :DESTINATION DATA INCORRECT<br>4755 :                               : OR SEQUENCE ERROR<br>4756 :<br>4757 :<br>4758 :<br>4759 :<br>4760 :<br>4761 :<br>4762 :<br>4763 :<br>4764 :<br>4765 :<br>4766 :<br>4767 :<br>4768 :<br>4769 :<br>4770 :<br>4771 :<br>4772 013764 005212              INC   (R2)          :UPDATE TEST NUMBER<br>4773 013766 022712 000155      CMP   #155,(R2)      :SEQUENCE ERROR?<br>4774 013772 001015              BNE  DOP4            :BR TO ERROR HALT ON SEQ ERROR<br>4775 013774 012700 014046      MOV   #TBL1,R0        :INITIALIZE R0<br>4776 014000 014037 014046      MOV   -(R0),@#TBL1   :TBL1=125252
```

```
*****<br>: THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.<br>: THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A<br>: TABLE OF OPERANDS. THE TABLE OF OPERANDS AND THE WORK LOCATION IS<br>: STORED FOLLOWING THE TEST CODE. A SERIES OF 5 DOP INSTRUCTIONS UTILIZES<br>: THE DATA IN THE TABLE TO CYCLE THE WORK LOCATION THRU A SET OF<br>: VALUE. THE DATA HAS BEEN CHOSEN TO INSURE THAT NO SINGLE ERROR WILL<br>: GO UNDETECTED. WORD AND BYTE INSTRUCTION ACCESSING BOTH EVEN AND<br>: ODD ADDRESSES ARE USED IN THE TEST. THE LISTING SHOWS THE<br>: EXPECTED INTERMEDIATE RESULT AS EACH INSTRUCTION IS EXECUTED.<br>*****
```

```
*****<br>: TEST 155 TEST MODE 4 W/ DOP INSTS.<br>*****
```

4777 014004 064037 014046  
4778 014010 144037 014046  
4779 014014 154037 014047  
4780 014020 024037 014046  
4781 014024 001411  
4782  
4783  
4784  
4785  
4786 014026  
4787 014026 012742 000350  
4788 014032 005242  
4789 014034 000000  
4790  
4791  
4792 014036 125252  
4793 014040 052652  
4794 014042 053125  
4795 014044 125252  
4796 014046 000000  
4797  
4798  
4799  
4800  
4801  
4802  
4803  
4804  
4805  
4806  
4807  
4808  
4809  
4810 014050 005212  
4811 014052 022712 000156  
4812 014056 001015  
4813 014060 012700 014134  
4814 014064 015037 014046  
4815 014070 065037 014046  
4816 014074 145037 014046  
4817 014100 155037 014047  
4818 014104 025037 014046  
4819 014110 001411  
4820  
4821  
4822  
4823  
4824 014112  
4825 014112 012742 000351  
4826 014116 005242  
4827 014120 000000  
4828  
4829 014122 014036  
4830 014124 014040  
4831 014126 014041  
4832 014130 014042

ADD -(R0),@#TBL1 ;TBL1=000377  
BICB -(R0),@#TBL1 ;TBL1=000252  
BISB -(R0),@#TBL1+1 ;TBL1=125252  
CMP -(R0),@#TBL1 ;CHECK RESULT  
BEQ TS156  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 762 <====  
DOP4: MOV #350,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 350 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF MODE 4 INSTS. INCORRECT  
; OR SEQUENCE ERROR

125252  
52652  
53125  
125252  
TBL1: 0

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 5 DOUBLE OPERAND INSTRUCTIONS.  
: THE TEST USES AN ADDRESS TABLE STORED FOLLOWING THE TEST CODE.  
: THIS TABLE IS SIMPLY A TABLE OF ADDRESS POINTERS WHICH ADDRESS  
: THE DATA TABLE USED IN THE PREVIOUS TEST. THE TEST IS IDENTICAL TO  
: THE PREVIOUS TEST EXCEPT THE DATA IS REFERENCED USING THIS ADDRESS  
: TABLE AND MODE 5 ADDRESSING. (SEE PREVIOUS TEST).  
: \*\*\*\*\*

TEST 156 TEST MODE 5 W/ DOP INSTS.  
\*\*\*\*\*

TS156: INC (R2) ;UPDATE TEST NUMBER  
CMP #156,(R2) ;SEQUENCE ERROR?  
BNE DOP5 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #TBL2+2,R0 ;INITIALIZE R0  
MOV @-(R0),@#TBL1 ;TBL1=125252  
ADD @-(R0),@#TBL1 ;TBL1=000377  
BICB @-(R0),@#TBL1 ;TBL1=000252  
BISB @-(R0),@#TBL1+1 ;TBL1=125252  
CMP @-(R0),@#TBL1 ;CHECK RESULT  
BEQ TS157  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 762 <====

DOP5: MOV #351,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 351 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF MODE 5 INSTS. INCORRECT  
; OR SEQUENCE ERROR  
TBL1-10  
TBL1-6  
TBL1-5  
TBL1-4

MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79<sup>D 8</sup> 11:31 PAGE 94  
T156 TEST MODE 5 W/ DOP INSTS.

SEQ 0094

4833 014132 014044

TBL2: TBL1-2



4834  
4835  
4836  
4837  
4838  
4839  
4840  
4841  
4842  
4843  
4844  
4845  
4846  
4847  
4848  
4849  
4850  
4851  
4852  
4853  
4854  
4855  
4856  
4857  
4858  
4859  
4860  
4861  
4862  
4863  
4864  
4865  
4866  
4867  
4868  
4869  
4870  
4871  
4872  
4873  
4874  
4875  
4876  
4877  
4878  
4879  
4880  
4881  
4882  
4883  
4884  
4885  
4886  
4887  
4888  
4889

014134 005212  
014136 022712 000157  
014142 001022  
014144 012700 014042  
014150 016037 000002 014046  
014156 066037 000000 014046  
014164 146037 177777 014046  
014172 156037 177776 014047  
014200 026037 177774 014046  
014206 001404  
  
014210 012742 000352  
014214 005242  
014216 000000  
  
014220 005212  
014222 022712 000160  
014226 001022  
014230 012700 014126  
014234 017037 000004 014046  
014242 067037 000002 014046  
014250 147037 000000 014046  
014256 157037 177776 014047  
014264 027037 177774 014046  
014272 001404

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 6 DOUBLE OPERAND INSTRUCTIONS.  
: IT USES THE SAME DATA AS THAT USED IN THE MODE 4 TESTS.  
: THIS TIME THE DATA IS ACCESSED USING MODE 6. R0 IS SET  
: TO POINT TO THE MIDDLE OF THE TABLE. THE TABLE IS ACCESSED FROM  
: BOTTOM TO TOP BY VARYING THE OFFSET IN THE MODE 6 INSTRUCTIONS.  
: THE DATA RESULTS ARE IDENTICAL TO THOSE EXPECTED IN THE MODE 4  
: TESTS.  
\*\*\*\*\*

\*\*\*\*\*  
: TEST 157 TEST MODE 6 W/ DOP INSTS.  
\*\*\*\*\*

TS157: INC (R2) ;UPDATE TEST NUMBER  
CMP #157,(R2) ;SEQUENCE ERROR?  
BNE TS160-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #TBL1-4,R0 ;INITIALIZE R0  
MOV 2(R0),@#TBL1 ;TBL1=125252  
ADD 0(R0),@#TBL1 ;TBL1=000377  
BICB -1(R0),@#TBL1 ;TBL1=000252  
BISB -2(R0),@#TBL1+1 ;TBL1=125252  
CMP -4(R0),@#TBL1 ;CHECK RESULT  
BEQ TS160  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 755 <====  
MOV #352,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 352 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF MODE 6 INSTS. INCORRECT  
; OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 7 DOUBLE OPERAND INSTRUCTIONS.  
: THIS TEST USES THE SAME ADDRESS TABLE AND DATA TABLE USED BY  
: THE MODE 5 TESTS. THIS TIME THE DATA IS ACCESSED USING MODE 7.  
: R0 IS SET TO POINT TO THE MIDDLE OF THE ADDRESS TABLE IN THE MODE 5  
: TEST. THE TABLE IS ACCESSED FROM BOTTOM TO TOP BY VARYING THE OFFSET  
: IN THE MODE 7 INSTRUCTIONS. THE DATA RESULTS ARE IDENTICAL TO  
: THOSE EXPECTED IN THE MODE 5 TESTS.  
\*\*\*\*\*

\*\*\*\*\*  
: TEST 160 TEST MODE 7 W/ DOP INSTS.  
\*\*\*\*\*

TS160: INC (R2) ;UPDATE TEST NUMBER  
CMP #160,(R2) ;SEQUENCE ERROR?  
BNE TS161-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #TBL2-4,R0 ;INITIALIZE R0  
MOV @4(R0),@#TBL1 ;TBL1=125252  
ADD @2(R0),@#TBL1 ;TBL1=000377  
BICB @0(R0),@#TBL1 ;TBL1=000252  
BISB @-2(R0),@#TBL1+1 ;TBL1=125252  
CMP @-4(R0),@#TBL1 ;CHECK RESULT  
BEQ TS161  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====

```
4890
4891
4892 014274 012742 000353      MOV    #353,-(R2)
4893 014300 005242              INC    -(R2)
4894 014302 000000              HALT
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907 014304 005212              TS161: INC    (R2)
4908 014306 022712 000161      CMP    #161,(R2)
4909 014312 001026              BNE   TS162-10
4910 014314 012700 125252      MOV    #125252,R0
4911 014320 000261              SEC
4912 014322 006100              ROL   R0
4913 014324 102004              BVC   R0TOA
4914 014326 103003              BCC   R0TOA
4915 014330 022700 052525      CMP    #052525,R0
4916 014334 001404              BEQ   R0TOB
4917
4918
4919
4920
4921 014336
4922 014336 012742 000354      R0TOA: MOV    #354,-(R2)
4923 014342 005242              INC    -(R2)
4924 014344 000000              HALT
4925 014346 012700 125252      R0TOB: MOV    #125252,R0
4926 014352 000261              SEC
4927 014354 106100              ROLB  R0
4928 014356 102004              BVC   R0TOC
4929 014360 103003              BCC   R0TOC
4930 014362 022700 125125      CMP    #125125,R0
4931 014366 001404              BEQ   TS162
4932
4933
4934
4935
4936 014370
4937 014370 012742 000355      R0TOC: MOV    #355,-(R2)
4938 014374 005242              INC    -(R2)
4939 014376 000000              HALT
4940
```

REPLACE THE MOVE INSTRUCTION WHICH FOLLOWS W/ 755  
MOVE TO MAILBOX # \*\*\*\*\* 353 \*\*\*\*\*  
SET MSGTYP TO FATAL ERROR  
RESULT OF MODE 7 INSTS INCORRECT OR SEQUENCE ERROR

\*\*\*\*\*  
THIS TEST VERIFIES THE ROTATE MODE 0 INSTRUCTIONS.  
;R0 IS LOADED WITH A DATA PATTERN, THE C-BIT IS LOADED, AND  
;AN ROL INSTRUCTION IS EXECUTED WITH MODE 0. THE OPERATION IS CHECKED  
;BY TESTING THE RESULTING DATA AND THE STATE OF THE C AND V BITS.  
;NEXT, THE SAME PROCEDURE IS EXECUTED TO TEST MODE 0 BYTE INSTRUCTIONS.  
\*\*\*\*\*

TEST 161 TEST ROTATE INSTRUCTIONS OF MODE 0  
\*\*\*\*\*

TS161: UPDATE TEST NUMBER  
SEQUENCE ERROR?  
BR TO ERROR HALT ON SEQ ERROR  
INITIALIZE DATA  
SET C-BIT  
TRY ROL W/ MODE 0  
CC=0011

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
CONDITIONAL BRANCH INST. AND  
REPLACE THE MOVE INSTRUCTION WHICH FOLLOWS W/ 766

R0TOA: MOVE TO MAILBOX # \*\*\*\*\* 354 \*\*\*\*\*  
SET MSGTYP TO FATAL ERROR  
ROL MODE 0 FAILED

R0TOB: INITIALIZE DATA  
SET C-BIT  
TRY ROL W/ MODE 0 EVEN BYTE  
CC=0011

CHECK DATA

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
CONDITIONAL BRANCH INST. AND  
REPLACE THE MOVE INSTRUCTION WHICH FOLLOWS W/ 751

R0TOC: MOVE TO MAILBOX # \*\*\*\*\* 355 \*\*\*\*\*  
SET MSGTYP TO FATAL ERROR  
ROLB MODE 0 FAILED  
OR SEQUENCE ERROR

4941  
4942  
4943  
4944  
4945  
4946  
4947  
4948  
4949  
4950  
4951  
4952  
4953  
4954 014400 005212  
4955 014402 022712 000162  
4956 014406 001051  
4957 014410 005000  
4958 014412 012710 052525  
4959 014416 000241  
4960 014420 006110  
4961 014422 102005  
4962 014424 103404  
4963 014426 023727 000000 125252  
4964 014434 001404  
4965  
4966  
4967  
4968  
4969 014436  
4970 014436 012742 000356  
4971 014442 005242  
4972 014444 000000  
4973 014446 000261  
4974 014450 012710 125252  
4975 014454 106110  
4976 014456 102005  
4977 014460 103004  
4978 014462 022737 125125 000000  
4979 014470 001404  
4980  
4981  
4982  
4983  
4984 014472  
4985 014472 012742 000357  
4986 014476 005242  
4987 014500 000000  
4988 014502 012710 125252  
4989 014506 005000  
4990 014510 005200  
4991 014512 000261  
4992 014514 106110  
4993 014516 102005  
4994 014520 103004  
4995 014522 022737 052652 000000  
4996 014530 001404

.....  
: THIS TEST VERIFIES THE ROTATE MODE 1 INSTRUCTIONS.  
: THE DATA TO BE ROTATED IS IN LOC 0. R0 IS USED AS THE  
: ADDRESSING REGISTER. THE C-BIT IS LOADED AND AN ROL IS EXECUTED.  
: THE RESULTS ARE CHECKED BY COMPARING THE DATA RESULTS AND TESTING  
: THE C AND V BITS. THIS PROCEDURE IS THEN REPEATED TWICE MORE  
: TO TEST THE BYTE ROTATES. FIRST ON BYTE 0, THEN ON BYTE 1.  
:.....

: TEST 162 TEST ROTATE INSTRUCTIONS W/ MODE 1  
:.....

TS162: INC (R2) ;UPDATE TEST NUMBER  
CMP #162,(R2) ;SEQUENCE ERROR?  
BNE TS163-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;POINT TO LOC. 0  
MOV #52525,(R0) ;INITIALIZE DATA  
CLC ;CLEAR C-BIT  
ROL (R0) ;TRY ROL W/ MODE 1  
BVC ROT1A ;CC=1010  
BCS ROT1A  
CMP @#0,#125252 ;CHECK RESULT  
BEQ ROT1B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 764 <====

ROT1A: MOV #356,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 356 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ROL MODE 1 FAILED

ROT1B: SEC  
MOV #125252,(R0) ;INITIALIZE DATA  
ROLB (R0) ;TRY ROLB W/ MODE 1 EVEN BYTE  
BVC ROT1C ;CC=1011  
BCC ROT1C  
CMP #125125,@#0 ;TEST RESULT  
BEQ ROT1D

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 746 <====

ROT1C: MOV #357,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 357 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ROLB W/ MODE 1 EVEN BYTE FAILED

ROT1D: MOV #125252,(R0)  
CLR R0 ;POINT TO ODD BYTE  
INC R0  
SEC ;SET C-BIT  
ROLB (R0) ;TRY ROLB W/ MODE 1 ODD BYTE  
BVC ROT1E ;CC=0011  
BCC ROT1E  
CMP #052652,@#0 ;CHECK DATA  
BEQ TS163

4997  
4998  
4999  
5000  
5001 014532  
5002 014532 012742 000360  
5003 014536 005242  
5004 014540 000000  
5005  
5006  
5007  
5008  
5009  
5010  
5011  
5012  
5013  
5014  
5015  
5016  
5017 014542 005212  
5018 014544 022712 000163  
5019 014550 001057  
5020 014552 005000  
5021 014554 012710 173737  
5022 014560 000241  
5023 014562 006120  
5024 014564 103007  
5025 014566 022737 167676 000000  
5026 014574 001003  
5027 014576 005300  
5028 014600 005300  
5029 014602 001404  
5030  
5031  
5032  
5033  
5034 014604  
5035 014604 012742 000361  
5036 014610 005242  
5037 014612 000000  
5038 014614 005000  
5039 014616 012710 004040  
5040 014622 000241  
5041 014624 106120  
5042 014626 103406  
5043 014630 022737 004100 000000  
5044 014636 001002  
5045 014640 005300  
5046 014642 001404  
5047  
5048  
5049  
5050  
5051 014644  
5052 014644 012742 000362

ROT1E:  
MOV #360,-(R2) ; MOVE TO MAILBOX # \*\*\*\*\* 360 \*\*\*\*\*  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; ROLB W/ MODE 1 ODD BYTE FAILED  
; OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 2 ROTATE INSTRUCTIONS.  
: THE SAME PROCEDURE AS IN THE OTHER ROTATE TESTS ARE USED. R0  
: IS USED AS THE ADDRESSING REGISTER AND IS CHECKED FOR PROPER  
: INCREMENTING. BYTE INSTRUCTIONS ARE ALSO CHECKED.  
\*\*\*\*\*

: TEST 163 TEST ROTATE INSTRUCTIONS W/ MODE 2  
\*\*\*\*\*

TS163: INC (R2) ; UPDATE TEST NUMBER  
CMP #163,(R2) ; SEQUENCE ERROR?  
BNE TS164-10 ; BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ; POINT TO LOC 0  
MOV #173737,(R0) ; INITIALIZE DATA  
CLC ; CLEAR C-BIT  
ROL (R0)+ ; TRY ROL W/ MODE 2  
BCC ROT2A ; CHECK C-BIT  
CMP #167676,@#0 ; CHECK DATA  
BNE ROT2A ; BRANCH IF RESULT INCORRECT  
DEC R0 ; TEST R0  
DEC R0  
BEQ ROT2B

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
: CONDITIONAL BRANCH INST. AND  
: REPLACE THE MOVE INSTRUCTION  
: WHICH FOLLOWS W/ 762 <====

ROT2A: MOV #361,-(R2) ; MOVE TO MAILBOX # \*\*\*\*\* 361 \*\*\*\*\*  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; ROL W/ MODE 2 FAILED

ROT2B: CLR R0 ; POINT TO LOC 0  
MOV #4040,(R0) ; INITIALIZE DATA  
CLC ; CLEAR C-BIT  
ROLB (R0)+ ; TRY ROLB W/ MODE 2 EVEN BYTE  
BCS ROT2C ; CHECK C-BIT  
CMP #4100,@#0 ; CHECK DATA  
BNE ROT2C ; BRANCH IF DATA INCORRECT  
DEC R0 ; CHECK R0  
BEQ ROT2D

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
: CONDITIONAL BRANCH INST. AND  
: REPLACE THE MOVE INSTRUCTION  
: WHICH FOLLOWS W/ 742 <====

ROT2C: MOV #362,-(R2) ; MOVE TO MAILBOX # \*\*\*\*\* 362 \*\*\*\*\*

|      |        |        |        |        |        |            |    |  |       |
|------|--------|--------|--------|--------|--------|------------|----|--|-------|
| 5053 | 014650 | 005242 |        |        | INC    | -(R2)      |    | :SET MSGTYP TO FATAL ERROR               |       |
| 5054 | 014652 | 000000 |        |        | HALT   |            |    | :ROLB W/ MODE 2 EVEN BYTE FAILED         |       |
| 5055 | 014654 | 005000 |        |        | ROT2D: | CLR        | R0 | :POINT TO LOC 0                          |       |
| 5056 | 014656 | 012710 | 004040 |        | MOV    | #4040,(R0) |    | :INITIALIZE DATA                         |       |
| 5057 | 014662 | 005200 |        |        | INC    | R0         |    | :POINT TO ODD BYTE OF DATA               |       |
| 5058 | 014664 | 000261 |        |        | SEC    |            |    | :SET C-BIT                               |       |
| 5059 | 014666 | 106120 |        |        | ROLB   | (R0)+      |    | :TRY ROL W/ MODE 2 ODD BYTE              |       |
| 5060 | 014670 | 103407 |        |        | BCS    | ROT2E      |    | :CHECK C-BIT                             |       |
| 5061 | 014672 | 022737 | 010440 | 000000 | CMP    | #10440,#0  |    | :CHECK DATA                              |       |
| 5062 | 014700 | 001003 |        |        | BNE    | ROT2E      |    | :BRANCH IF DATA INCORRECT                |       |
| 5063 | 014702 | 005300 |        |        | DEC    | R0         |    | :CHECK R0                                |       |
| 5064 | 014704 | 005300 |        |        | DEC    | R0         |    |  |       |
| 5065 | 014706 | 001404 |        |        | BEQ    | TS164      |    |  |       |
| 5066 |        |        |        |        |        |            |    | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 5067 |        |        |        |        |        |            |    | : CONDITIONAL BRANCH INST. AND           | <==== |
| 5068 |        |        |        |        |        |            |    | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 5069 |        |        |        |        |        |            |    | : WHICH FOLLOWS W/ 720                   | <==== |
| 5070 | 014710 |        |        |        | ROT2E: |            |    |  |       |
| 5071 | 014710 | 012742 | 000363 |        | MOV    | #363,-(R2) |    | :MOVE TO MAILBOX # ***** 363 *****       |       |
| 5072 | 014714 | 005242 |        |        | INC    | -(R2)      |    | :SET MSGTYP TO FATAL ERROR               |       |
| 5073 | 014716 | 000000 |        |        | HALT   |            |    | :ROLB W/ MODE 2 ODD BYTE FAILED          |       |
| 5074 |        |        |        |        |        |            |    | : OR SEQUENCE ERROR                      |       |

5075  
5076  
5077  
5078  
5079  
5080  
5081  
5082  
5083  
5084  
5085  
5086  
5087  
5088  
5089  
5090  
5091  
5092  
5093  
5094  
5095  
5096  
5097  
5098  
5099  
5100  
5101  
5102  
5103  
5104  
5105  
5106  
5107  
5108  
5109  
5110  
5111  
5112  
5113  
5114  
5115  
5116  
5117  
5118  
5119  
5120  
5121  
5122  
5123  
5124  
5125  
5126  
5127  
5128  
5129  
5130

014720 005212  
014722 022712 000164  
014726 001051  
014730 012737 052525 000000  
014736 000261  
014740 006137 000000  
014744 103404  
014746 022737 125253 000000  
014754 001404  
  
014756  
014756 012742 000364  
014762 005242  
014764 000000  
014766 012737 125252 000000  
014774 000241  
014776 106137 000000  
015002 103004  
015004 023727 000000 125124 48:  
015012 001404  
  
015014  
015014 012742 000365  
015020 005242  
015022 000000  
015024 012737 125252 000000  
015032 000261  
015034 106137 000001  
015040 103004  
015042 022737 052652 000000  
015050 001404  
  
015052  
015052 012742 000366  
015056 005242  
015060 000000

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 3 ROTATE INSTRUCTIONS.  
: THIS TEST USES THE SAME PROCEDURES AS IN THE OTHER ROTATE  
: TESTS. THE DATA IS STORED IN LOC. 0 AND IS ADDRESSED USING  
: MODE 37. BYTE ADDRESSING IS ALSO CHECKED FOR EVEN AND ODD BYTES.  
\*\*\*\*\*  
: TEST 164 TEST ROTATE INSTRUCTIONS /W MODE 3  
\*\*\*\*\*  
TS164: INC (R2) ;UPDATE TEST NUMBER  
CMP #164,(R2) ;SEQUENCE ERROR?  
BNE TS165-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #52525,@#0 ;INITIALIZE DATA IN LOC 0  
SEC ;SET C-BIT  
ROL @#0 ;TRO ROL W/ MODE 3  
BCS ROT3A ;CHECK C-BIT  
CMP #125253,@#0 ;CHECK DATA  
BEQ ROT3B  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 764 <====  
  
ROT3A: MOV #364,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 364 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ROL W/ MODE 3 FAILED  
ROT3B: MOV #125252,@#0 ;INITIALIZE DATA  
CLC ;CLEAR C-BIT  
ROLB @#0 ;TRY ROL W/ MODE 3 EVEN BYTE  
BCC ROT3C ;CHECK C-BIT  
48: CMP @#0,#125124 ;CHECK DATA  
BEQ ROT3D  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 745 <====  
  
ROT3C: MOV #365,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 365 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ROL W/ MODE 3 EVEN BYTE FAILED  
ROT3D: MOV #125252,@#0 ;INITIALIZE DATA IN LOC. 0  
SEC ;SET C-BIT  
ROLB @#1 ;TRY ROL W/ MODE 3 ODD BYTE  
BCC ROT3E ;CHECK C-BIT  
5121: CMP #052652,@#0 ;CHECK DATA  
BEQ TS165  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 726 <====  
  
ROT3E: MOV #366,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 366 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ROL W/ MODE 3 ODD BYTE FAILED

5131  
5132  
5133  
5134  
5135  
5136  
5137  
5138  
5139  
5140  
5141  
5142  
5143  
5144 015062 005212  
5145 015064 022712 000165  
5146 015070 001016  
5147 015072 012737 070707 000000  
5148 015100 012700 000002  
5149 015104 000261  
5150 015106 006140  
5151 015110 103406  
5152 015112 022737 161617 000000  
5153 015120 001002  
5154 015122 005700  
5155 015124 001404  
5156  
5157  
5158  
5159  
5160 015126  
5161 015126 012742 000367  
5162 015132 005242  
5163 015134 000000  
5164  
5165  
5166  
5167  
5168  
5169  
5170  
5171  
5172  
5173  
5174  
5175  
5176  
5177  
5178  
5179 015136 005212  
5180 015140 022712 000166  
5181 015144 001021  
5182 015146 012737 015220 000000  
5183 015154 012700 000002  
5184 015160 012767 107070 000032  
5185 015166 000241  
5186 015170 006150

: OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 4 ROTATE INSTRUCTIONS. THE DATA IS  
: STORED IN LOC. 0. R0 IS SET TO 2 AND THE CARRY IS SET. AN ROL MODE 4  
: IS USED TO ROTATE LOCATION 0 USING R0. THE DATA IS CHECKED  
: AND THE C AND V BITS ARE TESTED. THE PROPER DECREMENTING OF  
: R0 IS VERIFIED.  
\*\*\*\*\*

: TEST 165 TEST MODE 4 W/ ROTATE INSTRUCTIONS  
\*\*\*\*\*

TS165: INC (R2) ;UPDATE TEST NUMBER  
CMP #165,(R2) ;SEQUENCE ERROR?  
BNE TS166-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #070707,@#0 ;INITIALIZE DATA IN LOC. 0  
MOV #2,R0 ;INITIALIZE R0 AS POINTER  
SEC ;SET C-BIT  
ROL -(R0) ;TRY ROL W/ MODE 4  
BCS ROT4 ;CHECK C-BIT  
CMP #161617,@#0 ;CHECK DATA  
BNE ROT4 ;BRANCH IF DATA INCORRECT  
TST R0 ;CHECK MODE 4 REGISTER  
BEQ TS166

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 761 <====

ROT4: MOV #367,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 367 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ROL MODE 4 FAILED  
: OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 5 ROTATE INSTRUCTIONS.  
: THE DATA IS STORED IN A WORK LOCATION (ROTX) AT THE END OF THE  
: TEST CODE. LOC. 0 IS LOADED WITH THE ADDRESS OF THE DATA (ROTX).  
: R0 IS SET TO 2. THE CARRY IS CLEARED AND A MODE 5 ROL  
: IS EXECUTED USING R0 AS AN ADDRESSING REGISTER. THE DATA IS  
: CHECKED, THE C AND V BITS TESTED, AND R0 CHECKED FOR PROPER  
: DECREMENTING.  
\*\*\*\*\*

: TEST 166 TEST MODE 5 W/ ROTATE INSTRUCTIONS  
\*\*\*\*\*

TS166: INC (R2) ;UPDATE TEST NUMBER  
CMP #166,(R2) ;SEQUENCE ERROR?  
BNE ROT5 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #ROTX,@#0 ;MOVE POINTER TO LOC. 0  
MOV #2,R0 ;SET MODE 5 REG. TO LOC. 0  
MOV #107070,ROTX ;INITIALIZE DATA  
CLC ;CLEAR C-BIT  
ROL @-(R0) ;TRY ROL W/ MODE 5

5187 015172 103006  
5188 015174 022737 016160 015220  
5189 015202 001002  
5190 015204 005700  
5191 015206 001405

BCC ROT5 ;CHECK C-BIT  
CMP #016160,@#ROTX ;CHECK DATA  
BNE ROT5 ;BRANCH IF DATA INCORRECT  
TST RO ;CHECK MODE 5 REGISTER  
BEQ TS167

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 756 <====

5192  
5193  
5194  
5195  
5196 015210  
5197 015210 012742 000370  
5198 015214 005242  
5199 015216 000000

ROT5:  
MOV #370,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 370 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ROL MODE 5 FAILED  
; OR SEQUENCE ERROR

5200  
5201 015220 000000

ROTX: 0

5202  
5203  
5204  
5205  
5206  
5207  
5208  
5209

.....  
: THIS TEST VERIFIES MODE 6 ROTATE INSTRUCTIONS.  
: IT USES THE SAME PROCEDURE AS THE ABOVE TEST EXCEPT THE  
: ROTATE INSTRUCTION USES MODE 6 ADDRESSING WITH REGISTER 7.  
: THE DATA IS STILL OPERATED ON IN LOC. ROTX (SEE PREVIOUS TEST).  
: .....

5210  
5211

: TEST 167 TEST MODE 6 W/ ROTATE INSTRUCTIONS  
: .....

5212  
5213 015222 005212  
5214 015224 022712 000167

TS167: INC (R2) ;UPDATE TEST NUMBER  
CMP #167,(R2) ;SEQUENCE ERROR?  
BNE TS170-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #125252,@#ROTX ;INITIALIZE DATA  
SEC ;SET C-BIT  
ROL ROTX ;TRY ROL W/ MODE 6  
BCC ROT6 ;CHECK C-BIT  
CMP #52525,@#ROTX ;CHECK DATA  
BEQ TS170

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 764 <====

5215 015230 001013  
5216 015232 012737 125252 015220  
5217 015240 000261  
5218 015242 006167 177752  
5219 015246 103004  
5220 015250 022737 052525 015220  
5221 015256 001404

5222  
5223

ROT6:  
MOV #371,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 371 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ROL W/ MODE 6 FAILED  
; OR SEQUENCE ERROR

5224  
5225  
5226 015260  
5227 015260 012742 000371

5228 015264 005242  
5229 015266 000000

5230



5231  
5232  
5233  
5234  
5235  
5236  
5237  
5238  
5239  
5240  
5241  
5242 015270 005212  
5243 015272 022712 000170  
5244 015276 001016  
5245 015300 012737 052525 015220  
5246 015306 012737 015220 015344  
5247 015314 000241  
5248 015316 006177 000022  
5249 015322 103404  
5250 015324 023727 015220 125252  
5251 015332 001405  
5252  
5253  
5254  
5255  
5256 015334  
5257 015334 012742 000372  
5258 015340 005242  
5259 015342 000000  
5260  
5261 015344 000000  
5262  
5263  
5264  
5265  
5266  
5267  
5268  
5269  
5270  
5271  
5272  
5273  
5274 015346 005212  
5275 015350 022712 000171  
5276 015354 001013  
5277 015356 012700 177400  
5278 015362 000300  
5279 015364 100404  
5280  
5281  
5282  
5283  
5284 015366 012742 000373  
5285 015372 005242  
5286 015374 000000

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 7 ROTATE INSTRUCTIONS.  
: THE DATA IS SET IN LOC. ROTX, (SEE PREVIOUS TEST). THE ROL INSTRUCTION  
: ADDRESSES IT INDIRECTLY USING MODE 7 AND INDIRECT ADDRESS LOCATION  
: (ROTXAD) FOLLOWING THE TEST CODE.  
\*\*\*\*\*

: TEST 170 TEST MODE 7 W/ ROTATE INSTRUCTIONS  
\*\*\*\*\*

TS170: INC (R2) ; UPDATE TEST NUMBER  
CMP #170,(R2) ; SEQUENCE ERROR?  
BNE ROT7 ; BR TO ERROR HALT ON SEQ ERROR  
MOV #52525,@#ROTX ; INITIALIZE DATA  
MOV #ROTX,@#ROTXAD ; INITIALIZE ADDRESS POINTER  
CLC ; CLEAR C-BIT  
ROL @ROTXAD ; TRY ROL W/ MODE 7  
BCS ROT7 ; CHECK C-BIT  
CMP @#ROTX,#125252 ; CHECK DATA  
BEQ TS171  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 761 <====

ROT7: MOV #372,-(R2) ; MOVE TO MAILBOX # \*\*\*\*\* 372 \*\*\*\*\*  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; ROL W/ MODE 7 FAILED  
: OR SEQUENCE ERROR

ROTXAD: 0

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 0 SWAB INSTRUCTION. R0 IS SET TO  
: 177400. A SWAB MODE 0 IS EXECUTED AND THE CONDITIONAL BRANCH  
: IS USED TO CHECK THE SIGN OF THE RESULT. ALSO, A COMPARISON  
: IS MADE TO CHECK THE DATA RESULTS.  
\*\*\*\*\*

: TEST 171 TEST MODE 0 W/ SWAB INST.  
\*\*\*\*\*

TS171: INC (R2) ; UPDATE TEST NUMBER  
CMP #171,(R2) ; SEQUENCE ERROR?  
BNE TS172-10 ; BR TO ERROR HALT ON SEQ ERROR  
MOV #177400,R0 ; MOVE TEST PATTERN TO R0  
SWAB R0 ; TRY SWAB MODE 0  
BMI SBO  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 773 <====  
MOV #373,-(R2) ; MOVE TO MAILBOX # \*\*\*\*\* 373 \*\*\*\*\*  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; SWAB DID NOT SET CC'S CORRECT

5287 015376 022700 000377  
5288 015402 001404  
5289  
5290  
5291  
5292  
5293 015404 012742 000374  
5294 015410 005242  
5295 015412 000000  
5296  
5297  
5298  
5299  
5300  
5301  
5302  
5303  
5304  
5305  
5306  
5307  
5308 015414 005212  
5309 015416 022712 000172  
5310 015422 001011  
5311 015424 012737 125652 000000  
5312 015432 005000  
5313 015434 000310  
5314 015436 022737 125253 000000  
5315 015444 001404  
5316  
5317  
5318  
5319  
5320 015446 012742 000375  
5321 015452 005242  
5322 015454 000000  
5323

SBO: CMP #377,RO ;CHECK RESULT  
BEQ TS172 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 764 <====  
MOV #374,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 374 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF SWAB MODE 0 FAILED  
; OR SEQUENCE ERROR

.....  
: THIS TEST VERIFIES MODE 1 SWAB INSTRUCTION. THE TEST  
: PATTERN IS MOVED TO LOC 0. RO IS CLEARED AND USED AS THE ADDRESSING  
: REGISTER IN THE MODE 1 SWAB. THE DATA RESULTS ARE CHECKED WITH  
: A COMPARE.  
: .....

: TEST 172 TEST MODE 1 W/ SWAB INST  
: .....

TS172: INC (R2) ;UPDATE TEST NUMBER  
CMP #172,(R2) ;SEQUENCE ERROR?  
BNE TS173-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #125652,@#0 ;MOVE TEST PATTERN TO LOC. 0  
CLR RO ;RO=0  
SWAB (R0) ;TRY SWAB MODE 1  
CMP #125253,@#0 ;CHECK RESULT  
BEQ TS173

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 766 <====  
: MOVE TO MAILBOX # \*\*\*\*\* 375 \*\*\*\*\*  
: SET MSGTYP TO FATAL ERROR  
: RESULT OF SWAB MODE 1 FAILED  
: OR SEQUENCE ERROR

5324  
.325  
5326  
5327  
5328  
5329  
5330  
5331  
5332  
5333  
5334  
5335  
5336  
5337  
5338  
5339  
5340  
5341  
5342  
5343  
5344  
5345  
5346  
5347  
5348  
5349  
5350  
5351  
5352  
5353  
5354  
5355  
5356  
5357  
5358  
5359  
5360  
5361  
5362  
5363  
5364  
5365  
5366  
5367  
5368  
5369  
5370  
5371  
5372  
5373  
5374  
5375  
5376  
5377  
5378  
5379

015456 005212  
015460 022712 000173  
015464 001020  
015466 012737 125152 000000  
015474 005000  
015476 000320  
015500 022737 065252 000000  
015506 001404  
  
015510 012742 000376  
015514 005242  
015516 000000  
015520 162700 000002  
015524 001404  
  
015526 012742 000377  
015532 005242  
015534 000000  
  
015536 005212  
015540 022712 000174  
015544 001011  
015546 012737 000377 000000  
015554 000337 000000  
015560 022737 177400 000000  
015566 001404

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 2 SWAB INSTRUCTION. THE TEST  
: PATTERN IS MOVED TO LOC 0. R0 IS CLEARED AND USED AS THE MODE  
: 2 ADDRESSING REGISTER. THE RESULTS ARE CHECKED WITH A COMPARE.  
: R0 IS CHECKED FOR PROPER DECREMENTING.  
\*\*\*\*\*

\*\*\*\*\*  
: TEST 173 TEST MODE 2 W/ SWAB INST  
\*\*\*\*\*

TS173: INC (R2) ;UPDATE TEST NUMBER  
CMP #173,(R2) ;SEQUENCE ERROR?  
BNE TS174-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #125152,@#0 ;MOVE TEST PATTERN TO LOC. 0  
CLR R0 ;R0=0  
SWAB (R0)+ ;TRY SWAB MODE 2  
CMP #65252,@#0 ;CHECK RESULT  
BEQ SB2  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 766 <====  
: MOVE TO MAILBOX # \*\*\*\*\* 376 \*\*\*\*\*  
: SET MSGTYP TO FATAL ERROR  
: RESULT OF SWAB MODE 0 FAILED  
: CHECK EFFECT OF REG.  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 757 <====  
: MOVE TO MAILBOX # \*\*\*\*\* 377 \*\*\*\*\*  
: SET MSGTYP TO FATAL ERROR  
: REGISTER VALUE INCORRECT  
: OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 3 SWAB INSTRUCTION. THE TEST  
: PATTERN IS MOVED TO LOC 0. A MODE 3 SWAB INSTRUCTION IS EXECUTED  
: USING R7 AS THE ADDRESSING REGISTER. A COMPARE VERIFIES THE  
: DATA RESULTS.  
\*\*\*\*\*

\*\*\*\*\*  
: TEST 174 TEST MODE 3 W/SWAB INST.  
\*\*\*\*\*

TS174: INC (R2) ;UPDATE TEST NUMBER  
CMP #174,(R2) ;SEQUENCE ERROR?  
BNE TS175-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #377,@#0 ;MOVE TEST PATTERN TO LOC. 0  
SWAB @#0 ;TRY SWAB W/ MODE 3  
CMP #177400,@#0 ;CHECK RESULT  
BEQ TS175

5380  
5381  
5382  
5383  
5384 015570 012742 000400  
5385 015574 005242  
5386 015576 000000  
5387

MOV #400,-(R2)  
INC -(R2)  
HALT

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <== :  
: WHICH FOLLOWS W/ 766 <====  
: MOVE TO MAILBOX # \*\*\*\*\* 400 \*\*\*\*\*  
: SET MSGTYP TO FATAL ERROR  
: RESULT OF SWAB INCORRECT  
: OR SEQUENCE ERROR

5388  
5389  
5390  
5391  
5392  
5393  
5394  
5395  
5396  
5397  
5398  
5399  
5400  
5401  
5402  
5403  
5404  
5405  
5406  
5407  
5408  
5409  
5410  
5411  
5412  
5413  
5414  
5415  
5416  
5417  
5418  
5419  
5420  
5421  
5422  
5423  
5424  
5425

015600 005212  
015602 022712 000175  
015606 001020  
015610 012737 125652 000000  
015616 012700 000002  
015622 000340  
015624 022737 125253 000000  
015632 001404  
  
015634 012742 000401  
015640 005242  
015642 000000  
015644 005700  
015646 001404  
  
015650 012742 000402  
015654 005242  
015656 000000

```
*****
:      THIS TEST VERIFIES MODE 4 SWAB INSTRUCTIONS.  THE DATA
: IS MOVED TO LOC 0.  R0 IS SET TO 2 AND USED AS THE MODE 4 ADDRESSING
: REGISTER.  THE DATA IS CHECKED WITH A COMPARE AND R0 IS CHECKED
: FOR PROPER DECREMENTING.
*****
: TEST 175      TEST MODE 4 W/ SWAB INST
*****
TS175:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #175,(R2)    ;SEQUENCE ERROR?
        BNE     TS176-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #125652,a#0   ;MOVE TEST PATTERN TO LOC. 0
        MOV     #2,R0        ;SET UP REGISTER POINTER
        SWAB   -(R0)         ;TRY SWAB MODE 4
        CMP     #125253,a#0   ;CHECK RESULT
        BEQ    SB4

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 765 <====
        MOV     #401,-(R2)    ;MOVE TO MAILBOX # ***** 401 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT   ;RESULT OF SWAB INCORRECT
SB4:    TST     R0           ;CHECK EFFECT ON REG.
        BEQ    TS176

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;           CONDITIONAL BRANCH INST. AND <====
;           REPLACE THE MOVE INSTRUCTION <====
;           WHICH FOLLOWS W/ 757 <====
        MOV     #402,-(R2)    ;MOVE TO MAILBOX # ***** 402 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT   ;REGISTER VALUE INCORRECT
;           OR SEQUENCE ERROR
```

5426  
5427  
5428  
5429  
5430  
5431  
5432  
5433  
5434  
5435  
5436  
5437  
5438  
5439  
5440  
5441  
5442  
5443  
5444  
5445  
5446  
5447  
5448  
5449  
5450  
5451  
5452  
5453  
5454  
5455  
5456  
5457  
5458  
5459  
5460  
5461  
5462  
5463  
5464  
5465  
5466  
5467

015660 005212  
015662 022712 000176  
015666 001021  
015670 012700 015746  
015674 012767 125125 000040  
015702 000350  
015704 022767 052652 000030  
015712 001404  
  
015714 012742 000403  
015720 005242  
015722 000000  
015724 020027 015744  
015730 001406  
  
015732  
015732 012742 000404  
015736 005242  
015740 000000  
  
015742 000000  
015744 015742

```
*****
:
: THIS TEST VERIFIES MODE 5 SWAB INSTRUCTION. THE TEST USES
: TWO LOCATIONS FOLLOWING THE TEST CODE. SB5X HOLDS THE DATA;
: SB5XAD IS A POINTER TO THE DATA LOCATION. THE DATA IS MOVED TO
: SB5X AND R0 IS SET TO TWO PLUS THE ADDRESS OF SB5XAD. FOLLOWING
: THE MODE 5 SWAB SB5X IS CHECKED FOR THE PROPER DATA. R0 IS
: CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.
:
: *****
: TEST 176 TEST MODE 5 W/ SWAB INST.
: *****
TS176: INC (R2) ;UPDATE TEST NUMBER
      CMP #176,(R2) ;SEQUENCE ERROR?
      BNE SB5 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #SB5XAD+2,R0 ;SET UP POINTER TO WORK LOCATION
      MOV #125125,SB5X ;MOVE PATTERN TO WORK LOCATION
      SWAB @-(R0) ;TRY SWAB MODE 5
      CMP #52652,SB5X ;CHECK RESULT
      BEQ SB5A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 765 <====
      MOV #403,-(R2) ;MOVE TO MAILBOX # ***** 403 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;RESULT OF SWAB INCORRECT
SB5A: CMP R0,#SB5XAD ;CHECK RESULT OF REG.
      BEQ TS177
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 756 <====
SB5: MOV #404,-(R2) ;MOVE TO MAILBOX # ***** 404 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;REGISTER VALUE INCORRECT
; OR SEQUENCE ERROR
SB5X: 0 ;WORK LOCATION
SB5XAD: SB5X
```

5468  
5469  
5470  
5471  
5472  
5473  
5474  
5475  
5476  
5477  
5478  
5479  
5480  
5481  
5482  
5483  
5484  
5485  
5486  
5487  
5488  
5489  
5490  
5491  
5492  
5493  
5494  
5495  
5496  
5497  
5498  
5499

015746 005212  
015750 022712 000177  
015754 001013  
015756 012767 125125 000030  
015764 012700 016006  
015770 000360 000006  
015774 022760 052652 000006  
016002 001405  
  
016004  
016004 012742 000405  
016010 005242  
016012 000000  
  
016014 000000

```
*****
:
: THIS TEST VERIFIES MODE 6 SWAB INSTRUCTION. THIS TEST
: USES A WORK LOCATION (SB6X) FOLLOWING THE TEST CODE. TEST DATA
: IS LOADED INTO THE WORK LOCATION. R0, THE ADDRESSING REGISTER
: IS LOADED WITH 6 LESS THEN THE ADDRESS OF THE WORK LOCATION.
: THE MODE 6 SWAB IS EXECUTED WITH A +6 OFFSET. THE DATA IS
: VERIFIED WITH A COMPARE.
:
: *****
: TEST 177 TEST MODE 6 W/ SWAB INST.
: *****
TS177: INC (R2) ;UPDATE TEST NUMBER
      CMP #177,(R2) ;SEQUENCE ERROR?
      BNE SB6 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #125125,SB6X ;MOVE PATTERN TO WORK LOCATION
      MOV #SB6X-6,R0 ;MOVE OFFSET POINTER TO R0
      SWAB 6(R0) ;TRY SWAB W/ MODE 6
      CMP #52652,6(R0) ;CHECK RESULT
      BEQ TS200
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 764 <====
SB6: MOV #405,-(R2) ;MOVE TO MAILBOX # ***** 405 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;RESULT OF SWAB INCORRECT
: OR SEQUENCE ERROR
SB6X: 0 ;WORK LOCATION
```

5500  
5501  
5502  
5503  
5504  
5505  
5506  
5507  
5508  
5509  
5510  
5511  
5512  
5513  
5514  
5515  
5516  
5517  
5518  
5519  
5520  
5521  
5522  
5523  
5524  
5525  
5526  
5527  
5528  
5529  
5530  
5531  
5532  
5533

016016 005212  
016020 022712 000200  
016024 001013  
016026 012767 177400 000030  
016034 012700 015774  
016040 000370 000072  
016044 027027 000072 000377  
016052 001406  
  
016054  
016054 012742 000406  
016060 005242  
016062 000000  
  
016064 000000  
016066 016064

```
.....  
: THIS TEST VERIFIES MODE 7 SWAB INSTRUCTION. THIS TEST  
: USES TWO LOCATIONS FOLLOWING THE TEST CODE: A WORK LOCATION  
: (SB7X) AND A POINTER TO THE WORK LOCATION (SB7XAD). DATA IS MOVED  
: TO THE WORK LOCATION. R0 IS LOADED WITH 72 LESS THAN THE ADDRESS  
: OF THE ADDRESS POINTER. THE DATA IS SWAB'ED USING A MODE 7  
: INSTRUCTION WITH AN OFFSET OF +72. THE DATA IS VERIFIED WITH A  
: COMPARE.  
:.....  
: TEST 200 TEST MODE 7 W/ SWAB INST.  
:.....  
TS200: INC (R2) ;UPDATE TEST NUMBER  
CMP #200,(R2) ;SEQUENCE ERROR?  
BNE SB7 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #177400,SB7X ;MOVE PATTERN TO WORK LOCATION  
MOV #SB7XAD-72,R0 ;MOVE OFFSET POINTER TO R0  
SWAB @72(R0) ;TRY SWAB MODE 7  
CMP @72(R0),#377 ;CHECK RESULTS  
BEQ TS201  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 764 <====  
  
SB7: MOV #406,-(R2) ;MOVE TO MAILBOX # ***** 406 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF SWAB INCORRECT  
; OR SEQUENCE ERROR  
SB7X: 0 ;WORK LOCATION  
SB7XAD: SB7X ;POINTER TO WORK LOCATION
```



5534  
5535  
5536  
5537  
5538  
5539  
5540  
5541  
5542  
5543  
5544  
5545  
5546  
5547  
5548  
5549  
5550  
5551  
5552  
5553  
5554  
5555  
5556  
5557  
5558  
5559  
5560  
5561  
5562  
5563  
5564  
5565  
5566  
5567  
5568  
5569  
5570  
5571  
5572  
5573  
5574  
5575  
5576  
5577  
5578  
5579  
5580  
5581  
5582  
5583  
5584  
5585  
5586  
5587  
5588  
5589

\*\*\*\*\*

: THIS TEST VERIFIES ALL LEGAL MODES OF THE JMP INSTRUCTION.  
: BECAUSE OF THE NATURE OF THE INSTRUCTION UNDER TEST, THIS TEST  
: UTILIZES SEVERAL DIFFERENT TECHNIQUES. THE CODE IS NOT EXECUTED  
: IN A LINEAR FASHION. THE DIFFERENT MODES ARE EXECUTED IN ORDER  
: FROM 1-7; HOWEVER, THE CODE IS ARRANGED SO THAT CONTROL LEAP  
: FROGS THRU THE TEST CODE. THE ORDER OF APPEARANCE OF THE CODE  
: IS:

- JMP MODE 1
- JMP MODE 3
- JMP MODE 2
- JMP MODE 4
- JMP MODE 6
- JMP MODE 5
- JMP MODE 7

: AN INTERNAL SEQUENCE TEST (JMPSEQ) IS USED TO INSURE THAT THE  
: JUMPS ARE OCCURRING IN THE PROGRAMMED SEQUENCE.

: THE TEST IS MADE UP OF SEVERAL BLOCKS OF CODE. EACH CODE  
: BEGINS WITH A LABEL WHICH INDICATES THE MODE BEING EXECUTED IN  
: THAT BLOCK. A SIMPLE PROCEDURE IS FOLLOWED IN EACH BLOCK. FOR  
: EXAMPLE THE CODE BEGINNING AT JMP3 WILL FIRST COMPARE THE RESULTS  
: OF THE PREVIOUS MODE 2 JUMP. (ANY REGISTER CHANGES ARE VERIFIED  
: AND THE SEQUENCE CHECK IS MADE). THEN THE REGISTERS ARE SETUP  
: FOR A MODE 3 JUMP TO THE NEXT TEST BLOCK (HERE, JMP4), THE SEQUENCE  
: CHECKER IS UPDATED AND THE JUMP IS EXECUTED.

: IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN  
: DETERMINING JUST WHICH MODE FAILED. IF THE SEQUENCE IS CORRECT  
: THEN THE ERROR DETECTED WAS A MODE FAILURE (E.G. FAILURE OF THE  
: REGISTER TO BE INCREMENTED IN MODE 2 JUMP.)

\*\*\*\*\*

: TEST 201 TEST THE JMP INSTRUCTION IN ALL MODES

\*\*\*\*\*

```

TS201:  INC      (R2)           ;UPDATE TEST NUMBER
        CMP      #201,(R2)    ;SEQUENCE ERROR?
        BNE     JMPCK+6      ;BR TO ERROR HALT ON SEQ ERROR
        CLR     JMPSEQ       ;ESTABLISH A SEQUENCE CHECKER
        MOV     #JMP2,R0     ;SET R0=JUMP TARGET
        JMP     (R0)         ;TRY JMP MODE 1
JMP3:   CMP      #.+2,R0     ;CHECK RESULT OF MODE 2 JUMP
        BEQ     JMP3A

```

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 767 <====

```

        MOV     #407,-(R2)   ;MOVE TO MAILBOX # ***** 407 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                    ;REGISTER VALUE AFTER JMP MODE 2 INCORRECT
JMP3A:  CMP      JMPSEQ,#1   ;MAKE SURE JUMPS ARE IN SEQUENCE: JMPSEQ=1?
        BEQ     JMP3B

```

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 757 <====

|      |        |        |               |        |      |              |  |  |
|------|--------|--------|---------------|--------|------|--------------|--|--|
| 5590 | 016140 | 012742 | 000410        |        | MOV  | #410,-(R2)   |  | :MOVE TO MAILBOX # ***** 410 *****             |
| 5591 | 016144 | 005242 |               |        | INC  | -(R2)        |  | :SET MSGTYP TO FATAL ERROR                     |
| 5592 | 016146 | 000000 |               |        | HALT |              |  | :SHOULD BE HERE FROM JMP MODE 2 ONLY           |
| 5593 | 016150 | 012700 | 016162        | JMP3B: | MOV  | #1JMP4,R0    |  | :POINT R0 TO INDIRECT JMP ADDR.                |
| 5594 | 016154 | 005267 | 000252        |        | INC  | JMPSEQ       |  | :UPDATE SEQUENCE CHECKER                       |
| 5595 | 016160 | 000130 |               |        | JMP  | @(R0)+       |  | :TRY JMP MODE 3                                |
| 5596 | 016162 | 016214 |               | 1JMP4: | JMP4 |              |  | :ADDRESS INDIRECT JUMP                         |
| 5597 |        |        |               |        |      |              |  |  |
| 5598 | 016164 | 005767 | 000242        | JMP2:  | TST  | JMPSEQ       |  | :CHECK THAT JMPs ARE IN SEQUENCE: JMPSEQ=0?    |
| 5599 | 016170 | 001404 |               |        | BEQ  | JMP2A        |  |  |
| 5600 |        |        |               |        |      |              |  | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==== |
| 5601 |        |        |               |        |      |              |  | : CONDITIONAL BRANCH INST. AND <====           |
| 5602 |        |        |               |        |      |              |  | : REPLACE THE MOVE INSTRUCTION <====           |
| 5603 |        |        |               |        |      |              |  | : WHICH FOLLOWS W/ 742 <====                   |
| 5604 | 016172 | 012742 | 000411        |        | MOV  | #411,-(R2)   |  | :MOVE TO MAILBOX # ***** 411 *****             |
| 5605 | 016176 | 005242 |               |        | INC  | -(R2)        |  | :SET MSGTYP TO FATAL ERROR                     |
| 5606 | 016200 | 000000 |               |        | HALT |              |  | :SHOULD BE HERE FROM JMP MODE 1 ONLY           |
| 5607 | 016202 | 005267 | 000224        | JMP2A: | INC  | JMPSEQ       |  | :UPDATE SEQUENCE CHECKER                       |
| 5608 | 016206 | 012700 | 016112        |        | MOV  | #JMP3,R0     |  | :SET R0=JUMP TARGET                            |
| 5609 | 016212 | 000120 |               |        | JMP  | (R0)+        |  | :TRY A JUMP MODE 2 TO "JMP3"                   |
| 5610 | 016214 | 022700 | 016164        | JMP4:  | CMP  | #1JMP4+2,R0  |  | :CHECK RESULT OF REGISTER IN MODE 3 JUMP       |
| 5611 | 016220 | 001404 |               |        | BEQ  | JMP4A        |  |  |
| 5612 |        |        |               |        |      |              |  | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==== |
| 5613 |        |        |               |        |      |              |  | : CONDITIONAL BRANCH INST. AND <====           |
| 5614 |        |        |               |        |      |              |  | : REPLACE THE MOVE INSTRUCTION <====           |
| 5615 |        |        |               |        |      |              |  | : WHICH FOLLOWS W/ 726 <====                   |
| 5616 | 016222 | 012742 | 000412        |        | MOV  | #412,-(R2)   |  | :MOVE TO MAILBOX # ***** 412 *****             |
| 5617 | 016226 | 005242 |               |        | INC  | -(R2)        |  | :SET MSGTYP TO FATAL ERROR                     |
| 5618 | 016230 | 000000 |               |        | HALT |              |  | :REGISTER VALUE AFTER MODE 3 JUMP INCORRECT    |
| 5619 | 016232 | 022767 | 000002 000172 | JMP4A: | CMP  | #2,JMPSEQ    |  | :CHECK JUMP SEQUENCE: JMPSEQ=2?                |
| 5620 | 016240 | 001404 |               |        | BEQ  | JMP4B        |  |  |
| 5621 |        |        |               |        |      |              |  | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==== |
| 5622 |        |        |               |        |      |              |  | : CONDITIONAL BRANCH INST. AND <====           |
| 5623 |        |        |               |        |      |              |  | : REPLACE THE MOVE INSTRUCTION <====           |
| 5624 |        |        |               |        |      |              |  | : WHICH FOLLOWS W/ 716 <====                   |
| 5625 | 016242 | 012742 | 000413        |        | MOV  | #413,-(R2)   |  | :MOVE TO MAILBOX # ***** 413 *****             |
| 5626 | 016246 | 005242 |               |        | INC  | -(R2)        |  | :SET MSGTYP TO FATAL ERROR                     |
| 5627 | 016250 | 000000 |               |        | HALT |              |  | :SHOULD BE ONLY FROM MODE 3 JUMP               |
| 5628 | 016252 | 012700 | 016322        | JMP4B: | MOV  | #JMP5+2,R0   |  | :SET UP POINTER TO JUMP TARGET                 |
| 5629 | 016256 | 005267 | 000150        |        | INC  | JMPSEQ       |  | :UPDATE SEQUENCE CHECKER                       |
| 5630 | 016262 | 000140 |               |        | JMP  | -(R0)        |  | :TRY JUMP MODE 4 TO "JMP4"                     |
| 5631 |        |        |               |        |      |              |  |  |
| 5632 | 016264 | 022767 | 000004 000140 | JMP6:  | CMP  | #4,JMPSEQ    |  | :CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?   |
| 5633 | 016272 | 001404 |               |        | BEQ  | JMP6A        |  |  |
| 5634 |        |        |               |        |      |              |  | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==== |
| 5635 |        |        |               |        |      |              |  | : CONDITIONAL BRANCH INST. AND <====           |
| 5636 |        |        |               |        |      |              |  | : REPLACE THE MOVE INSTRUCTION <====           |
| 5637 |        |        |               |        |      |              |  | : WHICH FOLLOWS W/ 701 <====                   |
| 5638 | 016274 | 012742 | 000414        |        | MOV  | #414,-(R2)   |  | :MOVE TO MAILBOX # ***** 414 *****             |
| 5639 | 016300 | 005242 |               |        | INC  | -(R2)        |  | :SET MSGTYP TO FATAL ERROR                     |
| 5640 | 016302 | 000000 |               |        | HALT |              |  | :SHOULD BE HERE ONLY FROM MODE 5 JUMP          |
| 5641 | 016304 | 012700 | 016752        | JMP6A: | MOV  | #JMP7+376,R0 |  | :SET UP OFFSET POINTER TO JUMP TARGET          |
| 5642 | 016310 | 005267 | 000116        |        | INC  | JMPSEQ       |  | :UPDATE JUMP SEQUENCE                          |
| 5643 | 016314 | 000160 | 177402        |        | JMP  | -376(R0)     |  | :TRY MODE 6 JUMP                               |
| 5644 |        |        |               |        |      |              |  |  |
| 5645 | 016320 | 022767 | 000003 000104 | JMP5:  | CMP  | #3,JMPSEQ    |  | :CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?   |

```
5646 016326 001404          BEQ      JMP5A          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5647                                     ;          CONDITIONAL BRANCH INST. AND <====
5648                                     ;          REPLACE THE MOVE INSTRUCTION <====
5649                                     ;          WHICH FOLLOWS W/ 663 <====
5650                                     ;
5651 016330 012742 000415      MOV      #415,-(R2)      ;MOVE TO MAILBOX # ***** 415 *****
5652 016334 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5653 016336 000000              HALT                    ;SHOULD ONLY BE HERE FROM MODE 4 JUMP
5654 016340 012700 016354      JMP5A:  MOV      #1JMP5+2,R0 ;SET UP POINTER TO INDIRECT JUMP ADDR.
5655 016344 005267 000062              INC      JMPSEQ        ;UPDATE JUMP SEQUENCE
5656 016350 000150              JMP      @-(R0)         ;TRY JUMP MODE 5 TO "JMP6"
5657 016352 016264              1JMP5:  JMP6          ;INDIRECT ADDRESS POINTER
5658
5659 016354 022767 000005 000050 JMP7:   CMP      #5,JMPSEQ ;CHECK JUMPS IN SEQUENCE: JMPSEQ=5?
5660 016362 001404              BEQ      JMP7A
5661                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5662                                     ;          CONDITIONAL BRANCH INST. AND <====
5663                                     ;          REPLACE THE MOVE INSTRUCTION <====
5664                                     ;          WHICH FOLLOWS W/ 645 <====
5665 016364 012742 000416      MOV      #416,-(R2)      ;MOVE TO MAILBOX # ***** 416 *****
5666 016370 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5667 016372 000000              HALT                    ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5668 016374 012700 016420      JMP7A:  MOV      #1JMP+10,R0 ;SET UP OFFSET POINTER TO INDIRECT ADDR.
5669 016400 005267 000026              INC      JMPSEQ        ;UPDATE JUMP SEQUENCE
5670 016404 000170 177770              JMP      @-10(R0)       ;TRY MODE 7 JUMP
5671 016410 016412              1JMP:   JMPCK
5672
5673 016412 026727 000014 000006 JMPCK:  CMP      JMPSEQ,#6 ;CHECK JUMPS IN SEQUENCE: JMPSEQ
5674 016420 001405              BEQ      TS202
5675                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5676                                     ;          CONDITIONAL BRANCH INST. AND <====
5677                                     ;          REPLACE THE MOVE INSTRUCTION <====
5678                                     ;          WHICH FOLLOWS W/ 626 <====
5679 016422 012742 000417      MOV      #417,-(R2)      ;MOVE TO MAILBOX # ***** 417 *****
5680 016426 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5681 016430 000000              HALT                    ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5682                                     ; OR SEQUENCE ERROR
5683 016432 000000              JMPSEQ: 0
```

5684  
5685  
5686  
5687  
5688  
5689  
5690  
5691  
5692  
5693  
5694  
5695  
5696  
5697  
5698  
5699  
5700  
5701  
5702  
5703  
5704  
5705  
5706  
5707  
5708  
5709  
5710  
5711  
5712  
5713  
5714  
5715  
5716  
5717  
5718  
5719  
5720  
5721  
5722  
5723  
5724  
5725  
5726  
5727  
5728  
5729  
5730  
5731  
5732  
5733  
5734  
5735  
5736  
5737  
5738  
5739

016434 005212  
016436 022712 000202  
016442 001001  
016444 000402  
016446 000137 017102  
016452 012706 001000  
016456 012700 016564  
016462 005037 017062  
016466 005001  
016470 005101  
016472 004110  
016474  
016474 012742 000420  
016500 005242  
016502 000000  
016504 022737 000001 017062  
016512 001014  
016514 020127 016646  
016520 001011  
016522 022706 000776  
016526 001006  
016530 022716 125252  
016534 001003  
016536 022700 016506  
016542 001404  
016544  
016544 012742 000421  
016550 005242

```
*****
:
: THIS TEST VERIFIES ALL LEGAL MODES OF THE JSR INSTRUCTION.
: THE CONCEPT OF LEAP FROGGING AND SEQUENCE CHECKING (JSRSEQ) IS
: IDENTICAL TO THAT USED IN JMP TEST (SEE PREVIOUS TEST). EACH
: BLOCK OF CODE VERIFIES THE PREVIOUS JSR BY CHECKING THE SEQUENCE,
: CHECKING THAT THE PC WAS SAVED IN THE SPECIFIED REGISTER, CHECKING
: THAT THE SP WAS DECREMENTED, CHECKING THAT THE REGISTER WAS
: SAVED ON THE STACK, AND FINALLY CHECKING THAT ANY MODE ADDRESS
: REGISTER ALTERATIONS (E.G. INCREMENT REGISTER IN MODE 2) WERE
: SUCCESSFUL. R1 IS USED AS THE REGISTER IN ALL JSR INSTRUCTIONS.
: IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
: DETERMINING JUST WHICH MODE FAILED. IF THE SEQUENCE IS CORRECT
: THEN THE ERROR DETECTED WAS A FUNCTIONAL FAILURE (E.G., INCORRECT
: REGISTER SAVED).
:
:*****
:TEST 202 TEST JSR INSTRUCTION W/ ALL MODES
:*****
TS202: INC (R2) ;UPDATE TEST NUMBER
      CMP #202,(R2) ;SEQUENCE ERROR?
      BNE JSR0 ;BR TO ERROR HALT ON SEQ ERROR
      BR JSR1
JSR0: JMP @#JSRCK1
JSR1: MOV #STBOT,R6 ;SET STACK POINTER
      MOV #JSR2,R0 ;SET TARGET ADDRESS
      CLR @#JSRSEQ ;INITIALIZE SEQUENCE CHECKER
      CLR R1 ;INITIALIZE R1
      COM R1
      JSR R1,(R0) ;TRY JSR MODE 1
      ; TO SCOPE: REPLACE THE MOVE INSTRUCTION <====
      ; FOLLOWING W/ 774 <====
JSR1A: MOV #420,-(R2) ;MOVE TO MAILBOX # ***** 420 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;JSR MODE 1 FAILED
JSR3: CMP #1,@# JSRSEQ ;CHECK SEQUENCE: JSRSEQ=1?
      BNE JSR3A ;BRANCH IF OUT OF SEQUENCE
      CMP R1,#JSR4 ;PROPER PC SAVED?
      BNE JSR3A ;BRANCH IF PC WRONG
      CMP #STBOT-2,R6 ;STACK POINTER DECREMENTED?
      BNE JSR3A ;BRANCH IF SP WRONG
      CMP #125252,(R6) ;REG SAVED ON STACK?
      BNE JSR3A ;BRANCH IF REG. NOT SAVED
      CMP #JSR3+2,R0 ;MODE 2 INCREMENT CORRECT?
      BEQ JSR3B
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 737 <====
JSR3A: MOV #421,-(R2) ;MOVE TO MAILBOX # ***** 421 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
```

|      |        |        |               |        |      |             |  |  |  |  |  |
|------|--------|--------|---------------|--------|------|-------------|--|--|--|--|--|
| 5740 | 016552 | 000000 |               |        |      |             |  |  |  |  |  |
| 5741 | 016554 | 005237 | 017062        | JSR3B: | HALT |             |  |  |  |  |  |
| 5742 | 016560 | 004137 | 016646        |        | INC  | @#JSRSEQ    |  |  |  |  |  |
| 5743 |        |        |               |        | JSR  | R1,@#JSR4   |  |  |  |  |  |
| 5744 | 016564 | 005737 | 017062        | JSR2:  | TST  | @#JSRSEQ    |  |  |  |  |  |
| 5745 | 016570 | 001011 |               |        | BNE  | JSR2A       |  |  |  |  |  |
| 5746 | 016572 | 020127 | 016474        |        | CMP  | R1,#JSR1A   |  |  |  |  |  |
| 5747 | 016576 | 001006 |               |        | BNE  | JSR2A       |  |  |  |  |  |
| 5748 | 016600 | 022706 | 000776        |        | CMP  | #STBOT-2,R6 |  |  |  |  |  |
| 5749 | 016604 | 001003 |               |        | BNE  | JSR2A       |  |  |  |  |  |
| 5750 | 016606 | 021627 | 177777        |        | CMP  | (R6),#-1    |  |  |  |  |  |
| 5751 | 016612 | 001404 |               |        | BEQ  | JSR2B       |  |  |  |  |  |
| 5752 |        |        |               |        |      |             |  |  |  |  |  |
| 5753 |        |        |               |        |      |             |  |  |  |  |  |
| 5754 |        |        |               |        |      |             |  |  |  |  |  |
| 5755 |        |        |               |        |      |             |  |  |  |  |  |
| 5756 | 016614 |        |               | JSR2A: |      |             |  |  |  |  |  |
| 5757 | 016614 | 012742 | 000422        |        | MOV  | #422,-(R2)  |  |  |  |  |  |
| 5758 | 016620 | 005242 |               |        | INC  | -(R2)       |  |  |  |  |  |
| 5759 | 016622 | 000000 |               |        | HALT |             |  |  |  |  |  |
| 5760 | 016624 | 012706 | 001000        | JSR2B: | MOV  | #STBOT,R6   |  |  |  |  |  |
| 5761 | 016630 | 012701 | 125252        |        | MOV  | #125252,R1  |  |  |  |  |  |
| 5762 | 016634 | 005237 | 017062        |        | INC  | @#JSRSEQ    |  |  |  |  |  |
| 5763 | 016640 | 012700 | 016504        |        | MOV  | #JSR3,R0    |  |  |  |  |  |
| 5764 | 016644 | 004120 |               |        | JSR  | R1,(R0)+    |  |  |  |  |  |
| 5765 |        |        |               |        |      |             |  |  |  |  |  |
| 5766 | 016646 | 022737 | 000002 017062 | JSR4:  | CMP  | #2,@#JSRSEQ |  |  |  |  |  |
| 5767 | 016654 | 001003 |               |        | BNE  | JSR4A       |  |  |  |  |  |
| 5768 | 016656 | 022701 | 016564        |        | CMP  | #JSR2,R1    |  |  |  |  |  |
| 5769 | 016662 | 001404 |               |        | BEQ  | JSR4B       |  |  |  |  |  |
| 5770 |        |        |               |        |      |             |  |  |  |  |  |
| 5771 |        |        |               |        |      |             |  |  |  |  |  |
| 5772 |        |        |               |        |      |             |  |  |  |  |  |
| 5773 |        |        |               |        |      |             |  |  |  |  |  |
| 5774 | 016664 |        |               | JSR4A: |      |             |  |  |  |  |  |
| 5775 | 016664 | 012742 | 000423        |        | MOV  | #423,-(R2)  |  |  |  |  |  |
| 5776 | 016670 | 005242 |               |        | INC  | -(R2)       |  |  |  |  |  |
| 5777 | 016672 | 000000 |               |        | HALT |             |  |  |  |  |  |
| 5778 | 016674 | 005237 | 017062        | JSR4B: | INC  | @#JSRSEQ    |  |  |  |  |  |
| 5779 | 016700 | 012700 | 016754        |        | MOV  | #JSR5+2,R0  |  |  |  |  |  |
| 5780 | 016704 | 004140 |               |        | JSR  | R1,-(R0)    |  |  |  |  |  |
| 5781 |        |        |               |        |      |             |  |  |  |  |  |
| 5782 | 016706 | 022767 | 000004 000146 | JSR6:  | CMP  | #4,JSRSEQ   |  |  |  |  |  |
| 5783 | 016714 | 001006 |               |        | BNE  | JSR6A       |  |  |  |  |  |
| 5784 | 016716 | 022701 | 017020        |        | CMP  | #JSR7,R1    |  |  |  |  |  |
| 5785 | 016722 | 001003 |               |        | BNE  | JSR6A       |  |  |  |  |  |
| 5786 | 016724 | 022700 | 017056        |        | CMP  | #JSR6AD,R0  |  |  |  |  |  |
| 5787 | 016730 | 001404 |               |        | BEQ  | JSR6B       |  |  |  |  |  |
| 5788 |        |        |               |        |      |             |  |  |  |  |  |
| 5789 |        |        |               |        |      |             |  |  |  |  |  |
| 5790 |        |        |               |        |      |             |  |  |  |  |  |
| 5791 |        |        |               |        |      |             |  |  |  |  |  |
| 5792 | 016732 |        |               | JSR6A: |      |             |  |  |  |  |  |
| 5793 | 016732 | 012742 | 000424        |        | MOV  | #424,-(R2)  |  |  |  |  |  |
| 5794 | 016736 | 005242 |               |        | INC  | -(R2)       |  |  |  |  |  |
| 5795 | 016740 | 000000 |               |        | HALT |             |  |  |  |  |  |

```
5796 016742 005237 017062 JSR6B: INC @#JSRSEQ ;UPDATE SEQUENCE CHECKER
5797 016746 004167 000046 JSR5: JSR R1,JSR7 ;TRY JSR MODE 6
5798 016752 022767 000003 000102 JSR5: CMP #3,JSRSEQ ;CHECK SEQUENCE: JSRSEQ=3?
5799 016760 001006 BNE JSR5A ;BRANCH IF OUT OF SEQUENCE
5800 016762 022701 016706 CMP #JSR6,R1 ;PROPER PC SAVED?
5801 016766 001003 BNE JSR5A ;BRANCH IF PC WRONG
5802 016770 022700 016752 CMP #JSR5,R0 ;CHECK MODE 4 REGISTER
5803 016774 001404 BEQ JSR5B
5804
5805 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5806 ; CONDITIONAL BRANCH INST. AND <====
5807 ; REPLACE THE MOVE INSTRUCTION <====
5808 ; WHICH FOLLOWS W/ 622 <====
5808 016776 JSR5A:
5809 016776 012742 000425 MOV #425,-(R2) ;MOVE TO MAILBOX # ***** 425 *****
5810 017002 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5811 017004 000000 HALT ;JSR MODE 4 MALFUNCTIONED
5812 017006 005237 017062 JSR5B: INC @#JSRSEQ ;UPDATE SEQUENCE CHECKER
5813 017012 012700 017060 MOV #JSR6AD+2,R0 ;POINT R0 TO TARGET ADDRESS
5814 017016 004150 JSR R1,@-(R0) ;TRY JSR MODE 5
5815
5816 017020 022737 000005 017062 JSR7: CMP #5,@#JSRSEQ ;CHECK SEQUENCE: JSRSEQ=5?
5817 017026 001003 BNE JSR7A ;BRANCH IF OUT OF SEQUENCE
5818 017030 022701 016752 CMP #JSR5,R1 ;PROPER PC SAVED?
5819 017034 001404 BEQ JSR7B
5820
5821 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5822 ; CONDITIONAL BRANCH INST. AND <====
5823 ; REPLACE THE MOVE INSTRUCTION <====
5824 ; WHICH FOLLOWS W/ 602 <====
5824 017036 JSR7A:
5825 017036 012742 000426 MOV #426,-(R2) ;MOVE TO MAILBOX # ***** 426 *****
5826 017042 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5827 017044 000000 HALT ;JSR MODE 6 FAILED
5828 017046 005237 017062 JSR7B: INC @#JSRSEQ ;UPDATE SEQUENCE CHECKER
5829 017052 004177 000002 JSR R1,@JSRCKAD ;TRY JSR MODE 7
5830
5831 017056 016706 JSR6AD: JSR6 ;MODE 5 TARGET ADDRESS
5832 017060 017064 JSRCKAD:JSRCK ;MODE 7 TARGET ADDRESS
5833 017062 000000 JSRSEQ: 0 ;SEQUENCE CHECKER
5834
5835 017064 022767 000006 177770 JSRCK: CMP #6,JSRSEQ ;CHECK SEQUENCE: JSRSEQ=6?
5836 017072 001003 BNE JSRCK1 ;BRANCH IF OUT OF SEQUENCE
5837 017074 022701 017056 CMP #JSR6AD,R1 ;PROPER PC SAVED?
5838 017100 001404 BEQ TS203
5839
5840 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5841 ; CONDITIONAL BRANCH INST. AND <====
5842 ; REPLACE THE MOVE INSTRUCTION <====
5843 ; WHICH FOLLOWS W/ 560 <====
5843 017102 JSRCK1:
5844 017102 012742 000427 MOV #427,-(R2) ;MOVE TO MAILBOX # ***** 427 *****
5845 017106 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5846 017110 000000 HALT ;JSR MODE 7 MALFUNCTIONED
5847 ; OR SEQUENCE ERROR
5848
5849
```

5850  
 5851  
 5852  
 5853  
 5854  
 5855  
 5856  
 5857  
 5858  
 5859  
 5860 017112 005212  
 5861 017114 022712 000203  
 5862 017120 001016  
 5863 017122 012706 001000  
 5864 017126 012746 052525  
 5865 017132 012700 017150  
 5866 017136 000200  
 5867  
 5868  
 5869 017140 012742 000430  
 5870 017144 005242  
 5871 017146 000000  
 5872 017150 022700 052525  
 5873 017154 001404  
 5874  
 5875  
 5876  
 5877  
 5878 017156 012742 000431  
 5879 017162 005242  
 5880 017164 000000  
 5881

```

:*****
:
:   THIS TEST VERIFIES THE RTS INSTRUCTION.  THE STACK POINTER
: IS INITIALIZED AND A TEST PATTERN STORED ON STACK.  R0 IS LOADED
: WITH RETURN ADDRESS.  AN RTS IS EXECUTED, AND, AT THE TARGET
: ADDRESS, A CHECK IS MADE THAT R0 WAS PROPERLY RESTORED FROM THE
: STACK.
:*****
:TEST 203      TEST RTS INSTRUCTION
:*****
TS203:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #203,(R2)    ;SEQUENCE ERROR?
        BNE     TS204-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #STBOT,R6    ;INITIALIZE STACK POINTER
        MOV     #52525,-(R6) ;INITIALIZE TOP OF STACK
        MOV     #RTS1,R0     ;INITIALIZE RETURN REGISTER
        RTS     R0          ;TRY RTS THROUGH R0
        ; TO SCOPE: REPLACE THE MOVE INSTRUCTION <===
        ; FOLLOWING W/ 770 <===
        MOV     #430,-(R2)   ;MOVE TO MAILBOX # ***** 430 *****
        INC     -R2          ;SET MSGTYP TO FATAL ERROR
        HALT    ;RTS FAILED
RTS1:   CMP     #52525,R0    ;CHECK THAT R0 RESTORED FROM STACK
        BEQ     TS204
        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
        ; CONDITIONAL BRANCH INST. AND <===
        ; REPLACE THE MOVE INSTRUCTION <===
        ; WHICH FOLLOWS W/ 761 <===
        MOV     #431,-(R2)   ;MOVE TO MAILBOX # ***** 431 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT    ;RTS MALFUNCTIONED
        ; OR SEQUENCE ERROR
  
```

5882  
5883  
5884  
5885  
5886  
5887  
5888  
5889  
5890  
5891  
5892  
5893  
5894  
5895  
5896  
5897  
5898  
5899 017166 005212  
5900 017170 022712 000204  
5901 017174 001022  
5902 017176 000277  
5903 017200 000251  
5904 017202 012700 100000  
5905 017206 101402  
5906 017210 102401  
5907 017212 100404  
5908  
5909  
5910  
5911  
5912 017214  
5913 017214 012742 000432  
5914 017220 005242  
5915 017222 000000  
5916  
5917 017224 000277  
5918 017226 000244  
5919 017230 012700 000000  
5920 017234 101002  
5921 017236 102401  
5922 017240 100004  
5923  
5924  
5925  
5926  
5927 017242  
5928 017242 012742 000433  
5929 017246 005242  
5930 017250 000000  
5931  
5932  
5933  
5934  
5935 017252 005212  
5936 017254 022712 000205  
5937 017260 001024

```
*****
:
: THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF A GROUP
: OF FOUR INSTRUCTIONS. THE GROUP CONSISTS OF THE INSTRUCTIONS:
: MOV, BIC, BIT, AND BIS. THESE INSTRUCTIONS ARE SIMILAR IN THE
: WAY THEY EFFECT THE C AND V BITS. THEY ALL LEAVE THE V-BIT
: CLEAR AND THE C-BIT UNAFFECTED.
: THE TEST PROCEDURE IS AS FOLLOWS: THE N, Z, AND V BITS
: ARE LOADED WITH THE COMPLEMENT OF THE EXPECTED RESULTS, THE C-BIT
: IS LOADED WITH THE DESIRED RESULT. THE INSTRUCTION IS EXECUTED
: WITH DIFFERENT DATA PATTERNS AND THE RESULTS ARE VERIFIED WITH
: A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS. THE DATA IS CHOSEN
: TO PRODUCT ALL POSSIBLE COMBINATIONS OF THE C AND V BITS.
:
:*****
:TEST 204 TEST MOV INSTRUCTION
:*****
TS204: INC (R2) ;UPDATE TEST NUMBER
      CMP #204,(R2) ;SEQUENCE ERROR?
      BNE TS205-10 ;BR TO ERROR HALT ON SEQ ERROR
      SCC ;CC=0110
      +CLN!CLC
      MOV #100000,R0 ;CC=1000
      BLOS MOV1
      BVS MOV1
      BMI MOV2
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 770 <====
:
MOV1: MOV #432,-(R2) ;MOVE TO MAILBOX # ***** 432 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;MOV DID NOT SET CC'S CORRECTLY
:
MOV2: SCC ;CC=1011
      CLZ
      MOV #0,R0 ;CC=0101
      BHI MOV3 ;C OR Z = 0?
      BVS MOV3 ;V=1?
      BPL TS205
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 755 <====
:
MOV3: MOV #433,-(R2) ;MOVE TO MAILBOX # ***** 433 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;MOV DID NOT SET CC'S CORRECTLY
: OR SEQUENCE ERROR
:*****
:TEST 205 TEST BIT INSTRUCTION
:*****
TS205: INC (R2) ;UPDATE TEST NUMBER
      CMP #205,(R2) ;SEQUENCE ERROR?
      BNE TS206-10 ;BR TO ERROR HALT ON SEQ ERROR
```



```
5938 017262 012700 100001      MOV      #100001,R0
5939 017266 000277              SCC                      ;CC=0110
5940 017270 000251      +CLM!CLC
5941 017272 032700 100000      BIT      #100000,R0      ;CC=1000
5942 017276 101402      BLOS     BIT1
5943 017300 102401      BVS      BIT1
5944 017302 100404      BMI      BIT2
5945                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5946                                ;                               CONDITIONAL BRANCH INST. AND <====
5947                                ;                               REPLACE THE MOVE INSTRUCTION <====
5948                                ;                               WHICH FOLLOWS W/ 766 <====
5949 017304      BIT1:
5950 017304 012742 000434      MOV      #434,-(R2)      ;MOVE TO MAILBOX # ***** 434 *****
5951 017310 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5952 017312 000000      HALT                    ;BIT DID NOT SET CC'S CORRECTLY
5953
5954 017314 000277      BIT2:  SCC                      ;CC=1011
5955 017316 000244      CLZ
5956 017320 032700 077776      BIT      #77776,R0      ;CC=0101
5957 017324 101002      BHI      BIT3
5958 017326 102401      BVS      BIT3
5959 017330 100004      BPL      TS206
5960                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5961                                ;                               CONDITIONAL BRANCH INST. AND <====
5962                                ;                               REPLACE THE MOVE INSTRUCTION <====
5963                                ;                               WHICH FOLLOWS W/ 753 <====
5964 017332      BIT3:
5965 017332 012742 000435      MOV      #435,-(R2)      ;MOVE TO MAILBOX # ***** 435 *****
5966 017336 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5967 017340 000000      HALT                    ;BIT DID NOT SET CC'S CORRECTLY
5968                                ; OR SEQUENCE ERROR
5969
5970                                ;*****
5971                                ;TEST 206      TEST BIC INSTRUCTION
5972                                ;*****
5972 017342 005212      TS206: INC      (R2)          ;UPDATE TEST NUMBER
5973 017344 022712 000206      CMP      #206,(R2)      ;SEQUENCE_ERROR?
5974 017350 001024      BNE      TS207-10      ;BR TO ERROR HALT ON SEQ ERROR
5975 017352 012700 177777      MOV      #177777,R0
5976 017356 000277      SCC                      ;CC=0110
5977 017360 000251      +CLM!CLC
5978 017362 042700 077777      BIC      #77777,R0      ;CC=1000
5979 017366 101402      BLOS     BIC1
5980 017370 102401      BVS      BIC1
5981 017372 100404      BMI      BIC2
5982                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5983                                ;                               CONDITIONAL BRANCH INST. AND <====
5984                                ;                               REPLACE THE MOVE INSTRUCTION <====
5985                                ;                               WHICH FOLLOWS W/ 766 <====
5986 017374      BIC1:
5987 017374 012742 000436      MOV      #436,-(R2)      ;MOVE TO MAILBOX # ***** 436 *****
5988 017400 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5989 017402 000000      HALT                    ;BIC DID NOT SET CC'S CORRECTLY
5990 017404 000277      BIC2:  SCC                      ;CC=1011
5991 017406 000244      CLZ
5992 017410 042700 100000      BIC      #100000,R0      ;CC=0101
5993 017414 101002      BHI      BIC3
```

```
5994 017416 102401          BVS    BIC3
5995 017420 100004          BPL    TS207
5996                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5997                                     ;                                     <====
5998                                     ;                                     <====
5999                                     ;                                     <====
6000 017422                                     BIC3:
6001 017422 012742 000437      MOV    #437,-(R2)      ;MOVE TO MAILBOX # ***** 437 *****
6002 017426 005242             INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6003 017430 000000             HALT                   ;BIC DID NOT SET CC'S CORRECTLY
6004                                     ; OR SEQUENCE ERROR
6005 ;*****
6006 ;TEST 207          TEST BIS INSTRUCTION
6007 ;*****
6008 017432 005212             TS207: INC    (R2)          ;UPDATE TEST NUMBER
6009 017434 022712 000207      CMP    #207,(R2)      ;SEQUENCE ERROR?
6010 017440 001025             BNE    TS210-10       ;BR TO ERROR HALT ON SEQ ERROR
6011 017442 005000             CLR    R0             ;R0=0
6012 017444 000277             SCC                   ;CC=1010
6013 017446 000251             +CLN!CLC
6014 017450 052700 000000      BIS    #0,R0          ;CC=0100 R0=0
6015 017454 103403             BCS    BIS1
6016 017456 102402             BVS    BIS1
6017 017460 100401             BMI    BIS1
6018 017462 001404             BEQ    BIS2
6019                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6020                                     ;                                     <====
6021                                     ;                                     <====
6022                                     ;                                     <====
6023 017464                                     BIS1:
6024 017464 012742 000440      MOV    #440,-(R2)      ;MOVE TO MAILBOX # ***** 440 *****
6025 017470 005242             INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6026 017472 000000             HALT                   ;BIS DID NOT SET CC'S CORRECTLY
6027 017474 000277             BIS2: SCC                   ;CC=0111
6028 017476 000250             CLN
6029 017500 052700 177777      BIS    #177777,R0     ;CC=1001
6030 017504 103003             BCC    BIS3
6031 017506 102402             BVS    BIS3
6032 017510 001401             BEQ    BIS3
6033 017512 100404             BMI    TS210
6034                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6035                                     ;                                     <====
6036                                     ;                                     <====
6037                                     ;                                     <====
6038 017514                                     BIS3:
6039 017514 012742 000441      MOV    #441,-(R2)      ;MOVE TO MAILBOX # ***** 441 *****
6040 017520 005242             INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6041 017522 000000             HALT                   ;BIS DID NOT SET CC'S CORRECTLY
6042                                     ; OR SEQUENCE ERROR
```

6043  
6044  
6045  
6046  
6047  
6048  
6049  
6050  
6051  
6052  
6053  
6054  
6055  
6056  
6057  
6058  
6059  
6060  
6061  
6062  
6063  
6064  
6065  
6066  
6067  
6068  
6069  
6070  
6071  
6072  
6073  
6074  
6075  
6076  
6077  
6078  
6079  
6080  
6081  
6082  
6083  
6084  
6085  
6086  
6087  
6088  
6089  
6090  
6091  
6092  
6093  
6094  
6095  
6096  
6097  
6098

017524 005212  
017526 022712 000210  
017532 001037  
017534 012700 077777  
017540 000257  
017542 000264  
017544 005200  
017546 101402  
017550 100001  
017552 102404  
  
017554  
017554 012742 000442  
017560 005242  
017562 000000  
017564 052700 077777  
017570 000261  
017572 000244  
017574 005200  
017576 100403  
017600 102402  
017602 103001  
017604 001404  
  
017606  
017606 012742 000443  
017612 005242  
017614 000000  
  
017616 000277  
017620 000241  
017622 005200  
017624 101402  
017626 100401  
017630 100004

```
.....  
: THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE INC AND  
: DEC INSTRUCTIONS. THESE INSTRUCTIONS BOTH EFFECT THE C AND V  
: BITS THE SAME; THE C-BIT IS LEFT UNCHANGED AND THE V-BIT IS DEPENDENT  
: UPON THE DATA RESULTS. THE SAME PROCEDURE IS USED. THE CONDITION  
: CODE BITS ARE INITIALIZED, THE INSTRUCTION IS EXECUTED AND THE  
: RESULTS ARE VERIFIED WITH A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.  
: THIS PROCEDURE IS REPEATED WITH SEVERAL DATA PATTERNS TO PRODUCE  
: DIFFERENT COMBINATIONS OF THE C AND V BITS.  
:.....  
: TEST 210 TEST INC INSTRUCTION  
:.....  
TS210: INC (R2) ;UPDATE TEST NUMBER  
CMP #210,(R2) ;SEQUENCE ERROR?  
BNE TS211-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #077777,R0 ;R0=077777  
CCC ;CC=0100  
SEZ  
INC R0 ;CC=1010 R0=10000  
BLOS INC1  
BPL INC1  
BVS INC2  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 767 <====  
  
INC1: MOV #442,-(R2) ;MOVE TO MAILBOX # ***** 442 *****  
INC INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;INC DID NOT SET CC'S CORRECTLY  
INC2: BIS #77777,R0 ;R0=177777  
SEC ;CC=1011  
CLZ  
INC R0 ;CC=0101 R0=0  
BMI INC3  
BVS INC3  
BCC INC3  
BEQ INC4  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 752 <====  
  
INC3: MOV #443,-(R2) ;MOVE TO MAILBOX # ***** 443 *****  
INC INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;INC DID NOT SET CC'S CORRECTLY  
  
INC4: SCC ;CC=1110  
CLC  
INC R0 ;CC=0000 R0=1  
BLOS INC5  
BMI INC5  
BPL TS211
```

```
6099 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6100 ; CONDITIONAL BRANCH INST. AND <====
6101 ; REPLACE THE MOVE INSTRUCTION <====
6102 ; WHICH FOLLOWS W/ 740 <====
6103 017632 INCS:
6104 017632 012742 000444 MOV #444,-(R2) ;MOVE TO MAILBOX # ***** 444 *****
6105 017636 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6106 017640 000000 HALT ;INC DID NOT SET CC'S CORRECTLY
6107 ; OR SEQUENCE ERROR
6108
6109 ;*****
6110 ;TEST 211 TEST DEC INSTRUCTION
6111 ;*****
6112 017642 005212 TS211: INC (R2) ;UPDATE TEST NUMBER
6113 017644 022712 000211 CMP #211,(R2) ;SEQUENCE ERROR?
6114 017650 001051 BNE TS212-10 ;BR TO ERROR HALT ON SEQ ERROR
6115 017652 012700 000002 MOV #2,R0 ;R0=2
6116 017656 000277 SCC ;CC=1111
6117 017660 005300 DEC R0 ;CC=0001 R0=1
6118 017662 100403 BMI DEC1
6119 017664 001402 BEQ DEC1
6120 017666 102401 BVS DEC1
6121 017670 103404 BCS DEC2
6122 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6123 ; CONDITIONAL BRANCH INST. AND <====
6124 ; REPLACE THE MOVE INSTRUCTION <====
6125 ; WHICH FOLLOWS W/ 767 <====
6126 017672 DEC1:
6127 017672 012742 000445 MOV #445,-(R2) ;MOVE TO MAILBOX # ***** 445 *****
6128 017676 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6129 017700 000000 HALT ;DEC DID NOT SET CC'S CORRECTLY
6130 017702 000261 DEC2: SEC ;CC=1011
6131 017704 000244 CLZ
6132 017706 005300 DEC R0 ;CC=0101 R0=0
6133 017710 101002 BHI DEC3
6134 017712 100401 BMI DEC3
6135 017714 102004 BVC DEC4
6136 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6137 ; CONDITIONAL BRANCH INST. AND <====
6138 ; REPLACE THE MOVE INSTRUCTION <====
6139 ; WHICH FOLLOWS W/ 755 <====
6140 017716 DEC3:
6141 017716 012742 000446 MOV #446,-(R2) ;MOVE TO MAILBOX # ***** 446 *****
6142 017722 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6143 017724 000000 HALT ;DEC DID NOT SET CC'S CORRECTLY
6144 017726 000277 DEC4: SCC ;CC=0110
6145 017730 000251 +CLN!CLC
6146 017732 005300 DEC R0 ;CC=1000 R0=17777
6147 017734 101402 BLOS DEC5
6148 017736 102401 BVS DEC5
6149 017740 100404 BMI DEC6
6150 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6151 ; CONDITIONAL BRANCH INST. AND <====
6152 ; REPLACE THE MOVE INSTRUCTION <====
6153 ; WHICH FOLLOWS W/ 743 <====
6154 017742 DEC5:
```



6176  
6177  
6178  
6179  
6180  
6181  
6182  
6183  
6184  
6185  
6186  
6187  
6188  
6189  
6190  
6191  
6192  
6193  
6194  
6195  
6196  
6197  
6198  
6199  
6200  
6201  
6202  
6203  
6204  
6205  
6206  
6207  
6208  
6209  
6210  
6211  
6212  
6213  
6214  
6215  
6216  
6217  
6218  
6219  
6220  
6221  
6222  
6223  
6224  
6225  
6226  
6227  
6228  
6229  
6230  
6231

020004 005212  
020006 022712 000212  
020012 001007  
020014 000277  
020016 000244  
020020 005000  
020022 100403  
020024 102402  
020026 103401  
020030 001404  
  
020032  
020032 012742 000451  
020036 005242  
020040 000000  
  
  
  
020042 005212  
020044 022712 000213  
020050 001022  
020052 000277  
020054 000244  
020056 005700  
020060 100403  
020062 102402  
020064 103401  
020066 001404  
  
  
020070  
020070 012742 000452  
020074 005242  
020076 000000  
020100 005300  
020102 000277

\*\*\*\*\*  
: THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE CLR,  
: TST, AND SWAB INSTRUCTIONS. THESE THREE INSTRUCTIONS ALL LEAVE  
: THE C AND V BITS CLEARED. AGAIN, THE CONDITION CODES ARE PRESET,  
: THE INSTRUCTION EXECUTED AND THE RESULTS CHECKED WITH CONDITIONAL  
: BRANCH INSTRUCTIONS. THE PROCEDURE IS REPEATED TO PRODUCE OTHER  
: COMBINATIONS OF CONDITION CODES.  
\*\*\*\*\*

TEST 212 TEST CLR INSTRUCTION

\*\*\*\*\*  
: TS212: INC (R2) ;UPDATE TEST NUMBER  
: CMP #212,(R2) ;SEQUENCE ERROR?  
: BNE TS213-10 ;BR TO ERROR HALT ON SEQ ERROR  
: SCC ;CC=1011  
: CLZ  
: CLR R0 ;CC=0100 R0=0  
: BMI CLR1  
: BVS CLR1  
: BCS CLR1  
: BEQ TS213  
  
: ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: ; CONDITIONAL BRANCH INST. AND <====  
: ; REPLACE THE MOVE INSTRUCTION <====  
: ; WHICH FOLLOWS W/ 770 <====

CLR1: MOV #451,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 451 \*\*\*\*\*  
: INC -(R2) ;SET MSGTYP TO FATAL ERROR  
: HALT ;CLR DID NOT SET CC'S CORRECTLY  
: ; OR SEQUENCE ERROR

TEST 213 TEST TST INSTRUCTION

\*\*\*\*\*  
: TS213: INC (R2) ;UPDATE TEST NUMBER  
: CMP #213,(R2) ;SEQUENCE ERROR?  
: BNE TS214-10 ;BR TO ERROR HALT ON SEQ ERROR  
: SCC ;CC=1011  
: CLZ  
: TST R0 ;CC=0100  
: BMI TEST1  
: BVS TEST1  
: BCS TEST1  
: BEQ TEST2  
  
: ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: ; CONDITIONAL BRANCH INST. AND <====  
: ; REPLACE THE MOVE INSTRUCTION <====  
: ; WHICH FOLLOWS W/ 770 <====

TEST1: MOV #452,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 452 \*\*\*\*\*  
: INC -(R2) ;SET MSGTYP TO FATAL ERROR  
: HALT ;TEST DID NOT SET CC'S CORRECTLY  
TEST2: DEC R0 ;MAKE R0 NEGATIVE  
: SCC ;CC=0111

|      |        |        |        |           |                       |  |       |
|------|--------|--------|--------|-----------|-----------------------|--|-------|
| 6232 | 020104 | 000250 |        | CLN       |                       |  |       |
| 6233 | 020106 | 005700 |        | TST       | RO                    | :CC=1000                                 |       |
| 6234 | 020110 | 101402 |        | BLOS      | TEST3                 |  |       |
| 6235 | 020112 | 102401 |        | BVS       | TEST3                 |  |       |
| 6236 | 020114 | 100404 |        | BMI       | TS214                 |  |       |
| 6237 |        |        |        |           |                       | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 6238 |        |        |        |           |                       | : CONDITIONAL BRANCH INST. AND           | <==== |
| 6239 |        |        |        |           |                       | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 6240 |        |        |        |           |                       | : WHICH FOLLOWS W/ 755                   | <==== |
| 6241 | 020116 |        |        | TEST3:    |                       |  |       |
| 6242 | 020116 | 012742 | 000453 | MOV       | #453,-(R2)            | :MOVE TO MAILBOX # ***** 453 *****       |       |
| 6243 | 020122 | 005242 |        | INC       | -(R2)                 | :SET MSGTYP TO FATAL ERROR               |       |
| 6244 | 020124 | 000000 |        | HALT      |                       | :TEST DID NOT SET CC'S CORRECTLY         |       |
| 6245 |        |        |        |           |                       | : OR SEQUENCE ERROR                      |       |
| 6246 |        |        |        |           |                       | .....                                    |       |
| 6247 |        |        |        | :TEST 214 | TEST SWAB INSTRUCTION |  |       |
| 6248 |        |        |        |           |                       | .....                                    |       |
| 6249 | 020126 | 005212 |        | TS214:    | INC (R2)              | :UPDATE TEST NUMBER                      |       |
| 6250 | 020130 | 022712 | 000214 | CMP       | #214,(R2)             | :SEQUENCE ERROR?                         |       |
| 6251 | 020134 | 001023 |        | BNE       | TS215-10              | :BR TO ERROR HALT ON SEQ ERROR           |       |
| 6252 | 020136 | 012700 | 170000 | MOV       | #170000,RO            | :RO=170000                               |       |
| 6253 | 020142 | 000277 |        | SCC       |                       | :CC=0111                                 |       |
| 6254 | 020144 | 000250 |        | CLN       |                       |  |       |
| 6255 | 020146 | 000300 |        | SWAB      | RO                    | :CC=1000 RO=360                          |       |
| 6256 | 020150 | 101402 |        | BLOS      | SWB1                  |  |       |
| 6257 | 020152 | 102401 |        | BVS       | SWB1                  |  |       |
| 6258 | 020154 | 100404 |        | BMI       | SWB2                  |  |       |
| 6259 |        |        |        |           |                       | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 6260 |        |        |        |           |                       | : CONDITIONAL BRANCH INST. AND           | <==== |
| 6261 |        |        |        |           |                       | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 6262 |        |        |        |           |                       | : WHICH FOLLOWS W/ 767                   | <==== |
| 6263 | 020156 |        |        | SWB1:     |                       |  |       |
| 6264 | 020156 | 012742 | 000454 | MOV       | #454,-(R2)            | :MOVE TO MAILBOX # ***** 454 *****       |       |
| 6265 | 020162 | 005242 |        | INC       | -(R2)                 | :SET MSGTYP TO FATAL ERROR               |       |
| 6266 | 020164 | 000000 |        | HALT      |                       | :SWAB DID NOT SET CC'S CORRECTLY         |       |
| 6267 | 020166 | 000277 |        | SWB2:     | SCC                   | :CC=1011                                 |       |
| 6268 | 020170 | 000244 |        | CLZ       |                       |  |       |
| 6269 | 020172 | 000300 |        | SWAB      | RO                    | :CC=0100 RO=170000                       |       |
| 6270 | 020174 | 102403 |        | BVS       | SWB3                  |  |       |
| 6271 | 020176 | 103402 |        | BCS       | SWB3                  |  |       |
| 6272 | 020200 | 100401 |        | BMI       | SWB3                  |  |       |
| 6273 | 020202 | 001404 |        | BEQ       | TS215                 |  |       |
| 6274 |        |        |        |           |                       | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 6275 |        |        |        |           |                       | : CONDITIONAL BRANCH INST. AND           | <==== |
| 6276 |        |        |        |           |                       | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 6277 |        |        |        |           |                       | : WHICH FOLLOWS W/ 754                   | <==== |
| 6278 | 020204 |        |        | SWB3:     |                       |  |       |
| 6279 | 020204 | 012742 | 000455 | MOV       | #455,-(R2)            | :MOVE TO MAILBOX # ***** 455 *****       |       |
| 6280 | 020210 | 005242 |        | INC       | -(R2)                 | :SET MSGTYP TO FATAL ERROR               |       |
| 6281 | 020212 | 000000 |        | HALT      |                       |  |       |

6282  
6283  
6284  
6285  
6286  
6287  
6288  
6289  
6290  
6291  
6292  
6293  
6294  
6295  
6296  
6297  
6298  
6299  
6300  
6301  
6302  
6303  
6304  
6305  
6306  
6307  
6308  
6309  
6310  
6311  
6312  
6313  
6314  
6315  
6316  
6317  
6318  
6319  
6320  
6321  
6322  
6323  
6324  
6325  
6326  
6327  
6328  
6329  
6330  
6331  
6332  
6333  
6334  
6335  
6336  
6337

020214 005212  
020216 022712 000215  
020222 001062  
020224 012700 040000  
020230 000277  
020232 062700 030000  
020236 101402  
020240 102401  
020242 100004  
  
020244  
020244 012742 000456  
020250 005242  
020252 000000  
020254 000264  
  
020256 062700 010000  
020262 101402  
020264 102001  
020266 100404  
  
020270  
020270 012742 000457  
020274 005242  
020276 000000  
020300 000257  
020302 000270  
020304 062700 100000  
020310 101002  
020312 102001  
020314 100004  
  
020316

.....  
THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE ADD AND  
ADC INSTRUCTIONS. BOTH OF THESE INSTRUCTIONS HANDLE THE C AND  
V BITS IDENTICALLY. THE PROCEDURE IS TO PRESET THE CONDITION  
CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND  
THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL  
BRANCHES. THIS PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT  
DATA TO PRODUCE EVERY COMBINATION OF C AND V BITS.  
.....  
TEST 215 TEST ADD INSTRUCTION  
.....  
TS215: INC (R2) ;UPDATE TEST NUMBER  
CMP #215,(R2) ;SEQUENCE ERROR?  
BNE TS216-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #40000,R0 ;R0=40000  
SCC ;CC=1111  
ADD #30000,R0 ;CC=0000 R0=70000  
BLOS ADD1  
BVS ADD1  
BPL ADD2  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 767 <====  
  
ADD1: MOV #456,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 456 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ADD DID NOT SET CC'S CORRECTLY  
ADD2: SEZ ;CC=0100  
  
ADD #10000,R0 ;CC=1010 40=100000  
BLOS ADD3  
BVC ADD3  
BMI ADD4  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 755 <====  
  
ADD3: MOV #457,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 457 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ADD DID NOT SET CC'S CORRECTLY  
ADD4: CCC ;CC=1000  
SEN  
ADD #100000,R0 ;CC=0111 R0=0  
BHI ADD5  
BVC ADD5  
BPL ADD6  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 742 <====  
  
ADD5:



```
6338 020316 012742 000460      MOV      #460,-(R2)      ;MOVE TO MAILBOX # ***** 460 *****
6339 020322 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6340 020324 000000              HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6341 020326 062700 177777      ADD6:  ADD      #177777,R0 ;CC=1000  R0=177777
6342 020332 101402              BLOS    ADD7
6343 020334 102401              BVS     ADD7
6344 020336 100404              BMI     ADD8
6345                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6346                                ;                                     <====
6347                                ;                                     <====
6348                                ;                                     <====
6349 020340              ADD7:
6350 020340 012742 000461      MOV      #461,-(R2)      ;MOVE TO MAILBOX # ***** 461 *****
6351 020344 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6352 020346 000000              HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6353 020350 000277      ADD8:  SCC
6354 020352 000245      +CLC!CLZ                ;CC=1010
6355 020354 062700 000001      ADD      #1,R0          ;CC=0101  R=0
6356 020360 102403              BVS     ADD9
6357 020362 103002              BCC     ADD9
6358 020364 100401              BMI     ADD9
6359 020366 001404              BEQ     TS216
6360                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6361                                ;                                     <====
6362                                ;                                     <====
6363                                ;                                     <====
6364 020370              ADD9:
6365 020370 012742 000462      MOV      #462,-(R2)      ;MOVE TO MAILBOX # ***** 462 *****
6366 020374 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6367 020376 000000              HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6368                                ; OR SEQUENCE ERROR
6369
6370 ;*****
6371 ;TEST 216      TEST ADC INSTRUCTION
6372 ;*****
6373 020400 005212      TS216: INC      (R2)          ;UPDATE TEST NUMBER
6374 020402 022712 000216      CMP      #216,(R2)      ;SEQUENCE ERROR?
6375 020406 001037              BNE     TS217-10        ;BR TO ERROR HALT ON SEQ ERROR
6376 020410 012700 077777      MOV      #077777,R0
6377 020414 000277              SCC
6378 020416 000252      +CLN!CLV                ;CC=0101
6379 020420 005500              ADC      R0              ;CC=1010
6380 020422 101402              BLOS    ADC1
6381 020424 102001              BVC     ADC1
6382 020426 100404              BMI     ADC2
6383                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6384                                ;                                     <====
6385                                ;                                     <====
6386                                ;                                     <====
6387 020430              ADC1:
6388 020430 012742 000463      MOV      #463,-(R2)      ;MOVE TO MAILBOX # ***** 463 *****
6389 020434 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6390 020436 000000              HALT                    ;ADC DID NOT SET CC'S CORRECTLY
6391 020440 052700 077777      ADC2:  BIS      #77777,R0
6392 020444 000277              SCC
6393 020446 000244              CLZ
```

|      |        |        |        |       |            |  |          |      |  |
|------|--------|--------|--------|-------|------------|--|----------|------|--|
| 6394 | 020450 | 005500 |        | ADC   | R0         |  | :CC=0101 | R0=0 |  |
| 6395 | 020452 | 101002 |        | BHI   | ADC3       |  |          |      |  |
| 6396 | 020454 | 102401 |        | BVS   | ADC3       |  |          |      |  |
| 6397 | 020456 | 100004 |        | BPL   | ADC4       |  |          |      |  |
| 6398 |        |        |        |       |            |  |          |      |  |
| 6399 |        |        |        |       |            |  |          |      |  |
| 6400 |        |        |        |       |            |  |          |      |  |
| 6401 |        |        |        |       |            |  |          |      |  |
| 6402 | 020460 |        |        | ADC3: |            |  |          |      |  |
| 6403 | 020460 | 012742 | 000464 | MOV   | #464,-(R2) |  |          |      |  |
| 6404 | 020464 | 005242 |        | INC   | -(R2)      |  |          |      |  |
| 6405 | 020466 | 000000 |        | HALT  |            |  |          |      |  |
| 6406 | 020470 | 000277 |        | ADC4: | SCC        |  |          |      |  |
| 6407 | 020472 | 000245 |        |       | +CLZ!CLC   |  |          |      |  |
| 6408 | 020474 | 005500 |        | ADC   | R0         |  |          |      |  |
| 6409 | 020476 | 102403 |        | BVS   | ADC5       |  |          |      |  |
| 6410 | 020500 | 103402 |        | BCS   | ADC5       |  |          |      |  |
| 6411 | 020502 | 100401 |        | BMI   | ADC5       |  |          |      |  |
| 6412 | 020504 | 001404 |        | BEQ   | TS217      |  |          |      |  |
| 6413 |        |        |        |       |            |  |          |      |  |
| 6414 |        |        |        |       |            |  |          |      |  |
| 6415 |        |        |        |       |            |  |          |      |  |
| 6416 |        |        |        |       |            |  |          |      |  |
| 6417 | 020506 |        |        | ADC5: |            |  |          |      |  |
| 6418 | 020506 | 012742 | 000465 | MOV   | #465,-(R2) |  |          |      |  |
| 6419 | 020512 | 005242 |        | INC   | -(R2)      |  |          |      |  |
| 6420 | 020514 | 000000 |        | HALT  |            |  |          |      |  |
| 6421 |        |        |        |       |            |  |          |      |  |

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
: CONDITIONAL BRANCH INST. AND  
: REPLACE THE MOVE INSTRUCTION  
: WHICH FOLLOWS W/ 753

: MOVE TO MAILBOX # \*\*\*\*\* 464 \*\*\*\*\*  
: SET MSGTYP TO FATAL ERROR  
: ADC DID NOT SET CC'S CORRECTLY

:CC=1010  
:CC=0100

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
: CONDITIONAL BRANCH INST. AND  
: REPLACE THE MOVE INSTRUCTION  
: WHICH FOLLOWS W/ 740

: MOVE TO MAILBOX # \*\*\*\*\* 465 \*\*\*\*\*  
: SET MSGTYP TO FATAL ERROR  
: ADC DID NOT SET CC'S CORRECTLY  
: OR SEQUENCE ERROR

6422  
6423  
6424  
6425  
6426  
6427  
6428  
6429  
6430  
6431  
6432  
6433  
6434  
6435  
6436 020516 005212  
6437 020520 022712 000217  
6438 020524 001042  
6439 020526 012700 000001  
6440 020532 000277  
6441 020534 000251  
6442 020536 005400  
6443 020540 103003  
6444 020542 102402  
6445 020544 001401  
6446 020546 100404  
6447  
6448  
6449  
6450  
6451 020550  
6452 020550 012742 000466  
6453 020554 005242  
6454 020556 000000  
6455 020560 042700 077777  
6456 020564 000257  
6457 020566 000264  
6458 020570 005400  
6459 020572 102003  
6460 020574 103002  
6461 020576 001401  
6462 020600 100404  
6463  
6464  
6465  
6466  
6467 020602  
6468 020602 012742 000467  
6469 020606 005242  
6470 020610 000000  
6471 020612 005000  
6472 020614 000277  
6473 020616 000244  
6474 020620 005400  
6475 020622 102403  
6476 020624 103402  
6477 020626 001001

\*\*\*\*\*  
THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE NEG,  
CMP, AND COM INSTRUCTIONS. EACH OF THESE INSTRUCTIONS GENERATE  
THE C AND V BITS IDENTICALLY. THE CONDITION CODES ARE PRESET,  
THE INSTRUCTIONS EXECUTED, AND THE RESULTS CHECKED WITH A SERIES  
OF CONDITIONAL BRANCH INSTRUCTIONS. THIS PROCEDURE IS REPEATED  
SEVERAL TIMES WITH DIFFERENT DATA IN ORDER TO GENERATE DIFFERENT  
COMBINATIONS OF THE C AND V BITS.  
\*\*\*\*\*

TEST 217 TEST NEG INSTRUCTION  
\*\*\*\*\*

TS217: INC (R2) ;UPDATE TEST NUMBER  
CMP #217,(R2) ;SEQUENCE ERROR?  
BNE TS220-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #1,R0  
SCC ;CC=0110  
+CLN!CLC  
NEG RO ;CC=1001 RO=177777  
BCC NEG1  
BVS NEG1  
BEQ NEG1  
BMI NEG2  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 766 <====

NEG1: MOV #466,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 466 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;NEG DID NOT SET CC'S CORRECTLY  
NEG2: BIC #77777,R0  
CCC ;CC=0100  
SEZ  
NEG RO ;CC=1011 RO=100000  
BVC NEG3  
BCC NEG3  
BEQ NEG3  
BMI NEG4

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 751 <====

NEG3: MOV #467,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 467 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;NEG DID NOT SET CC'S CORRECTLY  
NEG4: CLR RO  
SCC ;CC=1011  
CLZ  
NEG RO ;CC=0100 RO=0  
BVS NEG5  
BCS NEG5  
BNE NEG5

```
6478 020630 100004          BPL      TS220
6479
6480
6481
6482
6483 020632          NEG5:
6484 020632 012742 000470    MOV      #470,-(R2)
6485 020636 005242          INC      -(R2)
6486 020640 000000          HALT
6487
6488
6489
6490
6491
6492 020642 005212          TS220:  INC      (R2)
6493 020644 022712 000220    CMP      #220,(R2)
6494 020650 001060          BNE     TS221-10
6495 020652 012700 000005    MOV      #5,R0
6496 020656 000257          CCC
6497 020660 000271          +SEN!SEC
6498 020662 022700 000005    CMP      #5,R0
6499 020666 101002          BHI     CMP1
6500 020670 102401          BVS     CMP1
6501 020672 100004          BPL     CMP2
6502
6503
6504
6505
6506 020674          CMP1:
6507 020674 012742 000471    MOV      #471,-(R2)
6508 020700 005242          INC      -(R2)
6509 020702 000000          HALT
6510 020704 012700 100000    CMP2:  MOV      #100000,R0
6511 020710 000277          SCC
6512 020712 000242          CLV
6513 020714 020027 077777    CMP      R0,#77777
6514 020720 101402          BLOS   CMP3
6515 020722 102001          BVC     CMP3
6516 020724 100004          BPL     CMP4
6517
6518
6519
6520
6521 020726          CMP3:
6522 020726 012742 000472    MOV      #472,-(R2)
6523 020732 005242          INC      -(R2)
6524 020734 000000          HALT
6525 020736 052700 040000    CMP4:  BIS      #40000,R0
6526 020742 000257          CCC
6527 020744 000264          SEZ
6528 020746 022700 040000    CMP      #40000,R0
6529 020752 102003          BVC     CMP5
6530 020754 103002          BCC     CMP5
6531 020756 001401          BEQ     CMP5
6532 020760 100404          BMI     CMP6
6533
```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
CONDITIONAL BRANCH INST. AND  
REPLACE THE MOVE INSTRUCTION  
WHICH FOLLOWS W/ 735 <====

\*\*\*\*\*  
;TEST 220 TEST CMP INSTRUCTION  
\*\*\*\*\*

;UPDATE TEST NUMBER  
;SEQUENCE ERROR?  
;BR TO ERROR HALT ON SEQ ERROR

;CC=1010  
;CC=0101

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
CONDITIONAL BRANCH INST. AND  
REPLACE THE MOVE INSTRUCTION  
WHICH FOLLOWS W/ 766 <====

;MOVE TO MAILBOX # \*\*\*\*\* 471 \*\*\*\*\*  
;SET MSGTYP TO FATAL ERROR  
;CMP DID NOT SET CC'S CORRECTLY

;CC=1101  
;CC=0010

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
CONDITIONAL BRANCH INST. AND  
REPLACE THE MOVE INSTRUCTION  
WHICH FOLLOWS W/ 751 <====

;MOVE TO MAILBOX # \*\*\*\*\* 472 \*\*\*\*\*  
;SET MSGTYP TO FATAL ERROR  
;CMP DID NOT SET CC'S CORRECTLY  
;R0=140000  
;CC=0100  
;CC=1011

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====

```

6534          :          CONDITIONAL BRANCH INST. AND <====
6535          :          REPLACE THE MOVE INSTRUCTION <====
6536          :          WHICH FOLLOWS W/ 733 <====
6537 020762    CMP5:
6538 020762 012742 000473    MOV #473,-(R2) ;MOVE TO MAILBOX # ***** 473 *****
6539 020766 005242          INC -(R2) ;SET MSGTYP TO FATAL ERROR
6540 020770 000000          HALT ;CMP DID NOT SET CC'S CORRECTLY
6541 020772 042700 040000    CMP6: BIC #40000,R0
6542 020776 000277          SCC ;CC=1111
6543 021000 022700 177777    CMP #1,R0 ;CC=0000
6544 021004 101402          BLOS CMP7
6545 021006 102401          BVS CMP7
6546 021010 100004          BPL TS221
6547          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6548          :          CONDITIONAL BRANCH INST. AND <====
6549          :          REPLACE THE MOVE INSTRUCTION <====
6550          :          WHICH FOLLOWS W/ 717 <====
6551 021012    CMP7:
6552 021012 012742 000474    MOV #474,-(R2) ;MOVE TO MAILBOX # ***** 474 *****
6553 021016 005242          INC -(R2) ;SET MSGTYP TO FATAL ERROR
6554 021020 000000          HALT ;CMP DID NOT SET CC'S CORRECTLY
6555          : OR SEQUENCE ERROR
6556
6557 ;*****
6558 ;TEST 221 TEST COM INSTRUCTION
6559 ;*****
6560 021022 005212    TS221: INC (R2) ;UPDATE TEST NUMBER
6561 021024 022712 000221    CMP #221,(R2) ;SEQUENCE ERROR?
6562 021030 001010          BNE TS222-10 ;BR TO ERROR HALT ON SEQ ERROR
6563 021032 012700 177777    MOV #1,R0
6564 021036 000257          CCC ;CC=1010
6565 021040 000265          +SEC!SEZ
6566 021042 005100          COM R0 ;CC=0101
6567 021044 101002          BHI -COM1
6568 021046 102401          BVS COM1
6569 021050 100004          BPL TS222
6570          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6571          :          CONDITIONAL BRANCH INST. AND <====
6572          :          REPLACE THE MOVE INSTRUCTION <====
6573          :          WHICH FOLLOWS W/ 767 <====
6574 021052    COM1:
6575 021052 012742 000475    MOV #475,-(R2) ;MOVE TO MAILBOX # ***** 475 *****
6576 021056 005242          INC -(R2) ;SET MSGTYP TO FATAL ERROR
6577 021060 000000          HALT ;COM DID NOT SET CC'S CORRECTLY
6578          : OR SEQUENCE ERROR
6579

```

6580  
6581  
6582  
6583  
6584  
6585  
6586  
6587  
6588  
6589  
6590  
6591  
6592  
6593  
6594  
6595  
6596  
6597  
6598  
6599  
6600  
6601  
6602  
6603  
6604  
6605  
6606  
6607  
6608  
6609  
6610  
6611  
6612  
6613  
6614  
6615  
6616  
6617  
6618  
6619  
6620  
6621  
6622  
6623  
6624  
6625  
6626  
6627  
6628  
6629  
6630  
6631  
6632  
6633  
6634  
6635

021062 005212  
021064 022712 000222  
021070 001055  
021072 012700 125252  
021076 000257  
021100 000271  
021102 162700 125252  
021106 101002  
021110 102401  
021112 100004  
  
021114  
021114 012742 000476  
021120 005242  
021122 000000  
021124 052700 100000  
021130 000277  
021132 000242  
021134 162700 077777  
021140 101402  
021142 102001  
021144 100004  
  
021146  
021146 012742 000477  
021152 005242  
021154 000000  
021156 005100  
021160 000277  
  
021162 162700 100000  
021166 101402  
021170 102401  
021172 100004

```
*****  
: THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE SUB  
: AND SBC INSTRUCTIONS. BOTH OF THESE INSTRUCTIONS HANDLE THE  
: C AND V BITS IDENTICALLY. THE PROCEDURE IS TO PRESET THE CONDITION  
: CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND  
: THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL  
: BRANCHES. THIS PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT  
: DATA PATTERNS TO PROVIDE EVERY COMBINATION OF THE C AND V BITS.  
*****  
: TEST 222 TEST SUB INSTRUCTION  
*****  
TS222: INC (R2) ;UPDATE TEST NUMBER  
CMP #222,(R2) ;SEQUENCE ERROR?  
BNE TS223-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #125252,R0  
CCC ;CC=1010  
+SEN!SEC  
SUB #125252,R0 ;CC=0101 R0=0  
BHI SUB1  
BVS SUB1  
BPL SUB2  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 766 <====  
  
SUB1: MOV #476,-(R2) ;MOVE TO MAILBOX # ***** 476 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;SUB DID NOT SET CC'S CORRECTLY  
  
SUB2: BIS #100000,R0  
SCC ;CC=1101  
CLV  
SUB #77777,R0 ;CC=0010 R0=1  
BLOS SUB3  
BVC SUB3  
BPL SUB4  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 751 <====  
  
SUB3: MOV #477,-(R2) ;MOVE TO MAILBOX # ***** 477 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT  
  
SUB4: COM R0 ;R0=177777  
SCC ;CC=11111  
  
SUB #100000,R0 ;CC=0000 R0=77777  
BLOS SUB5  
BVS SUB5  
BPL SUB6  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====
```

```
6636                                     :           REPLACE THE MOVE INSTRUCTION <====
6637                                     :           WHICH FOLLOWS W/ 736           <====
6638 021174                               SUB5:
6639 021174 012742 000500                 MOV    #500,-(R2)      ;MOVE TO MAILBOX # ***** 500 *****
6640 021200 005242                       INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6641 021202 000000                       HALT
6642 021204 000257                               SUB6:
6643 021206 000264                               CCC
6644 021210 162700 140000                 SUB    #140000,R0    ;CC=1011
6645 021214 102003                       BVC    SUB7
6646 021216 103002                       BCC    SUB7
6647 021220 001401                       BEQ    SUB7
6648 021222 100404                       BMI    TS223
6649                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6650                                     ;           CONDITIONAL BRANCH INST. AND <====
6651                                     ;           REPLACE THE MOVE INSTRUCTION <====
6652                                     ;           WHICH FOLLOWS W/ 722         <====
6653 021224                               SUB7:
6654 021224 012742 000501                 MOV    #501,-(R2)      ;MOVE TO MAILBOX # ***** 501 *****
6655 021230 005242                       INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6656 021232 000000                       HALT
6657
6658 ;*****
6659 ;TEST 223 TEST SBC INSTRUCTION
6660 ;*****
6661 021234 005212                               TS223:
6662 021236 022712 000223                 INC    (R2)          ;UPDATE TEST NUMBER
6663 021242 001053                       CMP    #223,(R2)     ;SEQUENCE ERROR?
6664 021244 012700 000001                 BNE    TS224-10      ;BR TO ERROR HALT ON SEQ ERROR
6665 021250 000277                       MOV    #1,R0
6666 021252 000244                       SCC
6667 021254 005600                       CLZ
6668 021256 103403                       SBC    R0            ;CC=1010 R=0
6669 021260 102402                       BCS    SBC1
6670 021262 100401                       BVS    SBC1
6671 021264 001404                       BMI    SBC1
6672                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6673                                     ;           CONDITIONAL BRANCH INST. AND <====
6674                                     ;           REPLACE THE MOVE INSTRUCTION <====
6675                                     ;           WHICH FOLLOWS W/ 766         <====
6676 021266                               SBC1:
6677 021266 012742 000502                 MOV    #502,-(R2)      ;MOVE TO MAILBOX # ***** 502 *****
6678 021272 005242                       INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6679 021274 000000                       HALT
6680 021276 000277                               SBC2:
6681 021300 000245                               SCC
6682 021302 005600                       +CLZ!CLC
6683 021304 103403                       SBC    R0            ;CC=0100 R=0
6684 021306 102402                       BCS    SBC3
6685 021310 100401                       BVS    SBC3
6686 021312 001404                       BMI    SBC3
6687                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6688                                     ;           CONDITIONAL BRANCH INST. AND <====
6689                                     ;           REPLACE THE MOVE INSTRUCTION <====
6690                                     ;           WHICH FOLLOWS W/ 753         <====
6691 021314                               SBC3:
```

|      |        |        |        |       |      |            |  |       |
|------|--------|--------|--------|-------|------|------------|--|-------|
| 6692 | 021314 | 012742 | 000503 |       | MOV  | #503,-(R2) | :MOVE TO MAILBOX # ***** 503 *****       |       |
| 6693 | 021320 | 005242 |        |       | INC  | -(R2)      | :SET MSGTYP TO FATAL ERROR               |       |
| 6694 | 021322 | 000000 |        |       | HALT |            | :SBC DID NOT SET CC'S CORRECTLY          |       |
| 6695 | 021324 | 000277 |        | SBC4: | SCC  |            | :CC=0111                                 |       |
| 6696 | 021326 | 000250 |        |       | CLN  |            |  |       |
| 6697 | 021330 | 005600 |        |       | SBC  | R0         | :CC=1001 R0=177777                       |       |
| 6698 | 021332 | 103003 |        |       | BCC  | SBC5       |  |       |
| 6699 | 021334 | 102402 |        |       | BVS  | SBC5       |  |       |
| 6700 | 021336 | 001401 |        |       | BEQ  | SBC5       |  |       |
| 6701 | 021340 | 100404 |        |       | BMI  | SBC6       |  |       |
| 6702 |        |        |        |       |      |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 6703 |        |        |        |       |      |            | : CONDITIONAL BRANCH INST. AND           | <==== |
| 6704 |        |        |        |       |      |            | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 6705 |        |        |        |       |      |            | : WHICH FOLLOWS W/ 740                   | <==== |
| 6706 | 021342 |        |        | SBC5: |      |            |  |       |
| 6707 | 021342 | 012742 | 000504 |       | MOV  | #504,-(R2) | :MOVE TO MAILBOX # ***** 504 *****       |       |
| 6708 | 021346 | 005242 |        |       | INC  | -(R2)      | :SET MSGTYP TO FATAL ERROR               |       |
| 6709 | 021350 | 000000 |        |       | HALT |            | :SBC DID NOT SET CC'S CORRECTLY          |       |
| 6710 | 021352 | 042700 | 077777 | SBC6: | BIC  | #77777,R0  | :R0=100000                               |       |
| 6711 | 021356 | 000277 |        |       | SCC  |            | :CC=1101                                 |       |
| 6712 | 021360 | 000242 |        |       | CLV  |            |  |       |
| 6713 | 021362 | 005600 |        |       | SBC  | R0         | :CC=0010                                 |       |
| 6714 | 021364 | 101402 |        |       | BLOS | SBC7       |  |       |
| 6715 | 021366 | 102001 |        |       | BVC  | SBC7       |  |       |
| 6716 | 021370 | 100004 |        |       | BPL  | TS224      |  |       |
| 6717 |        |        |        |       |      |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 6718 |        |        |        |       |      |            | : CONDITIONAL BRANCH INST. AND           | <==== |
| 6719 |        |        |        |       |      |            | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 6720 |        |        |        |       |      |            | : WHICH FOLLOWS W/ 724                   | <==== |
| 6721 | 021372 |        |        | SBC7: |      |            |  |       |
| 6722 | 021372 | 012742 | 000505 |       | MOV  | #505,-(R2) | :MOVE TO MAILBOX # ***** 505 *****       |       |
| 6723 | 021376 | 005242 |        |       | INC  | -(R2)      | :SET MSGTYP TO FATAL ERROR               |       |
| 6724 | 021400 | 000000 |        |       | HALT |            | :SBC DID NOT SET CC'S CORRECTLY          |       |
| 6725 |        |        |        |       |      |            | : OR SEQUENCE ERROR                      |       |
| 6726 |        |        |        |       |      |            |  |       |



6727  
6728  
6729  
6730  
6731  
6732  
6733  
6734  
6735  
6736  
6737  
6738  
6739  
6740  
6741  
6742  
6743  
6744  
6745  
6746  
6747  
6748  
6749  
6750  
6751  
6752  
6753  
6754  
6755  
6756  
6757  
6758  
6759  
6760  
6761  
6762  
6763  
6764  
6765  
6766  
6767  
6768  
6769  
6770  
6771  
6772  
6773  
6774  
6775  
6776  
6777  
6778  
6779  
6780  
6781  
6782

021402 005212  
021404 022712 000224  
021410 001053  
021412 012700 144000  
021416 000257  
021420 000266  
021422 006100  
021424 103003  
021426 102402  
021430 001401  
021432 100404  
  
021434  
021434 012742 000506  
021440 005242  
021442 000000  
021444 000277  
021446 000243  
021450 006100  
021452 103003  
021454 102002  
021456 001401  
021460 100004  
  
021462  
021462 012742 000507  
021466 005242  
021470 000000  
021472 000277  
021474 000250  
021476 006100  
021500 101402  
021502 102401  
021504 100004

```
.....  
: THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF THE ROL,  
: ROR, ASL AND ASR INSTRUCTIONS. SPECIAL DATA PATTERNS ARE LOADED  
: AND ROTATED SEVERAL TIMES FOR EACH TEST. THE CONDITION CODES  
: ARE PRESET BEFORE EACH ROTATION AND THE CONDITION CODES ARE  
: CHECKED AFTER EACH ROTATION. THE FINAL CHECK IN EACH TEST IS  
: TO VERIFY THE CUMULATIVE DATA RESULT. THE DATA PATTERNS HAVE  
: BEEN SELECTED TO PRODUCE ALL COMBINATIONS OF THE C AND V BITS.  
:.....  
: TEST 224 TEST ROL INSTRUCTION  
:.....  
TS224: INC (R2) ;UPDATE TEST NUMBER  
CMP #224,(R2) ;SEQUENCE ERROR?  
BNE TS225-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #144000,R0 ;R0=144000  
CCC ;CC=0110  
+SEZ!SEV  
ROL R0 ;CC=1001 R0=110000  
BCC ROL1  
BVS ROL1  
BEQ ROL1  
BMI ROL2  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 766 <====  
  
ROL1: MOV #506,-(R2) ;MOVE TO MAILBOX # ***** 506 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT  
ROL2: SCC ;CC=1100  
+CLV!CLC  
ROL R0 ;CC=0011 R0=020000  
BCC ROL3  
BVC ROL3  
BEQ ROL3  
BPL ROL4  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 753 <====  
  
ROL3: MOV #507,-(R2) ;MOVE TO MAILBOX # ***** 507 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT  
ROL4: SCC ;ROL DID NOT SET CC'S CORRECTLY  
CLN ;CC=0111  
ROL R0 ;CC=0000 R0=040001  
BLOS ROL5  
BVS ROL5  
BPL ROL6  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====
```

```
6783
6784 021506
6785 021506 012742 000510
6786 021512 005242
6787 021514 000000
6788 021516 000257
6789 021520 000265
6790 021522 006100
6791 021524 101405
6792 021526 102004
6793 021530 100003
6794 021532 022700 100003
6795 021536 001404
6796
6797
6798
6799
6800 021540
6801 021540 012742 000511
6802 021544 005242
6803 021546 000000
6804
6805
6806
6807
6808 021550 005212
6809 021552 022712 000225
6810 021556 001051
6811 021560 012700 000023
6812 021564 000277
6813 021566 000250
6814 021570 006000
6815 021572 102403
6816 021574 103002
6817 021576 001401
6818 021600 100404
6819
6820
6821
6822
6823 021602
6824 021602 012742 000512
6825 021606 005242
6826 021610 000000
6827 021612 000257
6828 021614 000274
6829 021616 006000
6830 021620 102003
6831 021622 103002
6832 021624 001401
6833 021626 100004
6834
6835
6836
6837
6838 021630
```

```

: WHICH FOLLOWS W/ 741 <====
ROL5: MOV #510,-(R2) ;MOVE TO MAILBOX # ***** 510 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;ROL DID NOT SET CC'S CORRECTLY
ROL6: CCC ;CC=0101
      +SEZ!SEC ;CC=1010 R0=100003
      ROL R0
      BLOS ROL7
      BVC ROL7
      BPL ROL7
      CMP #100003,R0
      BEQ TS225
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 724 <====
ROL7: MOV #511,-(R2) ;MOVE TO MAILBOX # ***** 511 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;ROL MALFUNCTIONED
      ; OR SEQUENCE ERROR
:*****
:TEST 225 TEST ROR INSTRUCTION
:*****
TS225: INC (R2) ;UPDATE TEST NUMBER
      CMP #225,(R2) ;SEQUENCE ERROR?
      BNE TS226-10 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #23,R0 ;R0=23
      SCC ;CC=0111
      CLN
      ROR R0 ;CC=1001 R0=100011
      BVS ROR1
      BCC ROR1
      BEQ ROR1
      BMI ROR2
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 766 <====
ROR1: MOV #512,-(R2) ;MOVE TO MAILBOX # ***** 512 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;ROR DID NOT SET CC'S CORRECTLY
ROR2: CCC ;CC=1100
      +SEN!SEZ ;CC=0011 R0=040004
      ROR R0
      BVC ROR3
      BCC ROR3
      BEQ ROR3
      BPL ROR4
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 753 <====
ROR3:
```

```
6839 021630 012742 000513      MOV    #513,-(R2)      ;MOVE TO MAILBOX # ***** 513 *****
6840 021634 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6841 021636 000000              HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6842 021640 000277      ROR4:  SCC            ;CC=1110
6843 021642 000241              CLC
6844 021644 006000              ROR    R0              ;CC=0000  R0=020002
6845 021646 101403              BLOS   ROR5
6846 021650 102402              BVS    ROR5
6847 021652 001401              BEQ    ROR5
6848 021654 100004              BPL    ROR6
6849
6850
6851
6852
6853 021656
6854 021656 012742 000514      ROR5:  MOV    #514,-(R2) ;MOVE TO MAILBOX # ***** 514 *****
6855 021662 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6856 021664 000000              HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6857 021666 000257      ROR6:  CCC            ;CC=0101
6858 021670 000265              +SEC!SEZ
6859 021672 006000              ROR    R0              ;CC=1010  R0=110001
6860 021674 101402              BLOS   ROR7
6861 021676 102001              BVC    ROR7
6862 021700 100404              BMI    TS226
6863
6864
6865
6866
6867 021702
6868 021702 012742 000515      ROR7:  MOV    #515,-(R2) ;MOVE TO MAILBOX # ***** 515 *****
6869 021706 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6870 021710 000000              HALT                    ;ROR DID NOT PRODUCE CORRECT RESULTS
6871
6872
6873
6874
6875 021712 005212
6876 021714 022712 000226      TS226: INC    (R2)          ;UPDATE TEST NUMBER
6877 021720 001054              CMP    #226,(R2)      ;SEQUENCE ERROR?
6878 021722 012700 144000      BNE    TS227-10        ;BR TO ERROR HALT ON SEQ ERROR
6879 021726 000257              MOV    #144000,R0     ;R0=14000
6880 021730 000271              CCC            ;CC=0110
6881 021732 006300              +SEN!SEC
6882 021734 103003              ASL    R0              ;CC=1001  R0=110000
6883 021736 102402              BCC    ASL1
6884 021740 001401              BVS    ASL1
6885 021742 100404              BEQ    ASL1
6886
6887
6888
6889
6890 021744
6891 021744 012742 000516      ASL1:  MOV    #516,-(R2) ;MOVE TO MAILBOX # ***** 516 *****
6892 021750 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6893 021752 000000              HALT
6894 021754 000277      ASL2:  SCC            ;CC=1100

;*****
;TEST 226 TEST ASL INSTRUCTION
;*****
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 740 <====

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 726 <====

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
```

```
6895 021756 000243 +CLV!CLC
6896 021760 006300 ASL R0 ;CC=0011 R0=020000
6897 021762 103003 BCC ASL3
6898 021764 102002 BVC ASL3
6899 021766 001401 BEQ ASL3
6900 021770 100004 BPL ASL4
6901 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6902 ; CONDITIONAL BRANCH INST. AND <====
6903 ; REPLACE THE MOVE INSTRUCTION <====
6904 ; WHICH FOLLOWS W/ 753 <====
6905 021772 ASL3:
6906 021772 012742 000517 MOV #517,-(R2) ;MOVE TO MAILBOX # ***** 517 *****
6907 021776 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6908 022000 000000 HALT ;ASL DID NOT SET CC'S CORRECTLY
6909 022002 000277 ASL4: SCC ;CC=0111
6910 022004 000250 CLN
6911 022006 006300 ASL R0 ;CC=0000 R0=040000
6912 022010 101402 BLOS ASL5
6913 022012 102401 BVS ASL5
6914 022014 100004 BPL ASL6
6915 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6916 ; CONDITIONAL BRANCH INST. AND <====
6917 ; REPLACE THE MOVE INSTRUCTION <====
6918 ; WHICH FOLLOWS W/ 741 <====
6919 022016 ASL5:
6920 022016 012742 000520 MOV #520,-(R2) ;MOVE TO MAILBOX # ***** 520 *****
6921 022022 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6922 022024 000000 HALT ;ASL DID NOT SET CC'S CORRECTLY
6923 022026 000257 ASL6: CCC ;CC=0101
6924 022030 000265 +SEZ!SEC
6925 022032 006300 ASL R0 ;CC=1010 R0=100000
6926 022034 103406 BCS ASL7
6927 022036 001405 BEQ ASL7
6928 022040 102004 BVC ASL7
6929 022042 100003 BPL ASL7
6930 022044 022700 100000 CMP #100000,R0
6931 022050 001404 BEQ TS227
6932 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6933 ; CONDITIONAL BRANCH INST. AND <====
6934 ; REPLACE THE MOVE INSTRUCTION <====
6935 ; WHICH FOLLOWS W/ 723 <====
6936 022052 ASL7:
6937 022052 012742 000521 MOV #521,-(R2) ;MOVE TO MAILBOX # ***** 521 *****
6938 022056 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6939 022060 000000 HALT ;ASL MALFUNCTIONED
6940 ; OR SEQUENCE ERROR
```

```
6941
6942
6943
6944 022062 005212
6945 022064 022712 000227
6946 022070 001060
6947 022072 012700 100023
6948 022076 000277
6949 022100 000250
6950 022102 006200
6951 022104 102403
6952 022106 103002
6953 022110 001401
6954 022112 100404
6955
6956
6957
6958
6959 022114
6960 022114 012742 000522
6961 022120 005242
6962 022122 000000
6963 022124 042700 100000
6964 022130 000277
6965 022132 000243
6966 022134 006200
6967 022136 102003
6968 022140 103002
6969 022142 001401
6970 022144 100004
6971
6972
6973
6974
6975 022146
6976 022146 012742 000523
6977 022152 005242
6978 022154 000000
6979 022156 000277
6980
6981 022160 006200
6982 022162 101403
6983 022164 102402
6984 022166 001401
6985 022170 100004
6986
6987
6988
6989
6990 022172
6991 022172 012742 000524
6992 022176 005242
6993 022200 000000
6994 022202 052700 100000
6995 022206 000257
6996 022210 000265

;*****
;TEST 227 TEST ASR INSTRUCTION
;*****
TS227: INC (R2) ;UPDATE TEST NUMBER
CMP #227,(R2) ;SEQUENCE ERROR?
BNE TS230-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #100023,R0 ;R0=100023
SCC ;CC=0110
CLN
ASR R0 ;CC=1001 RP=140011
BVS ASR1
BCC ASR1
BEQ ASR1
BMI ASR2

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====

ASR1: MOV #522,-(R2) ;MOVE TO MAILBOX # ***** 522 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ASR DID NOT SET CC'S CORRECTLY
ASR2: BIC #100000,R0 ;R0=40011
SCC ;CC=1100
+CLV!CLC
ASR R0 ;CC=0011 R0=020004
BVC ASR3
BCC ASR3
BEQ ASR3
BPL ASR4

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 751 <====

ASR3: MOV #523,-(R2) ;MOVE TO MAILBOX # ***** 523 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ASR DID NOT SET CC'S CORRECTLY
ASR4: SCC ;CC=1111

;CC=0000 R0=010002

ASR5: ASR R0
BLOS ASR5
BVS ASR5
BEQ ASR5
BPL ASR6

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 737 <====

ASR5: MOV #524,-(R2) ;MOVE TO MAILBOX # ***** 524 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ASR DID NOT SET CC'S CORRECTLY
ASR6: BIS #100000,R0 ;R0=110002
CCC ;CC=0101
+SEZ!SEC
```

```
6997 022212 006200 ASR R0 ;C=1010 R0=144001
6998 022214 101406 BLOS ASR7
6999 022216 102005 BVC ASR7
7000 022220 100004 BPL ASR7
7001 022222 001403 BEQ ASR7
7002 022224 022700 144001 CMP #144001,R0 ;CHECK RESULT OF ASR'S
7003 022230 001404 BEQ TS230
7004 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7005 ; CONDITIONAL BRANCH INST. AND <====
7006 ; REPLACE THE MOVE INSTRUCTION <====
7007 ; WHICH FOLLOWS W/ 717 <====
7008 022232 ASR7:
7009 022232 012742 000525 MOV #525,-(R2) ;MOVE TO MAILBOX # ***** 525 *****
7010 022236 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7011 022240 000000 HALT ;ASR DID NOT FUNCTION CORRECTLY
7012 ; OR SEQUENCE ERROR
7013
7014
7015
7016
7017
7018
7019
7020
7021
7022
7023
7024
7025
7026
7027 022242 005212 TS230: INC (R2) ;UPDATE TEST NUMBER
7028 022244 022712 000230 CMP #230,(R2) ;SEQUENCE ERROR?
7029 022250 001033 BNE TS231-10 ;BR TO ERROR HALT ON SEQ ERROR
7030 022252 005000 CLR R0
7031 022254 000277 SCC ;SET CC=1011
7032 022256 000244 CLZ
7033 022260 006700 SXT R0 ;TRY SXT
7034 022262 100006 BPL SXT0 ;TEST CC=1001
7035 022264 001405 BEQ SXT0
7036 022266 102404 BVS SXT0
7037 022270 103003 BCC SXT0
7038 022272 022700 177777 CMP #-1,R0 ;CHECK DATA RESULT
7039 022276 001404 BEQ SXT1
7040 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7041 ; CONDITIONAL BRANCH INST. AND <====
7042 ; REPLACE THE MOVE INSTRUCTION <====
7043 ; WHICH FOLLOWS W/ 764 <====
7044 022300 SXT0:
7045 022300 012742 000526 MOV #526,-(R2) ;MOVE TO MAILBOX # ***** 526 *****
7046 022304 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7047 022306 000000 HALT ;RESULTS OF SXT INCORRECT
7048 022310 005000 SXT1: CLR R0 ;R0=0
7049 022312 005010 CLR (R0) ;LOC. 0=0
7050 022314 005110 COM (R0) ;LOC. 0=177777
7051 022316 000257 CCC ;SET CC=0110
7052 022320 000266 +SEZ!SEV
```

7053 022322 006710  
7054 022324 001005  
7055 022326 103404  
7056 022330 102403  
7057 022332 100402  
7058 022334 005710  
7059 022336 001404  
7060  
7061  
7062  
7063  
7064 022340  
7065 022340 012742 000527  
7066 022344 005242  
7067 022346 000000  
7068

SXT (R0)  
BNE SXT2  
BCS SXT2  
BVS SXT2  
BMI SXT2  
TST (R0)  
BEQ TS231

;TEST CC=0100

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 744 <====

SXT2:

MOV #527,-(R2)  
INC -(R2)  
HALT

;MOVE TO MAILBOX # \*\*\*\*\* 527 \*\*\*\*\*  
;SET MSGTYP TO FATAL ERROR  
;RESULTS OF SXT INCORRECT  
; OR SEQUENCE ERROR

7069  
7070  
7071  
7072  
7073  
7074  
7075  
7076  
7077  
7078  
7079  
7080 022350 005212  
7081 022352 022712 000231  
7082 022356 001035  
7083 022360 012700 007463  
7084 022364 012701 031525  
7085 022370 000277  
7086 022372 000241  
7087 022374 074100  
7088 022376 101406  
7089 022400 102405  
7090 022402 001404  
7091 022404 100403  
7092 022406 022700 036146  
7093 022412 001404  
7094  
7095  
7096  
7097  
7098 022414  
7099 022414 012742 000530  
7100 022420 005242  
7101 022422 000000  
7102 022424 010104  
7103 022426 000261  
7104 022430 000241  
7105 022432 074400  
7106 022434 101406  
7107 022436 102405  
7108 022440 001404  
7109 022442 100403  
7110 022444 022700 007463  
7111 022450 001404  
7112  
7113  
7114  
7115  
7116 022452  
7117 022452 012742 000531  
7118 022456 005242  
7119 022460 000000  
7120

```
.....  
: THIS TEST VERIFIES THE XOR INSTRUCTION. UNIQUE PATTERNS  
: OF ONES AND ZEROES ARE MOVED TO DATA REGISTERS R0 AND R1.  
: AFTER THE FIRST XOR INSTRUCTION R0=36146. AN XOR IS THEN  
: EXECUTED WITH THIS NEW VALUE AND THE CONTENTS OF R1 TO  
: REPRODUCE THE ORIGINAL VALUE IF R0=31525.  
:.....  
: TEST 231 TEST THE XOR INSTRUCTION  
:.....  
TS231: INC (R2) ;UPDATE TEST NUMBER  
CMP #231,(R2) ;SEQUENCE ERROR?  
BNE TS232-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #7463,R0 ;SET UP R0  
MOV #31525,R1 ;SET UP R1  
SCC ;SET CC=1110  
CLC  
XOR R1,R0 ;TRY XOR  
BLOS XOR1 ;CC=0000?  
BVS XOR1  
BEQ XOR1  
BMI XOR1  
CMP #36146,R0 ;DATA RESULT CORRECT?  
BEQ XOR2  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 761 <====  
XOR1: MOV #530,-(R2) ;MOVE TO MAILBOX # ***** 530 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
: ;  
XOR2: MOV R1,R4  
SEC ;CC=1110  
CLC  
XOR R4,R0 ;TRY XOR MODE 0,0  
BLOS XOR3 ;CC=0000?  
BVS XOR3  
BEQ XOR3  
BMI XOR3  
CMP #7463,R0  
BEQ TS232  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 742 <====  
XOR3: MOV #531,-(R2) ;MOVE TO MAILBOX # ***** 531 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
: ;  
: RESULT OF XOR INCORRECT  
: OR SEQUENCE ERROR
```





7165  
7166  
7167  
7168  
7169  
7170  
7171  
7172  
7173  
7174  
7175 022550 005212  
7176 022552 022712 000233  
7177 022556 001061  
7178 022560 012706 001000  
7179 022564 012746 125252  
7180 022570 162706 000074  
7181 022574 012705 022620  
7182 022600 012746 006436  
7183 022604 000277  
7184 022606 000116  
7185 022610 012742 000534  
7186 022614 005242  
7187 022616 000000  
7188 022620 101010  
7189 022622 100007  
7190 022624 102006  
7191 022626 020527 125252  
7192 022632 001003  
7193 022634 022706 001000  
7194 022640 001404  
7195  
7196  
7197  
7198  
7199 022642  
7200 022642 012742 000535  
7201 022646 005242  
7202 022650 000000  
7203 022652 012746 052525  
7204 022656 012746 006400  
7205 022662 010605  
7206 022664 004737 022674  
7207 022670 000137 022706  
7208 022674 000205  
7209 022676 012742 000536  
7210 022702 005242  
7211 022704 000000  
7212 022706 022706 001000  
7213 022712 001003  
7214 022714 022705 052525  
7215 022720 001404  
7216  
7217  
7218  
7219  
7220 022722

```
*****
:
: THIS TEST VERIFIES THE MARK INSTRUCTION. THE EFFECTS
: OF THE MARK INSTRUCTION ARE SIMULATED BY THE PROGRAM INSTRUCTIONS.
: THE CONTENTS OF R5 AND THE STACK POINTER ARE CHECKED AFTER EACH
: OF THE TWO ROUTINES IN THE TEST.
:
:*****
:TEST 233 TEST MARK INSTRUCTION
:*****
1S233: INC (R2) ;UPDATE TEST NUMBER
      CMP #233,(R2) ;SEQUENCE ERROR?
      BNE TS234-10 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #STBOT,SP
      MOV #125252,-(SP) ;PUT R5 VALUE ON STACK
      SUB #74,SP ;EFFECTIVELY PUT 36 ARGUMENTS ON STACK
      MOV #MRK1,R5 ;SET NEW PC IN R5
      MOV #6436,-(SP) ;PUT MARK 36 INST. ON STACK
      SCC ;SET CC=1111
      JMP (SP) ;XFER CONTL TO MARK 36 INST. ON STACK
      MOV #534,-(R2) ;MOVE TO MAILBOX # ***** 534 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;MARK INST. SHOULD HAVE JUMPED TO MRK1
MRK1: BHI MRK2 ;TEST CC UNAFFECTED
      BPL MRK2 ;IE. CC=1111
      BVC MRK2
      CMP R5,#125252 ;CHECK R5 RESTORED FROM STACK
      BNE MRK2
      CMP #STBOT,R6 ;CHECK STACK POINTER READJUSTED CORRECTLY.
      BEQ MRK3
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 746 <====
MRK2: MOV #535,-(R2) ;MOVE TO MAILBOX # ***** 535 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;RESULTS OF MARK INCORRECT
MRK3: MOV #52525,-(SP) ;PUT MARK 0 INST. ON STACK
      MOV #6400,-(SP) ;SET ADDR. OF MARK INST. IN R5
      MOV SP,R5 ;DO JSR
      JSR @MRK4 ;DO JSR
      JMP @MRK5
MRK4: RTS R5 ;DO RTS WITH R5 TO MARK INST ON STACK
      MOV #536,-(R2) ;MOVE TO MAILBOX # ***** 536 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;RTS,MARK SEQUENCE FAILED
MRK5: CMP #STBOT,R6 ;STACK ADJUSTED CORRECTLY
      BNE MRK6 ;IF NOT: BR
      CMP #52525,R5 ;CHECK IF R5 RESTORED FROM STACK
      BEQ TS234
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 716 <====
MRK6:
```

7221 022722 012742 000537  
7222 022726 005242  
7223 022730 000000  
7224

MOV #537,-(R2)  
INC -(R2)  
HALT

:MOVE TO MAILBOX # \*\*\*\*\* 537 \*\*\*\*\*  
:SET MSGTYP TO FATAL ERROR  
:RESULTS OF MARK INCORRECT  
: OR SEQUENCE ERROR

7225 177776

PS=177776

7226  
7227  
7228  
7229  
7230  
7231  
7232  
7233  
7234  
7235  
7236  
7237  
7238  
7239  
7240  
7241  
7242  
7243  
7244  
7245  
7246  
7247  
7248  
7249  
7250  
7251  
7252  
7253  
7254  
7255  
7256  
7257  
7258  
7259  
7260  
7261  
7262  
7263  
7264  
7265  
7266  
7267  
7268  
7269  
7270  
7271  
7272  
7273  
7274  
7275  
7276  
7277  
7278  
7279  
7280

005212  
022712 000234  
001024  
012700 000377  
000257  
106400  
022767 000357 155016  
001404  
  
012742 000540  
005242  
000000  
005000  
005010  
000277  
106410  
100403  
102402  
103401  
001004  
  
012742 000541  
005242  
000000  
  
005212  
022712 000235  
001021  
005000  
012710 177777  
005037 177776  
106420

\*\*\*\*\*  
THESE NEXT SEVEN TESTS VERIFY THE MTPS INSTRUCTION IN ALL  
MODES. THE PSW IS DEFINED BY AN EQUATE STATEMENT BEFORE THE  
FIRST MTPS TEST. IN EACH TEST A PATTERN OF ONES AND  
ZEROS IS SET IN A DATA REGISTER AND MOVED TO THE PSW.  
THE DATA IN THE PSW, AND THE DATA REGISTER ADDRESS,  
ARE CHECKED TO VERIFY PROPER EXECUTION OF THE INSTRUCTION.  
\*\*\*\*\*

TEST 234 TEST MTPS INSTRUCTION

```
TS234: INC (R2) ;UPDATE TEST NUMBER
        CMP #234,(R2) ;SEQUENCE ERROR?
        BNE TS235-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #377,R0
        CCC
        MTPS R0
        CMP #357,PS
        BEQ MTPS1

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
        MOV #540,-(R2) ;MOVE TO MAILBOX # ***** 540 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;MTPS FAILED
MTPS1: CLR R0
        CLR (R0)
        SCC ;CC=1111
        MTPS (R0) ;TRY MTPS MODE 1
        BMI MTPS1A ;CHECK PS
        BVS MTPS1A
        BCS MTPS1A
        BNE TS235
```

```
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 753 <====
MTPS1A: MOV #541,-(R2) ;MOVE TO MAILBOX # ***** 541 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;MTPS FAILED
        ; OR SEQUENCE ERROR
```

TEST 235 TEST MTPS MODE 2

```
TS235: INC (R2) ;UPDATE TEST NUMBER
        CMP #235,(R2) ;SEQUENCE ERROR?
        BNE TS236-10 ;BR TO ERROR HALT ON SEQ ERROR
        CLR R0 ;R0=0
        MOV #-1,(R0) ;LOC. 0=-1
        CLR @#PS ;PS=0
        MTPS (R0)+ ;TRY MTPS W/MODE 2
```

```
7281 023046 022737 000357 177776      CMP      #357,@#PS      :CHECK DATA
7282 023054 001404                      BEQ      MTPS2
7283                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7284                                     :          CONDITIONAL BRANCH INST. AND <====
7285                                     :          REPLACE THE MOVE INSTRUCTION <====
7286                                     :          WHICH FOLLOWS W/ 765 <====
7287 023056 012742 000542      MOV      #542,-(R2)      :MOVE TO MAILBOX # ***** 542 *****
7288 023062 005242      INC      -(R2)          :SET MSGTYP TO FATAL ERROR
7289 023064 000000      HALT
7290 023066 022700 000001      MTPS2:  CMP      #1,R0      :DEST. DATA INCORRECT
7291 023072 001404      BEQ      TS236          :CHECK DEST. REGISTER.
7292                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7293                                     :          CONDITIONAL BRANCH INST. AND <====
7294                                     :          REPLACE THE MOVE INSTRUCTION <====
7295                                     :          WHICH FOLLOWS W/ 756 <====
7296 023074 012742 000543      MOV      #543,-(R2)      :MOVE TO MAILBOX # ***** 543 *****
7297 023100 005242      INC      -(R2)          :SET MSGTYP TO FATAL ERROR
7298 023102 000000      HALT          :DEST REGISTER NOT INCREMENTED BY 1
7299                                     : OR SEQUENCE ERROR
7300
7301
7302
7303
```

\*\*\*\*\*  
:TEST 236 TEST MTPS MODE 3  
\*\*\*\*\*

```
7304 023104 005212      TS236:  INC      (R2)          :UPDATE TEST NUMBER
7305 023106 022712 000236      CMP      #236,(R2)      :SEQUENCE ERROR?
7306 023112 001024      BNE      TS237-10      :BR TO ERROR HALT ON SEQ ERROR
7307 023114 012700 000402      MOV      #402,R0        :R0=402
7308 023120 005010      CLR      (R0)          :LOC. 402=0
7309 023122 012737 052652 000000      MOV      #52652,@#0     :LOC. 0=52652
7310 023130 005037 177776      CLR      @#PS          :PS=0
7311 023134 106430      MTPS    @ (R0)+        :TRY MTPS W/MODE 3
7312 023136 022737 000252 177776      CMP      #252,@#PS      :CHECK DEST. DATA
7313 023144 001404      BEQ      MTPS3
7314                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7315                                     :          CONDITIONAL BRANCH INST. AND <====
7316                                     :          REPLACE THE MOVE INSTRUCTION <====
7317                                     :          WHICH FOLLOWS W/ 762 <====
7318 023146 012742 000544      MOV      #544,-(R2)      :MOVE TO MAILBOX # ***** 544 *****
7319 023152 005242      INC      -(R2)          :SET MSGTYP TO FATAL ERROR
7320 023154 000000      HALT          :DEST. DATA INCORRECT
7321 023156 022700 000404      MTPS3:  CMP      #404,R0      :CHECK MODE 3 REGISTER.
7322 023162 001404      BEQ      TS237
7323                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7324                                     :          CONDITIONAL BRANCH INST. AND <====
7325                                     :          REPLACE THE MOVE INSTRUCTION <====
7326                                     :          WHICH FOLLOWS W/ 753 <====
7327 023164 012742 000545      MOV      #545,-(R2)      :MOVE TO MAILBOX # ***** 545 *****
7328 023170 005242      INC      -(R2)          :SET MSGTYP TO FATAL ERROR
7329 023172 000000      HALT          :MODE 3 REGISTER INCORRECT
7330                                     : OR SEQUENCE ERROR
7331
7332
```

\*\*\*\*\*  
:TEST 237 TEST MTPS MODE 4  
\*\*\*\*\*

```
7333
7334
7335 023174 005212      TS237:  INC      (R2)          :UPDATE TEST NUMBER
7336 023176 022712 000237      CMP      #237,(R2)      :SEQUENCE ERROR?
```

```
7337 023202 001022      BNE    TS240-10      ;BR TO ERROR HALT ON SEQ ERROR
7338 023204 012700 000001  MOV    #1,R0         ;RO=1
7339 023210 012737 125125 000000  MOV    #125125,@#0   ;LOC. 0 = 125125
7340 023216 005037 177776      CLR    @#PS          ;PS=0
7341 023222 106440      MTPS   -(R0)         ;TRY MTPS W/MODE 4
7342 023224 022737 000105 177776  CMP    #105,@#PS     ;CHECK DEST. DATA
7343 023232 001404      BEQ    MTPS4         ;
7344      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7345      ;          CONDITIONAL BRANCH INST. AND <====
7346      ;          REPLACE THE MOVE INSTRUCTION <====
7347      ;          WHICH FOLLOWS W/ 763 <====
7348 023234 012742 000546      MOV    #546,-(R2)    ;MOVE TO MAILBOX # ***** 546 *****
7349 023240 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
7350 023242 000000      HALT                   ;DEST. DATA INCORRECT
7351 023244 005700      MTPS4: TST   R0       ;CHECK MODE 4 REGISTER
7352 023246 001404      BEQ    TS240         ;
7353      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7354      ;          CONDITIONAL BRANCH INST. AND <====
7355      ;          REPLACE THE MOVE INSTRUCTION <====
7356      ;          WHICH FOLLOWS W/ 755 <====
7357 023250 012742 000547      MOV    #547,-(R2)    ;MOVE TO MAILBOX # ***** 547 *****
7358 023254 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
7359 023256 000000      HALT                   ;MODE 4 REGISTER NOT DECREMENTED BY 1
7360      ; OR SEQUENCE ERROR
7361
7362 ;*****
7363 ;TEST 240      TEST MTPS MODE 5
7364 ;*****
7365 023260 005212      TS240: INC   (R2)     ;UPDATE TEST NUMBER
7366 023262 022712 000240      CMP    #240,(R2)    ;SEQUENCE ERROR?
7367 023266 001021      BNE    TS241-10     ;BR TO ERROR HALT ON SEQ ERROR
7368 023270 012700 000404      MOV    #404,R0      ;RO=404
7369 023274 012737 177400 000000  MOV    #177400,@#0   ;LOC. 0=177400
7370 023302 000277      SCC                   ;SET ALL COND. CODES
7371 023304 106450      MTPS   @-(R0)       ;TRY MTPS W/MODE 5
7372 023306 005737 177776      TST   @#PS          ;CHECK DEST. DATA.
7373 023312 001404      BEQ    MTPS5         ;
7374      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7375      ;          CONDITIONAL BRANCH INST. AND <====
7376      ;          REPLACE THE MOVE INSTRUCTION <====
7377      ;          WHICH FOLLOWS W/ 765 <====
7378 023314 012742 000550      MOV    #550,-(R2)    ;MOVE TO MAILBOX # ***** 550 *****
7379 023320 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
7380 023322 000000      HALT                   ;DESTINATION DATA INCORRECT
7381 023324 022700 000402      MTPS5: CMP   #402,R0  ;CHECK MODE 5 REGISTER
7382 023330 001404      BEQ    TS241         ;
7383      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7384      ;          CONDITIONAL BRANCH INST. AND <====
7385      ;          REPLACE THE MOVE INSTRUCTION <====
7386      ;          WHICH FOLLOWS W/ 756 <====
7387 023332 012742 000551      MOV    #551,-(R2)    ;MOVE TO MAILBOX # ***** 551 *****
7388 023336 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
7389 023340 000000      HALT                   ;MODE 5 REGISTER NOT DECREMENTED BY 2
7390      ; OR SEQUENCE ERROR
7391
7392 ;*****
```

7393  
7394  
7395 023342 005212  
7396 023344 022712 000241  
7397 023350 001024  
7398 023352 012737 052652 000000  
7399 023360 012700 000406  
7400 023364 005037 177776  
7401 023370 106460 177372  
7402 023374 022737 000252 177776  
7403 023402 001404  
7404  
7405  
7406  
7407  
7408 023404 012742 000552  
7409 023410 005242  
7410 023412 000000  
7411 023414 022700 000406  
7412 023420 001404  
7413  
7414  
7415  
7416  
7417 023422 012742 000553  
7418 023426 005242  
7419 023430 000000  
7420  
7421  
7422  
7423  
7424  
7425 023432 005212  
7426 023434 022712 000242  
7427 023440 001024  
7428 023442 012737 052652 000000  
7429 023450 012700 000410  
7430 023454 005037 177776  
7431 023460 106470 177776  
7432 023464 022737 000105 177776  
7433 023472 001404  
7434  
7435  
7436  
7437  
7438 023474 012742 000554  
7439 023500 005242  
7440 023502 000000  
7441 023504 022700 000410  
7442 023510 001404  
7443  
7444  
7445  
7446  
7447 023512 012742 000555  
7448 023516 005242

```
;TEST 241      TEST MTPS MODE 6
:*****
TS241:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #241,(R2)    ;SEQUENCE ERROR?
        BNE     TS242-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #52652,@#0    ;LOC. 0=52652
        MOV     #406,R0      ;R0=406
        CLR     @#PS         ;PS=0
        MTPS   -406(R0)      ;TRY MTPS W/MODE 6
        CMP     #252,@#PS    ;CHECK DEST. DATA
        BEQ     MTPS6
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 762 <====
        MOV     #552,-(R2)    ;MOVE TO MAILBOX # ***** 552 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT
MTPS6:  CMP     #406,R0      ;DEST. DATA INCORRECT
        BEQ     TS242        ;CHECK MODE 6 REGISTER
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 753 <====
        MOV     #553,-(R2)    ;MOVE TO MAILBOX # ***** 553 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT                ;MODE 6 REGISTER MODIFIED
; OR SEQUENCE ERROR
:*****
;TEST 242      TEST MTPS MODE 7
:*****
TS242:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #242,(R2)    ;SEQUENCE ERROR?
        BNE     TS243-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #52652,@#0    ;LOC. 0=52652
        MOV     #410,R0      ;R0=410
        CLR     @#PS         ;PS=0
        MTPS   @-2(R0)       ;TRY MTPS W/MODE 7
        CMP     #105,@#PS    ;CHECK DEST. DATA
        BEQ     MTPS7
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 762 <====
        MOV     #554,-(R2)    ;MOVE TO MAILBOX # ***** 554 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT                ;DESTINATION DATA INCORRECT
MTPS7:  CMP     #410,R0      ;CHECK MODE 7 REGISTER
        BEQ     TS243
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 753 <====
        MOV     #555,-(R2)    ;MOVE TO MAILBOX # ***** 555 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
```

MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79<sup>H 12</sup> 11:31 PAGE 150  
1242 TEST MTPS MODE 7

SEQ 0150

7449 023520 000000  
7450  
7451

HALT

;MODE 7 REGISTER MODIFIED  
; OR SEQUENCE ERROR



7452  
7453  
7454  
7455  
7456  
7457  
7458  
7459  
7460  
7461  
7462  
7463 023522 005212  
7464 023524 022712 000243  
7465 023530 001025  
7466 023532 012737 000377 177776  
7467 023540 106700  
7468 023542 022700 177757  
7469 023546 001404  
7470  
7471  
7472  
7473  
7474 023550 012742 000556  
7475 023554 005242  
7476 023556 000000  
7477  
7478 023560 005000  
7479 023562 012737 177777 000000  
7480 023570 005037 177776  
7481 023574 106710  
7482 023576 105737 000000  
7483 023602 001404  
7484  
7485  
7486  
7487  
7488 023604 012742 000557  
7489 023610 005242  
7490 023612 000000  
7491  
7492  
7493  
7494  
7495  
7496 023614 005212  
7497 023616 022712 000244  
7498 023622 001031  
7499 023624 005000  
7500 023626 005010  
7501 023630 012737 000377 177776  
7502 023636 106720  
7503 023640 103003  
7504 023642 102402  
7505 023644 001401  
7506 023646 100404  
7507

.....  
: THESE NEXT SEVEN TESTS VERIFY THE MFPS INSTRUCTION IN ALL  
: MODES. IN EACH TEST, A PATTERN OF ONES AND ZEROS IS MOVED TO THE  
: PSW, AND AN MFPS INSTRUCTION MOVES THE DATA TO A LOCATION SETUP  
: BY R0, EITHER DIRECTLY OR INDIRECTLY. CONDITIONAL BRANCHES ARE  
: USED TO CHECK PROPER ADDRESSING AND DATA.  
: .....

: TEST 243 TEST MFPS INSTRUCTION  
: .....

TS243: INC (R2) ;UPDATE TEST NUMBER  
CMP #243,(R2) ;SEQUENCE ERROR?  
BNE TS244-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #377,@#PS  
MFPS R0  
CMP #177757,R0  
BEQ MFPS1

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 770 <====

MOV #556,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 556 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;MFPS FAILED

MFPS1: CLR R0  
MOV #-1,@#0  
CLR @#PS  
MFPS (R0)  
TSTB @#0  
BEQ TS244

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 752 <====

MOV #557,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 557 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;MFPS FAILED  
: OR SEQUENCE ERROR

.....  
: TEST 244 TEST MFPS MODE 2  
: .....

TS244: INC (R2) ;UPDATE TEST NUMBER  
CMP #244,(R2) ;SEQUENCE ERROR?  
BNE TS245-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;R0=0  
CLR (R0) ;LOC. 0=0  
MOV #377,@#PS ;SET PS=357  
MFPS (R0)+ ;TRY MFPS W/MODE 2  
BCC MFPS2A ;BR TO ERROR IF C BIT CLEAR  
BVS MFPS2A ;BR TO ERROR IF V BIT SET  
BEQ MFPS2A ;BR TO ERROR IF Z BIT SET  
BMI MFPS2B

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====

```
7508                                     :           CONDITIONAL BRANCH INST. AND <====
7509                                     :           REPLACE THE MOVE INSTRUCTION <====
7510                                     :           WHICH FOLLOWS W/ 765          <====
7511 023650                               MFPS2A:
7512 023650 012742 000560                 MOV    #560,-(R2)   ;MOVE TO MAILBOX # ***** 560 *****
7513 023654 005242                       INC    -(R2)       ;SET MSGTYP TO FATAL ERROR
7514 023656 000000                       HALT                               ;COND. CODES INCORRECT
7515 023660 022737 000357 000000         MFPS2B: CMP    #357,@#0 ;CHECK DEST. DATA
7516 023666 001404                       BEQ    MFPS2C
7517                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7518                                     :           CONDITIONAL BRANCH INST. AND <====
7519                                     :           REPLACE THE MOVE INSTRUCTION <====
7520                                     :           WHICH FOLLOWS W/ 755          <====
7521 023670 012742 000561                 MOV    #561,-(R2)   ;MOVE TO MAILBOX # ***** 561 *****
7522 023674 005242                       INC    -(R2)       ;SET MSGTYP TO FATAL ERROR
7523 023676 000000                       HALT                               ;DEST. DATA INCORRECT
7524 023700 022700 000001                 MFPS2C: CMP    #1,R0 ;CHECK MODE 2 REGISTER
7525 023704 001404                       BEQ    TS245
7526                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7527                                     :           CONDITIONAL BRANCH INST. AND <====
7528                                     :           REPLACE THE MOVE INSTRUCTION <====
7529                                     :           WHICH FOLLOWS W/ 746          <====
7530 023706 012742 000562                 MOV    #562,-(R2)   ;MOVE TO MAILBOX # ***** 562 *****
7531 023712 005242                       INC    -(R2)       ;SET MSGTYP TO FATAL ERROR
7532 023714 000000                       HALT                               ;MODE 2 REGISTER NOT INCREMENTED 1
7533                                     : OR SEQUENCE ERROR
7534
7535 ;*****
7536 ;TEST 245 TEST MFPS MODE 3
7537 ;*****
7538 023716 005212                               TS245: INC    (R2) ;UPDATE TEST NUMBER
7539 023720 022712 000245                   CMP    #245,(R2) ;SEQUENCE ERROR?
7540 023724 001033                           BNE    TS246-10 ;BR TO ERROR HALT ON SEQ ERROR
7541 023726 012700 000406                   MOV    #406,R0 ;R0=406
7542 023732 005037 000000                   CLR    @#0 ;LOC. 0=0
7543 023736 012737 000252 177776           MOV    #252,@#PS ;PS=252
7544 023744 106730                           MFPS   @(R0)+ ;TRY MFPS WITH MODE 3
7545 023746 103403                           BCS    MFPS3A ;BR TO ERROR IF C-BIT SET
7546 023750 102402                           BVS    MFPS3A ;BR TO ERROR IF V-BIT SET
7547 023752 001401                           BEQ    MFPS3A ;BR TO ERROR IF Z-BIT SET
7548 023754 100404                           BMI    MFPS3B
7549                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7550                                     :           CONDITIONAL BRANCH INST. AND <====
7551                                     :           REPLACE THE MOVE INSTRUCTION <====
7552                                     :           WHICH FOLLOWS W/ 763          <====
7553 023756                               MFPS3A:
7554 023756 012742 000563                 MOV    #563,-(R2)   ;MOVE TO MAILBOX # ***** 563 *****
7555 023762 005242                       INC    -(R2)       ;SET MSGTYP TO FATAL ERROR
7556 023764 000000                       HALT                               ;CONDITION CODES INCORRECT
7557 023766 022737 125000 000000         MFPS3B: CMP    #125000,@#0 ;CHECK DEST. DATA
7558 023774 001404                       BEQ    MFPS3C
7559                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7560                                     :           CONDITIONAL BRANCH INST. AND <====
7561                                     :           REPLACE THE MOVE INSTRUCTION <====
7562                                     :           WHICH FOLLOWS W/ 753          <====
7563 023776 012742 000564                 MOV    #564,-(R2)   ;MOVE TO MAILBOX # ***** 564 *****
```



7620  
7621  
7622 024132 005212  
7623 024134 022712 000247  
7624 024140 001033  
7625 024142 012700 000410  
7626 024146 012737 177777 000000  
7627 024154 005037 177776  
7628 024160 106750  
7629 024162 103403  
7630 024164 102402  
7631 024166 100401  
7632 024170 001404  
7633  
7634  
7635  
7636  
7637 024172  
7638 024172 012742 000571  
7639 024176 005242  
7640 024200 000000  
7641 024202 022737 000377 000000  
7642 024210 001404  
7643  
7644  
7645  
7646  
7647 024212 012742 000572  
7648 024216 005242  
7649 024220 000000  
7650 024222 020027 000406  
7651 024226 001404  
7652  
7653  
7654  
7655  
7656 024230 012742 000573  
7657 024234 005242  
7658 024236 000000  
7659  
7660  
7661  
7662  
7663  
7664 024240 005212  
7665 024242 022712 000250  
7666 024246 001034  
7667 024250 012700 000401  
7668 024254 005037 000000  
7669 024260 012737 000252 177776  
7670 024266 106760 177377  
7671 024272 102403  
7672 024274 103402  
7673 024276 001401  
7674 024300 100404  
7675

```
;TEST 247      TEST MFPS MODE 5
:*****
TS247:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #247,(R2)    ;SEQUENCE ERROR?
        BNE     TS250-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #410,R0      ;R0=410
        MOV     #-1,@#0     ;LOC. 0=-1
        CLR     @#PS        ;PS=0
        MFPS   @-(R0)       ;TRY MFPS W/MODE 5
        BCS    MFPS5A      ;BR TO ERROR IF C-BIT SET
        BVS    MFPS5A      ;BR TO ERROR IF V-BIT SET
        BMI    MFPS5A      ;BR TO ERROR IF N-BIT SET
        BEQ    MFPS5B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 763 <====

MFPS5A: MOV     #571,-(R2)    ;MOVE TO MAILBOX # ***** 571 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                    ;COND. CODES INCORRECT
MFPS5B: CMP     #377,@#0     ;CHECK DEST. DATA
        BEQ    MFPS5C

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 753 <====

MFPS5C: MOV     #572,-(R2)    ;MOVE TO MAILBOX # ***** 572 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                    ;DEST DATA INCORRECT
        CMP     R0,#406     ;CHECK MODE 5 REGISTER
        BEQ    TS250

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 744 <====

        MOV     #573,-(R2)    ;MOVE TO MAILBOX # ***** 573 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
; OR SEQUENCE ERROR

:*****
;TEST 250      TEST MFPS MODE 6
:*****
TS250:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #250,(R2)    ;SEQUENCE ERROR?
        BNE     TS251-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #401,R0      ;R0=410
        CLR     @#0         ;LOC. 0=0
        MOV     #252,@#PS    ;PS=252
        MFPS   -401(R0)     ;TRY MFPS W/MODE 6
        BVS    MFPS6A      ;BR TO ERROR IF V-BIT SET
        BCS    MFPS6A      ;BR TO ERROR IF C-BIT SET
        BEQ    MFPS6A      ;BR TO ERROR IF Z-BIT SET
        BMI    MFPS6B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
```

```
7676                                     :                               <====
7677                                     :                               <====
7678                                     :                               <====
7679 024302                               MFPS6A:                               <====
7680 024302 012742 000574                MOV    #574,-(R2)           ;MOVE TO MAILBOX # ***** 574 *****
7681 024306 005242                        INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
7682 024310 000000                        HALT                               ;COND. CODES INCORRECT
7683 024312 022737 000252 000000        MFPS6B: CMP    #252,@#0           ;CHECK DEST. DATA
7684 024320 001404                        BEQ    MFPS6C
7685                                     :                               <====
7686                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7687                                     :                               <====
7688                                     :                               <====
7689                                     :                               <====
7689 024322 012742 000575                MOV    #575,-(R2)           ;MOVE TO MAILBOX # ***** 575 *****
7690 024326 005242                        INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
7691 024330 000000                        HALT                               ;DEST. DATA INCORRECT
7692 024332 022700 000401                MFPS6C: CMP    #401,R0           ;CHECK DEST. REGISTER
7693 024336 001404                        BEQ    TS251
7694                                     :                               <====
7695                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7696                                     :                               <====
7697                                     :                               <====
7698 024340 012742 000576                MOV    #576,-(R2)           ;MOVE TO MAILBOX # ***** 576 *****
7699 024344 005242                        INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
7700 024346 000000                        HALT                               ;DEST. DATA INCORRECT
7701                                     : OR SEQUENCE ERROR
7702
7703
7704
7705
7706 024350 005212                               :*****
7707 024352 022712 000251                ;TEST 251          TEST MFPS MODE 7
7708 024356 001034                               :*****
7709 024360 012700 000777                TS251: INC    (R2)             ;UPDATE TEST NUMBER
7710 024364 005037 000000                CMP    #251,(R2)           ;SEQUENCE ERROR?
7711 024370 012737 000125 177776        BNE    TS252-10           ;BR TO ERROR HALT ON SEQ ERROR
7712 024376 106770 177407                MOV    #777,R0           ;RO=777
7713 024402 102403                               CLR    @#0                ;LOC. 0=0
7714 024404 103002                               MOV    #125,@#PS         ;PS=125
7715 024406 001401                               MFPS  @-371(R0)         ;TRY MFPS W/MODE 7
7716 024410 100004                               BVS    MFPS7A           ;BR TO ERROR IF V-BIT SET
7717                                     BCC    MFPS7A           ;BR TO ERROR IF C-BIT SET
7718                                     BEQ    MFPS7A           ;BR TO ERROR IF Z-BIT SET
7719                                     BPL    MFPS7B
7720                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7721                                     :                               <====
7722                                     :                               <====
7723                                     :                               <====
7723 024412 012742 000577                MFPS7A: MOV    #577,-(R2)           ;MOVE TO MAILBOX # ***** 577 *****
7724 024416 005242                        INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
7725 024422 022737 042400 000000        MFPS7B: CMP    #42400,@#0       ;CONDITION CODE INCORRECT
7726 024430 001404                        BEQ    MFPS7C           ;CHECK DESTINATION DATA
7727                                     :                               <====
7728                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7729                                     :                               <====
7730                                     :                               <====
7731 024432 012742 000600                MOV    #600,-(R2)         ;MOVE TO MAILBOX # ***** 600 *****
```

7732 024436 005242  
7733 024440 000000  
7734 024442 022700 000777  
7735 024446 001404

MFPS7C: INC -(R2)  
HALT  
CMP #777,R0  
BEQ TS252

:SET MSGTYP TO FATAL ERROR  
:DEST. DATA INCORRECT  
:CHECK MODE 7 REGISTER  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 743 <====

7740 024450 012742 000601  
7741 024454 005242  
7742 024456 000000

MOV #601,-(R2)  
INC -(R2)  
HALT

:MOVE TO MAILBOX # \*\*\*\*\* 601 \*\*\*\*\*  
:SET MSGTYP TO FATAL ERROR  
:MODE 7 REGISTER MODIFIED  
: OR SEQUENCE ERROR

7743  
7744  
7745  
7746  
7747  
7748  
7749  
7750  
7751

.....  
: THIS TEST VERIFIES THAT RESET DOES NOT CLEAR THE PSW.  
: THE PSW IS LOADED WITH ONES, A RESET IS ISSUED, AND THE  
: CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT THEY HAVE NOT  
: CHANGED. THIS TEST IS EXECUTED ONLY ONCE EVERY 240 (DECIMAL)  
: ITERATIONS OF PROGRAM.  
: .....

7752  
7753  
7754  
7755

: TEST 252 TEST THAT RESET DOES NOT CLEAR PSW  
: .....

7756 024460 005212  
7757 024462 022712 000252  
7758 024466 001017  
7759 024470 122767 000001 153622  
7760 024476 001003  
7761 024500 005767 153602  
7762 024504 001014

TS252: INC (R2) :UPDATE TEST NUMBER  
CMP #252,(R2) :SEQUENCE ERROR?  
BNE TS253-10 :BR TO ERROR HALT ON SEQ ERROR  
CMPB #APTENV,SENV :RUNING IN APT MODE?  
BNE 1\$ :IF NOT, DO THIS TEST  
TST \$PASS :IS THIS THE FIRST PASS?  
BNE REST :IF NOT FIRST PASS, SKIP TEST

7763 024506  
7764 024506 012737 000357 177776  
7765 024514 000005  
7766 024516 022737 000357 177776  
7767 024524 001404

1\$: MOV #357,@#PS :MOV ONES TO PSW  
RESET :  
CMP #357,@#PS :PSW CORRECT?  
BEQ TS253

7768  
7769  
7770  
7771

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 760 <====

7772 024526 012742 000602  
7773 024532 005242  
7774 024534 000000

MOV #602,-(R2)  
INC -(R2)  
HALT

:MOVE TO MAILBOX # \*\*\*\*\* 602 \*\*\*\*\*  
:SET MSGTYP TO FATAL ERROR  
:RESET ALTERED PSW  
: OR SEQUENCE ERROR

7775  
7776 024536  
7777

REST:

7778  
7779  
7780  
7781  
7782  
7783

.....  
: THE FOLLOWING TEST CHECKS THE INDEPENDENT FUNCTIONING OF BASIC  
: DATA PATH COMPONENTS WITH USER MODE SET.  
: .....

7784  
7785  
7786 024536 005212  
7787 024540 022712 000253

: TEST 253 TEST USER MODE R6 CAN HOLD A ONE IN EVERY POSITION  
: .....

TS253: INC (R2) :UPDATE TEST NUMBER  
CMP #253,(R2) :SEQUENCE ERROR?

|      |        |        |        |        |            |            |                                    |
|------|--------|--------|--------|--------|------------|------------|------------------------------------|
| 7788 | 024544 | 001017 |        |        | BNE        | TS254-10   | :BR TO ERROR HALT ON SEQ ERROR     |
| 7789 | 024545 | 052767 | 140000 | 153222 | BIS        | #USRM,PS   | :SET USER MODE                     |
| 7790 | 024554 | 012706 | 000001 |        | MOV        | #1,R6      | :SET BIT0                          |
| 7791 | 024560 | 000241 |        |        | CLC        |            | :CLEAR C-BIT                       |
| 7792 | 024562 | 006106 |        |        | USP1: ROL  | R6         | :ROTATE 1 POSITION                 |
| 7793 | 024564 | 103376 |        |        | BCC        | USP1       | :BR IF NOT ALL DONE                |
| 7794 | 024566 | 001407 |        |        | BEQ        | USP1A      | :BR IF NO BITS PICKED              |
| 7795 | 024570 | 042767 | 140000 | 153200 | BIC        | #USRM,PS   | :CLEAR USER MODE                   |
| 7796 | 024576 | 012742 | 000603 |        | MOV        | #603,-(R2) | :MOVE TO MAILBOX # ***** 603 ***** |
| 7797 | 024602 | 005242 |        |        | INC        | -(R2)      | :SET MSGTYP TO FATAL ERROR         |
| 7798 | 024604 | 000000 |        |        | HALT       |            | :USER MODE R6 PICKED A BIT         |
| 7799 | 024606 | 042767 | 140000 | 153162 | USP1A: BIC | #USRM,PS   | :CLEAR USER MODE                   |

```

:*****
:
:      THIS TEST CHECKS THE INDEPENDENT FUNCTIONING OF THE USER
:AND KERNEL MODE R6'S. R6 IS SETUP AND ADDRESSED IN EACH
:OF THE TWO MODES TO VERIFY THAT THE TWO R6'S ARE INDEPENDENT
:OF EACH OTHER.
:*****

```

```

:TEST 254      TEST INDEPENDENCE OF USER AND KERNEL MODE R6'S
:*****

```

|      |        |        |        |        |            |            |  |
|------|--------|--------|--------|--------|------------|------------|--|
| 7812 | 024614 | 005212 |        |        | TS254: INC | (R2)       | :UPDATE TEST NUMBER                            |
| 7813 | 024616 | 022712 | 000254 |        | CMP        | #254,(R2)  | :SEQUENCE ERROR?                               |
| 7814 | 024622 | 001043 |        |        | BNE        | USP4-10    | :BR TO ERROR HALT ON SEQ ERROR                 |
| 7815 | 024624 | 052767 | 140000 | 153144 | BIS        | #USRM,PS   | :SET USER MODE                                 |
| 7816 | 024632 | 012706 | 177777 |        | MOV        | #-1,R6     | :SET USER R6 TO ALL ONES                       |
| 7817 | 024636 | 022706 | 177777 |        | CMP        | #-1,R6     | :READ AND CHECK USER R6                        |
| 7818 | 024642 | 001407 |        |        | BEQ        | USP2       | :BR IF NO ERROR                                |
| 7819 | 024644 | 042767 | 140000 | 153124 | BIC        | #USRM,PS   | :CLEAR USER MODE                               |
| 7820 | 024652 | 012742 | 000604 |        | MOV        | #604,-(R2) | :MOVE TO MAILBOX # ***** 604 *****             |
| 7821 | 024656 | 005242 |        |        | INC        | -(R2)      | :SET MSGTYP TO FATAL ERROR                     |
| 7822 | 024660 | 000000 |        |        | HALT       |            | :USER R6 WILL NOT HOLD ALL ONES                |
| 7823 | 024662 | 042767 | 140000 | 153106 | USP2: BIC  | #USRM,PS   | :SET KERNEL MODE                               |
| 7824 | 024670 | 022706 | 177777 |        | CMP        | #-1,R6     | :KERNEL MODE R6 ADDR. FROM USER MODE?>>        |
| 7825 | 024674 | 001004 |        |        | BNE        | USP3       |  |
| 7826 |        |        |        |        |            |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==== |
| 7827 |        |        |        |        |            |            | :      CONDITIONAL BRANCH INST. AND <====      |
| 7828 |        |        |        |        |            |            | :      REPLACE THE MOVE INSTRUCTION <====      |
| 7829 |        |        |        |        |            |            | :      WHICH FOLLOWS W/ 752 <====              |
| 7830 | 024676 | 012742 | 000605 |        | MOV        | #605,-(R2) | :MOVE TO MAILBOX # ***** 605 *****             |
| 7831 | 024702 | 005242 |        |        | INC        | -(R2)      | :SET MSGTYP TO FATAL ERROR                     |
| 7832 | 024704 | 000000 |        |        | HALT       |            | :DUAL ADDRESSING ERROR USER/KERNEL R6          |
| 7833 | 024706 | 005006 |        |        | USP3: CLR  | R6         | :CLEAR KERNEL MODE SP                          |
| 7834 | 024710 | 052767 | 140000 | 153060 | BIS        | #USRM,PS   | :SET USER MODE                                 |
| 7835 | 024716 | 022706 | 177777 |        | CMP        | #-1,R6     | :CHECK USER R6 NOT ADDR. FROM KERNEL MODE      |
| 7836 | 024722 | 042767 | 140000 | 153046 | BIC        | #USRM,PS   | :CLEAR USER MODE                               |
| 7837 | 024730 | 001404 |        |        | BEQ        | USP4       | :BR IF NO ERROR                                |
| 7838 | 024732 | 012742 | 000606 |        | MOV        | #606,-(R2) | :MOVE TO MAILBOX # ***** 606 *****             |
| 7839 | 024736 | 005242 |        |        | INC        | -(R2)      | :SET MSGTYP TO FATAL ERROR                     |
| 7840 | 024740 | 000000 |        |        | HALT       |            | :DUAL ADDRESSING ERROR OR SEQUENCE ERROR       |
| 7841 | 024742 | 012706 | 001000 |        | USP4: MOV  | #STBOT,R6  | :RESTORE SP USER                               |
| 7842 | 024746 | 042767 | 140000 | 153022 | BIC        | #USRM,PS   | :SET KERNEL MODE                               |
| 7843 | 024754 | 012706 | 001000 |        | MOV        | #STBOT,R6  | :RESTORE SP KERNEL                             |

7844  
7845  
7846  
7847  
7848  
7849  
7850  
7851  
7852  
7853 024760 005212  
7854 024762 022712 000255  
7855 024766 001032  
7856 024770 012706 001000  
7857 024774 012767 140000 152774  
7858 025002 012706 027352  
7859 025006 006506  
7860 025010 022767 140000 152760  
7861 025016 001407  
7862 025020 042767 140000 152750  
7863 025026 012742 000607  
7864 025032 005242  
7865 025034 000000  
7866 025036 042767 140000 152732  
7867 025044 022767 001000 002276  
7868 025052 001404  
7869 025054 012742 000610  
7870 025060 005242  
7871 025062 000000  
7872 025064  
7873  
7874  
7875  
7876  
7877 025064 005212  
7878 025066 022712 000256  
7879 025072 001033  
7880 025074 005067 152676  
7881 025100 005006  
7882 025102 012767 140000 152666  
7883 025110 012706 027352  
7884 025114 012746 001000  
7885 025120 006606  
7886 025122 022767 140000 152646  
7887 025130 001407  
7888 025132 042767 140000 152636  
7889 025140 012742 000611  
7890 025144 005242  
7891 025146 000000  
7892 025150 005067 152622  
7893 025154 020627 001000  
7894 025160 001404  
7895  
7896  
7897  
7898  
7899 025162 012742 000612

```
*****
:
: THESE NEXT TWO TESTS VERIFY MFPI AND MTP1 INSTRUCTIONS
: WITH R6 IN MODE 0.
:
:*****
: TEST 255 TEST MFPI WITH R6 IN MODE 0
:*****
TS255:  INC      (R2)           ;UPDATE TEST NUMBER
        CMP      #255,(R2)    ;SEQUENCE ERROR?
        BNE      TS256-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV      #STBOT,R6    ;INITIALIZE KERNEL STACK POINTER
        MOV      #USRM,PS     ;SET USER MODE/PREVIOUS KERNEL
        MOV      #USTBOT,R6   ;INITIALIZE USER STACK POINTER
        MFPI     R6           ;TRY MFPI WITH MODE 0
        CMP      #140000,PS   ;CHECK PSW
        BEQ      MFPI0       ;BR IF NO ERROR
        BIC      #USRM,PS     ;CLEAR USER MODE
        MOV      #607,-(R2)   ;MOVE TO MAILBOX # ***** 607 *****
        INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                    ;INCORRECT PSW FROM MFPI
MFPI0:  BIC      #USRM,PS     ;CLEAR USER MODE
        CMP      #STBOT,USTBOT-2 ;CHECK DATA ON STACK
        BEQ      MFPI0A      ;BR IF NO ERROR
        MOV      #610,-(R2)   ;MOVE TO MAILBOX # ***** 610 *****
        INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                    ;INCORRECT DATA FROM MFPI
MFPI0A:
:*****
: TEST 256 TEST MTP1 WITH R6 IN MODE 0
:*****
TS256:  INC      (R2)           ;UPDATE TEST NUMBER
        CMP      #256,(R2)    ;SEQUENCE ERROR?
        BNE      TS257-10     ;BR TO ERROR HALT ON SEQ ERROR
        CLR      PS          ;SET KERNEL MODE
        CLR      R6          ;INITIALIZE KERNEL R6
        MOV      #USRM,PS     ;SET USER MODE/PREVIOUS KERNEL
        MOV      #USTBOT,R6   ;INITIALIZE USER STACK POINTER
        MOV      #STBOT,-(R6) ;SET UP TARGET DATA
        MTP1     R6          ;TRY MODE 0 MTP1
        CMP      #USRM,PS     ;CHECK PSW
        BEQ      MTP10       ;BR IF NO ERROR
        BIC      #USRM,PS     ;CLEAR USER MODE
        MOV      #611,-(R2)   ;MOVE TO MAILBOX # ***** 611 *****
        INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                    ;PS INCORRECT FOLLOWING MTP1
MTP10:  CLR      PS          ;SET KERNEL MODE
        CMP      R6,#STBOT    ;CHECK TARGET DATA
        BEQ      TS257
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
: CONDITIONAL BRANCH INST. AND <===
: REPLACE THE MOVE INSTRUCTION <===
: WHICH FOLLOWS W/ 744 <===
:
: MOVE TO MAILBOX # ***** 612 *****
```



MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79 11:31 PAGE 159  
T256 TEST MTP1 WITH R6 IN MODE 0

D 13

SEQ 0159

7900 025166 005242  
7901 025170 000000  
7902  
7903

INC -(R2)  
HALT

:SET MSGTYP TO FATAL ERROR  
:DATA INCORRECT FOLLOWING MTP1  
: OR SEQUENCE ERROR

7904  
7905  
7906  
7907  
7908  
7909  
7910  
7911  
7912  
7913  
7914  
7915  
7916  
7917  
7918  
7919  
7920  
7921  
7922  
7923  
7924  
7925  
7926  
7927  
7928  
7929  
7930  
7931  
7932  
7933  
7934  
7935  
7936  
7937  
7938  
7939  
7940  
7941  
7942  
7943  
7944  
7945  
7946  
7947  
7948  
7949  
7950  
7951  
7952  
7953  
7954  
7955  
7956  
7957  
7958  
7959

025172 005212  
025174 022712 000257  
025200 001062  
025202 012700 027242  
025206 012704 027300  
025212 012767 000017 000142  
025220 012067 000110  
025224 012401  
025226 012767 177777 000074  
025234 012703 000020  
025240 005267 000064  
025244 032701 100000  
025250 013705 177776  
025254 042705 177773  
025260 000165 025264  
025264 000167 0C0020  
025270 012767 025364 000042  
025276 012767 025346 000040  
025304 000167 000014  
025310 012767 025346 000022  
025316 012767 025364 000020  
025324 006101  
025326 012737  
025330 000000  
025332 177776  
025334 000000  
025336 000137  
025340 000000  
025342 000137

\*\*\*\*\*  
: THIS TEST EXECUTES EVERY POSSIBLE BRANCH WITH EVERY POSSIBLE  
: CONDITION CODE COMBINATION.  
: THE ROUTINE USES TWO TABLES. THE BRANCH TABLE HOLDS ALL THE  
: POSSIBLE BRANCH INSTRUCTIONS, THE OTHER TABLE (YNTAB) HOLDS BIT MAPS FOR  
: EACH BRANCH. A ONE IN THE BIT MAP INDICATES THAT THE CORRESPONDING  
: BRANCH INSTRUCTION SHOULD BRANCH FOR THE CONDITION CODE SETTING WHICH  
: CORRESPONDS TO THE BIT POSITION WITHIN THE MAP. FOR EXAMPLE IF THE LEFT  
: MOST BIT IS A ONE THEN THE CORRESPONDING BRANCH INSTRUCTION SHOULD BRANCH  
: WHEN THE CONDITION CODES ARE 0.  
: THE ROUTINE CONSISTS OF NESTED LOOPS; THE OUTER LOOP SETS UP  
: ALL THE POSSIBLE BRANCH INSTRUCTIONS. THE INNER LOOP SETS UP EVERY POSSIBLE  
: CONDITION CODE FOR EACH BRANCH.  
: THE BIT MAP IS USED TO SET THE ADDRESS LOCATION IN TWO  
: JUMP MODE 3 INSTRUCTIONS. THE ADDRESSES ARE CHANGED TO ALLOW THE  
: PROGRAM TO CONTINUE OR JUMP TO AN ERROR ROUTINE DEPENDING UPON  
: WHETHER IT HANDLED THE BRANCH INSTRUCTION CORRECTLY.  
: AT ANY ERROR HALT, LOCATION, BRH, HOLDS THE BRANCH INSTRUCTION  
: UNDER TEST AND LOCATION, CC, HOLDS THE VALUE OF THE CONDITION CODES  
: AT THE TIME THE BRANCH WAS EXECUTED.  
\*\*\*\*\*  
: TEST 257 TEST THE BRANCH ROM  
\*\*\*\*\*  
TS257: INC (R2) ;UPDATE TEST NUMBER  
CMP #257,(R2) ;SEQUENCE ERROR?  
BNE ER ;BR TO ERROR HALT ON SEQ ERROR  
SETUP: MOV #BRTAB,R0 ;INITIALIZE BRANCH TABLE POINTER  
MOV #YNTAB,R4 ;INITIALIZE YES/NO BRANCH MAP POINTER  
MOV #15.,BRCT ;INITIALIZE BRANCH TABLE COUNT  
SETBR: MOV (R0)+,BRH ;GET NEXT BRANCH INST.  
MOV (R4)+,R1 ;GET NEXT BRANCH MAP  
MOV #-1,CC1 ;INITIALIZE CONDITION CODE VALUE  
MOV #16.,R3 ;INITIALIZE CONDITION CODE COUNT  
SETCC: INC CC1 ;SET FOR NEXT CC VALUE  
BIT #100000,R1 ;SEE IF SHOULD BR W/ THESE CC'S  
MOV @#177776,R5 ;SIMULATE A JNE  
BIC #177773,R5 ; (JUMP NOT EQUAL)  
JMP .+4(R5) ; TO SET2BR  
JMP SET2BR  
MOV #CONT,NBR ;SET TO CONTINUE IF NO BRANCH  
MOV #ER,YBR ;SET TO REPORT ERROR IF BRANCH  
JMP AROUND ;GO AROUND OPPOSITE CONDITION  
SET2BR: MOV #ER,NBR ;SET TO REPORT ERROR IF NO BRANCH  
MOV #CONT,YBR ;SET TO CONTINUE IF BRANCH  
AROUN: ROL R1 ;UPDATE BIT MAP  
  
MOV (PC)+,@(PC)+ ;SET CONDITION CODE  
CC1: 0 ;NEW CC VALUE GOES HERE  
177776  
BRH: 0 ;BRANCH INST. GOES HERE  
JMP @(PC)+ ;THIS JUMP IF NO BRANCH  
NBR: 0 ;WHERE TO GO IF NO BRANCH OCCURS  
JMP @(PC)+ ;THIS JUMP IF BRANCH OCCURS

7960 025344 000000  
7961 025346 012702 000304  
7962 025352 012742 000613  
7963 025356 005242  
7964 025360 000000  
7965 025362 000000  
7966 025364 005303  
7967 025366 013705 177776  
7968 025372 042705 177773  
7969 025376 000165 025402  
7970 025402 000167 177632  
7971 025406 005367 177750  
7972 025412 013705 177776  
7973 025416 042705 177773  
7974 025422 000165 025426  
7975 025426 000167 177566

YBR: 0 ;WHERE TO GO IF BRANCH OCCURS  
ER: MOV #8TESTN,R2 ;RESTORE POINTER  
MOV #613,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 613 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;  
BRCT: 0 ;  
CONT: DEC R3 ;CC'S DONE?  
MOV @#177776,R5 ;SIMULATE A JNE  
BIC #177773,R5 ; (JUMP NOT EQUAL)  
JMP .+4(R5) ; TO SETCC  
JMP SETCC ;  
DEC BRCT ;BR'S DONE?  
MOV @#177776,R5 ;SIMULATE A JNE  
BIC #177773,R5 ; (JUMP NOT EQUAL)  
JMP .+4(R5) ; TO SETBR  
JMP SETBR ;

7976  
7977  
7978  
7979  
7980  
7981  
7982  
7983  
7984  
7985  
7986  
7987 025432 005212  
7988 025434 022712 000260  
7989 025440 001052  
7990 025442 005000  
7991 025444 005001  
7992 025446 005002  
7993 025450 005003  
7994 025452 005004  
7995 025454 005005  
7996 025456 005006  
7997 025460 052700 000001  
7998 025464 052701 000002  
7999 025470 052702 000004  
8000 025474 052703 000010  
8001 025500 052704 000020  
8002 025504 052705 000040  
8003 025510 052706 000100  
8004 025514 022706 000100  
8005 025520 001022  
8006 025522 022705 000040  
8007 025526 001017  
8008 025530 022704 000020  
8009 025534 001014  
8010 025536 022703 000010  
8011 025542 001011  
8012 025544 022702 000004  
8013 025550 001006  
8014 025552 022701 000002  
8015 025556 001003  
8016 025560 022700 000001  
8017 025564 001404  
8018  
8019  
8020  
8021  
8022 025566  
8023 025566 012742 000614  
8024 025572 005242  
8025 025574 000000  
8026 025576 012702 000304  
8027

\*\*\*\*\*  
: THE FOLLOWING TEST VERIFIES THAT NO DUAL ADDRESSING OF THE GENERAL  
: REGISTERS OCCURS. ALL REGISTERS ARE CLEARED, AND A UNIQUE BIT IS SET  
: IN EACH. CMP INSTRUCTIONS CHECK THAT ONLY ONE BIT IS SET IN EACH  
: REGISTER.  
\*\*\*\*\*

TEST 260 DUAL REGISTER ADDRESSING TEST  
\*\*\*\*\*

TS260: INC (R2) ; UPDATE TEST NUMBER  
CMP #260,(R2) ; SEQUENCE ERROR?  
BNE DAERR ; BR TO ERROR HALT ON SEQ ERROR  
BITCLR: CLR R0 ; INITIALIZE ALL REGISTERS  
CLR R1  
CLR R2  
CLR R3  
CLR R4  
CLR R5  
CLR R6  
BITSET: BIS #1,R0 ; SET R0=1  
BIS #2,R1 ; R1=2  
BIS #4,R2 ; R2=4  
BIS #10,R3 ; R3=10  
BIS #20,R4 ; R4=20  
BIS #40,R5 ; R5=40  
BIS #100,R6 ; R6=100  
BITCHK: CMP #100,R6 ; TEST THAT NO DUAL ADDRESSING OCCURRED  
BNE DAERR ; BR TO ERROR HALT IF ANY OTHER BITS ARE SET  
CMP #40,R5  
BNE DAERR  
CMP #20,R4  
BNE DAERR  
CMP #10,R3  
BNE DAERR  
CMP #4,R2  
BNE DAERR  
CMP #2,R1  
BNE DAERR  
CMP #1,R0  
BEQ BITCON

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 725 <====

DAERR: MOV #614,-(R2) ; MOVE TO MAILBOX # \*\*\*\*\* 614 \*\*\*\*\*  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; DUAL ADDRESSING ERROR  
BITCON: MOV #TESTN,R2 ; RESTORE POINTER

8028  
8029  
8030  
8031  
8032  
8033  
8034  
8035  
8036  
8037 025602 005212  
8038 025604 022712 000261  
8039 025610 001012  
8040 025612 052737 170357 177776  
8041 025620 105037 177776  
8042 025624 013700 177776  
8043 025630 032700 170000  
8044 025634 001006  
8045 025636 005037 177776  
8046 025642 012742 000615  
8047 025646 005242  
8048 025650 000000  
8049 025652 005037 177776  
8050  
8051  
8052  
8053  
8054  
8055  
8056  
8057  
8058  
8059  
8060 025656 005212  
8061 025660 022712 000262  
8062 025664 001010  
8063 025666 000277  
8064 025670 000252  
8065 025672 000167 000000  
8066 025676 100403  
8067 025700 001002  
8068 025702 102401  
8069 025704 103404  
8070  
8071  
8072  
8073  
8074 025706  
8075 025706 012742 000616  
8076 025712 005242  
8077 025714 000000  
8078

\*\*\*\*\*  
: THIS TEST VERIFIES THAT THE UPPER BYTE OF THE PSW IS NOT AFFECTED  
: WHEN THE PRIORITY LEVEL OR CC'S ARE CHANGED. ALL BITS ARE  
: INITIALLY SET IN THE PSW, AND THE LOW BYTE IS CLEARED. A BIT  
: INSTRUCTION VERIFIES THE DATA.  
\*\*\*\*\*

: TEST 261 TEST BYTE INSTRUCTION ON PSW  
\*\*\*\*\*

TS261: INC (R2) ;UPDATE TEST NUMBER  
CMP #261,(R2) ;SEQUENCE ERROR?  
BNE BTERR ;BR TO ERROR HALT ON SEQ ERROR  
BIS #170357,@#PS ;SET ALL POSSIBLE BITS IN PSW  
CLRB @#PS ;CLR PR LEVEL AND CC'S  
MOV @#PS,R0 ;COPY CONTENTS OF PSW  
BIT #170000,R0 ;TEST THAT UPPER BYTE IS UNAFFECTED  
BNE BTCON ;CONTINUE IF OK  
BTERR: CLR @#PS ;RETURN TO KERNEL MODE  
MOV #615,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 615 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;BYTE INSTRUCTION ALTERED PSW  
BTCON: CLR @#PS ;RETURN TO KERNEL MODE

\*\*\*\*\*  
: THIS TEST VERIFIES THAT A JMP INSTRUCTION DOES NOT ALTER THE  
: CONDITION CODES IN THE PSW. THE CC'S ARE PRESET,THE JMP IS  
: EXECUTED, AND CONDITIONAL BRANCHES VERIFY THE STATE OF THE CC'S.  
\*\*\*\*\*

: TEST 262 TEST THAT JMP INSTRUCTION DOES NOT AFFECT CONDITION CODES  
\*\*\*\*\*

TS262: INC (R2) ;UPDATE TEST NUMBER  
CMP #262,(R2) ;SEQUENCE ERROR?  
BNE TS263-10 ;BR TO ERROR HALT ON SEQ ERROR  
SCC  
+CLN!CLV ;CC=0101  
JMPT: JMT JMPT ;JUMP TO TEST PSW  
BMI JMPERR ;BR TO ERROR HALT IF N-BIT IS SET  
BNE JMPERR ;BR TO ERROR HALT IF Z-BIT IS CLEAR  
BVS JMPERR ;BR TO ERROR HALT IF V-BIT IS SET  
BCS TS263  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 767 <====  
JMPT: MOV #616,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 616 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;JMP INSTRUCTION AFFECTED CC'S  
: OR SEQUENCE ERROR

8079  
8080  
8081  
8082  
8083  
8084  
8085  
8086  
8087  
8088  
8089  
8090  
8091  
8092  
8093  
8094  
8095  
8096  
8097  
8098  
8099  
8100  
8101  
8102  
8103  
8104  
8105  
8106  
8107  
8108  
8109  
8110  
8111  
8112  
8113  
8114  
8115  
8116  
8117  
8118  
8119  
8120  
8121  
8122  
8123  
8124  
8125  
8126  
8127  
8128  
8129  
8130  
8131  
8132  
8133  
8134

025716 005212  
025720 022712 000263  
025724 001062  
025726 012767 000240 000024  
025734 012767 000017 000032  
025742 012767 000261 000102  
025750 012767 000001 000110  
025756 000277  
025760 000000  
025762 013704 177776  
025766 042704 177760  
025772 022704  
025774 000000  
025776 001404  
  
026000 012742 000617  
026004 005242  
026006 000000  
026010 005367 177760  
026014 005267 177740  
026020 026727 177734 000257  
026026 003753  
026030 026727 177724 000260  
026036 001004  
026040 012767 000017 177726  
026046 000743  
026050 000257  
026052 000000  
026054 013704 177776  
026060 042704 177760  
026064 022704  
026066 000000  
026070 001404

```
*****
:
: THIS TEST VERIFIES THE SET AND CLEAR CONDITION CODE INSTRUCTIONS.
: THE TEST CONSISTS OF TWO ROUTINES, ONE TO TEST ALL CLEAR CC
: INSTRUCTIONS, AND THE SECOND TO TEST ALL SET CC INSTRUCTIONS. ALL
: POSSIBLE COMBINATIONS OF CONDITION CODES ARE TESTED, INCLUDING NOP'S.
: TO TEST THE CLEAR CC INSTRUCTIONS, ALL CONDITION CODES ARE
: INITIALLY SET. THE INSTRUCTION IS EXECUTED, AND THE PSW IS CHECKED
: TO VERIFY THE PROPER COMBINATION OF CONDITION CODES.
: TO TEST THE SET CC INSTRUCTIONS, THE CONDITION CODES ARE
: INITIALLY CLEARED, AND ONLY THE REQUIRED BITS ARE SET BY THE SET CC
: INSTRUCTION. THE CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT
: ONLY THE REQUIRED BITS WERE SET.
:
: *****
: TEST 263 TEST SET CC AND CLEAR CC INSTRUCTIONS
: *****
TS263: INC (R2) ;UPDATE TEST NUMBER
      CMP #263,(R2) ;SEQUENCE ERROR?
      BNE CCERR ;BR TO ERROR HALT ON SEQ ERROR
      MOV #240,CC3 ;INITIALIZE CLR CC INSTRUCTION CODES
      MOV #17,CC2 ;INITIALIZE OCTAL MAP
      MOV #261,SC3 ;INITIALIZE SET CC INSTRUCTION CODES
      MOV #1,SC4 ;INITIALIZE OCTAL MAP
CLRCD: SCC ;SET ALL CONDITION CODES
CC3: 0 ;CONDITION CODE INSTRUCTION
      MOV @#PS,R4 ;COPY THE PSW
      BIC #177760,R4 ;ISOLATE CONDITION CODES
      CMP (PC)+,R4 ;CHECK THAT PROPER CC'S WERE CLEARED
CC2: 0 ;OCTAL REPRESENTATION OF CC'S
      BEQ CON1
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 752 <====
      MOV #617,-(R2) ;MOVE TO MAILBOX # ***** 617 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;CLEAR CC INSTRUCTION FAILED
CON1: DEC CC2 ;SET NEXT OCTAL MAP OF CC'S
      INC CC3 ;GET NEXT CLEAR CC INSTRUCTION
      CMP CC3,#257 ;TEST FOR CCC INSTRUCTION
      BLE CLRCD ;GO TEST NEXT INSTRUCTION IF NOT FOUND
      CMP CC3,#260 ;CHECK FOR NOP=260
      BNE SETCD ;GO TEST SET CC INSTRUCTIONS
      MOV #17,CC2 ;SET OCTAL MAP TO TEST NOP
      BR CLRCD ;GO TEST NOP
SETCD: CCC ;CLEAR ALL CONDITION CODES
SC3: 0 ;CONDITION CODE INSTRUCTION
      MOV @#PS,R4 ;COY PSW
      BIC #177760,R4 ;CLEAR AWAY UNWANTED BITS
      CMP (PC)+,R4 ;CHECK THAT PROPER CC'S WERE SET
SC4: 0 ;OCTAL REPRESENTATION OF CC'S
      BEQ CON2
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
```

8135  
8136 026072  
8137 026072 012742 000620  
8138 026076 005242  
8139 026100 000000  
8140 026102 005267 177760  
8141 026106 005267 177740  
8142 026112 026727 177734 000277  
8143 026120 003753  
8144 026122 000167 000006

CCERR:

CON2:

MOV #620,-(R2)  
INC -(R2)  
HALT  
INC SC4  
INC SC3  
CMP SC3,#277  
BLE SETCD  
JMP MORO

WHICH FOLLOWS W/ 715 <====  
:MOVE TO MAILBOX # \*\*\*\*\* 620 \*\*\*\*\*  
:SET MSGTYP TO FATAL ERROR  
:SET CC FAILED OR SEQUENCE ERROR  
:SET NEXT OCTAL MAP  
:PREPARE NEXT SET CC INSTRUCTION  
:FINISHED?  
:BR IF NO  
:JUMP TO NEXT TESTS

8145  
8146  
8147  
8148  
8149  
8150  
8151  
8152  
8153  
8154 026126 000000 000000 000000  
8155 026134  
8156  
8157  
8158  
8159 026134 005212  
8160 026136 022712 000264  
8161 026142 001020  
8162 026144 005037 026126  
8163 026150 012700 026126  
8164 026154 060020  
8165  
8166 026156 022700 026130  
8167 026162 001404  
8168  
8169  
8170  
8171  
8172 026164 012742 000621  
8173 026170 005242  
8174 026172 000000  
8175  
8176 026174 022737 026130 026126 MOR1:  
8177  
8178  
8179 026202 001404  
8180  
8181  
8182  
8183  
8184 026204 012742 000622  
8185 026210 005242  
8186 026212 000000  
8187  
8188  
8189  
8190  
8191  
8192 026214 005212  
8193 026216 022712 000265  
8194 026222 001020  
8195 026224 005037 026126  
8196 026230 012700 026130  
8197 026234 060040  
8198  
8199 026236 022700 026126  
8200 026242 001404

\*\*\*\*\*  
:SBTTL TEST INSTRUCTIONS USING SAME REGISTER FOR SOURCE & DESTINATION  
:IN AUTO INCREMENT (DECREMENT) MODES AND  
:AUTO INCREMENT (DECREMENT) DEFERRED MODES,  
:CONTENTS OF THE REGISTER IN USED ARE  
:INCREMENTED (DECREMENTED) BY 2  
:BEFORE USED AS THE SOURCE OPERAND.  
:A: .WORD 0,0,0  
MORO:  
\*\*\*\*\*  
:TEST 264 TEST AUTO-INCREMENT MODE, USING R0  
\*\*\*\*\*  
TS264: INC (R2) ;UPDATE TEST NUMBER  
CMP #264,(R2) ;SEQUENCE ERROR?  
BNE TS265-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR @#A ;CLEAR LOC A  
MOV #A,R0 ;R0 STORES ADDR OF A  
ADD R0,(R0)+ ;CHECK THAT R0 IS INCR BY 2 BEFORE  
;BEING USED AS THE SOURCE OPERAND  
CMP #A+2,R0 ;R0 INCR BY 2?  
BEQ MOR1  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 767 <====  
MOV #621,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 621 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;R0 WAS NOT INCREMENTED BY 2  
:CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE  
:BEING USED IN THE "ADD" INSTR  
:LOC A CONTAINS (A+2)?  
BEQ TS265  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 757 <====  
MOV #622,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 622 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;WRONG SUM IN LOC A  
: OR SEQUENCE ERROR  
\*\*\*\*\*  
:TEST 265 AUTO-DECREMENT MODE, USING R0  
\*\*\*\*\*  
TS265: INC (R2) ;UPDATE TEST NUMBER  
CMP #265,(R2) ;SEQUENCE ERROR?  
BNE TS266-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR @#A ;CLEAR LOC A  
MOV #A+2,R0 ;R0 STORES ADDR OF A+2  
ADD R0,-(R0) ;CHECK THAT R0 IS DECR BY 2 BEFORE  
;BEING USED AS THE SOURCE OPERAND  
CMP #A,R0 ;R0 DECR BY 2?  
BEQ MOR2



```
8201 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8202 : CONDITIONAL BRANCH INST. AND <====
8203 : REPLACE THE MOVE INSTRUCTION <====
8204 : WHICH FOLLOWS W/ 767 <====
8205 026244 012742 000623 MOV #623,-(R2) :MOVE TO MAILBOX # ***** 623 *****
8206 026250 005242 INC -(R2) :SET MSGTYP TO FATAL ERROR
8207 026252 000000 HALT :RO WAS NOT DECREMENTED BY 2
8208
8209 026254 022737 026126 026126 MOR2: CMP #A,@#A :CONTENT OF R0 WAS DECR BY 2 BEFORE
8210 :BEING USED IN THE "ADD" INSTR
8211 :LOC A CONTAINS (R0)
8212 026262 001404 BEQ TS266
8213 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8214 : CONDITIONAL BRANCH INST. AND <====
8215 : REPLACE THE MOVE INSTRUCTION <====
8216 : WHICH FOLLOWS W/ 757 <====
8217 026264 012742 000624 MOV #624,-(R2) :MOVE TO MAILBOX # ***** 624 *****
8218 026270 005242 INC -(R2) :SET MSGTYP TO FATAL ERROR
8219 026272 000000 HALT :WRONG SUM IN LOC A
8220 : OR SEQUENCE ERROR
8221
8222 :*****
8223 :TEST 266 TEST AUTO-INCREMENT DEFERRED MODE, USING R0
8224 :*****
8225 026274 005212 TS266: INC (R2) :UPDATE TEST NUMBER
8226 026276 022712 000266 CMP #266,(R2) :SEQUENCE ERROR?
8227 026302 001044 BNE TS267-10 :BR TO ERROR HALT ON SEQ ERROR
8228 026304 005037 026126 CLR @#A :CLEAR LOC A
8229 026310 005037 026132 CLR @#A+4 :CLEAR LOC A+4
8230 026314 012737 026126 026130 MOV #A,@#A+2 :STORE ADDR A IN LOC A+2
8231 026322 012700 026130 MOV #A+2,R0 :R0 STORES ADDR A+2
8232 026326 060030 ADD R0,@(R0)+ :CHECK THAT R0 IS INCR BY 2 BEFORE
8233 :BEING USED AS THE SOURCE OPERAND
8234 026330 022700 026132 CMP #A+4,R0 :R0 INCR BY 2?
8235 026334 001404 BEQ MOR3
8236 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8237 : CONDITIONAL BRANCH INST. AND <====
8238 : REPLACE THE MOVE INSTRUCTION <====
8239 : WHICH FOLLOWS W/ 762 <====
8240 026336 012742 000625 MOV #625,-(R2) :MOVE TO MAILBOX # ***** 625 *****
8241 026342 005242 INC -(R2) :SET MSGTYP TO FATAL ERROR
8242 026344 000000 HALT :RO WAS NOT INCREMENTED BY 2
8243
8244 026346 022737 026126 026130 MOR3: CMP #A,@#A+2 :LOC A+2 STILL STORES ADDR A?
8245 026354 001404 BEQ MOR4
8246 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8247 : CONDITIONAL BRANCH INST. AND <====
8248 : REPLACE THE MOVE INSTRUCTION <====
8249 : WHICH FOLLOWS W/ 752 <====
8250 026356 012742 000626 MOV #626,-(R2) :MOVE TO MAILBOX # ***** 626 *****
8251 026362 005242 INC -(R2) :SET MSGTYP TO FATAL ERROR
8252 026364 000000 HALT :LOC A+2 STORES WRONG DATA
8253
8254 026366 022737 026132 026126 MOR4: CMP #A+4,@#A :CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
8255 :BEING USED IN THE "ADD" INSTR
8256 026374 001404 BEQ MOR5
```

```
8257                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8258                                     ;          CONDITIONAL BRANCH INST. AND <====
8259                                     ;          REPLACE THE MOVE INSTRUCTION <====
8260                                     ;          WHICH FOLLOWS W/ 742 <====
8261 026376 012742 000627                MOV #627,-(R2) ;MOVE TO MAILBOX # ***** 627 *****
8262 026402 005242                       INC -(R2) ;SET MSGTYP TO FATAL ERROR
8263 026404 000000                       HALT ;LOC A STORES WRONG DATA
8264
8265 026406 005737 026132                MOR5: TST @#A+4 ;LOC A+4 STILL STORES 0?
8266 026412 001404                       BEQ TS267
8267
8268                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8269                                     ;          CONDITIONAL BRANCH INST. AND <====
8270                                     ;          REPLACE THE MOVE INSTRUCTION <====
8271                                     ;          WHICH FOLLOWS W/ 733 <====
8272 026414 012742 000630                MOV #630,-(R2) ;MOVE TO MAILBOX # ***** 630 *****
8273 026420 005242                       INC -(R2) ;SET MSGTYP TO FATAL ERROR
8274 026422 000000                       HALT ;LOC A+4 DID NOT STAY CLEAR
8275                                     ; OR SEQUENCE ERROR
8276
8277 :*****
8278 :TEST 267 TEST AUTO-DECREMENT DEFERRED, USING R0
8279 :*****
8279 026424 005212                        TS267: INC (R2) ;UPDATE TEST NUMBER
8280 026426 022712 000267                CMP #267,(R2) ;SEQUENCE ERROR?
8281 026432 001044                        BNE TS270-10 ;BR TO ERROR HALT ON SEQ ERROR
8282 026434 005037 026126                CLR @#A ;CLEAR LOC A
8283 026440 005037 026132                CLR @#A+4 ;CLEAR LOC A+4
8284 026444 012700 026132                MOV #A+4,R0 ;R0 STORES ADDR A+4
8285 026450 012737 026126 026130        MOV #A,@#A+2 ;STORE ADDR A IN LOC A+2
8286 026456 060050                        ADD R0,@-(R0) ;CHECK THAT R0 IS DECR BY 2 BEFORE
8287                                     ;BEING USED AS THE SOURCE OPERAND
8288 026460 022700 026130                CMP #A+2,R0 ;R0 DECREMENTED BY 2?
8289 026464 001404                        BEQ MOR6
8290
8291                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8292                                     ;          CONDITIONAL BRANCH INST. AND <====
8293                                     ;          REPLACE THE MOVE INSTRUCTION <====
8294                                     ;          WHICH FOLLOWS W/ 762 <====
8295 026466 012742 000631                MOV #631,- R2) ;MOVE TO MAILBOX # ***** 631 *****
8296 026472 005242                       INC -(R2) ;SET MSGTYP TO FATAL ERROR
8297 026474 000000                       HALT ;R0 WAS NOT DECREMENTED BY 2
8298 026476 022737 026130 026126        MOR6: CMP #A+2,@#A ;CHECK CONTENT OF R0 WAS DECR BY 2 BEFORE
8299                                     ;BEING USED IN THE "ADD" INSTR
8300 026504 001404                        BEQ MOR7
8301
8302                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8303                                     ;          CONDITIONAL BRANCH INST. AND <====
8304                                     ;          REPLACE THE MOVE INSTRUCTION <====
8305                                     ;          WHICH FOLLOWS W/ 752 <====
8306 026506 012742 000632                MOV #632,-(R2) ;MOVE TO MAILBOX # ***** 632 *****
8307 026512 005242                       INC -(R2) ;SET MSGTYP TO FATAL ERROR
8308 026514 000000                       HALT ;LOC A STORES WRONG DATA
8309
8310 026516 022737 026126 026130        MOR7: CMP #A,@#A+2 ;LOC A+2 STILL STORES A?
8311 026524 001404                        BEQ MOR8
8312                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
```

|      |        |        |        |       |     |  |                           |
|------|--------|--------|--------|-------|-----|--|---------------------------|
| 8313 |        |        |        |       | :   | CONDITIONAL BRANCH INST. AND           | <====                     |
| 8314 |        |        |        |       | :   | REPLACE THE MOVE INSTRUCTION           | <====                     |
| 8315 |        |        |        |       | :   | WHICH FOLLOWS W/ 742                   | <====                     |
| 8316 | 026526 | 012742 | 000633 |       | :   | MOVE TO MAILBOX # ***** 633 *****      |                           |
| 8317 | 026532 | 005242 |        |       | :   | SET MSGTYP TO FATAL ERROR              |                           |
| 8318 | 026534 | 000000 |        |       | :   | LOC A+2 STORES WRONG DATA              |                           |
| 8319 |        |        |        |       | :   |  |                           |
| 8320 | 026536 | 005737 | 026132 | MOR8: | TST | @A+4                                   | : LOC A+4 STILL STORES 0? |
| 8321 | 026542 | 001404 |        |       | BEQ | TS270                                  |                           |
| 8322 |        |        |        |       |     |  |                           |
| 8323 |        |        |        |       | :   | TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <====                     |
| 8324 |        |        |        |       | :   | CONDITIONAL BRANCH INST. AND           | <====                     |
| 8325 |        |        |        |       | :   | REPLACE THE MOVE INSTRUCTION           | <====                     |
| 8326 | 026544 | 012742 | 000634 |       | :   | WHICH FOLLOWS W/ 733                   | <====                     |
| 8327 | 026550 | 005242 |        |       | :   | MOVE TO MAILBOX # ***** 634 *****      |                           |
| 8328 | 026552 | 000000 |        |       | :   | SET MSGTYP TO FATAL ERROR              |                           |
| 8329 |        |        |        |       | :   | LOC A+4 DID NOT STAY CLEAR             |                           |
| 8330 |        |        |        |       | :   | OR SEQUENCE ERROR                      |                           |

8331  
8332  
8333  
8334  
8335  
8336  
8337  
8338  
8339  
8340  
8341 026554 005212  
8342 026556 022712 000270  
8343 026562 001006  
8344 026564 012700 177777  
8345 026570 010700  
8346 026572 022700 026572  
8347 026576 001404  
8348  
8349  
8350  
8351  
8352 026600 012742 000635  
8353 026604 005242  
8354 026606 000000  
8355  
8356  
8357  
8358  
8359  
8360 026610 005212  
8361 026612 022712 000271  
8362 026616 001010  
8363 026620 012700 026126  
8364 026624 010760 000004  
8365 026630 022737 026630 026132  
8366 026636 001404  
8367  
8368  
8369  
8370  
8371 026640 012742 000636  
8372 026644 005242  
8373 026646 000000  
8374  
8375  
8376  
8377  
8378  
8379 026650 005212  
8380 026652 022712 000272  
8381 026656 001013  
8382 026660 012737 026126 026132  
8383 026666 012700 026126  
8384 026672 010770 000004  
8385 026676 022737 026676 026126  
8386 026704 001404

```
*****  
:SBTTL INSTRUCTION USING PC AS SOURCE REGISTER  
:IN INDEX, INDEX DEFERRED, RELATIVE, AND  
:RELATIVE DEFERRED MODES, DESTINATION WILL CONTAIN  
:THE PC COUNT OF THE CURRENT INSTRUCTION +4.  
*****  
:TEST 270 TEST PC AS SOURCE IN MODE 0, USING RO  
*****  
TS270: INC (R2) ;UPDATE TEST NUMBER  
CMP #270,(R2) ;SEQUENCE ERROR?  
BNE TS271-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #-1,RO ;SET ALL 1 IN RO  
PCN01: MOV PC,RO ;STORES PC IN RO  
CMP #PCN01+2,RO ;RO STORES PC+2?  
BEQ TS271  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 771 <====  
MOV #635,-(R2) ;MOVE TO MAILBOX # ***** 635 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RO STORED WRONG VALUE  
: OR SEQUENCE ERROR  
*****  
:TEST 271 TEST PC AS SOURCE IN MODE 6, USING RO  
*****  
TS271: INC (R2) ;UPDATE TEST NUMBER  
CMP #271,(R2) ;SEQUENCE ERROR?  
BNE TS272-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #A,RO ;RO STORES ADDR A  
PCN2: MOV PC,4(R0) ;EFFECTIVE ADDR IS A+4  
CMP #PCN2+4,@#A+4 ;LOC A+4 STORES PC+4?  
BEQ TS272  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 767 <====  
MOV #636,-(R2) ;MOVE TO MAILBOX # ***** 636 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;LOC A+4 STORED WRONG VALUE  
: OR SEQUENCE ERROR  
*****  
:TEST 272 TEST PC AS SOURCE IN MODE 7, USING RO  
*****  
TS272: INC (R2) ;UPDATE TEST NUMBER  
CMP #272,(R2) ;SEQUENCE ERROR?  
BNE TS273-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #A,@#A+4 ;LOC A+4 STORES ADDR A  
PCN3: MOV #A,RO ;RO STORES ADDR A  
MOV PC,@4(R0) ;EFFECTIVE ADDR IS A  
CMP #PCN3+4,@#A ;LOC A STORES PC+4?  
BEQ TS273
```

```
8387      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
8388      ;                               CONDITIONAL BRANCH INST. AND <====  
8389      ;                               REPLACE THE MOVE INSTRUCTION <====  
8390      ;                               WHICH FOLLOWS W/ 764 <====  
8391 026706 012742 000637      MOV #637,-(R2) ;MOVE TO MAILBOX # ***** 637 *****  
8392 026712 005242      INC -(R2) ;SET MSGTYP TO FATAL ERROR  
8393 026714 000000      HALT ;LOC A STORED WRONG VALUE  
8394      ; OR SEQUENCE ERROR
```

```
8396      ;*****  
8397 ;TEST 273 TEST PC AS SOURCE IN RELATIVE DEFERRED MODE ,USING R0  
8398      ;*****
```

```
8399 026716 005212      TS273: INC (R2) ;UPDATE TEST NUMBER  
8400 026720 022712 000273      CMP #273,(R2) ;SEQUENCE ERROR?  
8401 026724 001011      BNE TS274-10 ;BR TO ERROR HALT ON SEQ ERROR  
8402 026726 012737 026130 026126      MOV #A+2,@#A ;LOC A STORES ADDR A+2  
8403 026734 010777 177166      PCN4: MOV PC,@A ;EFFECTIVE ADDR IS A+2  
8404 026740 022737 026740 026130      CMP #PCN4+4,@#A+2 ;LOC A+2 STORES PC+4?  
8405 026746 001404      BEQ TS274
```

```
8406      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
8407      ;                               CONDITIONAL BRANCH INST. AND <====  
8408      ;                               REPLACE THE MOVE INSTRUCTION <====  
8409      ;                               WHICH FOLLOWS W/ 766 <====  
8410 026750 012742 000640      MOV #640,-(R2) ;MOVE TO MAILBOX # ***** 640 *****  
8411 026754 005242      INC -(R2) ;SET MSGTYP TO FATAL ERROR  
8412 026756 000000      HALT ;LOC A+2 STORED WRONG VALUE  
8413      ; OR SEQUENCE ERROR
```

```
8415      ;*****  
8416 ;TEST 274 TEST PC AS SOURCE IN RELATIVE MODE ,USING R0  
8417      ;*****
```

```
8418 026760 005212      TS274: INC (R2) ;UPDATE TEST NUMBER  
8419 026762 022712 000274      CMP #274,(R2) ;SEQUENCE ERROR?  
8420 026766 001010      BNE TS275-10 ;BR TO ERROR HALT ON SEQ ERROR  
8421 026770 005037 026126      CLR @#A ;CLEAR A  
8422 026774 010767 177126      PCN5: MOV PC,A ;EFFECTIVE ADDR IS A  
8423 027000 022737 027000 026126      CMP #PCN5+4,@#A ;LOC A STORES PC+4?  
8424 027006 001404      BEQ TS275
```

```
8425      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
8426      ;                               CONDITIONAL BRANCH INST. AND <====  
8427      ;                               REPLACE THE MOVE INSTRUCTION <====  
8428      ;                               WHICH FOLLOWS W/ 767 <====  
8429 027010 012742 000641      MOV #641,-(R2) ;MOVE TO MAILBOX # ***** 641 *****  
8430 027014 005242      INC -(R2) ;SET MSGTYP TO FATAL ERROR  
8431 027016 000000      HALT ;LOCATION A STORED WRONG VALUE  
8432      ; OR SEQUENCE ERROR
```

```
8434      ;*****  
8435 ;SBTTL THE NEXT THREE TESTS EXERCISE MASKING ACTION OF MICROCODES.  
8436      ;*****
```

```
8437 ;TEST 275 TEST SUB INSTRUCTION, SM=0, DM=2  
8438      ;*****
```

```
8439 027020 005212      TS275: INC (R2) ;UPDATE TEST NUMBER  
8440 027022 022712 000275      CMP #275,(R2) ;SEQUENCE ERROR?  
8441 027026 001013      BNE TS276-10 ;BR TO ERROR HALT ON SEQ ERROR  
8442 027030 012737 052525 000000      MOV #052525,@#0 ;SET UP LOC 0
```

```
8443 027036 012701 050505      MOV      #050505,R1      ;SET UP R1
8444 027042 005000              CLR      RO              ;CLEAR RO
8445 027044 160120              SUB      R1,(RO)+        ;SUBTRACTION, SM=0,DM=2
8446 027046 022737 002020 000000  CMP      #2020,@#0      ;CHECK DIFFERENCE AT LOC 0
8447 027054 001404              BEQ      TS276
8448                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8449                                ;          CONDITIONAL BRANCH INST. AND <====
8450                                ;          REPLACE THE MOVE INSTRUCTION <====
8451                                ;          WHICH FOLLOWS W/ 764 <====
8452 027056 012742 000642      MOV      #642,-(R2)     ;MOVE TO MAILBOX # ***** 642 *****
8453 027062 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
8454 027064 000000              HALT                    ;WRONG RESULT FROM SUBTRACTION
8455                                ; OR SEQUENCE ERROR
8456
8457                                ;*****
8458                                ;TEST 276      TEST MFPD WITH RO, IN MODE 2
8459                                ;*****
8460 027066 005212              TS276:  INC      (R2)      ;UPDATE TEST NUMBER
8461 027070 022712 000276      CMP      #276,(R2)     ;SEQUENCE ERROR?
8462 027074 001020              BNE     TS277-10       ;BR TO ERROR HALT ON SEQ ERROR
8463 027076 012737 052525 000000  MOV      #052525,@#0   ;SET UP LOC 0
8464 027104 005000              CLR      RO              ;CLEAR RO
8465 027106 012767 170000 150662  MOV      #170000,PS    ;SET USER MODE ON, CURRENT & PREVIOUS
8466 027114 012706 027352      MOV      #USTBOT,R6    ;SET USER STACK POINTER
8467 027120 106520              MFPD    (RO)+          ;MODE 2, MFPD
8468 027122 005067 150650      CLR      PS            ;SET KERNEL MODE
8469 027126 022767 052525 000214  CMP      #052525,USTBOT-2 ;CHECK DATA ON STACK
8470 027134 001404              BEQ     BRMFPD         ;BR IF NO ERROR
8471 027136 012742 000643      MOV      #643,-(R2)   ;MOVE TO MAILBOX # ***** 643 *****
8472 027142 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
8473 027144 000000              HALT                    ;INCORRECT DATA FROM MFPD
8474 027146
8475
8476                                ;*****
8477                                ;TEST 277      TEST MTPD WITH RO, IN MODE 2
8478                                ;*****
8479 027146 005212              TS277:  INC      (R2)      ;UPDATE TEST NUMBER
8480 027150 022712 000277      CMP      #277,(R2)     ;SEQUENCE ERROR?
8481 027154 001026              BNE     END1           ;BR TO ERROR HALT ON SEQ ERROR
8482 027156 012767 170000 150612  MOV      #170000,PS    ;SET USER MODE ON, CURRENT & PREVIOUS
8483 027164 012706 027352      MOV      #USTBOT,R6    ;SET USER STACK POINTER
8484 027170 012746 125252      MOV      #125252,-(R6) ;PUSH DATA IN USER STACK
8485 027174 012737 000000 000000  MOV      #0,@#0        ;CLEAR LOC 0
8486 027202 005000              CLR      RO              ;CLEAR RO
8487 027204 106620              MTPD    (RO)+          ;MODE 2, MTPD
8488 027206 005067 150564      CLR      PS            ;SET KERNEL MODE
8489 027212 022737 125252 000000  CMP      #125252,@#0   ;CHECK DATA ON LOC 0
8490 027220 001514              BEQ     SECPRT         ;BR TO TRAP TEST IF NO ERROR
8491 027222 012742 000644      MOV      #644,-(R2)   ;MOVE TO MAILBOX # ***** 644 *****
8492 027226 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
8493 027230 000000              HALT                    ;INCORRECT DATA FROM MTPD
8494 027232
8495 027232 012742 000645      END1:  MOV      #645,-(R2)   ;MOVE TO MAILBOX # ***** 645 *****
8496 027236 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
8497 027240 000000              HALT                    ;SEQUENCE ERROR
8498
```

8499  
8500 027242 000402  
8501 027244 001002  
8502 027246 001402  
8503 027250 002002  
8504 027252 002402  
8505 027254 003002  
8506 027256 003402  
8507 027260 100002  
8508 027262 100402  
8509 027264 101002  
8510 027266 101402  
8511 027270 102002  
8512 027272 102402  
8513 027274 103002  
8514 027276 103402  
8515  
8516 000002  
8517 027300 177777  
8518 027302 170360  
8519 027304 007417  
8520 027306 146063  
8521 027310 031714  
8522 027312 140060  
8523 027314 037717  
8524  
8525 027316 177400  
8526 027320 000377  
8527 027322 120240  
8528 027324 057537  
8529 027326 146314  
8530 027330 031463  
8531 027332 125252  
8532 027334 052525  
8533 000010  
8534  
8535  
8536 027336 000006  
8537 027352  
8538  
8539  
8540  
8541  
8542  
8543  
8544 027352  
8545 027352 012742 000646  
8546 027356 005242  
8547 027360 000000  
8548 027362  
8549 027362 012742 000647  
8550 027366 005242  
8551 027370 000000  
8552 027372  
8553 027372 012742 000650  
8554 027376 005242

BRTAB: BR .+6  
BNE .+6  
BEQ .+6  
BGE .+6  
BLT .+6  
BGT .+6  
BLE .+6  
BPL .+6  
BMI .+6  
BHI .+6  
BLOS .+6  
BVC .+6  
BVS .+6  
BCC .+6  
BCS .+6

;SAME AS BHIS  
;SAME AS BLO

.RADIX 2  
YNTAB: 1111111111111111  
1111000011110000  
0000111100001111  
1100110000110011  
0011001111001100  
1100000000110000  
0011111111001111  
1111111100000000  
0000000011111111  
1010000010100000  
0101111101011111  
1100110011001100  
0011001100110011  
1010101010101010  
0101010101010101

;BR  
;BNE: Z=0  
;BEQ: Z=1  
;BGE: N XOR V =0  
;BLT: N XOR V =1  
;BGT: Z+(N XOR V) =0  
;BLE: Z+(N XOR V) =1

;BPL: N=0  
;BMI: N=1  
;BHI: C+Z=0  
;BLOS: C+Z=1  
;BVC: V=0  
;BVS: V=1  
;BCC: C=0  
;BCS: C=1

.RADIX 8

.EVEN  
.BLKW 6

USTBOT:

\*\*\*\*\*  
; THE FOLLOWING ARE SPECIAL CPU TRAP  
; HANDLERS TO TRAP AND REPORT SPECIAL TRAPS.  
\*\*\*\*\*

T04: MOV #646, -(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 646 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TRAPPED THRU LOC. 4  
T010: MOV #647, -(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 647 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TRAPPED THRU LOC. 10  
T014: MOV #650, -(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 650 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR

```
8555 027400 000000
8556 027402
8557 027402 012742 000651
8558 027406 005242
8559 027410 000000
8560 027412
8561 027412 012742 000652
8562 027416 005242
8563 027420 000000
8564 027422
8565 027422 012742 000653
8566 027426 005242
8567 027430 000000
8568 027432
8569 027432 012742 000654
8570 027436 005242
8571 027440 000000
8572 027442
8573 027442 012742 000655
8574 027446 005242
8575 027450 000000
8576
8577 027452
```

```
T030: HALT ;TRAPPED THRU LOC. 14
      MOV #651,-(R2) ;MOVE TO MAILBOX # ***** 651 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;TRAPPED THRU LOC. 30
T034: MOV #652,-(R2) ;MOVE TO MAILBOX # ***** 652 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;TRAPPED THRU LOC. 34
T0114: MOV #653,-(R2) ;MOVE TO MAILBOX # ***** 653 *****
       INC -(R2) ;SET MSGTYP TO FATAL ERROR
       HALT ;TRAPPED THRU LOC. 114
T0244: MOV #654,-(R2) ;MOVE TO MAILBOX # ***** 654 *****
       INC -(R2) ;SET MSGTYP TO FATAL ERROR
       HALT ;TRAPPED THRU LOC. 244
T0250: MOV #655,-(R2) ;MOVE TO MAILBOX # ***** 655 *****
       INC -(R2) ;SET MSGTYP TO FATAL ERROR
       HALT ;TRAPPED THRU LOC. 250
.SBTTL ** STARTING OF TRAP TEST **
SECPR:
```



8578  
8579 000000  
8580  
8581  
8582  
8583  
8584  
8585  
8586  
8587  
8588  
8589  
8590  
8591  
8592  
8593  
8594  
8595  
8596  
8597 000006  
8598 000006  
8599 000003  
8600 000001  
8601 000005  
8602 000002  
8603 000000  
8604 000003  
8605 000004  
8606 000004  
8607 000014  
8608 000030  
8609 000020  
8610 000034  
8611 177564  
8612 177560  
8613 177564  
8614 177566  
8615 000240  
8616 000240  
8617 177776  
8618 000077  
8619 000010  
8620 004700  
8621 000100  
8622 177776  
8623 001000  
8624  
8625  
8626

.REPT 0

PART TWO:

F11 TRAP TEST, THIS IS THE SECOND  
PART OF THE MAIN PROGRAM.

ABSTRACT

THIS IS A TEST OF ALL OPERATIONS AND INSTRUCTIONS THAT CAUSE  
TRAPS. ALSO TESTED ARE TRAP OVERFLOW CONDITIONS,  
ODDITIES OF REGISTER 6, INTERRUPTS , THE RESET AND WAIT INSTRUCTIONS.

.ENDR

.LIST ME

.NLIST MC,MD,CND

.ABS

SP=%6

R6=%6

TAB=%3

LAST=%1

FIRST=%5

R2=%2

HLT=HALT

TRT=3

ITRAP5=4

RTRAP5=4

RTRAP4=14

RTRAP3=30

RTRAP2=20

RTRAP1=34

TICSR=177564

TRCSR=177560

TPS=177564

TPB=177566

BELL=240

NOP=240

STATUS=177776

TRAPA=77

RTRAP=10

ILLA=004700

ILLB=100

CC=177776

BUFF=STBOT

;RESERVED INST AND ILLEGAL ADDRESSES  
;FOR TRACE TRAP  
;FOR EMULATOR TRAP  
;FOR IOT TRAP  
;FOR TRAP INST

8627  
8628  
8629 000000  
8630  
8631 027452 000167 000024  
8632 027456 000000  
8633 027460 000000  
8634 027462 000000  
8635 027464 000000  
8636 027466 000000  
8637 027470 000000  
8638 027472 052525  
8639 027474 052400  
8640 027476 000000  
8641 027500 000000  
8642  
8643 027502  
8644  
8645  
8646  
8647 027502 005212  
8648 027504 022712 000300  
8649 027510 001127  
8650 027512 005006  
8651 027514 112667 150260  
8652 027520 020627 000002  
8653 027524 001404  
8654  
8655  
8656  
8657  
8658 027526 012742 000656  
8659 027532 005242  
8660 027534 000000  
8661  
8662 027536 012706 001000  
8663 027542 114627 000000  
8664 027546 020627 000776  
8665 027552 001404  
8666  
8667  
8668  
8669  
8670 027554 012742 000657  
8671 027560 005242  
8672 027562 000000  
8673  
8674 027564 005006  
8675 027566 112626  
8676 027570 020627 000004  
8677 027574 001404  
8678  
8679  
8680  
8681  
8682 027576 012742 000660

:SPECIAL CASE OF ODD;.EVEN .BYTE AND REGISTER 6  
HERE=0

JMP TESTN1  
K1: 0  
K2: 0  
K3: 0  
K4: 0  
K5: 0  
K6: 0  
K7: 052525  
K10: 052400  
K11: 0  
K12: 0

TESTN1:

\*\*\*\*\*  
:TEST 300 TEST AUTO INCREMENT AND DECREMENT OF R6 FOR WORD AND BYTES  
\*\*\*\*\*

TS300: INC (R2) ;UPDATE TEST NUMBER  
CMP #300,(R2) ;SEQUENCE ERROR?  
BNE TS301-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR %6  
MOVB (6)+,HERE ;SIX SHOULD INCREMENT BY TWO  
CMP %6,#2  
BEQ BR1

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 771 <====

MOV #656,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 656 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;R6 DID NOT AUTO INCREMENT BY TWO

BR1: MOV #1000,%6  
MOVB -(6),#HERE ;SHOULD DECREMENT BY TWO  
CMP %6,#776  
BEQ BR2

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 756 <====

MOV #657,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 657 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;R6 DID NOT AUTO DECREMENT BY 2

BR2: CLR %6  
MOVB (6)+,(6)+ ;DOUBLES AUTO INCREMENT OF R6  
CMP %6,#4  
BEQ BR3

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 745 <====

MOV #660,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 660 \*\*\*\*\*

|      |        |        |        |      |       |            |  |       |
|------|--------|--------|--------|------|-------|------------|--|-------|
| 8683 | 027602 | 005242 |        | INC  | -(R2) |            | ;SEI MSGTYP TO FATAL ERROR               |       |
| 8684 | 027604 | 000000 |        | HALT |       |            | ;WRONG AUTO INCREMENT OF R6              |       |
| 8685 |        |        |        |      |       |            |  |       |
| 8686 | 027606 | 005006 |        | BR3: | CLR   | %6         |  |       |
| 8687 | 027610 | 005004 |        |      | CLR   | %4         |  |       |
| 8688 | 027612 | 122624 |        |      | CMPB  | (6)+,(4)+  | ;TEST INCREMENT OF R6                    |       |
| 8689 | 027614 | 020627 | 000002 |      | CMP   | %6,#2      |  |       |
| 8690 | 027620 | 001404 |        |      | BEQ   | BR4        |  |       |
| 8691 |        |        |        |      |       |            | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 8692 |        |        |        |      |       |            | CONDITIONAL BRANCH INST. AND             | <==== |
| 8693 |        |        |        |      |       |            | REPLACE THE MOVE INSTRUCTION             | <==== |
| 8694 |        |        |        |      |       |            | WHICH FOLLOWS W/ 733                     | <==== |
| 8695 | 027622 | 012742 | 000661 |      | MOV   | #661,-(R2) | ;MOVE TO MAILBOX # ***** 661 *****       |       |
| 8696 | 027626 | 005242 |        |      | INC   | -(R2)      | ;SET MSGTYP TO FATAL ERROR               |       |
| 8697 | 027630 | 000000 |        |      | HALT  |            | ;WRONG INCREMENT OF R6                   |       |
| 8698 |        |        |        |      |       |            |  |       |
| 8699 | 027632 | 005006 |        | BR4: | CLR   | %6         |  |       |
| 8700 | 027634 | 005004 |        |      | CLR   | %4         |  |       |
| 8701 | 027636 | 122426 |        |      | CMPB  | (4)+,(6)+  | ;TEST INCREMENT OF R6                    |       |
| 8702 | 027640 | 020627 | 000002 |      | CMP   | %6,#2      |  |       |
| 8703 | 027644 | 001404 |        |      | BEQ   | BR5        |  |       |
| 8704 |        |        |        |      |       |            | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 8705 |        |        |        |      |       |            | CONDITIONAL BRANCH INST. AND             | <==== |
| 8706 |        |        |        |      |       |            | REPLACE THE MOVE INSTRUCTION             | <==== |
| 8707 |        |        |        |      |       |            | WHICH FOLLOWS W/ 721                     | <==== |
| 8708 | 027646 | 012742 | 000662 |      | MOV   | #662,-(R2) | ;MOVE TO MAILBOX # ***** 662 *****       |       |
| 8709 | 027652 | 005242 |        |      | INC   | -(R2)      | ;SET MSGTYP TO FATAL ERROR               |       |
| 8710 | 027654 | 000000 |        |      | HALT  |            | ;WRONG INCREMENT OF R6                   |       |
| 8711 |        |        |        |      |       |            |  |       |
| 8712 | 027656 | 005006 |        | BR5: | CLR   | %6         |  |       |
| 8713 | 027660 | 005004 |        |      | CLR   | %4         |  |       |
| 8714 | 027662 | 122624 |        |      | CMPB  | (6)+,(4)+  | ;TEST INCREMENT OF R4                    |       |
| 8715 | 027664 | 020427 | 000001 |      | CMP   | %4,#1      |  |       |
| 8716 | 027670 | 001404 |        |      | BEQ   | BR6        |  |       |
| 8717 |        |        |        |      |       |            | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 8718 |        |        |        |      |       |            | CONDITIONAL BRANCH INST. AND             | <==== |
| 8719 |        |        |        |      |       |            | REPLACE THE MOVE INSTRUCTION             | <==== |
| 8720 |        |        |        |      |       |            | WHICH FOLLOWS W/ 707                     | <==== |
| 8721 | 027672 | 012742 | 000663 |      | MOV   | #663,-(R2) | ;MOVE TO MAILBOX # ***** 663 *****       |       |
| 8722 | 027676 | 005242 |        |      | INC   | -(R2)      | ;SET MSGTYP TO FATAL ERROR               |       |
| 8723 | 027700 | 000000 |        |      | HALT  |            | ;WRONG INCREMENT OF R4                   |       |
| 8724 | 027702 | 005006 |        | BR6: | CLR   | %6         |  |       |
| 8725 | 027704 | 005004 |        |      | CLR   | %4         |  |       |
| 8726 | 027706 | 122426 |        |      | CMPB  | (4)+,(6)+  | ;TEST INCREMENT OF R6                    |       |
| 8727 | 027710 | 020627 | 000002 |      | CMP   | %6,#2      |  |       |
| 8728 | 027714 | 001404 |        |      | BEQ   | BR7        |  |       |
| 8729 |        |        |        |      |       |            | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 8730 |        |        |        |      |       |            | CONDITIONAL BRANCH INST. AND             | <==== |
| 8731 |        |        |        |      |       |            | REPLACE THE MOVE INSTRUCTION             | <==== |
| 8732 |        |        |        |      |       |            | WHICH FOLLOWS W/ 675                     | <==== |
| 8733 | 027716 | 012742 | 000664 |      | MOV   | #664,-(R2) | ;MOVE TO MAILBOX # ***** 664 *****       |       |
| 8734 | 027722 | 005242 |        |      | INC   | -(R2)      | ;SET MSGTYP TO FATAL ERROR               |       |
| 8735 | 027724 | 000000 |        |      | HALT  |            | ;WRONG INCREMENT OF R6                   |       |
| 8736 |        |        |        |      |       |            |  |       |
| 8737 | 027726 | 005006 |        | BR7: | CLR   | %6         |  |       |
| 8738 | 027730 | 005004 |        |      | CLR   | %4         |  |       |

```
8739 027732 122426          CMPB  (4)+,(6)+      ;TEST INCREMENT OF R4
8740 027734 020427 000001    CMP   %4,#1
8741 027740 001404          BEQ   BR10
8742                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8743                               ;          CONDITIONAL BRANCH INST. AND <====
8744                               ;          REPLACE THE MOVE INSTRUCTION <====
8745                               ;          WHICH FOLLOWS W/ 663 <====
8746 027742 012742 000665    MOV   #665,-(R2)     ;MOVE TO MAILBOX # ***** 665 *****
8747 027746 005242          INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8748 027750 000000          HALT                ;WRONG INCREMENT OF R4
8749
8750 027752 012706 001000    BR10: MOV  #1000,%6
8751 027756 124627 000000    CMPB  -(6),#HERE    ;TEST DECREMENT OF R6
8752 027762 022706 000776    CMP   #776,%6
8753 027766 001404          BEQ   TS301
8754                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8755                               ;          CONDITIONAL BRANCH INST. AND <====
8756                               ;          REPLACE THE MOVE INSTRUCTION <====
8757                               ;          WHICH FOLLOWS W/ 650 <====
8758 027770 012742 000666    MOV   #666,-(R2)     ;MOVE TO MAILBOX # ***** 666 *****
8759 027774 005242          INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8760 027776 000000          HALT                ;WRONG DECREMENT OF R6,OR WRONG $TSTNM
8761                               ; OR SEQUENCE ERROR
8762
8763 ;*****
8764 ;TEST 301 TEST TRANSFER OF .BYTE USING R6
8765 ;*****
8765 030000 005212          TS301: INC  (R2)      ;UPDATE TEST NUMBER
8766 030002 022712 000301    CMP   #301,(R2)     ;SEQUENCE ERROR?
8767 030006 001133          BNE   TS302-10     ;BR TO ERROR HALT ON SEQ ERROR
8768 030010 012767 123456 177450  MOV   #123456,K5
8769 030016 012767 050505 177432  MOV   #050505,K1
8770 030024 012705 027456          MOV   #K1,%5       ;%5=(050505)K1
8771 030030 012706 027466          MOV   #K5,%6       ;%6=(123456)K5
8772 030034 112625          MOVVB (6)+,(5)+    ;LOW .BYTE OF R6 TO R5
8773 030036 022767 050456 177412  CMP   #050456,K1
8774 030044 001404          BEQ   BR11
8775                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8776                               ;          CONDITIONAL BRANCH INST. AND <====
8777                               ;          REPLACE THE MOVE INSTRUCTION <====
8778                               ;          WHICH FOLLOWS W/ 760 <====
8779 030046 012742 000667    MOV   #667,-(R2)     ;MOVE TO MAILBOX # ***** 667 *****
8780 030052 005242          INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8781 030054 000000          HALT                ;FALSE TRANSFER OF .BYTE
8782
8783 030056 012767 123456 177402  BR11: MOV  #123456,K5
8784 030064 012767 050505 177364  MOV   #050505,K1
8785 030072 012705 027456          MOV   #K1,%5       ;%5(050505)K1
8786 030076 012706 027470          MOV   #K6,%6       ;%6(123456)K5
8787 030102 114625          MOVVB -(6),(5)+    ;LOW .BYTE OF R6 TO R5 (DECREMENT)
8788 030104 026727 177346 050456  CMP   K1,#050456
8789 030112 001404          BEQ   BR12
8790                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8791                               ;          CONDITIONAL BRANCH INST. AND <====
8792                               ;          REPLACE THE MOVE INSTRUCTION <====
8793                               ;          WHICH FOLLOWS W/ 735 <====
8794 030114 012742 000670    MOV   #670,-(R2)     ;MOVE TO MAILBOX # ***** 670 *****
```

```
8795 030120 005242          INC      -(R2)          ;SEI MSGTYP TO FATAL ERROR
8796 030122 000000          HALT                    ;FALSE R6 .BYTE TRANSFER
8797
8798 030124 012767 123456 177324 BR12:  MOV      #123456,K1
8799 030132 012767 050505 177326      MOV      #050505,K5
8800 030140 012705 027456          MOV      #K1,%5          ;(123456)
8801 030144 012706 027466          MOV      #K5,%6          ;(050505)
8802 030150 112526          MOVVB   (5)+,(6)+        ;LOW OF R5 TO LOW OF R6
8803 030152 022767 050456 177306      CMP      #050456,K5
8804 030160 001404          BEQ      BR13
8805
8806
8807
8808
8809 030162 012742 000671          MOV      #671,-(R2)      ; MOVE TO MAILBOX # ***** 671 *****
8810 030166 005242          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
8811 030170 000000          HALT                    ; FALSE R6 .BYTE TRANSFER
8812
8813 030172 012767 123456 177256 BR13:  MOV      #123456,K1
8814 030200 012767 050505 177260      MOV      #050505,K5
8815 030206 012705 027457          MOV      #K1+1,%5        ;123456
8816 030212 012706 027466          MOV      #K5,%6          ;050505
8817 030216 112526          MOVVB   (5)+,(6)+        ;HIGH OF R5 TO LOW OF R6
8818 030220 026727 177242 050647      CMP      K5,#050647
8819 030226 001404          BEQ      BR14
8820
8821
8822
8823
8824 030230 012742 000672          MOV      #672,-(R2)      ; MOVE TO MAILBOX # ***** 672 *****
8825 030234 005242          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
8826 030236 000000          HALT                    ; FALSE R6 .BYTE TRANSFER
8827
8828 030240 012767 123456 177210 BR14:  MOV      #123456,K1
8829 030246 012767 050505 177212      MOV      #050505,K5
8830 030254 012705 027457          MOV      #K1+1,%5        ;R5-123456-ODD ADDRESS
8831 030260 012706 027466          MOV      #K5,%6          ;R6-050505--.EVEN ADDRESS
8832 030264 112625          MOVVB   (6)+,(5)+        ;LOW OF R6 TO HIGH OF R5
8833 030266 022767 042456 177162      CMP      #042456,K1
8834 030274 001404          BEQ      TS302
8835
8836
8837
8838
8839 030276 012742 000673          MOV      #673,-(R2)      ; MOVE TO MAILBOX # ***** 673 *****
8840 030302 005242          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
8841 030304 000000          HALT                    ; FAILED LOW OF 6 TO HIGH OF 5,OR WRONG $TSTNM
8842
8843
8844
8845
8846 030306 005212          TS302: INC      (R2)          ; UPDATE TEST NUMBER
8847 030310 022712 000302          CMP      #302,(R2)      ; SEQUENCE ERROR?
8848 030314 001074          BNE     TS303-10        ; BR TO ERROR HALT ON SEQ ERROR
8849 030316 126767 177150 177147      CMPB   K7,K7+1          ; SAME .WORD LOW TO HIGH
8850 030324 001404          BEQ      BR15
```

|      |        |        |        |        |       |            |            |  |       |
|------|--------|--------|--------|--------|-------|------------|------------|--|-------|
| 8851 |        |        |        |        |       |            |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 8852 |        |        |        |        |       |            |            | : CONDITIONAL BRANCH INST. AND           | <==== |
| 8853 |        |        |        |        |       |            |            | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 8854 |        |        |        |        |       |            |            | : WHICH FOLLOWS W/ 773                   | <==== |
| 8855 | 030326 | 012742 | 000674 |        | MOV   | #674,-(R2) |            | : MOVE TO MAILBOX # ***** 674 *****      |       |
| 8856 | 030332 | 005242 |        |        | INC   | -(R2)      |            | : SET MSGTYP TO FATAL ERROR              |       |
| 8857 | 030334 | 000000 |        |        | HALT  |            |            | : SHOULD COMPARE LOW TO HIGH             |       |
| 8858 |        |        |        |        |       |            |            |  |       |
| 8859 | 030336 | 126767 | 177131 | 177126 | BR15: | CMPB       | K7+1,K7    | : COMPARE ODD TO .EVEN SAME .WORD        |       |
| 8860 | 030344 | 001404 |        |        | BEQ   | BR16       |            |  |       |
| 8861 |        |        |        |        |       |            |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 8862 |        |        |        |        |       |            |            | : CONDITIONAL BRANCH INST. AND           | <==== |
| 8863 |        |        |        |        |       |            |            | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 8864 |        |        |        |        |       |            |            | : WHICH FOLLOWS W/ 763                   | <==== |
| 8865 | 030346 | 012742 | 000675 |        | MOV   | #675,-(R2) |            | : MOVE TO MAILBOX # ***** 675 *****      |       |
| 8866 | 030352 | 005242 |        |        | INC   | -(R2)      |            | : SET MSGTYP TO FATAL ERROR              |       |
| 8867 | 030354 | 000000 |        |        | HALT  |            |            | : ODD TO .EVEN .BYTE FAILURE             |       |
| 8868 |        |        |        |        |       |            |            |  |       |
| 8869 | 030356 | 126767 | 177113 | 177106 | BR16: | CMPB       | K10+1,K7   | : SEQUENTIAL .BYTES                      |       |
| 8870 | 030364 | 001404 |        |        | BEQ   | BR17       |            |  |       |
| 8871 |        |        |        |        |       |            |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 8872 |        |        |        |        |       |            |            | : CONDITIONAL BRANCH INST. AND           | <==== |
| 8873 |        |        |        |        |       |            |            | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 8874 |        |        |        |        |       |            |            | : WHICH FOLLOWS W/ 753                   | <==== |
| 8875 | 030366 | 012742 | 000676 |        | MOV   | #676,-(R2) |            | : MOVE TO MAILBOX # ***** 676 *****      |       |
| 8876 | 030372 | 005242 |        |        | INC   | -(R2)      |            | : SET MSGTYP TO FATAL ERROR              |       |
| 8877 | 030374 | 000000 |        |        | HALT  |            |            | : ODD TO .EVEN FAILED                    |       |
| 8878 |        |        |        |        |       |            |            |  |       |
| 8879 | 030376 | 126767 | 177072 | 177064 | BR17: | CMPB       | K10,K6     |  |       |
| 8880 | 030404 | 001404 |        |        | BEQ   | BR20       |            |  |       |
| 8881 |        |        |        |        |       |            |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 8882 |        |        |        |        |       |            |            | : CONDITIONAL BRANCH INST. AND           | <==== |
| 8883 |        |        |        |        |       |            |            | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 8884 |        |        |        |        |       |            |            | : WHICH FOLLOWS W/ 743                   | <==== |
| 8885 | 030406 | 012742 | 000677 |        | MOV   | #677,-(R2) |            | : MOVE TO MAILBOX # ***** 677 *****      |       |
| 8886 | 030412 | 005242 |        |        | INC   | -(R2)      |            | : SET MSGTYP TO FATAL ERROR              |       |
| 8887 | 030414 | 000000 |        |        | HALT  |            |            | : .EVEN TO EVEN FAILED                   |       |
| 8888 | 030416 | 126767 | 177051 | 177051 | BR20: | CMPB       | K7+1,K10+1 |  |       |
| 8889 | 030424 | 001404 |        |        | BEQ   | BR21       |            |  |       |
| 8890 |        |        |        |        |       |            |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 8891 |        |        |        |        |       |            |            | : CONDITIONAL BRANCH INST. AND           | <==== |
| 8892 |        |        |        |        |       |            |            | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 8893 |        |        |        |        |       |            |            | : WHICH FOLLOWS W/ 733                   | <==== |
| 8894 | 030426 | 012742 | 000700 |        | MOV   | #700,-(R2) |            | : MOVE TO MAILBOX # ***** 700 *****      |       |
| 8895 | 030432 | 005242 |        |        | INC   | -(R2)      |            | : SET MSGTYP TO FATAL ERROR              |       |
| 8896 | 030434 | 000000 |        |        | HALT  |            |            | : ODD TO ODD FAILED                      |       |
| 8897 |        |        |        |        |       |            |            |  |       |
| 8898 | 030436 | 126767 | 177032 | 177031 | BR21: | CMPB       | K10,K10+1  |  |       |
| 8899 | 030444 | 001004 |        |        | BNE   | BR22       |            |  |       |
| 8900 |        |        |        |        |       |            |            | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 8901 |        |        |        |        |       |            |            | : CONDITIONAL BRANCH INST. AND           | <==== |
| 8902 |        |        |        |        |       |            |            | : REPLACE THE MOVE INSTRUCTION           | <==== |
| 8903 |        |        |        |        |       |            |            | : WHICH FOLLOWS W/ 723                   | <==== |
| 8904 | 030446 | 012742 | 000701 |        | MOV   | #701,-(R2) |            | : MOVE TO MAILBOX # ***** 701 *****      |       |
| 8905 | 030452 | 005242 |        |        | INC   | -(R2)      |            | : SET MSGTYP TO FATAL ERROR              |       |
| 8906 | 030454 | 000000 |        |        | HALT  |            |            | : LOW TO HIGH IN SAME .WORD FAILED       |       |

```
8907
8908 030456 126767 177013 177011 BR22:  CMPB  K10+1,K10+1
8909 030464 001404                BEQ   BR23
8910
8911                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8912                                ;          CONDITIONAL BRANCH INST. AND <====
8913                                ;          REPLACE THE MOVE INSTRUCTION <====
8914                                ;          WHICH FOLLOWS W/ 713 <====
8914 030466 012742 000702                MOV   #702,-(R2) ;MOVE TO MAILBOX # ***** 702 *****
8915 030472 005242                INC   -(R2)      ;SET MSGTYP TO FATAL ERROR
8916 030474 000000                HALT                ;HIGH TO LOW IN SAME .WORD FAILED
8917
8918 030476 126767 176772 176767 BR23:  CMPB  K10,K7+1
8919 030504 001004                BNE   TS303
8920
8921                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8922                                ;          CONDITIONAL BRANCH INST. AND <====
8923                                ;          REPLACE THE MOVE INSTRUCTION <====
8924                                ;          WHICH FOLLOWS W/ 703 <====
8924 030506 012742 000703                MOV   #703,-(R2) ;MOVE TO MAILBOX # ***** 703 *****
8925 030512 005242                INC   -(R2)      ;SET MSGTYP TO FATAL ERROR
8926 030514 000000                HALT                ;.EVEN TO ODD FAILED,OR WRONG $TSTNM
8927
8928                                ; OR SEQUENCE ERROR
8929
8930
8931 ;*****
8931 ;TEST 303 TEST THE CC BITS
8932 ;*****
8933 TS303: INC (R2) ;UPDATE TEST NUMBER
8934 030516 005212                CMP   #303,(R2) ;SEQUENCE ERROR?
8935 030520 022712 000303                BNE  TS304-10 ;BR TO ERROR HALT ON SEQ ERROR
8936 030524 001053                SCC
8937 030526 000277                ;SET STATUS
8938 030530 005067 147242                CLR  STATUS ;CLEAR STATUS
8939 030534 103004                BCC  BR33
8940
8941                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8942                                ;          CONDITIONAL BRANCH INST. AND <====
8943                                ;          REPLACE THE MOVE INSTRUCTION <====
8944                                ;          WHICH FOLLOWS W/ 773 <====
8943 030536 012742 000704                MOV   #704,-(R2) ;MOVE TO MAILBOX # ***** 704 *****
8944 030542 005242                INC   -(R2)      ;SET MSGTYP TO FATAL ERROR
8945 030544 000000                HALT                ;C NOT CLEAR
8946 030546                BR33:  BVC   BR34
8947 030546 102004
8948
8949                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8950                                ;          CONDITIONAL BRANCH INST. AND <====
8951                                ;          REPLACE THE MOVE INSTRUCTION <====
8952                                ;          WHICH FOLLOWS W/ 766 <====
8952 030550 012742 000705                MOV   #705,-(R2) ;MOVE TO MAILBOX # ***** 705 *****
8953 030554 005242                INC   -(R2)      ;SET MSGTYP TO FATAL ERROR
8954 030556 000000                HALT                ;V NOT CLEAR
8955 030560                BR34:  BNE   BR35
8956 030560 001004
8957
8958                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8959                                ;          CONDITIONAL BRANCH INST. AND <====
8960                                ;          REPLACE THE MOVE INSTRUCTION <====
8961                                ;          WHICH FOLLOWS W/ 761 <====
8961 030562 012742 000706                MOV   #706,-(R2) ;MOVE TO MAILBOX # ***** 706 *****
8962 030566 005242                INC   -(R2)      ;SET MSGTYP TO FATAL ERROR
```

```
8963 030570 000000
8964 030572
8965 030572 100004
8966
8967
8968
8969
8970 030574 012742 000707
8971 030600 005242
8972 030602 000000
8973 030604 000257
8974 030606 052767 000017 147162
8975
8976 030614 103404
8977
8978
8979
8980
8981 030616 012742 000710
8982 030622 005242
8983 030624 000000
8984 030626
8985 030626 102404
8986
8987
8988
8989
8990 030630 012742 000711
8991 030634 005242
8992 030636 000000
8993 030640
8994 030640 001404
8995
8996
8997
8998
8999 030642 012742 000712
9000 030646 005242
9001 030650 000000
9002 030652
9003 030652 100404
9004
9005
9006
9007
9008 030654 012742 000713
9009 030660 005242
9010 030662 000000
9011
9012
9013
9014
9015 030664 005212
9016 030666 022712 000304
9017 030672 001006
9018 030674 012706 001000
```

BR35: HALT ;Z NOT CLEAR  
BPL BR36  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 754 <====  
MOV #707,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 707 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;N NOT CLEAR  
BR36: CCC ;CLEAR CONDITION CODES  
BIS #17,STATUS ;SET STATUS TO ONES  
BCS BR37  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 743 <====  
MOV #710,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 710 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;C NOT SET  
BR37: BVS BR40  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 736 <====  
MOV #711,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 711 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;V NOT SET  
BR40: BEQ BR41  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 731 <====  
MOV #712,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 712 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;Z NOT SET  
BR41: BMI TS304  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 724 <====  
MOV #713,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 713 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;N NOT SET,OR WRONG \$STNM  
: OR SEQUENCE ERROR

\*\*\*\*\*  
:TEST 304 TEST THAT A TRAP OCCURS ON A RESERVED INSTRUCTION  
\*\*\*\*\*

TS304: INC (R2) ;UPDATE TEST NUMBER  
CMP #304,(R2) ;SEQUENCE ERROR?  
BNE RETA ;BR TO ERROR HALT ON SEQ ERROR  
MOV #BUFF,SP ;STACK POINTER SETUP



9019 030700 012767 030720 147102  
9020 030706 000077  
9021 030710  
9022 030710 012742 000714  
9023 030714 005242  
9024 030716 000000  
9025 030720  
9026  
9027  
9028  
9029 030720 005212  
9030 030722 022712 000305  
9031 030726 001011  
9032 030730 012706 001000  
9033 030734 012767 030744 147046  
9034 030742 000077  
9035 030744 020627 000774  
9036 030750 001404  
9037  
9038  
9039  
9040  
9041 030752 012742 000715  
9042 030756 005242  
9043 030760 000000  
9044  
9045  
9046  
9047  
9048 030762 005212  
9049 030764 022712 000306  
9050 030770 001012  
9051 030772 012706 001000  
9052 030776 012767 031006 147004  
9053 031004 000077  
9054 031006 022767 031006 147760  
9055 031014 001404  
9056  
9057  
9058  
9059  
9060 031016 012742 000716  
9061 031022 005242  
9062 031024 000000  
9063  
9064  
9065  
9066  
9067 031026 005212  
9068 031030 022712 000307  
9069 031034 001037  
9070 031036 012706 001000  
9071 031042 012767 031060 146740  
9072 031050 005067 146722  
9073 031054 000257  
9074 031056 000077

MOV #RETAH,RTRAP ;RETURN LOCATION  
TRAPA ;RESERVED INSTRUCTION, SHOULD TRAP  
RETA: MOV #714,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 714 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESERVE INSTRUCTION DIDN'T TRAP,OR WRONG \$STNM  
RETAH:  
:\*\*\*\*\*  
:TEST 305 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION  
:\*\*\*\*\*  
TS305: INC (R2) ;UPDATE TEST NUMBER  
CMP #305,(R2) ;SEQUENCE ERROR?  
BNE TS306-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #BUFF,SP ;STACK POINTER SETUP  
MOV #RETB,RTRAP ;RETURN POINTER  
TRAPA ;RESERVED INSTRUCTION  
RETB: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP  
BEQ TS306  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 766 <====  
MOV #715,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 715 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;NOT DECREMENTED TWO WORDS,OR WRONG \$STNM  
: OR SEQUENCE ERROR  
:\*\*\*\*\*  
:TEST 306 TEST THAT PROPER P.C. IS SAVED  
:\*\*\*\*\*  
TS306: INC (R2) ;UPDATE TEST NUMBER  
CMP #306,(R2) ;SEQUENCE ERROR?  
BNE TS307-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #BUFF,SP ;STACK POINTER SETUP  
MOV #RETC,RTRAP ;RETURN FROM TRAP POINTER  
INSTC: TRAPA ;TRAP ON THIS INSTRUCTION  
RETC: CMP #,BUFF-4 ;CHECK FOR INCREMENTED P.C.  
BEQ TS307  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 765 <====  
MOV #716,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 716 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;INCORRECT P.C.,OR WRONG \$STNM  
: OR SEQUENCE ERROR  
:\*\*\*\*\*  
:TEST 307 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK  
:\*\*\*\*\*  
TS307: INC (R2) ;UPDATE TEST NUMBER  
CMP #307,(R2) ;SEQUENCE ERROR?  
BNE TS310-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #BUFF,SP ;SET UP  
MOV #RETD,RTRAP ;SET UP  
CLR CC ;CLEAR CC AND PRIORITY  
CCC  
TRAPA ;TRAP

```

9075 031060 026727 147712 000000 RETD:  CMP      BUFF-2,#0      ;TEST THAT OLD STATUS WENT TO STACK
9076 031066 001404                BEQ      1$
9077                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9078                                ;          CONDITIONAL BRANCH INST. AND <====
9079                                ;          REPLACE THE MOVE INSTRUCTION <====
9080                                ;          WHICH FOLLOWS W/ 762 <====
9081 031070 012742 000717                MOV      #717,-(R2) ;MOVE TO MAILBOX # ***** 717 *****
9082 031074 005242                INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
9083 031076 000000                HALT                    ;INCORRECT STATUS
9084 031100 012706 001000                1$:  MOV      #BUFF,SP   ;SET UP
9085 031104 012767 031124 146676        MOV      #RETE,RTRAP ;SET UP
9086 031112 012767 000357 146656        MOV      #357,CC     ;SET PRIORITY
9087 031120 000277                SCC                    ;SET CC
9088 031122 000077                TRAPA                   ;TRAP
9089 031124 026727 147646 000357 RETE:  CMP      BUFF-2,#357 ;COMPARES STATUS ON STACK
9090 031132 001404                BEQ      TS310
9091                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9092                                ;          CONDITIONAL BRANCH INST. AND <====
9093                                ;          REPLACE THE MOVE INSTRUCTION <====
9094                                ;          WHICH FOLLOWS W/ 740 <====
9095 031134 012742 000720                MOV      #720,-(R2) ;MOVE TO MAILBOX # ***** 720 *****
9096 031140 005242                INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
9097 031142 000000                HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
9098                                ; OR SEQUENCE ERROR
9099                                ;*****
9100                                ;TEST 310 TEST THAT 'NEW' STATUS IS CORRECT
9101                                ;*****
9102 031144 005212                TS310: INC      (R2)      ;UPDATE TEST NUMBER
9103 031146 022712 000310                CMP      #310,(R2)   ;SEQUENCE ERROR?
9104 031152 001110                BNE     STPP          ;BR TO ERROR HALT ON SEQ ERROR
9105 031154 012706 001000                MOV      #BUFF,SP   ;CLEAR FUTURE PRIORITY AND CC
9106 031160 012767 031174 146622        MOV      #RETF,RTRAP
9107 031166 005067 146620                CLR     RTRAP+2
9108 031172 000077                TRAPA                   ;TEST FOR 'C' CLEARED
9109 031174                RETF:
9110 031174 100004                BPL     1$
9111                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9112                                ;          CONDITIONAL BRANCH INST. AND <====
9113                                ;          REPLACE THE MOVE INSTRUCTION <====
9114                                ;          WHICH FOLLOWS W/ 766 <====
9115 031176 012742 000721                MOV      #721,-(R2) ;MOVE TO MAILBOX # ***** 721 *****
9116 031202 005242                INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
9117 031204 000000                HALT                    ;N NOT CLEARED
9118 031206                1$:
9119 031206 001004                BNE     2$
9120                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9121                                ;          CONDITIONAL BRANCH INST. AND <====
9122                                ;          REPLACE THE MOVE INSTRUCTION <====
9123                                ;          WHICH FOLLOWS W/ 761 <====
9124 031210 012742 000722                MOV      #722,-(R2) ;MOVE TO MAILBOX # ***** 722 *****
9125 031214 005242                INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
9126 031216 000000                HALT                    ;Z NOT CLEARED
9127 031220                2$:
9128 031220 102004                BVC     3$
9129                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9130                                ;          CONDITIONAL BRANCH INST. AND <====
    
```



```
9187 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9188 ; CONDITIONAL BRANCH INST. AND <====
9189 ; REPLACE THE MOVE INSTRUCTION <====
9190 ; WHICH FOLLOWS W/ 702 <====
9191 031346 012742 000731 MOV #731,-(R2) ;MOVE TO MAILBOX # ***** 731 *****
9192 031352 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9193 031354 000000 HALT ;C NOT SET
9194 031356 016706 146414 4$: MOV CC,SP
9195 031362 042706 000017 BIC #17,SP
9196 031366 022706 000340 CMP #340,SP
9197 031372 001404 BEQ STPPA
9198 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9199 ; CONDITIONAL BRANCH INST. AND <====
9200 ; REPLACE THE MOVE INSTRUCTION <====
9201 ; WHICH FOLLOWS W/ 667 <====
9202 031374 STPP:
9203 031374 012742 000732 MOV #732,-(R2) ;MOVE TO MAILBOX # ***** 732 *****
9204 031400 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9205 031402 000000 HALT ;PRIORITY WAS CHANGED,OR WRONG $STNM
9206 031404 012767 000012 146376 STPPA: MOV #12,10
9207 031412 005067 146374 CLR 12
9208 ;*****
9209 ;TEST 311 TEST THAT A TRAP OCCURS FOR A 'TRAP' INSTRUCTION
9210 ;*****
9211 031416 005212 TS311: INC (R2) ;UPDATE TEST NUMBER
9212 031420 022712 000311 CMP #311,(R2) ;SEQUENCE ERROR?
9213 031424 001013 BNE TS312-10 ;BR TO ERROR HALT ON SEQ ERROR
9214 031426 012767 000012 146354 MOV #12,10
9215 031434 005067 146352 CLR 12
9216 031440 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9217 031444 012767 031464 146362 MOV #RETA1,RTRAP1 ;RETURN LOCATION
9218 031452 104400 TRAP ;RESERVED INSTRUCTION, SHOULD TRAP
9219 031454 012742 000733 MOV #733,-(R2) ;MOVE TO MAILBOX # ***** 733 *****
9220 031460 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9221 031462 000000 HALT ;TRAP DIDN'T TRAP,OR WRONG $STNM
9222 031464 RETA1:
9223 ;*****
9224 ;TEST 312 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9225 ;*****
9226 031464 005212 TS312: INC (R2) ;UPDATE TEST NUMBER
9227 031466 022712 000312 CMP #312,(R2) ;SEQUENCE ERROR?
9228 031472 001011 BNE TS313-10 ;BR TO ERROR HALT ON SEQ ERROR
9229 031474 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9230 031500 012767 031510 146326 MOV #RETB1,RTRAP1 ;RETURN POINTER
9231 031506 104400 TRAP ;RESERVED INSTRUCTION
9232 031510 020627 000774 RETB1: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
9233 031514 001404 BEQ TS313
9234 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9235 ; CONDITIONAL BRANCH INST. AND <====
9236 ; REPLACE THE MOVE INSTRUCTION <====
9237 ; WHICH FOLLOWS W/ 766 <====
9238 031516 012742 000734 MOV #734,-(R2) ;MOVE TO MAILBOX # ***** 734 *****
9239 031522 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9240 031524 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $STNM
9241 ; OR SEQUENCE ERROR
9242 ;*****
```

```
9243 :TEST 313 TEST THAT PROPER P.C. IS SAVED
9244 :*****
9245 031526 005212 TS313: INC (R2) :UPDATE TEST NUMBER
9246 031530 022712 000313 CMP #313,(R2) :SEQUENCE ERROR?
9247 031534 001012 BNE TS314-10 :BR TO ERROR HALT ON SEQ ERROR
9248 031536 012706 001000 MOV #BUFF,SP :STACK POINTER SETUP
9249 031542 012767 031552 146264 MOV #RETC1,RTRAP1 :RETURN FROM TRAP POINTER
9250 031550 104400 TRAP :TRAP ON THIS INSTRUCTION
9251 031552 022767 031552 147214 RETC1: CMP #,BUFF-4 :CHECK INCREMENTED P.C.
9252 031560 001404 BEQ TS314
9253 :
9254 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9255 : CONDITIONAL BRANCH INST. AND <====
9256 : REPLACE THE MOVE INSTRUCTION <====
9257 : WHICH FOLLOWS W/ 765 <====
9257 031562 012742 000735 MOV #735,-(R2) :MOVE TO MAILBOX # ***** 735 *****
9258 031566 005242 INC -(R2) :SET MSGTYP TO FATAL ERROR
9259 031570 000000 HALT :INCORRECT P.C.,OR WRONG $STNM
9260 : OR SEQUENCE ERROR
9261 :*****
9262 :TEST 314 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
9263 :*****
9264 031572 005212 TS314: INC (R2) :UPDATE TEST NUMBER
9265 031574 022712 000314 CMP #314,(R2) :SEQUENCE ERROR?
9266 031600 001036 BNE TS315-10 :BR TO ERROR HALT ON SEQ ERROR
9267 031602 012706 001000 MOV #BUFF,SP :SET UP
9268 031606 012767 031624 146220 MOV #RETD1,RTRAP1 :SET UP
9269 031614 005067 146156 CLR CC :CLEAR CC AND PRIORITY
9270 031620 000257 CCC
9271 031622 104400 TRAP :TRAP
9272 031624 026727 147146 000000 RETD1: CMP BUFF-2,#0 :TEST THAT OLD STATUS WENT TO STACK
9273 031632 001404 BEQ 1$
9274 :
9275 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9276 : CONDITIONAL BRANCH INST. AND <====
9277 : REPLACE THE MOVE INSTRUCTION <====
9278 : WHICH FOLLOWS W/ 762 <====
9278 031634 012742 000736 MOV #736,-(R2) :MOVE TO MAILBOX # ***** 736 *****
9279 031640 005242 INC -(R2) :SET MSGTYP TO FATAL ERROR
9280 031642 000000 HALT :INCORRECT STATUS
9281 031644 012706 001000 1$: MOV #BUFF,SP :SET UP
9282 031650 012767 031666 146156 MOV #RETE1,RTRAP1 :SET UP
9283 031656 012767 000357 146112 MOV #357,CC :SET PRIORITY
9284 031664 104400 TRAP :SET CC
9285 031666 026727 147104 000357 RETE1: CMP BUFF-2,#357 :COMPARES STATUS ON STACK
9286 031674 001404 BEQ TS315
9287 :
9288 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9289 : CONDITIONAL BRANCH INST. AND <====
9290 : REPLACE THE MOVE INSTRUCTION <====
9291 : WHICH FOLLOWS W/ 741 <====
9291 031676 012742 000737 MOV #737,-(R2) :MOVE TO MAILBOX # ***** 737 *****
9292 031702 005242 INC -(R2) :SET MSGTYP TO FATAL ERROR
9293 031704 000000 HALT :INCORRECT STATUS ON STACK,OR WRONG $STNM
9294 : OR SEQUENCE ERROR
9295 :*****
9296 :TEST 315 TEST THAT 'NEW' STATUS IS CORRECT
9297 :*****
9298 031706 005212 TS315: INC (R2) :UPDATE TEST NUMBER
```

|      |        |        |        |        |        |  |  |  |      |               |  |       |
|------|--------|--------|--------|--------|--------|--|--|--|------|---------------|--|-------|
| 9299 | 031710 | 022712 | 000315 |        |        |  |  |  | CMP  | #315,(R2)     | ;SEQUENCE ERROR?                         |       |
| 9300 | 031714 | 001110 |        |        |        |  |  |  | BNE  | TS316-10      | ;BR TO ERROR HALT ON SEQ ERROR           |       |
| 9301 | 031716 | 012706 | 001000 |        |        |  |  |  | MOV  | #BUFF,SP      |  |       |
| 9302 | 031722 | 012767 | 031736 | 146104 |        |  |  |  | MOV  | #RETF1,RTRAP1 |  |       |
| 9303 | 031730 | 005067 | 146102 |        |        |  |  |  | CLR  | RTRAP1+2      | ;CLEAR FUTURE PRIORITY AND CC            |       |
| 9304 | 031734 | 104400 |        |        |        |  |  |  | TRAP |               |  |       |
| 9305 | 031736 |        |        |        | RETF1: |  |  |  |      |               | ;TEST FOR 'C' CLEARED                    |       |
| 9306 | 031736 | 100004 |        |        |        |  |  |  | BPL  | 1\$           |  |       |
| 9307 |        |        |        |        |        |  |  |  |      |               | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9308 |        |        |        |        |        |  |  |  |      |               | CONDITIONAL BRANCH INST. AND             | <==== |
| 9309 |        |        |        |        |        |  |  |  |      |               | REPLACE THE MOVE INSTRUCTION             | <==== |
| 9310 |        |        |        |        |        |  |  |  |      |               | WHICH FOLLOWS W/ 766                     | <==== |
| 9311 | 031740 | 012742 | 000740 |        |        |  |  |  | MOV  | #740,-(R2)    | ;MOVE TO MAILBOX # ***** 740 *****       |       |
| 9312 | 031744 | 005242 |        |        |        |  |  |  | INC  | -(R2)         | ;SET MSGTYP TO FATAL ERROR               |       |
| 9313 | 031746 | 000000 |        |        |        |  |  |  | HALT |               | ;C NOT CLEARED                           |       |
| 9314 | 031750 |        |        |        | 1\$:   |  |  |  |      |               |  |       |
| 9315 | 031750 | 001004 |        |        |        |  |  |  | BNE  | 2\$           |  |       |
| 9316 |        |        |        |        |        |  |  |  |      |               | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9317 |        |        |        |        |        |  |  |  |      |               | CONDITIONAL BRANCH INST. AND             | <==== |
| 9318 |        |        |        |        |        |  |  |  |      |               | REPLACE THE MOVE INSTRUCTION             | <==== |
| 9319 |        |        |        |        |        |  |  |  |      |               | WHICH FOLLOWS W/ 761                     | <==== |
| 9320 | 031752 | 012742 | 000741 |        |        |  |  |  | MOV  | #741,-(R2)    | ;MOVE TO MAILBOX # ***** 741 *****       |       |
| 9321 | 031756 | 005242 |        |        |        |  |  |  | INC  | -(R2)         | ;SET MSGTYP TO FATAL ERROR               |       |
| 9322 | 031760 | 000000 |        |        |        |  |  |  | HALT |               | ;Z NOT CLEARED                           |       |
| 9323 | 031762 |        |        |        | 2\$:   |  |  |  |      |               |  |       |
| 9324 | 031762 | 102004 |        |        |        |  |  |  | BVC  | 3\$           |  |       |
| 9325 |        |        |        |        |        |  |  |  |      |               | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9326 |        |        |        |        |        |  |  |  |      |               | CONDITIONAL BRANCH INST. AND             | <==== |
| 9327 |        |        |        |        |        |  |  |  |      |               | REPLACE THE MOVE INSTRUCTION             | <==== |
| 9328 |        |        |        |        |        |  |  |  |      |               | WHICH FOLLOWS W/ 754                     | <==== |
| 9329 | 031764 | 012742 | 000742 |        |        |  |  |  | MOV  | #742,-(R2)    | ;MOVE TO MAILBOX # ***** 742 *****       |       |
| 9330 | 031770 | 005242 |        |        |        |  |  |  | INC  | -(R2)         | ;SET MSGTYP TO FATAL ERROR               |       |
| 9331 | 031772 | 000000 |        |        |        |  |  |  | HALT |               | ;V NOT CLEARED                           |       |
| 9332 | 031774 |        |        |        | 3\$:   |  |  |  |      |               |  |       |
| 9333 | 031774 | 103004 |        |        |        |  |  |  | BCC  | 4\$           |  |       |
| 9334 |        |        |        |        |        |  |  |  |      |               | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9335 |        |        |        |        |        |  |  |  |      |               | CONDITIONAL BRANCH INST. AND             | <==== |
| 9336 |        |        |        |        |        |  |  |  |      |               | REPLACE THE MOVE INSTRUCTION             | <==== |
| 9337 |        |        |        |        |        |  |  |  |      |               | WHICH FOLLOWS W/ 747                     | <==== |
| 9338 | 031776 | 012742 | 000743 |        |        |  |  |  | MOV  | #743,-(R2)    | ;MOVE TO MAILBOX # ***** 743 *****       |       |
| 9339 | 032002 | 005242 |        |        |        |  |  |  | INC  | -(R2)         | ;SET MSGTYP TO FATAL ERROR               |       |
| 9340 | 032004 | 000000 |        |        |        |  |  |  | HALT |               | ;C NOT CLEARED                           |       |
| 9341 | 032006 | 032767 | 000340 | 145762 | 4\$:   |  |  |  | BIT  | #340,CC       | ;TEST PRIORITY                           |       |
| 9342 | 032014 | 001404 |        |        |        |  |  |  | BEQ  | 5\$           |  |       |
| 9343 |        |        |        |        |        |  |  |  |      |               | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9344 |        |        |        |        |        |  |  |  |      |               | CONDITIONAL BRANCH INST. AND             | <==== |
| 9345 |        |        |        |        |        |  |  |  |      |               | REPLACE THE MOVE INSTRUCTION             | <==== |
| 9346 |        |        |        |        |        |  |  |  |      |               | WHICH FOLLOWS W/ 737                     | <==== |
| 9347 | 032016 | 012742 | 000744 |        |        |  |  |  | MOV  | #744,-(R2)    | ;MOVE TO MAILBOX # ***** 744 *****       |       |
| 9348 | 032022 | 005242 |        |        |        |  |  |  | INC  | -(R2)         | ;SET MSGTYP TO FATAL ERROR               |       |
| 9349 | 032024 | 000000 |        |        |        |  |  |  | HALT |               | ;PRIORITY NOT ZERO                       |       |
| 9350 | 032026 | 012706 | 001000 |        | 5\$:   |  |  |  | MOV  | #BUFF,SP      |  |       |
| 9351 | 032032 | 012767 | 032050 | 145774 |        |  |  |  | MOV  | #RETF1,RTRAP1 |  |       |
| 9352 | 032040 | 012767 | 000357 | 145770 |        |  |  |  | MOV  | #357,RTRAP1+2 | ;SET NEW 'CC' AND PRIORITY               |       |
| 9353 | 032046 | 104400 |        |        |        |  |  |  | TRAP |               | ;TRAP HERE                               |       |
| 9354 | 032050 |        |        |        | RETG1: |  |  |  |      |               |  |       |

```
9355 032050 100404      BMI      1$
9356
9357
9358
9359
9360 032052 012742 000745      MOV      #745,-(R2)
9361 032056 005242      INC      -(R2)
9362 032060 000000      HALT
9363 032062
9364 032062 001404      1$:     BEQ      2$
9365
9366
9367
9368
9369 032064 012742 000746      MOV      #746,-(R2)
9370 032070 005242      INC      -(R2)
9371 032072 000000      HALT
9372 032074
9373 032074 102404      2$:     BVS      3$
9374
9375
9376
9377
9378 032076 012742 000747      MOV      #747,-(R2)
9379 032102 005242      INC      -(R2)
9380 032104 000000      HALT
9381 032106
9382 032106 103404      3$:     BCS      4$
9383
9384
9385
9386
9387 032110 012742 000750      MOV      #750,-(R2)
9388 032114 005242      INC      -(R2)
9389 032116 000000      HALT
9390 032120 016706 145652      4$:     MOV      CC,SP
9391 032124 042706 000017      BIC      #17,SP
9392 032130 022706 000340      CMP      #340,SP
9393 032134 001404      BEQ      TS316
9394
9395
9396
9397
9398 032136 012742 000751      MOV      #751,-(R2)
9399 032142 005242      INC      -(R2)
9400 032144 000000      HALT
9401
9402
9403
9404
9405 032146 005212
9406 032150 022712 000316      TS316:  INC      (R2)
9407 032154 001011      CMP      #316,(R2)
9408
9409 032156 012767 104776 000012      BNE      BR45
9410 032164 012767 032210 145642      MOV      #TRAP+376,RB1
      MOV      #RA1,34
```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
; CONDITIONAL BRANCH INST. AND  
; REPLACE THE MOVE INSTRUCTION  
; WHICH FOLLOWS W/ 721  
; MOVE TO MAILBOX # \*\*\*\*\* 745 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; N NOT SET

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
; CONDITIONAL BRANCH INST. AND  
; REPLACE THE MOVE INSTRUCTION  
; WHICH FOLLOWS W/ 714  
; MOVE TO MAILBOX # \*\*\*\*\* 746 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; Z NOT SET

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
; CONDITIONAL BRANCH INST. AND  
; REPLACE THE MOVE INSTRUCTION  
; WHICH FOLLOWS W/ 707  
; MOVE TO MAILBOX # \*\*\*\*\* 747 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; V NOT SET

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
; CONDITIONAL BRANCH INST. AND  
; REPLACE THE MOVE INSTRUCTION  
; WHICH FOLLOWS W/ 702  
; MOVE TO MAILBOX # \*\*\*\*\* 750 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; C NOT SET

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  
; CONDITIONAL BRANCH INST. AND  
; REPLACE THE MOVE INSTRUCTION  
; WHICH FOLLOWS W/ 667  
; MOVE TO MAILBOX # \*\*\*\*\* 751 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; PRIORITY WAS CHANGED,OR WRONG \$STNM  
; OR SEQUENCE ERROR

\*\*\*\*\*  
;TEST 316 TEST THAT ALL COMBINATION OF 'TRAP' WILL CAUSE A TRAP  
\*\*\*\*\*

;UPDATE TEST NUMBER  
;SEQUENCE ERROR?  
;BR TO ERROR HALT ON SEQ ERROR  
;\*\*\*\*\* F11 \*\*\*\* ADD +376 TO SHORTEN TEST  
;INITIALIZE BASE TRAP INSTRUCTION  
;RETURN FROM TRAP TO RA1

9411 032172 012706 001000  
9412 032176 104400  
9413 032200  
9414 032200 012742 000752  
9415 032204 005242  
9416 032206 000000  
9417 032210 005267 177762  
9418 032214 022767 104777 177754  
9419 032222 103363  
9420 032224 012767 000036 145602  
9421 032232 005067 145600  
9422  
9423  
9424  
9425 032236 005212  
9426 032240 022712 000317  
9427 032244 001006  
9428 032246 012706 001000  
9429 032252 012767 032272 145540  
9430 032260 000004  
9431 032262 012742 000753  
9432 032266 005242  
9433 032270 000000  
9434 032272  
9435  
9436  
9437  
9438 032272 005212  
9439 032274 022712 000320  
9440 032300 001011  
9441 032302 012706 001000  
9442 032306 012767 032316 145504  
9443 032314 000004  
9444 032316 020627 000774  
9445 032322 001404  
9446  
9447  
9448  
9449  
9450 032324 012742 000754  
9451 032330 005242  
9452 032332 000000  
9453  
9454  
9455  
9456  
9457 032334 005212  
9458 032336 022712 000321  
9459 032342 001012  
9460 032344 012706 001000  
9461 032350 012767 032360 145442  
9462 032356 000004  
9463 032360 022767 032360 146406  
9464 032366 001404  
9465  
9466

RC1: MOV #BUFF,SP ;SET UP STACK POINTER  
RB1: TRAP ;TRAP INST WILL BE MODIFIED TO TRAP+377  
BR45:  
MOV #752,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 752 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;PREVIOUS INST FAILED TO TRAP,OR WRONG \$STNM  
RA1: INC RB1 ;INCREMENT TRAP INSTRUCTION  
CMP #104777,RB1 ;TRAP+377 TO UPPER LIMIT  
BHIS RC1 ;HAVE WE TESTED ALL  
MOV #36,34  
CLR 36

\*\*\*\*\*  
;TEST 317 TEST THAT A TRAP OCCURES ON AN 'IOT' INSTRUCTION  
\*\*\*\*\*

TS317: INC (R2) ;UPDATE TEST NUMBER  
CMP #317,(R2) ;SEQUENCE ERROR?  
BNE TS320-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #BUFF,SP ;STACK POINTER SETUP  
MOV #RETA2,RTRAP2 ;RETURN LOCATION  
IOT ;RESERVE INSTRUCTION, SHOULD TRAP  
MOV #753,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 753 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;IOT DIDN'T TRAP,OR WRONG \$STNM

RETA2:  
\*\*\*\*\*  
;TEST 320 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION  
\*\*\*\*\*

TS320: INC (R2) ;UPDATE TEST NUMBER  
CMP #320,(R2) ;SEQUENCE ERROR?  
BNE TS321-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #BUFF,SP ;STACK POINTER SETUP  
MOV #RETB2,RTRAP2 ;RETURN POINTER  
IOT ;RESERVED INSTRUCTION  
RETB2: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP  
BEQ TS321

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
; CONDITIONAL BRANCH INST. AND <=====  
; REPLACE THE MOVE INSTRUCTION <=====  
; WHICH FOLLOWS W/ 766 <=====  
;

MOV #754,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 754 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;NOT DECREMENTED TWO WORDS,OR WRONG \$STNM  
; OR SEQUENCE ERROR

\*\*\*\*\*  
;TEST 321 TEST THAT PROPER P.C. IS SAVED  
\*\*\*\*\*

TS321: INC (R2) ;UPDATE TEST NUMBER  
CMP #321,(R2) ;SEQUENCE ERROR?  
BNE TS322-10 ;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 TS322

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
; CONDITIONAL BRANCH INST. AND <=====  
;



```
9467                                     :                                     <====
9468                                     :                                     <====
9469 032370 012742 000755                MOV    #755,-(R2)                : MOVE TO MAILBOX # ***** 755 *****
9470 032374 005242                        INC    -(R2)                    : SET MSGTYP TO FATAL ERROR
9471 032376 000000                        HALT                          : INCORRECT P.C.,OR WRONG $STNM
9472                                     : OR SEQUENCE ERROR
9473                                     :
9474                                     : *****
9475 :TEST 322 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
9476 :*****
9476 032400 005212                          TS322: INC    (R2)                : UPDATE TEST NUMBER
9477 032402 022712 000322                  CMP    #322,(R2)              : SEQUENCE ERROR?
9478 032406 001037                          BNE    TS323-10               : BR TO ERROR HALT ON SEQ ERROR
9479 032410 012706 001000                  MOV    #BUFF,SP              : SET UP
9480 032414 012767 032432 145376          MOV    #RETD2,RTRAP2         : SET UP
9481 032422 005067 145350                  CLR    CC                    : CLEAR CC AND PRIORITY
9482 032426 000257                          CCC
9483 032430 000004                          IOT
9484 032432 026727 146340 000000 RETD2: CMP    BUFF-2,#0          : TRAP
9485 032440 001404                          BEQ    1$                    : TEST THAT OLD STATUS WENT TO STACK
9486                                     :
9487                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9488                                     : CONDITIONAL BRANCH INST. AND <====
9489                                     : REPLACE THE MOVE INSTRUCTION <====
9490 032442 012742 000756                MOV    #756,-(R2)                : MOVE TO MAILBOX # ***** 756 *****
9491 032446 005242                        INC    -(R2)                    : SET MSGTYP TO FATAL ERROR
9492 032450 000000                        HALT                          : INCORRECT STATUS
9493 032452 012706 001000 1$: MOV    #BUFF,SP              : SET UP
9494 032456 012767 032476 145334          MOV    #RETE2,RTRAP2         : SET UP
9495 032464 012767 000357 145304          MOV    #357,CC               : SET PRIORITY
9496 032472 000277                          SCC
9497 032474 000004                          IOT
9498 032476 026727 146274 000357 RETE2: CMP    BUFF-2,#357        : COMPARES STATUS ON STACK
9499 032504 001404                          BEQ    TS323
9500                                     :
9501                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9502                                     : CONDITIONAL BRANCH INST. AND <====
9503                                     : REPLACE THE MOVE INSTRUCTION <====
9504 032506 012742 000757                MOV    #757,-(R2)                : MOVE TO MAILBOX # ***** 757 *****
9505 032512 005242                        INC    -(R2)                    : SET MSGTYP TO FATAL ERROR
9506 032514 000000                        HALT                          : INCORRECT STATUS ON STACK,OR WRONG $STNM
9507                                     : OR SEQUENCE ERROR
9508                                     :
9509 :TEST 323 TEST THAT 'NEW' STATUS IS CORRECT
9510 :*****
9511 032516 005212                          TS323: INC    (R2)                : UPDATE TEST NUMBER
9512 032520 022712 000323                  CMP    #323,(R2)              : SEQUENCE ERROR?
9513 032524 001110                          BNE    BR46                   : BR TO ERROR HALT ON SEQ ERROR
9514 032526 012706 001000                  MOV    #BUFF,SP              :
9515 032532 012767 032546 145260          MOV    #RETF2,RTRAP2         :
9516 032540 005067 145256                  CLR    RTRAP2+2              : CLEAR FUTURE PRIORITY AND CC
9517 032544 000004                          IOT
9518 032546                          RETF2:                          : TEST FOR 'C' CLEARED
9519 032546 100004                          BPL    1$
9520                                     :
9521                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9522                                     : CONDITIONAL BRANCH INST. AND <====
9523                                     : REPLACE THE MOVE INSTRUCTION <====
```

```
9523                                     : WHICH FOLLOWS W/ 766 <====
9524 032550 012742 000760             MOV #760,-(R2)      ;MOVE TO MAILBOX # ***** 760 *****
9525 032554 005242                     INC -(R2)          ;SET MSGTYP TO FATAL ERROR
9526 032556 000000                     HALT              ;N NOT CLEARED
9527 032560                               1$:
9528 032560 001004                     BNE 2$
9529                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9530                                     : CONDITIONAL BRANCH INST. AND <====
9531                                     : REPLACE THE MOVF INSTRUCTION <====
9532                                     : WHICH FOLLOWS W/ 761 <====
9533 032562 012742 000761             MOV #761,-(R2)      ;MOVE TO MAILBOX # ***** 761 *****
9534 032566 005242                     INC -(R2)          ;SET MSGTYP TO FATAL ERROR
9535 032570 000000                     HALT              ;Z NOT CLEARED
9536 032572                               2$:
9537 032572 102004                     BVC 3$
9538                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9539                                     : CONDITIONAL BRANCH INST. AND <====
9540                                     : REPLACE THE MOVE INSTRUCTION <====
9541                                     : WHICH FOLLOWS W/ 754 <====
9542 032574 012742 000762             MOV #762,-(R2)      ;MOVE TO MAILBOX # ***** 762 *****
9543 032600 005242                     INC -(R2)          ;SET MSGTYP TO FATAL ERROR
9544 032602 000000                     HALT              ;V NOT CLEARED
9545 032604                               3$:
9546 032604 103004                     BCC 4$
9547                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9548                                     : CONDITIONAL BRANCH INST. AND <====
9549                                     : REPLACE THE MOVE INSTRUCTION <====
9550                                     : WHICH FOLLOWS W/ 747 <====
9551 032606 012742 000763             MOV #763,-(R2)      ;MOVE TO MAILBOX # ***** 763 *****
9552 032612 005242                     INC -(R2)          ;SET MSGTYP TO FATAL ERROR
9553 032614 000000                     HALT              ;C NOT CLEARED
9554 032616 032767 000340 145152 4$:  BIT #340,CC
9555 032624 001404                     BEQ 5$            ;TEST PRIORITY
9556                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9557                                     : CONDITIONAL BRANCH INST. AND <====
9558                                     : REPLACE THE MOVE INSTRUCTION <====
9559                                     : WHICH FOLLOWS W/ 737 <====
9560 032626 012742 000764             MOV #764,-(R2)      ;MOVE TO MAILBOX # ***** 764 *****
9561 032632 005242                     INC -(R2)          ;SET MSGTYP TO FATAL ERROR
9562 032634 000000                     HALT              ;PRIORITY NOT ZERO
9563 032636 012706 001000             MOV #BUFF,SP
9564 032642 012767 032660 145150     MOV #RETG2,RTRAP2
9565 032650 012767 000357 145144     MOV #357,RTRAP2+2 ;SET NEW 'CC' AND PRIORITY
9566 032656 000004                     IOT              ;TRAP HERE
9567 032660                               RETG2:
9568 032660 100404                     BMI 1$
9569                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9570                                     : CONDITIONAL BRANCH INST. AND <====
9571                                     : REPLACE THE MOVE INSTRUCTION <====
9572                                     : WHICH FOLLOWS W/ 721 <====
9573 032662 012742 000765             MOV #765,-(R2)      ;MOVE TO MAILBOX # ***** 765 *****
9574 032666 005242                     INC -(R2)          ;SET MSGTYP TO FATAL ERROR
9575 032670 000000                     HALT              ;N NOT SET
9576 032672                               1$:
9577 032672 001404                     BEQ 2$
9578                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
```

```
9579          :          CONDITIONAL BRANCH INST. AND <====
9580          :          REPLACE THE MOVE INSTRUCTION <====
9581          :          WHICH FOLLOWS W/ 714 <====
9582 032674 012742 000766      MOV #766,-(R2) :MOVE TO MAILBOX # ***** 766 *****
9583 032700 005242      INC -(R2) :SET MSGTYP TO FATAL ERROR
9584 032702 000000      HALT :Z NOT SET
9585 032704          :
9586 032704 102404 2$:      BVS 3$
9587          :
9588          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9589          :          CONDITIONAL BRANCH INST. AND <====
9590          :          REPLACE THE MOVE INSTRUCTION <====
9591          :          WHICH FOLLOWS W/ 707 <====
9591 032706 012742 000767      MOV #767,-(R2) :MOVE TO MAILBOX # ***** 767 *****
9592 032712 005242      INC -(R2) :SET MSGTYP TO FATAL ERROR
9593 032714 000000      HALT :V NOT SET
9594 032716          :
9595 032716 103404 3$:      BCS 4$
9596          :
9597          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9598          :          CONDITIONAL BRANCH INST. AND <====
9599          :          REPLACE THE MOVE INSTRUCTION <====
9600          :          WHICH FOLLOWS W/ 702 <====
9600 032720 012742 000770      MOV #770,-(R2) :MOVE TO MAILBOX # ***** 770 *****
9601 032724 005242      INC -(R2) :SET MSGTYP TO FATAL ERROR
9602 032726 000000      HALT :C NOT SET
9603 032730 016706 145042 4$:      MOV CC,SP
9604 032734 042706 000017      BIC #17,SP
9605 032740 022706 000340      CMP #340,SP
9606 032744 001404      BEQ BR46A
9607          :
9608          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9609          :          CONDITIONAL BRANCH INST. AND <====
9610          :          REPLACE THE MOVE INSTRUCTION <====
9611          :          WHICH FOLLOWS W/ 667 <====
9611 032746          BR46:
9612 032746 012742 000771      MOV #771,-(R2) :MOVE TO MAILBOX # ***** 771 *****
9613 032752 005242      INC -(R2) :SET MSGTYP TO FATAL ERROR
9614 032754 000000      HALT :PRIORITY WAS CHANGED,OR WRONG $STNM
9615 032756 012767 000022 145034 BR46A: MOV #22,20 :.+2
9616 032764 005067 145032      CLR 22 :HALT
9617          :
9618          :*****
9618          :TEST 324 TEST THAT A TRAP OCCURS ON AN EMT INSTRUCTION
9619          :*****
9620 032770 005212          TS324: INC (R2) :UPDATE TEST NUMBER
9621 032772 022712 000324      CMP #324,(R2) :SEQUENCE ERROR?
9622 032776 001006      BNE TS325-10 :BR TO ERROR HALT ON SEQ ERROR
9623 033000 012706 001000      MOV #BUFF,SP :STACK POINTER SETUP
9624 033004 012767 033024 145016      MOV #RETA3,RTRAP3 :RETURN LOCATION
9625 033012 104000      EMT :RESERVE INSTRUCTION, SHOULD TRAP
9626 033014 012742 000772      MOV #772,-(R2) :MOVE TO MAILBOX # ***** 772 *****
9627 033020 005242      INC -(R2) :SET MSGTYP TO FATAL ERROR
9628 033022 000000      HALT :EMT DIDN'T TRAP,OR WRONG $STNM
9629 033024          RETA3:
9630          :*****
9631          :TEST 325 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9632          :*****
9633 033024 005212          TS325: INC (R2) :UPDATE TEST NUMBER
9634 033026 022712 000325      CMP #325,(R2) :SEQUENCE ERROR?
```

|      |        |        |        |        |            |               |   |
|------|--------|--------|--------|--------|------------|---------------|---|
| 9635 | 033032 | 001011 |        |        | BNE        | TS326-10      | :BR TO ERROR HALT ON SEQ ERROR                                |
| 9636 | 033034 | 012706 | 001000 |        | MOV        | #BUFF,SP      | :STACK POINTER SETUP  |
| 9637 | 033040 | 012767 | 033050 | 144762 | MOV        | #RETB3,RTRAP3 | :RETURN POINTER   |
| 9638 | 033046 | 104000 |        |        | EMT        |               | :RESERVED INSTRUCTION   |
| 9639 | 033050 | 020627 | 000774 |        | RETB3: CMP | SP,#BUFF-4    | :TEST DECREMENT OF SP   |
| 9640 | 033054 | 001404 |        |        | BEQ        | TS326         |   |
| 9641 |        |        |        |        |            |               | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====                |
| 9642 |        |        |        |        |            |               | : CONDITIONAL BRANCH INST. AND <====                          |
| 9643 |        |        |        |        |            |               | : REPLACE THE MOVE INSTRUCTION <====                          |
| 9644 |        |        |        |        |            |               | : WHICH FOLLOWS W/ 766 <====                                  |
| 9645 | 033056 | 012742 | 000773 |        | MOV        | #773,-(R2)    | :MOVE TO MAILBOX # ***** 773 *****                            |
| 9646 | 033062 | 005242 |        |        | INC        | -(R2)         | :SET MSGTYP TO FATAL ERROR                                    |
| 9647 | 033064 | 000000 |        |        | HALT       |               | :NOT DECREMENTED TWO WORDS,OR WRONG STSTNM                    |
| 9648 |        |        |        |        |            |               | : OR SEQUENCE ERROR   |
| 9649 |        |        |        |        |            |               | :*****  |
| 9650 |        |        |        |        |            |               | :TEST 326 TEST THAT PROPER P.C. IS SAVED                      |
| 9651 |        |        |        |        |            |               | :*****  |
| 9652 | 033066 | 005212 |        |        | TS326: INC | (R2)          | :UPDATE TEST NUMBER   |
| 9653 | 033070 | 022712 | 000326 |        | CMP        | #326,(R2)     | :SEQUENCE ERROR?  |
| 9654 | 033074 | 001012 |        |        | BNE        | TS327-10      | :BR TO ERROR HALT ON SEQ ERROR                                |
| 9655 | 033076 | 012706 | 001000 |        | MOV        | #BUFF,SP      | :STACK POINTER SETUP  |
| 9656 | 033102 | 012767 | 033112 | 144720 | MOV        | #RETC3,RTRAP3 | :RTURN FROM TRAP POINTER                                      |
| 9657 | 033110 | 104000 |        |        | EMT        |               | :TRAP ON THIS INSTRUCTION                                     |
| 9658 | 033112 | 022767 | 033112 | 145654 | RETC3: CMP | #,BUFF-4      | :CHECK FOR INCREMENTED P.C.                                   |
| 9659 | 033120 | 001404 |        |        | BEQ        | TS327         |   |
| 9660 |        |        |        |        |            |               | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====                |
| 9661 |        |        |        |        |            |               | : CONDITIONAL BRANCH INST. AND <====                          |
| 9662 |        |        |        |        |            |               | : REPLACE THE MOVE INSTRUCTION <====                          |
| 9663 |        |        |        |        |            |               | : WHICH FOLLOWS W/ 765 <====                                  |
| 9664 | 033122 | 012742 | 000774 |        | MOV        | #774,-(R2)    | :MOVE TO MAILBOX # ***** 774 *****                            |
| 9665 | 033126 | 005242 |        |        | INC        | -(R2)         | :SET MSGTYP TO FATAL ERROR                                    |
| 9666 | 033130 | 000000 |        |        | HALT       |               | :INCORRECT P.C.,OR WRONG STSTNM                               |
| 9667 |        |        |        |        |            |               | : OR SEQUENCE ERROR   |
| 9668 |        |        |        |        |            |               | :*****  |
| 9669 |        |        |        |        |            |               | :TEST 327 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK |
| 9670 |        |        |        |        |            |               | :*****  |
| 9671 | 033132 | 005212 |        |        | TS327: INC | (R2)          | :UPDATE TEST NUMBER   |
| 9672 | 033134 | 022712 | 000327 |        | CMP        | #327,(R2)     | :SEQUENCE ERROR?  |
| 9673 | 033140 | 001037 |        |        | BNE        | TS330-10      | :BR TO ERROR HALT ON SEQ ERROR                                |
| 9674 | 033142 | 012706 | 001000 |        | MOV        | #BUFF,SP      | :SET UP   |
| 9675 | 033146 | 012767 | 033164 | 144654 | MOV        | #RETD3,RTRAP3 | :SET UP   |
| 9676 | 033154 | 005067 | 144616 |        | CLR        | CC            | :CLEAR CC AND PRIORITY  |
| 9677 | 033160 | 000257 |        |        | CCC        |               |   |
| 9678 | 033162 | 104000 |        |        | EMT        |               | :TRAP   |
| 9679 | 033164 | 026727 | 145606 | 000000 | RETD3: CMP | BUFF-2,#0     | :TEST THAT OLD STATUS WENT TO STACK                           |
| 9680 | 033172 | 001404 |        |        | BEQ        | 18            |   |
| 9681 |        |        |        |        |            |               | : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====                |
| 9682 |        |        |        |        |            |               | : CONDITIONAL BRANCH INST. AND <====                          |
| 9683 |        |        |        |        |            |               | : REPLACE THE MOVE INSTRUCTION <====                          |
| 9684 |        |        |        |        |            |               | : WHICH FOLLOWS W/ 762 <====                                  |
| 9685 | 033174 | 012742 | 000775 |        | MOV        | #775,-(R2)    | :MOVE TO MAILBOX # ***** 775 *****                            |
| 9686 | 033200 | 005242 |        |        | INC        | -(R2)         | :SET MSGTYP TO FATAL ERROR                                    |
| 9687 | 033202 | 000000 |        |        | HALT       |               | :INCORRECT STATUS   |
| 9688 | 033204 | 012706 | 001000 |        | 18: MOV    | #BUFF,SP      | :SET UP   |
| 9689 | 033210 | 012767 | 033230 | 144612 | MOV        | #RETE3,RTRAP3 | :SET UP   |
| 9690 | 033216 | 012767 | 000357 | 144552 | MOV        | #357,CC       | :SET PRIORITY   |

```
9691 033224 000277          SCC          :SEI CC
9692 033226 104000          EMT          :TRAP
9693 033230 026727 145542 000357 RETE3: CMP          :COMPARES STATUS ON STACK
9694 033236 001404          BEQ          TS330
9695          :          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9696          :          :          CONDITIONAL BRANCH INST. AND <====
9697          :          :          REPLACE THE MOVE INSTRUCTION <====
9698          :          :          WHICH FOLLOWS W/ 740 <====
9699 033240 012742 000776          MOV          #776,-(R2) :MOVE TO MAILBOX # ***** 776 *****
9700 033244 005242          INC          -(R2)   :SET MSGTYP TO FATAL ERROR
9701 033246 000000          HALT         :INCORRECT STATUS ON STACK,OR WRCNG $TSTMP
9702          :          : OR SEQUENCE ERROR
9703          :.....:
9704          :TEST 330      TEST THAT 'NEW' STATUS IS CORRECT
9705          :.....:
9706 033250 005212          TS330: INC          (R2)   :UPDATE TEST NUMBER
9707 033252 022712 000330          CMP          #330,(R2) :SEQUENCE ERROR?
9708 033256 001106          BNE          TS331-10 :BR TO ERROR HALT ON SEQ ERROR
9709 033260 012706 001000          MOV          #BUFF,SP
9710 033264 012767 033300 144536          MOV          #RETF3,RTRAP3
9711 033272 005067 144534          CLR          RTRAP3+2 :CLEAR FUTURE PRIORITY AND CC
9712 033276 104000          EMT
9713          RETF3:          :TEST FOR 'C' CLEARED
9714 033300 100004          BPL          1$
9715          :          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9716          :          :          CONDITIONAL BRANCH INST. AND <====
9717          :          :          REPLACE THE MOVE INSTRUCTION <====
9718          :          :          WHICH FOLLOWS W/ 766 <====
9719 033302 012742 000777          MOV          #777,-(R2) :MOVE TO MAILBOX # ***** 777 *****
9720 033306 005242          INC          -(R2)   :SET MSGTYP TO FATAL ERROR
9721 033310 000000          HALT         :C NOT CLEARED
9722 033312
9723 033312 001004          1$: BNE          2$
9724          :          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9725          :          :          CONDITIONAL BRANCH INST. AND <====
9726          :          :          REPLACE THE MOVE INSTRUCTION <====
9727          :          :          WHICH FOLLOWS W/ 761 <====
9728 033314 012742 001000          MOV          #1000,-(R2) :MOVE TO MAILBOX # ***** 1000 *****
9729 033320 005242          INC          -(R2)   :SET MSGTYP TO FATAL ERROR
9730 033322 000000          HALT         :Z NOT CLEARED
9731 033324
9732 033324 102004          2$: BVC          3$
9733          :          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9734          :          :          CONDITIONAL BRANCH INST. AND <====
9735          :          :          REPLACE THE MOVE INSTRUCTION <====
9736          :          :          WHICH FOLLOWS W/ 754 <====
9737 033326 012742 001001          MOV          #1001,-(R2) :MOVE TO MAILBOX # ***** 1001 *****
9738 033332 005242          INC          -(R2)   :SET MSGTYP TO FATAL ERROR
9739 033334 000000          HALT         :V NOT CLEARED
9740 033336
9741 033336 103004          3$: BCC          4$
9742          :          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9743          :          :          CONDITIONAL BRANCH INST. AND <====
9744          :          :          REPLACE THE MOVE INSTRUCTION <====
9745          :          :          WHICH FOLLOWS W/ 747 <====
9746 033340 012742 001002          MOV          #1002,-(R2) :MOVE TO MAILBOX # ***** 1002 *****
```

|      |        |        |        |        |        |      |               |  |  |       |
|------|--------|--------|--------|--------|--------|------|---------------|--|--|-------|
| 9747 | 033344 | 005242 |        |        |        | INC  | -(R2)         |  | ;SET MSGTYP TO FATAL ERROR               |       |
| 9748 | 033346 | 000000 |        |        |        | HALT |               |  | ;C NOT CLEARED                           |       |
| 9749 | 033350 | 032767 | 000340 | 144420 | 4\$:   | BIT  | #340,CC       |  | ;TEST PRIORITY                           |       |
| 9750 | 033356 | 001404 |        |        |        | BEQ  | 5\$           |  |  |       |
| 9751 |        |        |        |        |        |      |               |  | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9752 |        |        |        |        |        |      |               |  | CONDITIONAL BRANCH INST. AND             | <==== |
| 9753 |        |        |        |        |        |      |               |  | REPLACE THE MOVE INSTRUCTION             | <==== |
| 9754 |        |        |        |        |        |      |               |  | WHICH FOLLOWS W/ 737                     | <==== |
| 9755 | 033360 | 012742 | 001003 |        |        | MOV  | #1003,-(R2)   |  | ;MOVE TO MAILBOX # ***** 1003 *****      |       |
| 9756 | 033364 | 005242 |        |        |        | INC  | -(R2)         |  | ;SET MSGTYP TO FATAL ERROR               |       |
| 9757 | 033366 | 000000 |        |        |        | HALT |               |  | ;PRIORITY NOT ZERO                       |       |
| 9758 | 033370 | 012706 | 001000 |        | 5\$:   | MOV  | #BUFF,SP      |  |  |       |
| 9759 | 033374 | 012767 | 033412 | 144426 |        | MOV  | #RETG3,RTRAP3 |  |  |       |
| 9760 | 033402 | 012767 | 000357 | 144422 |        | MOV  | #357,RTRAP3+2 |  | ;SET NEW 'CC' AND PRIORITY               |       |
| 9761 | 033410 | 104000 |        |        |        | EMT  |               |  | ;TRAP HERE                               |       |
| 9762 | 033412 |        |        |        | RETG3: |      |               |  |  |       |
| 9763 | 033412 | 100404 |        |        |        | BMI  | 1\$           |  |  |       |
| 9764 |        |        |        |        |        |      |               |  | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9765 |        |        |        |        |        |      |               |  | CONDITIONAL BRANCH INST. AND             | <==== |
| 9766 |        |        |        |        |        |      |               |  | REPLACE THE MOVE INSTRUCTION             | <==== |
| 9767 |        |        |        |        |        |      |               |  | WHICH FOLLOWS W/ 721                     | <==== |
| 9768 | 033414 | 012742 | 001004 |        |        | MOV  | #1004,-(R2)   |  | ;MOVE TO MAILBOX # ***** 1004 *****      |       |
| 9769 | 033420 | 005242 |        |        |        | INC  | -(R2)         |  | ;SET MSGTYP TO FATAL ERROR               |       |
| 9770 | 033422 | 000000 |        |        |        | HALT |               |  | ;N NOT SET                               |       |
| 9771 | 033424 |        |        |        | 1\$:   |      |               |  |  |       |
| 9772 | 033424 | 001404 |        |        |        | BEQ  | 2\$           |  |  |       |
| 9773 |        |        |        |        |        |      |               |  | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9774 |        |        |        |        |        |      |               |  | CONDITIONAL BRANCH INST. AND             | <==== |
| 9775 |        |        |        |        |        |      |               |  | REPLACE THE MOVE INSTRUCTION             | <==== |
| 9776 |        |        |        |        |        |      |               |  | WHICH FOLLOWS W/ 714                     | <==== |
| 9777 | 033426 | 012742 | 001005 |        |        | MOV  | #1005,-(R2)   |  | ;MOVE TO MAILBOX # ***** 1005 *****      |       |
| 9778 | 033432 | 005242 |        |        |        | INC  | -(R2)         |  | ;SET MSGTYP TO FATAL ERROR               |       |
| 9779 | 033434 | 000000 |        |        |        | HALT |               |  | ;Z NOT SET                               |       |
| 9780 | 033436 |        |        |        | 2\$:   |      |               |  |  |       |
| 9781 | 033436 | 102404 |        |        |        | BVS  | 3\$           |  |  |       |
| 9782 |        |        |        |        |        |      |               |  | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9783 |        |        |        |        |        |      |               |  | CONDITIONAL BRANCH INST. AND             | <==== |
| 9784 |        |        |        |        |        |      |               |  | REPLACE THE MOVE INSTRUCTION             | <==== |
| 9785 |        |        |        |        |        |      |               |  | WHICH FOLLOWS W/ 707                     | <==== |
| 9786 | 033440 | 012742 | 001006 |        |        | MOV  | #1006,-(R2)   |  | ;MOVE TO MAILBOX # ***** 1006 *****      |       |
| 9787 | 033444 | 005242 |        |        |        | INC  | -(R2)         |  | ;SET MSGTYP TO FATAL ERROR               |       |
| 9788 | 033446 | 000000 |        |        |        | HALT |               |  | ;V NOT SET                               |       |
| 9789 | 033450 |        |        |        | 3\$:   |      |               |  |  |       |
| 9790 | 033450 | 103404 |        |        |        | BCS  | 4\$           |  |  |       |
| 9791 |        |        |        |        |        |      |               |  | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9792 |        |        |        |        |        |      |               |  | CONDITIONAL BRANCH INST. AND             | <==== |
| 9793 |        |        |        |        |        |      |               |  | REPLACE THE MOVE INSTRUCTION             | <==== |
| 9794 |        |        |        |        |        |      |               |  | WHICH FOLLOWS W/ 702                     | <==== |
| 9795 | 033452 | 012742 | 001007 |        |        | MOV  | #1007,-(R2)   |  | ;MOVE TO MAILBOX # ***** 1007 *****      |       |
| 9796 | 033456 | 005242 |        |        |        | INC  | -(R2)         |  | ;SET MSGTYP TO FATAL ERROR               |       |
| 9797 | 033460 | 000000 |        |        |        | HALT |               |  | ;C NOT SET                               |       |
| 9798 | 033462 | 000257 |        |        | 4\$:   | CCC  |               |  |  |       |
| 9799 | 033464 | 022767 | 000340 | 144304 |        | CMP  | #340,CC       |  |  |       |
| 9800 | 033472 | 001404 |        |        |        | BEQ  | TS331         |  |  |       |
| 9801 |        |        |        |        |        |      |               |  | ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS | <==== |
| 9802 |        |        |        |        |        |      |               |  | CONDITIONAL BRANCH INST. AND             | <==== |

```
9803                                     :           REPLACE THE MOVE INSTRUCTION <====
9804                                     :           WHICH FOLLOWS W/ 671           <====
9805 033474 012742 001010                MOV   #1010,-(R2)  ;MOVE TO MAILBOX # ***** 1010 *****
9806 033500 005242                        INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
9807 033502 000000                        HALT              ;PRIORITY WAS CHANGED,OR WRONG $STNM
9808                                     :           OR SEQUENCE ERROR
9809                                     :*****
9810 ;TEST 331 TEST THAT ALL COMBINATION OF EMT WILL CAUSE A TRAP
9811 ;*****
9812 033504 005212 TS331: INC (R2) ;UPDATE TEST NUMBER
9813 033506 022712 000331                CMP   #331,(R2)  ;SEQUENCE ERROR?
9814 033512 001011                        BNE   BR47       ;BR TO ERROR HALT ON SEQ ERROR
9815                                     :***** F11 ***** ADD +376 TO SHORTEN TEST
9816 033514 012767 104376 000012          MOV   #EMT+376,RB ;INITIALIZE BASE EMT INSTRUCTION
9817 033522 012767 033546 144300          MOV   #RA,30     ;RETURN FROM TRAP TO RA
9818 033530 012706 001000 RC: MOV #BUFF,SP ;SET UP STACK POINTER
9819 033534 104000 RB: EMT ;TRAP INST. WILL BE MODIFIED TO EMT+377
9820 033536 BR47:
9821 033536 012742 001011                MOV   #1011,-(R2) ;MOVE TO MAILBOX # ***** 1011 *****
9822 033542 005242                        INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
9823 033544 000000                        HALT              ;PREVIOUS INST FAILED TO TRAP,OR WRONG $STNM
9824 033546 005267 177762 RA: INC RB ;INCREMENT TRAP INSTRUCTION
9825 033552 022767 104377 177754          CMP   #104377,RB ;EMT+377 TO EMT?
9826 033560 103363                        BHIS  RC          ;HAVE WE TESTED ALL
9827                                     :YES
9828 033562 012767 000032 144240          MOV   #32,30     ;/.+
9829 033570 005067 144236                        CLR   32         ;HALT
9830 ;*****
9831 ;TEST 332 TEST THAT A TRAP OCCURES ON AN 'TRACE-TRT' INSTRUCTION
9832 ;*****
9833 033574 005212 TS332: INC (R2) ;UPDATE TEST NUMBER
9834 033576 022712 000332                CMP   #332,(R2)  ;SEQUENCE ERROR?
9835 033602 001006                        BNE   TS333-10   ;BR TO ERROR HALT ON SEQ ERROR
9836 033604 012706 001000                MOV   #BUFF,SP   ;STACK POINTER SETUP
9837 033610 012767 033630 144176          MOV   #RETA4,RTRAP4 ;RETURN LOCATION
9838 033616 000003                        TRT              ;RESERVED INSTRUCTION, SHOULD TRAP
9839 033620 012742 001012                MOV   #1012,-(R2) ;MOVE TO MAILBOX # ***** 1012 *****
9840 033624 005242                        INC   -(R2)       ;SET MSGTYP TO FATAL ERROR
9841 033626 000000                        HALT              ;TRT DIDN'T TRAP,OR WRONG $STNM
9842 033630 RETA4:
9843 ;*****
9844 ;TEST 333 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9845 ;*****
9846 033630 005212 TS333: INC (R2) ;UPDATE TEST NUMBER
9847 033632 022712 000333                CMP   #333,(R2)  ;SEQUENCE ERROR?
9848 033636 001011                        BNE   TS334-10   ;BR TO ERROR HALT ON SEQ ERROR
9849 033640 012706 001000                MOV   #BUFF,SP   ;STACK POINTER SETUP
9850 033644 012767 033654 144142          MOV   #RETB4,RTRAP4 ;RETURN POINTER
9851 033652 000003                        TRT              ;RESERVED INSTRUCTION
9852 033654 020627 000774 RETB4: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
9853 033660 001404                        BEQ   TS334
9854                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9855                                     :           CONDITIONAL BRANCH INST. AND <====
9856                                     :           REPLACE THE MOVE INSTRUCTION <====
9857                                     :           WHICH FOLLOWS W/ 766         <====
9858 033662 012742 001013                MOV   #1013,-(R2) ;MOVE TO MAILBOX # ***** 1013 *****
```

```
9859 033666 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
9860 033670 000000          HALT                   ;NOT DECREMENTED TWO WORDS,OR WRONG $STNM
9861                                     ; OR SEQUENCE ERROR
9862                                     ;*****
9863 ;TEST 334          TEST THAT PROPER P.C. IS SAVED
9864                                     ;*****
9865 033672 005212          TS334: INC    (R2)          ;UPDATE TEST NUMBER
9866 033674 022712 000334    CMP    #334,(R2)       ;SEQUENCE ERROR?
9867 033700 001012          BNE    TS335-10        ;BR TO ERROR HALT ON SEQ ERROR
9868 033702 012706 001000    MOV    #BUFF,SP        ;STACK POINTER SETUP
9869 033706 012767 033716 144100  MOV    #RETC4,RTRAP4    ;RETURN FROM TRAP POINTER
9870 033714 000003          TRT                   ;TRAP ON THIS INSTRUCTION
9871 033716 022767 033716 145050  RETC4: CMP    #,BUFF-4    ;CHECK FOR INCREMENTED P.C.
9872 033724 001404          BEQ    TS335
9873                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9874                                     ;          CONDITIONAL BRANCH INST. AND <====
9875                                     ;          REPLACE THE MOVE INSTRUCTION <====
9876                                     ;          WHICH FOLLOWS W/ 765 <====
9877 033726 012742 001014    MOV    #1014,-(R2)     ;MOVE TO MAILBOX # ***** 1014 *****
9878 033732 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
9879 033734 000000          HALT                   ;INCORRECT P.C.,OR WRONG $STNM
9880                                     ; OR SEQUENCE ERROR
9881                                     ;*****
9882 ;TEST 335          TEST THAT "OLD" CC AND PRIORITY ARE PLACED ON STACK
9883                                     ;*****
9884 033736 005212          TS335: INC    (R2)          ;UPDATE TEST NUMBER
9885 033740 022712 000335    CMP    #335,(R2)       ;SEQUENCE ERROR?
9886 033744 001037          BNE    TS336-10        ;BR TO ERROR HALT ON SEQ ERROR
9887 033746 012706 001000    MOV    #BUFF,SP        ;SET UP
9888 033752 012767 033770 144034    MOV    #RETD4,RTRAP4    ;SET UP
9889 033760 005067 144012    CLR    CC              ;CLEAR CC AND PRIORITY
9890 033764 000257          CCC
9891 033766 000003          TRT                   ;TRAP
9892 033770 026727 145002 000000  RETD4: CMP    BUFF-2,#0    ;TEST THAT OLD STATUS WENT TO STACK
9893                                     ;TEST FOR ALL ZEROS
9894 033776 001404          BEQ    1$
9895                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9896                                     ;          CONDITIONAL BRANCH INST. AND <====
9897                                     ;          REPLACE THE MOVE INSTRUCTION <====
9898                                     ;          WHICH FOLLOWS W/ 762 <====
9899 034000 012742 001015    MOV    #1015,-(R2)     ;MOVE TO MAILBOX # ***** 1015 *****
9900 034004 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
9901 034006 000000          HALT                   ;INCORRECT STATUS
9902 034010 012706 001000    1$: MOV    #BUFF,SP        ;SET UP
9903 034014 012767 034034 143772    MOV    #RETE4,RTRAP4    ;SET UP
9904 034022 012767 000357 143746    MOV    #357,CC         ;SET PRIORITY
9905 034030 000277          SCC                   ;SET-SET CC
9906 034032 000003          TRT                   ;TRAP
9907 034034 026727 144736 000357  RETE4: CMP    BUFF-2,#357    ;COMPARES STATUS ON STACK
9908 034042 001404          BEQ    TS336
9909                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9910                                     ;          CONDITIONAL BRANCH INST. AND <====
9911                                     ;          REPLACE THE MOVE INSTRUCTION <====
9912                                     ;          WHICH FOLLOWS W/ 740 <====
9913 034044 012742 001016    MOV    #1016,-(R2)     ;MOVE TO MAILBOX # ***** 1016 *****
9914 034050 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
```



```
9915 034052 000000          HALT          ;INCORRECT STATUS ON STACK,OR WRONG $STNM
9916                                     ; OR SEQUENCE ERROR
9917                                     ;*****
9918 :TEST 336          TEST THAT 'NEW' STATUS IS CORRECT
9919 :*****
9920 034054 005212          TS336:  INC      (R2)          ;UPDATE TEST NUMBER
9921 034056 022712 000336    CMP      #336,(R2)         ;SEQUENCE ERROR?
9922 034062 001110          BNE      BR51             ;BR TO ERROR HALT ON SEQ ERROR
9923 034064 012706 001000    MOV      #BUFF,SP
9924 034070 012767 034104 143716  MOV      #RETF4,RTRAP4
9925 034076 005067 143714    CLR      RTRAP4+2        ;CLEAR FUTURE PRIORITY AND CC
9926 034102 000003          TRT
9927 034104          RETF4:          ;TEST FOR 'C' CLEARED
9928 034104 100004          BPL      1$
9929                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9930                                     ;          CONDITIONAL BRANCH INST. AND <====
9931                                     ;          REPLACE THE MOVE INSTRUCTION <====
9932                                     ;          WHICH FOLLOWS W/ 766 <====
9933 034106 012742 001017    MOV      #1017,-(R2)     ;MOVE TO MAILBOX # ***** 1017 *****
9934 034112 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9935 034114 000000          HALT          ;C NOT CLEARED
9936 034116          1$:
9937 034116 001004          BNE      2$
9938                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9939                                     ;          CONDITIONAL BRANCH INST. AND <====
9940                                     ;          REPLACE THE MOVE INSTRUCTION <====
9941                                     ;          WHICH FOLLOWS W/ 761 <====
9942 034120 012742 001020    MOV      #1020,-(R2)     ;MOVE TO MAILBOX # ***** 1020 *****
9943 034124 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9944 034126 000000          HALT          ;Z NOT CLEARED
9945 034130          2$:
9946 034130 102004          BVC      3$
9947                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9948                                     ;          CONDITIONAL BRANCH INST. AND <====
9949                                     ;          REPLACE THE MOVE INSTRUCTION <====
9950                                     ;          WHICH FOLLOWS W/ 754 <====
9951 034132 012742 001021    MOV      #1021,-(R2)     ;MOVE TO MAILBOX # ***** 1021 *****
9952 034136 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9953 034140 000000          HALT          ;V NOT CLEARED
9954 034142          3$:
9955 034142 103004          BCC      4$
9956                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9957                                     ;          CONDITIONAL BRANCH INST. AND <====
9958                                     ;          REPLACE THE MOVE INSTRUCTION <====
9959                                     ;          WHICH FOLLOWS W/ 747 <====
9960 034144 012742 001022    MOV      #1022,-(R2)     ;MOVE TO MAILBOX # ***** 1022 *****
9961 034150 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9962 034152 000000          HALT          ;C NOT CLEARED
9963 034154 032767 000340 143614 4$:  BIT      #340,CC        ;TEST PRIORITY
9964 034162 001404          BEQ      5$
9965                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9966                                     ;          CONDITIONAL BRANCH INST. AND <====
9967                                     ;          REPLACE THE MOVE INSTRUCTION <====
9968                                     ;          WHICH FOLLOWS W/ 737 <====
9969 034164 012742 001023    MOV      #1023,-(R2)     ;MOVE TO MAILBOX # ***** 1023 *****
9970 034170 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
```

```
9971 034172 000000
9972 034174 012706 001000
9973 034200 012767 034216 143606
9974 034206 012767 000357 143602
9975 034214 000003
9976 034216
9977 034216 100404
9978
9979
9980
9981
9982 034220 012742 001024
9983 034224 005242
9984 034226 000000
9985 034230
9986 034230 001404
9987
9988
9989
9990
9991 034232 012742 001025
9992 034236 005242
9993 034240 000000
9994 034242
9995 034242 102404
9996
9997
9998
9999
10000 034244 012742 001026
10001 034250 005242
10002 034252 000000
10003 034254
10004 034254 103404
10005
10006
10007
10008
10009 034256 012742 001027
10010 034262 005242
10011 034264 000000
10012 034266 016706 143504
10013 034272 042706 000017
10014 034276 022706 000340
10015 034302 001404
10016
10017
10018
10019
10020 034304
10021 034304 012742 001030
10022 034310 005242
10023 034312 000000
10024 034314 012767 000016 143472
10025 034322 005067 143470
10026
```

5\$: HALT ;PRIORITY NOT ZERO  
MOV #BUFF,SP  
MOV #RETG4,RTRAP4  
MOV #357,RTRAP4+2 ;SET NEW "CC" AND PRIORITY  
TRT ;TRAP HERE

RETG4: BMI 1\$  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 721 <====  
; MOVE TO MAILBOX # \*\*\*\*\* 1024 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; N NOT SET

1\$: BEQ 2\$  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 714 <====  
; MOVE TO MAILBOX # \*\*\*\*\* 1025 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; Z NOT SET

2\$: BVS 3\$  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 707 <====  
; MOVE TO MAILBOX # \*\*\*\*\* 1026 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; V NOT SET

3\$: BCS 4\$  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 702 <====  
; MOVE TO MAILBOX # \*\*\*\*\* 1027 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; C NOT SET

4\$: MOV CC,SP  
BIC #17,SP  
CMP #340,SP  
BEQ BR51A  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 667 <====

BR51: MOV #1030,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 1030 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;PRIORITY WAS CHANGED,OR WRONG \$STNM

BR51A: MOV #16,14  
CLR 16

10027  
10028  
10029  
10030  
10031  
10032  
10033  
10034  
10035  
10036 034326 005212  
10037 034330 022712 000337  
10038 034334 001006  
10039 034336 012706 001000  
10040 034342 012767 034362 143434  
10041 034350 000100  
10042 034352 012742 001031  
10043 034356 005242  
10044 034360 000000  
10045 034362  
10046  
10047  
10048  
10049 034362 005212  
10050 034364 022712 000340  
10051 034370 001011  
10052 034372 012706 001000  
10053 034376 012767 034406 143400  
10054 034404 000100  
10055 034406 020627 000774  
10056 034412 001404  
10057  
10058  
10059  
10060  
10061 034414 012742 001032  
10062 034420 005242  
10063 034422 000000  
10064  
10065  
10066  
10067  
10068 034424 005212  
10069 034426 022712 000341  
10070 034432 001012  
10071 034434 012706 001000  
10072 034440 012767 034450 143336  
10073 034446 000100  
10074 034450 022767 034450 144316  
10075 034456 001404  
10076  
10077  
10078  
10079  
10080 034460 012742 001033  
10081 034464 005242  
10082 034466 000000

;PDP-11 ILLEGAL AND ADDRESS INSTRUCTION TEST  
;ALL INSTRUCTIONS THAT ARE RESERVED  
;SHOULD TRAP TO LOCATION 4, AND THE  
;PC THAT POINTS TO THE TRAPPING INSTRUCTION  
;SHOULD BE PLACED ON THE STACK

\*\*\*\*\*  
;TEST 337 TEST THAT A TRAP OCCURS ON AN ILLEGAL INSTRUCTION  
\*\*\*\*\*

TS337: INC (R2) ;UPDATE TEST NUMBER  
CMP #337,(R2) ;SEQUENCE ERROR?  
BNE TS340-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #BUFF,SP ;STACK POINTER SETUP  
MOV #RETA5,RTRAP5 ;RETURN LOCATION  
JMP %0 ;ILLEGAL INSTRUCTION, SHOULD TRAP  
MOV #1031,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 1031 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;ILLEGAL INSTRUCTION DIDN'T TRAP,OR WRONG \$STNM

RETA5:  
\*\*\*\*\*  
;TEST 340 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION  
\*\*\*\*\*

TS340: INC (R2) ;UPDATE TEST NUMBER  
CMP #340,(R2) ;SEQUENCE ERROR?  
BNE TS341-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #BUFF,SP ;STACK POINTER SETUP  
MOV #RETB5,RTRAP5 ;RETURN POINTER  
JMP %0 ;RESERVED INSTRUCTION  
RETB5: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP  
BEQ TS341

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 766 <====  
; MOVE TO MAILBOX # \*\*\*\*\* 1032 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; NOT DECREMENTED TWO WORDS,OR WRONG \$STNM  
; OR SEQUENCE ERROR

\*\*\*\*\*  
;TEST 341 TEST THAT PROPER P.C. IS SAVED  
\*\*\*\*\*

TS341: INC (R2) ;UPDATE TEST NUMBER  
CMP #341,(R2) ;SEQUENCE ERROR?  
BNE TS342-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #BUFF,SP ;STACK POINTER SETUP  
MOV #RETC5,RTRAP5 ;RETURN FROM TRAP POINTER  
JMP %0 ;TRAP ON THIS INSTRUCTION  
RETC5: CMP #,BUFF-4 ;CHECK FOR INCREMENTED P.C.  
BEQ TS342

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 765 <====  
; MOVE TO MAILBOX # \*\*\*\*\* 1033 \*\*\*\*\*  
; SET MSGTYP TO FATAL ERROR  
; INCORRECT P.C.,OR WRONG \$STNM



```

10139 034650 001004          BNE      2$
10140
10141
10142
10143
10144 034652 012742 001037    MOV      #1037,-(R2)
10145 034656 005242          INC      -(R2)
10146 034660 000000          HALT
10147 034662          2$:
10148 034662 102004          BVC      3$
10149
10150
10151
10152
10153 034664 012742 001040    MOV      #1040,-(R2)
10154 034670 005242          INC      -(R2)
10155 034672 000000          HALT
10156 034674          3$:
10157 034674 103004          BCC      4$
10158
10159
10160
10161
10162 034676 012742 001041    MOV      #1041,-(R2)
10163 034702 005242          INC      -(R2)
10164 034704 000000          HALT
10165 034706 032767 000357 143062 4$:
10166 034714 001404          BIT      #357,CC
10167
10168
10169
10170
10171 034716 012742 001042    MOV      #1042,-(R2)
10172 034722 005242          INC      -(R2)
10173 034724 000000          HALT
10174 034726 012706 001000    MOV      #BUFF,SP
10175 034732 012767 034750 143044 5$:
10176 034740 012767 000357 143040    MOV      #RETG5,RTRAP5
10177 034746 000100          MOV      #357,RTRAP5+2
10178 034750          JMP      %0
10179 034750 100404    RETG5:
10180
10181
10182
10183
10184 034752 012742 001043    BMI      1$
10185 034756 005242          BNE      2$
10186 034760 000000          BVC      3$
10187 034762          BCC      4$
10188 034762 001404          BEQ      5$
10189
10190
10191
10192
10193 034764 012742 001044    MOV      #1044,-(R2)
10194 034770 005242          INC      -(R2)

```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 761
<====
<====
<====
<====
; MOVE TO MAILBOX # ***** 1037 *****
; SET MSGTYP TO FATAL ERROR
; Z NOT CLEARED

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 754
<====
<====
<====
<====
; MOVE TO MAILBOX # ***** 1040 *****
; SET MSGTYP TO FATAL ERROR
; V NOT CLEARED

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 747
<====
<====
<====
<====
; MOVE TO MAILBOX # ***** 1041 *****
; SET MSGTYP TO FATAL ERROR
; C NOT CLEARED
; TEST PRIORITY

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 737
<====
<====
<====
<====
; MOVE TO MAILBOX # ***** 1042 *****
; SET MSGTYP TO FATAL ERROR
; PRIORITY NOT ZERO

; SET NEW "CC" AND PRIORITY
; TRAP HERE

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 721
<====
<====
<====
<====
; MOVE TO MAILBOX # ***** 1043 *****
; SET MSGTYP TO FATAL ERROR
; N NOT SET

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 714
<====
<====
<====
<====
; MOVE TO MAILBOX # ***** 1044 *****
; SET MSGTYP TO FATAL ERROR

```

```
10195 034772 000000
10196 034774
10197 034774 102404
10198
10199
10200
10201
10202 034776 012742 001045
10203 035002 005242
10204 035004 000000
10205 035006
10206 035006 103404
10207
10208
10209
10210
10211 035010 012742 001046
10212 035014 005242
10213 035016 000000
10214 035020 016706 142752
10215 035024 022706 000357
10216 035030 001404
10217
10218
10219
10220
10221 035032 012742 001047
10222 035036 005242
10223 035040 000000
10224
10225
10226
10227
10228 035042 005212
10229 035044 022712 000344
10230 035050 001006
10231 035052 012706 001000
10232 035056 012767 035076 142720
10233 035064 004000
10234 035066 012742 001050
10235 035072 005242
10236 035074 000000
10237 035076
10238
10239
10240
10241 035076 005212
10242 035100 022712 000345
10243 035104 001011
10244 035106 012706 001000
10245 035112 012767 035122 142664
10246 035120 004000
10247 035122 020627 000774
10248 035126 001404
10249
10250
```

```

      HALT                ;Z NOT SET
2$:   BVS      3$
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ;         CONDITIONAL BRANCH INST. AND <====
      ;         REPLACE THE MOVE INSTRUCTION <====
      ;         WHICH FOLLOWS W/ 707 <====
      MOV      #1045,-(R2) ;MOVE TO MAILBOX # ***** 1045 *****
      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
      HALT
3$:   BCS      4$
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ;         CONDITIONAL BRANCH INST. AND <====
      ;         REPLACE THE MOVE INSTRUCTION <====
      ;         WHICH FOLLOWS W/ 702 <====
      MOV      #1046,-(R2) ;MOVE TO MAILBOX # ***** 1046 *****
      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
      HALT
4$:   MOV      CC,SP
      CMP      #357,SP
      BEQ      TS344
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ;         CONDITIONAL BRANCH INST. AND <====
      ;         REPLACE THE MOVE INSTRUCTION <====
      ;         WHICH FOLLOWS W/ 671 <====
      MOV      #1047,-(R2) ;MOVE TO MAILBOX # ***** 1047 *****
      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
      HALT
      ;PRIORITY WAS CHANGED,OR WRONG $STNM
      ; OR SEQUENCE ERROR
      ;*****
      ;TEST 344 TEST THAT A TRAP OCCURES ON ALL ILLEGAL INSTRUCTION
      ;*****
TS344: INC      (R2)        ;UPDATE TEST NUMBER
      CMP      #344,(R2)   ;SEQUENCE ERROR?
      BNE     TS345-10    ;BR TO ERROR HALT ON SEQ ERROR
      MOV     #BUFF,SP    ;STACK POINTER SETUP
      MOV     #RETH5,RTRAP5 ;RETURN LOCATION
      JSR     %0,%0       ;RESERVED INSTRUCTION, SHOULD TRAP
      MOV     #1050,-(R2) ;MOVE TO MAILBOX # ***** 1050 *****
      INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
      HALT
      ;DIDN'T TRAP,OR WRONG $STNM
RETH5:
      ;*****
      ;TEST 345 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
      ;*****
TS345: INC      (R2)        ;UPDATE TEST NUMBER
      CMP      #345,(R2)   ;SEQUENCE ERROR?
      BNE     TS346-10    ;BR TO ERROR HALT ON SEQ ERROR
      MOV     #BUFF,SP    ;STACK POINTER SETUP
      MOV     #RETJ,RTRAP5 ;RETURN POINTER
      JSR     %0,%0       ;RESERVED INSTRUCTION
      RETJ:  CMP      SP,#BUFF-4 ;TEST DECREMENT OF SP
      BEQ     TS346
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ;         CONDITIONAL BRANCH INST. AND <====
```

```
10251                                     :           REPLACE THE MOVE INSTRUCTION <====
10252                                     :           WHICH FOLLOWS W/ 766           <====
10253 035130 012742 001051             MOV   #1051,-(R2)   :MOVE TO MAILBOX # ***** 1051 *****
10254 035134 005242                   INC   -(R2)       :SET MSGTYP TO FATAL ERROR
10255 035136 000000                   HALT              :NOT DECREMENTED TWO WORDS,OR WRONG $STNM
10256                                     :           OR SEQUENCE ERROR
10257                                     :
10258 :TEST 346          TEST THAT PROPER P.C. IS SAVED
10259 :
10260 035140 005212             TS346: INC   (R2)           :UPDATE TEST NUMBER
10261 035142 022712 000346             CMP   #346,(R2)   :SEQUENCE ERROR?
10262 035146 001012             BNE   TS347-10    :BR TO ERROR HALT ON SEQ ERROR
10263 035150 012706 001000             MOV   #BUFF,SP   :STACK POINTER SETUP
10264 035154 012767 035164 142622     MOV   #RETK,RTRAP5 :RETURN FROM TRAP POINTER
10265 035162 004000             INSTK: JSR   %0,%0 :TRAP ON THIS INSTRUCTION
10266 035164 022767 035164 143602     RETK:  CMP   #INSTK+2,BUFF-4 :CHECK FOR INCREMENTED P.C.
10267 035172 001404             BEQ   TS347
10268                                     :
10269                                     :           TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10270                                     :           CONDITIONAL BRANCH INST. AND <====
10271                                     :           REPLACE THE MOVE INSTRUCTION <====
10272                                     :           WHICH FOLLOWS W/ 765           <====
10272 035174 012742 001052             MOV   #1052,-(R2) :MOVE TO MAILBOX # ***** 1052 *****
10273 035200 005242                   INC   -(R2)       :SET MSGTYP TO FATAL ERROR
10274 035202 000000                   HALT              :INCORRECT P.C.,OR WRONG $STNM
10275                                     :           OR SEQUENCE ERROR
10276                                     :
10277 :
10278 :TEST 347          TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
10279 :
10280 035204 005212             TS347: INC   (R2)           :UPDATE TEST NUMBER
10281 035206 022712 000347             CMP   #347,(R2)   :SEQUENCE ERROR?
10282 035212 001037             BNE   TS350-10    :BR TO ERROR HALT ON SEQ ERROR
10283 035214 012706 001000             MOV   #BUFF,SP   :SET UP
10284 035220 012767 035236 142556     MOV   #RETL,RTRAP5 :SET UP
10285 035226 005067 142544             CLR   CC          :CLEAR CC AND PRIORITY
10286 035232 000257             CCC
10287 035234 004000             JSR   %0,%0       :TRAP
10288 035236 026727 143534 000000     RETL:  CMP   BUFF-2,%0 :TEST THAT OLD STATUS WENT TO STACK
10289 035244 001404             BEQ   1$
10290                                     :
10291                                     :           TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10292                                     :           CONDITIONAL BRANCH INST. AND <====
10293                                     :           REPLACE THE MOVE INSTRUCTION <====
10294                                     :           WHICH FOLLOWS W/ 762           <====
10294 035246 012742 001053             MOV   #1053,-(R2) :MOVE TO MAILBOX # ***** 1053 *****
10295 035252 005242                   INC   -(R2)       :SET MSGTYP TO FATAL ERROR
10296 035254 000000                   HALT              :INCORRECT STATUS
10297 035256 012706 001000             1$:  MOV   #BUFF,SP   :SET UP
10298 035262 012767 035302 142514     MOV   #RETM,RTRAP5 :SET UP
10299 035270 012767 000357 142500     MOV   #357,CC     :SET PRIORITY
10300 035276 000277             SCC
10301 035300 004000             JSR   %0,%0       :TRAP
10302 035302 026727 143470 000357     RETM:  CMP   BUFF-2,#357 :COMPARES STATUS ON STACK
10303 035310 001404             BEQ   TS350
10304                                     :
10305                                     :           TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10306                                     :           CONDITIONAL BRANCH INST. AND <====
10306                                     :           REPLACE THE MOVE INSTRUCTION <====
```

```
10307
10308 035312 012742 001054      MOV    #1054,-(R2)      ; WHICH FOLLOWS W/ 740      <====
10309 035316 005242              INC    -(R2)           ; MOVE TO MAILBOX # ***** 1054 *****
10310 035320 000000              HALT                    ; SET MSGTYP TO FATAL ERROR
10311                                     ; INCORRECT STATUS ON STACK,OR WRONG $STNM
10312                                     ; OR SEQUENCE ERROR
10313 :*****
10313 :TEST 350      TEST THAT 'NEW' STATUS IS CORRECT
10314 :*****
10315 035322 005212      TS350: INC    (R2)           ; UPDATE TEST NUMBER
10316 035324 022712 000350      CMP    #350,(R2)       ; SEQUENCE ERROR?
10317 035330 001105              BNE    TS351-10        ; BR TO ERROR HALT ON SEQ ERROR
10318 035332 012706 001000      MOV    #BUFF,SP
10319 035336 012767 035352 142440  MOV    #RETN,RTRAPS
10320 035344 005067 142436      CLR    RTRAPS+2       ; CLEAR FUTURE PRIORITY AND CC
10321 035350 004000      JSR    %0,%0
10322 035352              RETN:                  ; TEST FOR 'C' CLEARED
10323 035352 100004      BPL    1$
10324                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
10325                                     ; CONDITIONAL BRANCH INST. AND      <====
10326                                     ; REPLACE THE MOVE INSTRUCTION      <====
10327                                     ; WHICH FOLLOWS W/ 766      <====
10328 035354 012742 001055      MOV    #1055,-(R2)     ; MOVE TO MAILBOX # ***** 1055 *****
10329 035360 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
10330 035362 000000              HALT                    ; C NOT CLEARED
10331 035364              1$:
10332 035364 001004      BNE    2$
10333                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
10334                                     ; CONDITIONAL BRANCH INST. AND      <====
10335                                     ; REPLACE THE MOVE INSTRUCTION      <====
10336                                     ; WHICH FOLLOWS W/ 761      <====
10337 035366 012742 001056      MOV    #1056,-(R2)     ; MOVE TO MAILBOX # ***** 1056 *****
10338 035372 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
10339 035374 000000              HALT                    ; Z NOT CLEARED
10340 035376              2$:
10341 035376 102004      BVC    3$
10342                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
10343                                     ; CONDITIONAL BRANCH INST. AND      <====
10344                                     ; REPLACE THE MOVE INSTRUCTION      <====
10345                                     ; WHICH FOLLOWS W/ 754      <====
10346 035400 012742 001057      MOV    #1057,-(R2)     ; MOVE TO MAILBOX # ***** 1057 *****
10347 035404 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
10348 035406 000000              HALT                    ; V NOT CLEARED
10349 035410              3$:
10350 035410 103004      BCC    4$
10351                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
10352                                     ; CONDITIONAL BRANCH INST. AND      <====
10353                                     ; REPLACE THE MOVE INSTRUCTION      <====
10354                                     ; WHICH FOLLOWS W/ 747      <====
10355 035412 012742 001060      MOV    #1060,-(R2)     ; MOVE TO MAILBOX # ***** 1060 *****
10356 035416 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
10357 035420 000000              HALT                    ; C NOT CLEARED
10358 035422 016700 142350      4$: MOV    CC,%0          ; TEMP STORAGE
10359 035426 001404      BEQ    5$
10360                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
10361                                     ; CONDITIONAL BRANCH INST. AND      <====
10362                                     ; REPLACE THE MOVE INSTRUCTION      <====
```



```
10363
10364 035430 012742 001061      MOV      #1061,-(R2)      ; WHICH FOLLOWS W/ 740 <====
10365 035434 005242              INC      -(R2)           ; MOVE TO MAILBOX # ***** 1061 *****
10366 035436 000000              HALT                    ; SET MSGTYP TO FATAL ERROR
10367 035440 012706 001000      5$:    MOV      #BUFF,SP    ; PRIORITY NOT ZERO
10368 035444 012767 035462 142332  MOV      #RETO,RTRAP5
10369 035452 012767 000357 142326  MOV      #357,RTRAP5+2 ; SET NEW "CC" AND PRIORITY
10370 035460 004000              JSR      %0,%0          ; TRAP HERE
10371 035462
10372 035462 100404      RETO:   BMI      1$
10373
10374                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10375                          ; CONDITIONAL BRANCH INST. AND <====
10376                          ; REPLACE THE MOVE INSTRUCTION <====
10377                          ; WHICH FOLLOWS W/ 722 <====
10377 035464 012742 001062      MOV      #1062,-(R2)    ; MOVE TO MAILBOX # ***** 1062 *****
10378 035470 005242              INC      -(R2)           ; SET MSGTYP TO FATAL ERROR
10379 035472 000000              HALT                    ; N NOT SET
10380 035474
10381 035474 001404      1$:    BEQ      2$
10382
10383                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10384                          ; CONDITIONAL BRANCH INST. AND <====
10385                          ; REPLACE THE MOVE INSTRUCTION <====
10386                          ; WHICH FOLLOWS W/ 715 <====
10386 035476 012742 001063      MOV      #1063,-(R2)    ; MOVE TO MAILBOX # ***** 1063 *****
10387 035502 005242              INC      -(R2)           ; SET MSGTYP TO FATAL ERROR
10388 035504 000000              HALT                    ; Z NOT SET
10389 035506
10390 035506 102404      2$:    BVS      3$
10391
10392                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10393                          ; CONDITIONAL BRANCH INST. AND <====
10394                          ; REPLACE THE MOVE INSTRUCTION <====
10395                          ; WHICH FOLLOWS W/ 710 <====
10395 035510 012742 001064      MOV      #1064,-(R2)    ; MOVE TO MAILBOX # ***** 1064 *****
10396 035514 005242              INC      -(R2)           ; SET MSGTYP TO FATAL ERROR
10397 035516 000000              HALT                    ; V NOT SET
10398 035520
10399 035520 103404      3$:    BCS      4$
10400
10401                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10402                          ; CONDITIONAL BRANCH INST. AND <====
10403                          ; REPLACE THE MOVE INSTRUCTION <====
10404                          ; WHICH FOLLOWS W/ 703 <====
10404 035522 012742 001065      MOV      #1065,-(R2)    ; MOVE TO MAILBOX # ***** 1065 *****
10405 035526 005242              INC      -(R2)           ; SET MSGTYP TO FATAL ERROR
10406 035530 000000              HALT                    ; C NOT SET
10407 035532 016700 142240      4$:    MOV      CC,%0
10408 035536 022700 000357      CMP      #357,%0
10409 035542 001404      BEQ      TS351
10410
10411                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10412                          ; CONDITIONAL BRANCH INST. AND <====
10413                          ; REPLACE THE MOVE INSTRUCTION <====
10414                          ; WHICH FOLLOWS W/ 672 <====
10414 035544 012742 001066      MOV      #1066,-(R2)    ; MOVE TO MAILBOX # ***** 1066 *****
10415 035550 005242              INC      -(R2)           ; SET MSGTYP TO FATAL ERROR
10416 035552 000000              HALT                    ; PRIORITY WAS CHANGED,OR WRONG $1STNM
10417
10418                          ; OR SEQUENCE ERROR
```

10419  
10420  
10421  
10422 035554 005212  
10423 035556 022712 000351  
10424 035562 001006  
10425 035564 012706 000150  
10426 035570 012767 035610 142206  
10427 035576 005746  
10428 035600 012742 001067  
10429 035604 005242  
10430 035606 000000  
10431 035610  
10432  
10433  
10434  
10435  
10436 035610 005212  
10437 035612 022712 000352  
10438 035616 001011  
10439 035620 012706 000150  
10440 035624 012767 035634 142152  
10441 035632 005746  
10442 035634 020627 000142  
10443 035640 001404  
10444  
10445  
10446  
10447  
10448 035642 012742 001070  
10449 035646 005242  
10450 035650 000000  
10451  
10452  
10453  
10454  
10455  
10456 035652 005212  
10457 035654 022712 000353  
10458 035660 001041  
10459 035662 012706 000150  
10460 035666 005067 142254  
10461 035672 012767 035702 142104  
10462 035700 005246  
10463 035702 005767 142240  
10464 035706 001004  
10465  
10466  
10467  
10468  
10469 035710 012742 001071  
10470 035714 005242  
10471 035716 000000  
10472 035720 012705 001000  
10473 035724 012706 000400  
10474 035730 012767 035750 142046

```
*****  
:TEST 351 TEST THAT DECREMENT R6 TO A VALUE LESS THAN 400 TRAPS  
*****  
TS351: INC (R2) ;UPDATE TEST NUMBER  
CMP #351,(R2) ;SEQUENCE ERROR?  
BNE TS352-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #150,%6 ;R6 = 150  
MOV #TDEC1,4 ;STACK OVERFLOW TRAP POINTER  
TST -(6) ;WITH R6 = 150 SHOULD TRAP  
MOV #1067,-(R2) ;MOVE TO MAILBOX # ***** 1067 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;SHOULD HAVE TRAPPED,OR WRONG $STNM  
  
TDEC1:  
  
*****  
:TEST 352 TEST FOR DECREMENT OF R6 ON OVERFLOW TRAP  
*****  
TS352: INC (R2) ;UPDATE TEST NUMBER  
CMP #352,(R2) ;SEQUENCE ERROR?  
BNE TS353-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #150,%6 ;R6 = 150  
MOV #TDEC2,4 ;TRAP POINTER  
TST -(6) ;WITH R6 = 150 SHOULD TRAP  
TDEC2: CMP %6,#142 ;DID R6 DECREMENT  
BEQ TS353  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 766 <====  
MOV #1070,-(R2) ;MOVE TO MAILBOX # ***** 1070 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;R6 NOT = 142,OR WRONG $STNM  
; OR SEQUENCE ERROR  
  
*****  
:TEST 353 TEST DIFFERENT TYPES OF OVERFLOW  
*****  
TS353: INC (R2) ;UPDATE TEST NUMBER  
CMP #353,(R2) ;SEQUENCE ERROR?  
BNE TS354-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #150,%6  
CLR 146 ;STATUS WORD OF LOC 10  
MOV #TDEC3,4 ;RETURN TO LOC 4  
INC -(6)  
TDEC3: TST 146  
BNE 1$  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 764 <====  
MOV #1071,-(R2) ;MOVE TO MAILBOX # ***** 1071 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;INCREMENT OPERATION NOT INHIBITED  
1$: MOV #1000,%5  
MOV #400,%6  
MOV #TDEC4,4
```

```
10475 035736 124645          CMPB    -(6),-(5)
10476 035740 012742 001072    MOV     #1072,-(R2)      ;MOVE TO MAILBOX # ***** 1072 *****
10477 035744 005242          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10478 035746 000000          HALT                    ;STACK = 400 AND DECREMENTED, SHOULD TRAP
10479 035750 012706 000400    TDEC4: MOV     #400,%6
10480 035754 012767 035774 142022  MOV     #TDEC7,4
10481 035762 134546          BITB    -(5),-(6)
10482 035764
10483 035764 012742 001073    TDEC6: MOV     #1073,-(R2) ;MOVE TO MAILBOX # ***** 1073 *****
10484 035770 005242          INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
10485 035772 000000          HALT                    ;NO STACK OVERFLOW,OR WRONG $STNM
10486 035774
10487
10488
10489
10490
10491 035774 005212          ;*****
10492 035776 022712 000354    ;TEST 354      TEST THAT AN 77 CAUSES AN OVERFLOW TRAP
10493 036002 001011          ;*****
10494 036004 012706 000400    TS354: INC     (R2)       ;UPDATE TEST NUMBER
10495 036010 012767 036026 141772  CMP     #354,(R2)      ;SEQUENCE ERROR?
10496 036016 012767 036036 141760  BNE     VDEC2          ;BR TO ERROR HALT ON SEQ ERROR
10497 036024 000077          MOV     #400,%6       ;SET UP STACK TO OVERFLOW
10498 036026          MOV     #VDEC2,10     ;SET UP 77 VECTOR
10499 036026 012742 001074    MOV     #1074,-(R2)   ;MOVE TO MAILBOX # ***** 1074 *****
10500 036032 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
10501 036034 000000          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $STNM
10502 036036 012767 000012 141744  VDEC1: MOV     #10+2,10
10503
10504
10505
10506 036044 005212          ;*****
10507 036046 022712 000355    ;TEST 355      TEST THAT AN IOT CAUSES AN OVERFLOW TRAP
10508 036052 001011          ;*****
10509 036054 012706 000400    TS355: INC     (R2)       ;UPDATE TEST NUMBER
10510 036060 012767 036076 141732  CMP     #355,(R2)      ;SEQUENCE ERROR?
10511 036066 012767 036106 141710  BNE     VDEC4          ;BR TO ERROR HALT ON SEQ ERROR
10512 036074 000004          MOV     #400,%6       ;SET UP STACK TO OVERFLOW
10513 036076          MOV     #VDEC4,20     ;SET UP IOT VECTOR
10514 036076 012742 001075    MOV     #1075,-(R2)   ;MOVE TO MAILBOX # ***** 1075 *****
10515 036102 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
10516 036104 000000          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $STNM
10517 036106 012767 000022 14170-  VDEC3: MOV     #20+2,20
10518
10519
10520
10521 036114 005212          ;*****
10522 036116 022712 000356    ;TEST 356      TEST THAT AN EMT CAUSES AN OVERFLOW TRAP
10523 036122 001011          ;*****
10524 036124 012706 000400    TS356: INC     (R2)       ;UPDATE TEST NUMBER
10525 036130 012767 036146 141672  CMP     #356,(R2)      ;SEQUENCE ERROR?
10526 036136 012767 036156 141640  BNE     VDEC6          ;BR TO ERROR HALT ON SEQ ERROR
10527 036144 104000          MOV     #400,%6       ;SET UP STACK TO OVERFLOW
10528 036146          MOV     #VDEC6,30     ;SET UP EMT VECTOR
10529 036146 012742 001076    MOV     #1076,-(R2)   ;MOVE TO MAILBOX # ***** 1076 *****
10530 036152 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
```

```
10531 036154 000000          HALT          ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10532 036156 012767 000032 141644 VDEC5: MOV      #30+2,30
10533          ;*****
10534          ;TEST 357          TEST THAT AN TRT CAUSES AN OVERFLOW TRAP
10535          ;*****
10536 036164 005212          TS357: INC      (R2)          ;UPDATE TEST NUMBER
10537 036166 022712 000357          CMP      #357,(R2)         ;SEQUENCE ERROR?
10538 036172 001011          BNE      VDEC8            ;BR TO ERROR HALT ON SEQ ERROR
10539 036174 012706 000400          MOV      #400,%6         ;SET UP STACK TO OVERFLOW
10540 036200 012767 036216 141626          MOV      #VDEC8,34       ;SET UP TRAP VECTOR
10541 036206 012767 036226 141570          MOV      #VDEC7,4        ;SET UP OVERFLOW VECTOR
10542 036214 104400          TRAP                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10543 036216          VDEC8:
10544 036216 012742 001077          MOV      #1077,-(R2)     ;MOVE TO MAILBOX # ***** 1077 *****
10545 036222 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10546 036224 000000          HALT                     ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10547 036226 012767 000036 141600 VDEC7: MOV      #34+2,34
10548          ;*****
10549          ;TEST 360          TEST THAT AN TRT CAUSES AN OVERFLOW TRAP
10550          ;*****
10551 036234 005212          TS360: INC      (R2)          ;UPDATE TEST NUMBER
10552 036236 022712 000360          CMP      #360,(R2)         ;SEQUENCE ERROR?
10553 036242 001011          BNE      VDEC10           ;BR TO ERROR HALT ON SEQ ERROR
10554 036244 012706 000400          MOV      #400,%6         ;SET UP STACK TO OVERFLOW
10555 036250 012767 036266 141536          MOV      #VDEC10,14      ;SET UP TRT VECTOR
10556 036256 012767 036276 141520          MOV      #VDEC9,4        ;SET UP OVERFLOW VECTOR
10557 036264 000003          TRT                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10558 036266          VDEC10:
10559 036266 012742 001100          MOV      #1100,-(R2)     ;MOVE TO MAILBOX # ***** 1100 *****
10560 036272 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10561 036274 000000          HALT                     ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10562 036276 012767 000016 141510 VDEC9: MOV      #14+2,14
10563          ;*****
10564          ;TEST 361          TEST THAT AN ILLA CAUSES AN OVERFLOW TRAP
10565          ;*****
10566 036304 005212          TS361: INC      (R2)          ;UPDATE TEST NUMBER
10567 036306 022712 000361          CMP      #361,(R2)         ;SEQUENCE ERROR?
10568 036312 001011          BNE      VDEC11           ;BR TO ERROR HALT ON SEQ ERROR
10569 036314 012706 000400          MOV      #400,%6         ;SET UP STACK TO OVERFLOW
10570 036320 012767 036336 141456          MOV      #VDEC11,4       ;SET UP ILLA VECTOR
10571 036326 012767 036346 141450          MOV      #VDEC12,4       ;SET UP OVERFLOW VECTOR
10572 036334 004700          ILLA                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10573 036336          VDEC11:
10574 036336 012742 001101          MOV      #1101,-(R2)     ;MOVE TO MAILBOX # ***** 1101 *****
10575 036342 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10576 036344 000000          HALT                     ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10577 036346 012767 000006 141430 VDEC12: MOV      #4+2,4
10578 036354 020627 000370          CMP      %6,#370        ;STACK PUSHED FOUR WORDS?
10579 036360 001404          BEQ      TS362
10580          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10581          ;          CONDITIONAL BRANCH INST. AND <====
10582          ;          REPLACE THE MOVE INSTRUCTION <====
10583          ;          WHICH FOLLOWS W/ 754 <====
10584 036362 012742 001102          MOV      #1102,-(R2)     ;MOVE TO MAILBOX # ***** 1102 *****
10585 036366 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10586 036370 000000          HALT                     ;TRAP OVERFLOW DID NOT OCCUR
```

```
10587                                     ; OR SEQUENCE ERROR
10588                                     ;*****
10589 :TEST 362 TEST THAT AN ILLB CAUSES AN OVERFLOW TRAP
10590 :*****
10591 036372 005212 TS362: INC (R2) ;UPDATE TEST NUMBER
10592 036374 022712 000362 CMP #362,(R2) ;SEQUENCE ERROR?
10593 036400 001011 BNE VDEC13 ;BR TO ERROR HALT ON SEQ ERROR
10594 036402 012706 000400 MOV #400,%6 ;SET UP STACK TO OVERFLOW
10595 036406 012767 036424 141370 MOV #VDEC13,4 ;SET UP ILLB VECTOR
10596 036414 012767 036434 141362 MOV #VDEC14,4 ;SET UP OVERFLOW VECTOR
10597 036422 000100 ILLB ;THIS TRAP SHOULD CAUSE OVERFLOW
10598 036424 VDEC13:
10599 036424 012742 001103 MOV #1103,-(R2) ;MOVE TO MAILBOX # ***** 1103 *****
10600 036430 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10601 036432 000000 HALT ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10602 036434 012767 000006 141342 VDEC14: MOV #4+2,4
10603
10604 :*****
10605 :TEST 363 TEST FOR FALSE OVERFLOW TRAP
10606 :*****
10607 036442 005212 TS363: INC (R2) ;UPDATE TEST NUMBER
10608 036444 022712 000363 CMP #363,(R2) ;SEQUENCE ERROR?
10609 036450 001023 BNE FOVER ;BR TO ERROR HALT ON SEQ ERROR
10610
10611 036452 012767 036520 141324 MOV #FOVER,4 ;SET UP OVERFLOW POINTER
10612 036460 012706 001002 MOV #1002,%6
10613 036464 005746 TST -(6) ;SHOULD NOT OVERFLOW
10614 036466 012706 002002 MOV #2002,%6
10615 036472 005746 TST -(6) ;SHOULD NOT OVERFLOW
10616 036474 012706 004002 MOV #4002,%6
10617 036500 005746 TST -(6) ;SHOULD NOT OVERFLOW
10618 036502 012706 010002 MOV #10002,%6
10619 036506 005746 TST -(6)
10620 036510 012706 020000 MOV #20000,%6 ;SHOULD NOT OVERFLOW
10621 036514 005746 TST -(6)
10622 036516 000404 BR STP
10623 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10624 ; CONDITIONAL BRANCH INST. AND <====
10625 ; REPLACE THE MOVE INSTRUCTION <====
10626 ; WHICH FOLLOWS W/ 754 <====
10627 036520 FOVER:
10628 036520 012742 001104 MOV #1104,-(R2) ;MOVE TO MAILBOX # ***** 1104 *****
10629 036524 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10630 036526 000000 HALT ;IT OVERFLOWED,OR WRONG $TSTNM
```

```
10631 036530 012767 000006 141246 STP:  MOV #6,4
10632 036536 005067 141244          CLR  6
10633                                     ;*****
10634                                     ;TEST 364      TEST THAT BIT 4 PSW WILL CAUSE A TRAP TO 14
10635                                     ;*****
10636 036542 005212          TS364:  INC  (R2)          ;UPDATE TEST NUMBER
10637 036544 022712 000364          CMP  #364,(R2)      ;SEQUENCE ERROR?
10638 036550 001013          BNE  TS365-10      ;BR TO ERROR HALT ON SEQ ERROR
10639 036552 012706 001000          MOV  #BUFF,SP
10640 036556 012767 036610 141230  MOV  #RETAT,RTRAP4 ;SET UP TO TRAP TO 14
10641 036564 012746 000020          MOV  #20,-(SP)     ;PUSH T BIT
10642 036570 012746 036576          MOV  #.+6,-(SP)    ;PUSH PC
10643 036574 000002          RTI                ;SET T BIT
10644 036576 000240          NOP                ;TRAP HERE
10645 036600 012742 001105          MOV  #1105,-(R2)   ;MOVE TO MAILBOX # ***** 1105 *****
10646 036604 005242          INC  -(R2)         ;SET MSGTYP TO FATAL ERROR
10647 036606 000000          HALT               ;TRACE BIT DID NOT TRAP!,OR WRONG $TESTN
10648 036610
10649
10650                                     ;*****
10651                                     ;TEST 365      TEST STACK POINTER DECREMENTS
10652                                     ;*****
10652 036610 005212          TS365:  INC  (R2)          ;UPDATE TEST NUMBER
10653 036612 022712 000365          CMP  #365,(R2)      ;SEQUENCE ERROR?
10654 036616 001022          BNE  TS366-10      ;BR TO ERROR HALT ON SEQ ERROR
10655 036620 012706 001000          MOV  #BUFF,SP
10656 036624 012767 036656 141162  MOV  #RETBT,RTRAP4 ;
10657 036632 012746 000020          MOV  #20,-(SP)     ;PUSH T BIT
10658 036636 012746 036644          MOV  #.+6,-(SP)    ;PUSH PC
10659 036642 000002          RTI                ;SET T BIT
10660 036644 000240          NOP                ;TRAP HERE
10661 036646 012742 001106          MOV  #1106,-(R2)   ;MOVE TO MAILBOX # ***** 1106 *****
10662 036652 005242          INC  -(R2)         ;SET MSGTYP TO FATAL ERROR
10663 036654 000000          HALT               ;TRACE BIT DID NOT TRAP!
10664 036656 020627 000774          RETBT:  CMP  SP,#BUFF-4
10665 036662 001404          BEQ  TS366
10666                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10667                                     ;          CONDITIONAL BRANCH INST. AND <====
10668                                     ;          REPLACE THE MOVE INSTRUCTION <====
10669                                     ;          WHICH FOLLOWS W/ 755 <====
10670 036664 012742 001107          MOV  #1107,-(R2)   ;MOVE TO MAILBOX # ***** 1107 *****
10671 036670 005242          INC  -(R2)         ;SET MSGTYP TO FATAL ERROR
10672 036672 000000          HALT               ;STACK POINTER WAS NOT PUSHED BY TRAP,OR WRONG $TESTN
10673                                     ; OR SEQUENCE ERROR
10674                                     ;*****
10675                                     ;TEST 366      TEST FOR PROPER PC ON STACK
10676                                     ;*****
10677 036674 005212          TS366:  INC  (R2)          ;UPDATE TEST NUMBER
10678 036676 022712 000366          CMP  #366,(R2)      ;SEQUENCE ERROR?
10679 036702 001016          BNE  TS367-10      ;BR TO ERROR HALT ON SEQ ERROR
10680 036704 012706 001000          MOV  #BUFF,SP
10681 036710 012767 036730 141076  MOV  #RETCT,RTRAP4 ;
10682 036716 012746 000020          MOV  #20,-(SP)     ;PUSH T BIT
10683 036722 012746 036730          MOV  #.+6,-(SP)    ;PUSH PC
10684 036726 000002          RTI                ;SET T BIT
10685                                     ;TRAP HERE
10686 036730 022767 036730 142036  RETCT:  CMP  #.,BUFF-4
```

```
10687 036736 001404      BEQ      TS367
10688
10689                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10690                    ;          CONDITIONAL BRANCH INST. AND <====
10691                    ;          REPLACE THE MOVE INSTRUCTION <====
10692                    ;          WHICH FOLLOWS W/ 761 <====
10692 036740 012742 001110  MOV      #1110,-(R2)  ;MOVE TO MAILBOX # ***** 1110 *****
10693 036744 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10694 036746 000000      HALT           ;CORRECT PC WAS NOT SAVED ON STACK,OR WRONG $TESTN
10695
10696
10697
10698
10699                    ;*****
10700                    ;TEST 367      TEST THAT RTT POPS T- BIT
10701                    ;*****
10701 036750 005212      TS367:  INC      (R2)          ;UPDATE TEST NUMBER
10702 036752 022712 000367  CMP      #367,(R2)      ;SEQUENCE ERROR?
10703 036756 001015      BNE     TS370-10      ;BR TO ERROR HALT ON SEQ ERROR
10704
10705 036760 012706 001000      MOV      #BUFF,SP
10706 036764 005001      CLR     R1           ;CLEAR R1
10707 036766 012746 000020      MOV      #20,-(SP)
10708 036772 012746 037006      MOV      #RTT1,-(SP)
10709 036776 012767 037022 141010  MOV      #RTT2,14
10710 037004 000006      RTT:   RTT
10711 037006 000240      NOP
10712 037010 001404      BEQ     TS370
10713
10714                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10715                    ;          CONDITIONAL BRANCH INST. AND <====
10716                    ;          REPLACE THE MOVE INSTRUCTION <====
10717                    ;          WHICH FOLLOWS W/ 762 <====
10717 037012 012742 001111  MOV      #1111,-(R2)  ;MOVE TO MAILBOX # ***** 1111 *****
10718 037016 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10719 037020 000000      HALT           ;T-BIT DID NOT TRAP,OR WRONG $TESTN
10720
10721
10722                    ; OR SEQUENCE ERROR
10722 037022      RTT2:
10723                    ;*****
10724                    ;TEST 370      TEST THAT RTT ALLOWS ONE INST. BEFORE TRAP
10725                    ;*****
10726 037022 005212      TS370:  INC      (R2)          ;UPDATE TEST NUMBER
10727 037024 022712 000370  CMP      #370,(R2)      ;SEQUENCE ERROR?
10728 037030 001030      BNE     TS371-10      ;BR TO ERROR HALT ON SEQ ERROR
10729 037032 012705 177777      MOV      #177777,%5
10730 037036 012706 001000      RTT5:  MOV      #BUFF,SP
10731 037042 012746 000020      MOV      #20,-(SP)
10732 037046 012746 037064      MOV      #RTT3,-(SP)
10733 037052 012767 037102 140734  MOV      #RTT4,14
10734 037060 005001      CLR     R1           ;CLEAR R0
10735 037062 000006      RTT:   RTT           ;SET T-BIT
10736 037064 005201      RTT3:  INC      R1
10737 037066 005205      INC     %5
10738 037070 001762      BEQ     RTT5         ;DO THIS TEST NO MORE THAN 2 TIMES
10739 037072 012742 001112  MOV      #1112,-(R2)  ;MOVE TO MAILBOX # ***** 1112 *****
10740 037076 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10741 037100 000000      HALT           ;DID NOT TRAP
10742 037102 005301      RTT4:  DEC     R1     ;SEE IF RTT ALLOWS 1 INST.
```

10743 037104 001406  
10744 037106 005205  
10745 037110 001752  
10746  
10747  
10748  
10749  
10750 037112 012742 001113  
10751 037116 005242  
10752 037120 000000  
10753 037122  
10754  
10755  
10756  
10757 037122 005212  
10758 037124 022712 000371  
10759 037130 001022  
10760 037132 012706 001000  
10761 037136 012746 000020  
10762 037142 012746 037160  
10763 037146 012767 037172 140640  
10764 037154 005001  
10765 037156 000002  
10766 037160 005201  
10767 037162 012742 001114  
10768 037166 005242  
10769 037170 000000  
10770 037172 005701  
10771  
10772 037174 001404  
10773  
10774  
10775  
10776  
10777 037176 012742 001115  
10778 037202 005242  
10779 037204 000000  
10780  
10781  
10782  
10783  
10784  
10785 037206 005212  
10786 037210 022712 000372  
10787 037214 001026  
10788  
10789 037216 012706 001000  
10790 037222 012767 037262 140564  
10791 037230 005027 000016  
10792 037234 005027 000022  
10793 037240 012767 037302 140552  
10794 037246 012746 000020  
10795 037252 012746 037260  
10796 037256 000006  
10797 037260 000004  
10798 037262

```
BEQ RTT6
INC Z5 ;DO THIS TEST NO MORE THAN TWO TIMES
BEQ RTT5 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 747 <====
MOV #1113,-(R2) ;MOVE TO MAILBOX # ***** 1113 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RTT DID NOT ALLOW 1 INST.,OR WRONG $TESTN

RTT6:
;*****
;TEST 371 TEST THAT RTI DOES NOT ALLOW 1 INST.
;*****
TS371: INC (R2) ;UPDATE TEST NUMBER
CMP #371,(R2) ;SEQUENCE ERROR?
BNE TS372-10 ;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
INC R1 ;RTI SHOULD NOT ALLOW THIS
MOV #1114,-(R2) ;MOVE TO MAILBOX # ***** 1114 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;T- BIT DID NOT CAUSE TRAP

RTI2: TST R1 ;RTI SHOULD NOT ALLOW 1 INST. BEFORE TRAP
BEQ TS372 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 755 <====
MOV #1115,-(R2) ;MOVE TO MAILBOX # ***** 1115 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RTI DID ALLOW 1 INST. BEFORE TRAP,OR WRONG $TESTN
; OR SEQUENCE ERROR

;*****
;TEST 372 TEST TRAP ON TRAP THAT TRACE BIT TRAPS ARE INHIBITED ON TRAP INST
;*****
TS372: INC (R2) ;UPDATE TEST NUMBER
CMP #372,(R2) ;SEQUENCE ERROR?
BNE BR70 ;BR TO ERROR HALT ON SEQ ERROR
MOV #BUFF,%6
MOV #TRACE,14 ;TRACE TRAP
CLR #16
CLR #22
MOV #TONT1,20 ;IOT TRAP
MOV #20,-(SP) ;PUSH T BIT
MOV #.+6,-(SP) ;PUSH PC
RTT
IOT ;TRAP, NEW CC HAVE TRACE RESET

TRACE:
```



10799 037262 012742 001116  
10800 037266 005242  
10801 037270 000000  
10802 037272  
10803 037272 012742 001117  
10804 037276 005242  
10805 037300 000000  
10806 037302 012767 000016 140504  
10807 037310 012767 000022 140502  
10808  
10809  
10810  
10811 037316 005212  
10812 037320 022712 000373  
10813 037324 001020  
10814 037326 012706 001000  
10815 037332 012767 037356 140454  
10816 037340 005067 140452  
10817 037344 012746 000020  
10818 037350 012746 037356  
10819 037354 000002  
10820 037356 036727 141414 000020  
10821 037364 001004  
10822  
10823  
10824  
10825  
10826 037366  
10827 037366 012742 001120  
10828 037372 005242  
10829 037374 000000  
10830 037376 012767 000016 140410  
10831  
10832  
10833  
10834  
10835  
10836  
10837  
10838  
10839  
10840  
10841  
10842  
10843  
10844  
10845 037404 005212  
10846 037406 022712 000374  
10847 037412 001150  
10848 037414 042737 010000 037464  
10849 037422 032737 000002 000322  
10850 037430 001403  
10851 037432 052737 010000 037464  
10852 037440 005000  
10853 037442 005067 140340  
10854 037446 012767 037554 140330

MOV #1116,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 1116 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TRACE TRAP WAS NOT INHIBITED  
BR70:  
MOV #1117,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 1117 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;WRONG TSTNM,OR WRONG \$TSTNM  
TONT1: MOV #16,14  
MOV #22,20  
;\*\*\*\*\*  
;TEST 373 TEST THAT THE TRACE BIT IS SAVED IN THE STACK  
;\*\*\*\*\*  
TS373: INC (R2) ;UPDATE TEST NUMBER  
CMP #373,(R2) ;SEQUENCE ERROR?  
BNE STP3 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #BUFF,%6 ;SET UP STACK POINTER  
MOV #TRC1,14 ;TRACE TRAP RETURN  
CLR 16  
MOV #20,-(SP) ;SET THE T BIT  
MOV #TRC1,-(SP)  
RTI  
TRC1: BIT BUFF-2,#20 ;CHECK FOR T BIT ON STACK  
BNE STP3D  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
; ; CONDITIONAL BRANCH INST. AND <=====  
; ; REPLACE THE MOVE INSTRUCTION <=====  
; ; WHICH FOLLOWS W/ 757 <=====  
STP3: MOV #1120,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 1120 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;T BIT NOT SAVED ON THE STACK,OR WRONG \$TSTNM  
STP3D: MOV #16,14  
;\*\*\*\*\*  
;THIS ROUTINE TESTS THAT NO LEGAL ADDRESS TRAPS AND THAT AN ILLEGAL  
;ADDRESS TRAPS TO LOCATION 4. THIS WILL RUN ON 30K SYSTEM. BUT IF  
;SWITCH REGISTER BIT 1=0, THEN THE MEMORY FROM 28K-30K IS NOT LOOKED  
;AT, SINCE IT MAY HAVE I/O DEVICES. IF SWR BIT 1=1, THEN THAT AREA IS  
;CHECKED. (IT SHOULD EITHER ALL TRAP OR ALL NOT TRAP). LOC 160000  
;IS NO LONGER GUARANTEED TO TRAP, SINCE IT MAY CONTAIN MEMORY. LOCATION  
;177700 (THE UNIBUS ADDRESS FOR R0 ON OLDER SYSTEMS) IS USED FOR FORCING  
;A TIMEOUT IN THE EVENT THAT THERE WAS NO TIMEOUT FROM 0K-28K OR 30K.  
;THIS ROUTINE TESTS MEMORY UNTIL IT DOES A NXM STOP  
;\*\*\*\*\*  
;TEST 374 TEST NON-EXISTENT ADDRESS TRAPS  
;\*\*\*\*\*  
TS374: INC (R2) ;UPDATE TEST NUMBER  
CMP #374,(R2) ;SEQUENCE ERROR?  
BNE TS375-10 ;BR TO ERROR HALT ON SEQ ERROR  
BIC #10000,%#HICORE ;SET HIGHT CORE LIMIT TO 160000  
BIT #2,%#SSWREG ;CHECK IF BIT 1 IS SET  
BEQ 1\$ ;BRANCH IF IT IS, LEAVE LIMIT=160000  
BIS #10000,%#HICORE ;SET UPPER CORE LIMIT TO 30K (170000)  
1\$: CLR R0  
CLR 6  
MOV #ATRAP,4 ;SET UP ADDRESS TRAP ENTRANCE

```

10855 037454 012706 001000      MOV      #BUFF,SP      ;SET STACK POINTER
10856 037460 105720      NOR:    TSTB      (0)+    ;IF OUTSIDE OF CORE, TRAP TO 4
10857 037462 020027      MICORE: CMP      RO,(PC)+ ;IS POINTER INSIDE 28K (30K) CORE
10858 037464 160000      BLO     NOR           ;MAY BE CHANGED TO 170000 IF 30K
10859 037466 103774      MOV     #ROTRAP,@#4   ;TEST THE REST OF CORE
10860 037470 012737 037512 000004  TSTB    @#177700      ;SET UP NEW VECTOR POINTER
10861 037476 105737 177700      TRPADR: ;SHOULD CAUSE A TRAP
10862 037502
10863 037502 012742 001121      MOV     #1121,-(R2)   ;MOVE TO MAILBOX # ***** 1121 *****
10864 037506 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
10865 037510 000000      HALT                               ;SHOULD HAVE TRAPED
10866
;TRAP TO HERE IF FORCING TRAP BY TESTING 177700
10867 037512 106767 140260      ROTRAP: MFPS  STATUS   ;TEST PSW
10868 037516 005767 140254      TST    STATUS
10869 037522 001404      BEQ    1$
10870
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10871
;          CONDITIONAL BRANCH INST. AND <====
10872
;          REPLACE THE MOVE INSTRUCTION <====
10873
;          WHICH FOLLOWS W/ 733 <====
10874 037524 012742 001122      MOV     #1122,-(R2)   ;MOVE TO MAILBOX # ***** 1122 *****
10875 037530 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
10876 037532 000000      HALT                               ;NEW PSW SHOULD HAVE BEEN ZERO
10877 037534 026727 141234 037502 1$:  CMP    BUFF-4,#TRPADR ;TEST OLD PC AT STACK
10878 037542 001453      BEQ    TRAPB
10879
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10880
;          CONDITIONAL BRANCH INST. AND <====
10881
;          REPLACE THE MOVE INSTRUCTION <====
10882
;          WHICH FOLLOWS W/ 723 <====
10883 037544 012742 001123      MOV     #1123,-(R2)   ;MOVE TO MAILBOX # ***** 1123 *****
10884 037550 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
10885 037552 000000      HALT                               ;OLD PC WAS NOT SAVED
10886
;RETURN HERE ON AN ADDRESS TRAP FROM MEMORY BELOW 28K (OR 30K)
10887 037554 005300      ATRAP: DEC    RO
10888 037556 010067 000032      MOV    RO,CORH       ;MOVE THE FIRST NXM LOCATION IN CORH
10889
;THIS ROUTINE DOES NXM TRAPS UNTIL IT FINDS AN EXISTENT MEMORY LOCATION
10890 037562 013700 037464      MOV    @#MICORE,RO   ;SET UP THE HIGHEST MEM LOCATION
10891 037566 005300      DEC    RO            ;MAKE 1 LESS THAN THE HIGHEST CORE BOUNDARY
10892 037570 000402      BR     NOSUB         ;DON'T SUBTRACT 1K FIRST TIME
10893 037572 162700 001000      CTRAP: SUB   #1000,RO ;SUBTRACT 1K OCTAL BYTE FROM ADDRESS
10894
;TO SPEED UP TESTING
10895 037576 012767 037630 140200  NOSUB: MOV    #BTRAP,4  ;SET UP THE VECTOR
10896 037604 012706 001000      MOV    #BUFF,SP
10897 037610 005710      TST    (RO)         ;DOES THIS MEMORY EXIST?
10898
;IF NXM, TRAP TO BTRAP
10899 037612 020027      DTRAP1: CMP   RO,(PC)+ ;IF EXISTS, IS THIS THE SAME TRAP THAT CAUSED
10900
;TRAP TO ATRAP
10901 037614 000000      CORH:  .WORD  0
10902 037616 101425      BLOS  TRAPB
10903
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10904
;          CONDITIONAL BRANCH INST. AND <====
10905
;          REPLACE THE MOVE INSTRUCTION <====
10906
;          WHICH FOLLOWS W/ 675 <====
10907 037620 012742 001124      MOV     #1124,-(R2)   ;MOVE TO MAILBOX # ***** 1124 *****
10908 037624 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
10909 037626 000000      HALT                               ;CONTENTS OF RO SHOULD BE LESS THAN OR EQUAL TO CORH
10910
;IF THIS COMPARISON FAILS IT MEANS

```

```
10911                                     ;THAT SOME LEGAL ADDRESS TRAPPED, OR
10912                                     ;THAT AN ILLEGAL ADDRESS DID NOT TRAP
10913 037630 106767 140142      BTRAP: MFPS      STATUS
10914 037634 005767 140136      TST        STATUS
10915 037640 001404              BEQ        1$
10916                                     ;
10917                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10918                                     ;          CONDITIONAL BRANCH INST. AND <====
10919                                     ;          REPLACE THE MOVE INSTRUCTION <====
10920                                     ;          WHICH FOLLOWS W/ 664 <====
10920 037642 012742 001125      MOV        #1125,-(R2) ;MOVE TO MAILBOX # ***** 1125 *****
10921 037646 005242              INC        -(R2)      ;SET MSGTYP TO FATAL ERROR
10922 037650 000000              HALT
10923 037652 026727 141116 037612 1$: CMP        BUFF-4,#DTRAP1 ;NEW PSW SHOULD HAVE BEEN ZERO
10924 037660 001744              BEQ        CTRAP      ;CHECK IF TRAP PC IS OK
10925                                     ;
10926                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10927                                     ;          CONDITIONAL BRANCH INST. AND <====
10928                                     ;          REPLACE THE MOVE INSTRUCTION <====
10929                                     ;          WHICH FOLLOWS W/ 654 <====
10929 037662                      AUTO1:
10930 037662 012742 001126      MOV        #1126,-(R2) ;MOVE TO MAILBOX # ***** 1126 *****
10931 037666 005242              INC        -(R2)      ;SET MSGTYP TO FATAL ERROR
10932 037670 000000              HALT
10933 037672 012767 000006 140104 TRAPB: MOV        #6,4 ;OLD PC WAS NOT SAVED OR WRONG $TESTN
10934 037700 005067 140102      CLR        6 ;RESET TRAP CATCHER
10935
10936
10937                                     ;THIS ROUTINE WILL FIGURE OUT IF YOU HAVE A DL11W
10938 037704 005067 000020      CLR        PROFTE
10939 037710 012706 001000      MOV        #BUFF,SP ;SET UP THE STACK POINTER
10940 037714 012767 037732 140062 MOV        #DL11W,4 ;SET UP THE TRAP VECTOR
10941 037722 005767 137636      TST        TPS ;TEST THE PUNCH STATUS REGISTER
10942 037726 000403              BR         DL11W1
10943 037730 000000      PROFTE: 000000
10944 037732 005267 177772      DL11W: INC        PROFTE ;INCR IF NO DL11W
10945 037736 012767 000006 140040 DL11W1: MOV        #6,4
10946
10947 037744                      SKP104:
10948                                     ;*****
10949                                     ;TEST 375 TEST THAT A TTY INTERRUPT CAUSES AN OVERFLOW TRAP
10950                                     ;*****
10951 037744 005212              TS375: INC        (R2) ;UPDATE TEST NUMBER
10952 037746 022712 000375      CMP        #375,(R2) ;SEQUENCE ERROR?
10953 037752 001037              BNE        TDEC8 ;BR TO ERROR HALT ON SEQ ERROR
10954 037754 005767 177750      TST        PROFTE
10955 037760 001047              BNE        R7TRX
10956 037762 122767 000001 140330 CMPB       #APTENV,$ENV ;RUNING IN APT MODE?
10957 037770 001003              BNE        2$ ;IF NOT, DO THIS TEST
10958 037772 005767 140310      TST        $PASS ;IS THIS THE FIRST PASS?
10959 037776 001040              BNE        R7TRX ;IF NOT FIRST PASS, SKIP TEST
10960 040000                      2$:
10961 040000 000005              RESET
10962 040002 012767 000340 137766 MOV        #340,STATUS ;LOCK OUT INTERRUPT
10963 040010 012706 000400      MOV        #400,$6 ;SET UP STACK TO OVERFLOW
10964 040014 012767 040062 137762 MOV        #TDEC77,4 ;SET UP OVERFLOW TRAP
10965 040022 012767 040052 140034 MOV        #TDEC8,64 ;SET UP INTERRUPT VECTOR
10966 040030 012767 000100 137526 MOV        #100,TTCSR ;SET INTERRUPT ENABLE
```

10967 040036 005067 137734  
10968 040042 012742 001127  
10969 040046 005242  
10970 040050 000000  
10971 040052  
10972 040052 012742 001130  
10973 040056 005242  
10974 040060 000000  
10975 040062 005067 137476  
10976 040066 012767 000006 137710  
10977 040074 005067 137706  
10978 040100  
10979  
10980  
10981  
10982 040100 005212  
10983 040102 022712 000376  
10984 040106 001045  
10985 040110 005767 177614  
10986 040114 001053  
10987 040116 122767 000001 140174  
10988 040124 001003  
10989 040126 005767 140154  
10990 040132 001044  
10991 040134  
10992 040134 012706 001000  
10993 040140 012767 000340 137630  
10994 040146 012767 040212 137710  
10995 040154 012767 000100 137402  
10996 040162 012767 040222 137644  
10997 040170 012767 040232 137666  
10998 040176 012767 000340 137632  
10999 040204 005067 137566  
11000 040210 104400  
11001 040212  
11002 040212 012742 001131  
11003 040216 005242  
11004 040220 000000  
11005 040222  
11006 040222 012742 001132  
11007 040226 005242  
11008 040230 000000  
11009 040232 005067 137600  
11010 040236 042767 000100 137320  
11011 040244  
11012  
11013  
11014  
11015 040244 005212  
11016 040246 022712 000377  
11017 040252 001043  
11018 040254 005767 177450  
11019 040260 001063  
11020 040262 122767 000001 140030  
11021 040270 001003  
11022 040272 005767 140010

CLR STATUS ;ALLOW INTERRUPT TO OCCUR  
MOV #1127,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 1127 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;NO INTERRUPT OCCURRED  
TDEC8:  
MOV #1130,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 1130 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;OVERFLOW TRAP DID NOT OCCUR OR WRONG \$STNM  
TDEC77: CLR TTCSR ;CLEAR INTERRUPT ENABLE  
MOV #6,4  
CLR 6  
R7TRX:  
;\*\*\*\*\*  
;TEST 376 TEST THAT A PENDING INTERRUPT OCCURS BEFORE TRAP  
;\*\*\*\*\*  
T376: INC (R2) ;UPDATE TEST NUMBER  
CMP #376,(R2) ;SEQUENCE ERROR?  
BNE BR71 ;BR TO ERROR HALT ON SEQ ERROR  
TST PROFTE  
BNE NODL  
CMPB #APTENV,\$ENV ;RUNING IN APT MODE?  
BNE 2\$ ;IF NOT, DO THIS TEST  
TST \$PASS ;IS THIS THE FIRST PASS?  
BNE NODL ;IF NOT FIRST PASS, SKIP TEST  
2\$:  
MOV #BUFF,%6  
MOV #340,STATUS ;SET TO A HIGH PRIORITY LEVEL  
MOV #TR0,%6  
MOV #100,TTCSR ;INTERRUPT FOR TTY PUNCH/PRINTER  
MOV #BR71,%6 ;TRAP VECTOR  
MOV #TR2,%6 ;TTY VECTOR  
MOV #340,%6 ;IF TRAP TRAPS, MOVE 340 TO PRIORITY  
CLR STATUS ;SHOULD INTERRUPT AT END OF CLR INST  
TRAP ;TTY INTERRUPT SHOULD OVERRIDE TRAP  
TR0:  
MOV #1131,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 1131 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TTY SHOULDN'T HAVE INTERRUPTED  
BR71:  
MOV #1132,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 1132 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;TRAP OCCURRED FIRST,OR WRONG \$STNM  
TR2: CLR 36  
BIC #100,TTCSR  
NODL:  
;\*\*\*\*\*  
;TEST 377 TEST THAT A PENDING INTERRUPT, INTERRUPTS BETWEEN TRAPS  
;\*\*\*\*\*  
T377: INC (R2) ;UPDATE TEST NUMBER  
CMP #377,(R2) ;SEQUENCE ERROR?  
BNE TR5 ;BR TO ERROR HALT ON SEQ ERROR  
TST PROFTE  
BNE NODL1  
CMPB #APTENV,\$ENV ;RUNING IN APT MODE?  
BNE 2\$ ;IF NOT, DO THIS TEST  
TST \$PASS ;IS THIS THE FIRST PASS?

```
11023 040276 001054          BNE      MODL1          ;IF NOT FIRST PASS, SKIP TEST
11024 040300
11025 040300 012706 001000    2$:      MOV      #BUFF,26
11026 040304 012767 000340 137464    MOV      #340,STATUS
11027 040312 012767 000100 137244    MOV      #100,TTCSR
11028 040320 012767 040360 137506    MOV      #TR3,34          ;TRAP
11029 040326 012767 040372 137530    MOV      #TR4,64          ;TTY OUTPUT
11030 040334 012767 000340 137524    MOV      #340,66          ;TTY OUTPUT PRIORITY
11031 040342 012767 040362 137450    MOV      #TR5,20          ;IOT
11032 040350 012767 000340 137444    MOV      #340,22          ;IOT PRIORITY
11033 040356 104400
11034 040360 000004          TRAP          ;THE ACT OF TRAPPING LOWER PRIORITY
11035 040362          TR3:      IOT          ;INTERRUPT SHOULD OCCUR IN PLACE OF IOT TRAP
11036 040362 012742 001133    TR5:      MOV      #1133,-(R2) ;MOVE TO MAILBOX # ***** 1133 *****
11037 040366 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
11038 040370 000000          HALT          ;NO INTERRUPT BETWEEN TRAPS,OR WRONG $TSTM
11039 040372 005067 137424    TR4:      CLR      22          ;CLR IOT PRIORITY
11040 040376 005067 137464          CLR      66
11041 040402 012767 000036 137424    MOV      #36,34
11042 040410 012767 000066 137446    MOV      #66,64
11043 040416 012767 000022 137374    MOV      #22,20
11044 040424 005067 137134          CLR      TTCSR
11045 040430
11046
11047
11048          ;*****
11049          ;TEST 400      TEST THAT "RESET" GOES TO OUTSIDE WORLD
11050          ;*****
11050 040430 005212          TS400:    INC      (R2)          ;UPDATE TEST NUMBER
11051 040432 022712 000400    CMP      #400,(R2)      ;SEQUENCE ERROR?
11052 040436 001035          BNE      TS401-10      ;BR TO ERROR HALT ON SEQ ERROR
11053 040440 005767 177264    TST     PROFTE
11054 040444 001036          BNE      MODL2
11055 040446 122767 000001 137644    CMPB    #APTENV,$ENV    ;RUNING IN APT MODE?
11056 040454 001003          BNE      2$           ;IF NOT, DO THIS TEST
11057 040456 005767 137624    TST     $PASS          ;IS THIS THE FIRST PASS?
11058 040462 001027          BNE      MODL2          ;IF NOT FIRST PASS, SKIP TEST
11059 040464
11060 040464 012767 000100 137072    2$:      MOV      #100,TTCSR    ;SET INTERRUPT ENABLE
11061 040472 012767 000100 137060    MOV      #100,TRCSR    ;SET INTERRUPT ENABLE
11062 040500 000005          RESET          ;SHOULD CLEAR INTERRUPT ENABLE
11063 040502 032767 000100 137054    BIT     #100,TTCSR    ;TEST FOR CLEAR
11064 040510 001404          BEQ      1$
11065          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
11066          ;          CONDITIONAL BRANCH INST. AND <===
11067          ;          REPLACE THE MOVE INSTRUCTION <===
11068          ;          WHICH FOLLOWS W/ 752 <===
11069 040512 012742 001134          MOV      #1134,-(R2)    ;MOVE TO MAILBOX # ***** 1134 *****
11070 040516 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
11071 040520 000000          HALT          ;RESET FAILED TO CLEAR TTCSR
11072 040522 032767 000100 137030    1$:      BIT     #100,TRCSR    ;TEST FOR CLEAR
11073 040530 001404          BEQ      TS401
11074          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
11075          ;          CONDITIONAL BRANCH INST. AND <===
11076          ;          REPLACE THE MOVE INSTRUCTION <===
11077          ;          WHICH FOLLOWS W/ 742 <===
11078 040532 012742 001135          MOV      #1135,-(R2)    ;MOVE TO MAILBOX # ***** 1135 *****
```

```
11079 040536 005242          INC      -(R2)          ;SEI MSGTYP TO FATAL ERROR
11080 040540 000000          HALT                    ;RESET FAILED TO CLEAR TRCSR,OR WRONG $STMM
11081                                     ; OR SEQUENCE ERROR
11082 040542          MODL2:
11083          ;.....
11084          ;TEST 401      TEST THAT RESET HAS NO EFFECT ON THE TRACE TRAP
11085          ;.....
11086 040542 005212          TS401: INC      (R2)          ;UPDATE TEST NUMBER
11087 040544 022712 000401  CMP      #401,(R2)      ;SEQUENCE ERROR?
11088 040550 001023          BNE     RESET3         ;BR TO ERROR HALT ON SEQ ERROR
11089 040552 122767 000001 137540  CMPB   #APTENV,$ENV    ;RUNING IN APT MODE?
11090 040560 001003          BNE     2$             ;IF NOT, DO THIS TEST
11091 040562 005767 137520  TST    $PASS          ;IS THIS THE FIRST PASS?
11092 040566 001027          BNE     SKTST2        ;IF NOT FIRST PASS, SKIP TEST
11093 040570
11094 040570 012706 001000          2$:  MOV    #BUFF,26     ;SET STACK
11095 040574 012767 040630 137212  MOV    #RESET,14      ;SET UP TRACE VECTOR
11096 040602 012746 000020          MOV    #20,-(R6)     ;SET THE T-BIT ON STACK
11097 040606 012746 040614          MOV    #18,-(R6)     ;MOVE NEW PC ON STACK
11098 040612 000006          RTI
11099 040614 000005          1$:  RESET
11100 040616 000005          RESET
11101 040620
11102 040620 012742 001136          RESE13: MOV   #1136,-(R2)    ;MOVE TO MAILBOX # ***** 1136 *****
11103 040624 005242          INC      -(R2)
11104 040626 000000          HALT                    ;SET MSGTYP TO FATAL ERROR
11105 040630 005067 137142          RESE2: CLR   STATUS    ;TRACE TRAP FAILED,OR WRONG $STMM
11106 040634 005067 137156          CLR    16              ;CLEAR TRACK
11107 040640 012767 000016 137146  MOV    #16,14         ;TRACE STATUS
11108 040646          SKTST2:
11109          ;.....
11110          ;TEST 402      TEST THAT WHEN TTY INTERRUPTS IT POPS NEW STATUS
11111          ;.....
11112 040646 005212          TS402: INC      (R2)          ;UPDATE TEST NUMBER
11113 040650 022712 000402  CMP      #402,(R2)      ;SEQUENCE ERROR?
11114 040654 001057          BNE     TTY11         ;BR TO ERROR HALT ON SEQ ERROR
11115 040656 005767 177046  TST    PROFTE
11116 040662 001062          BNE     NODL3
11117 040664 122767 000001 137426  CMPB   #APTENV,$ENV    ;RUNING IN APT MODE?
11118 040672 001003          BNE     2$             ;IF NOT, DO THIS TEST
11119 040674 005767 137406  TST    $PASS          ;IS THIS THE FIRST PASS?
11120 040700 001053          BNE     NODL3        ;IF NOT FIRST PASS, SKIP TEST
11121 040702
11122 040702 000005          2$:  RESET
11123 040704 012706 001000          MOV    #BUFF,26     ;SET UP STACK
11124 040710 012767 040734 137146  MOV    #TTY3,64      ;INTERRUPT VECTOR
11125 040716 005067 137054          CLR   STATUS         ;DROP PROCESSOR PRIORITY
11126 040722 012767 000357 137136  MOV    #357,66      ;HIGH PRIORITY ON INTERRUPT
11127 040730 005167 136630          COM   TTCSR         ;SHOULD SET INTERRUPT ENABLE & INTERRUPT
11128 040734 026727 137036 000357  TTY3:  CMP    STATUS,#357
11129 040742 001404          BEQ   1$
11130          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
11131          ;          CONDITIONAL BRANCH INST. AND <===
11132          ;          REPLACE THE MOVE INSTRUCTION <===
11133          ;          WHICH FOLLOWS W/ 744 <===
11134 040744 012742 001137          MOV    #1137,-(R2)  ;MOVE TO MAILBOX # ***** 1137 *****
```

```
11135 040750 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
11136 040752 000000          HALT                    ;INTERRUPT DID NOT POP CORRECT STATUS
11137 040754 000005          1$:  RESET              ;CLR INTERRUPT ENABLE
11138 040756 012706 001000    MOV      #BUFF,26      ;STACK SET UP
11139 040762 012767 041006 137074  MOV      #TTY4,64      ;INTERRUPT VECTOR
11140 040770 005067 137072    CLR      66            ;CLR NEW STATUS
11141 040774 012767 000157 136774  MOV      #157,STATUS   ;PROCESSOR STATUS
11142 041002 005167 136556    COM      TTCR          ;SET INTERRUPT ENABLE
11143 041006 005767 136764    TTY4:  TST      STATUS
11144 041012 001404          BEQ      TTT37
11145                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11146                          ;          CONDITIONAL BRANCH INST. AND <====
11147                          ;          REPLACE THE MOVE INSTRUCTION <====
11148                          ;          WHICH FOLLOWS W/ 720 <====
11149 041014          TTY11:
11150 041014 012742 001140    MOV      #1140,-(R2)   ;MOVE TO MAILBOX # ***** 1140 *****
11151 041020 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
11152 041022 000000          HALT                    ;INCORRECT STATUS,OR WRONG $TSTNM
11153 041024 005067 136534    TTT37: CLR      TTCR
11154 041030          NODL3:
11155
11156          ;*****
11157          ;TEST 403      TEST THE 'WAIT' INSTRUCTION
11158          ;*****
11159 041030 005212          TS403: INC      (R2)          ;UPDATE TEST NUMBER
11160 041032 022712 000403    CMP      #403,(R2)     ;SEQUENCE ERROR?
11161 041036 001062          BNE      STP4          ;BR TO ERROR HALT ON SEQ ERROR
11162 041040 122767 000001 137252  CMPB    #APTENV,$ENV   ;RUNING IN APT MODE?
11163 041046 001003          BNE      1$           ;IF NOT, DO THIS TEST
11164 041050 005767 137232    TST      $PASS        ;IS THIS THE FIRST PASS?
11165 041054 001057          BNE      STP4E        ;IF NOT FIRST PASS, SKIP TEST
11166 041056          1$:
11167 041056 042767 000100 136500  BIC      #100,TPS      ;CLEAR INTERRUPT ENABLE
11168 041064 012706 001000    MOV      #BUFF,SP     ;SET UP THE STACK
11169 041070 012767 041156 136766  MOV      #WATE,64     ;SET UP THE INTERRUPT VECTOR
11170 041076 005067 136764    CLR      66
11171 041102 105767 136456    WATE1: TSTB    TPS     ;WAIT FOR READY
11172 041106 100375          BPL      WATE1        ;TO BE UP
11173 041110 012767 000015 136450  MOV      #15,TPB      ;DO A CARRIAGE RETURN
11174 041116 105767 136442    WATE2: TSTB    TPS     ;WAIT FOR READY TO COME UP
11175 041122 100375          BPL      WATE2
11176 041124 012767 000015 136434  MOV      #15,TPB      ;DO ANOTHER CARRIAGE RETURN
11177 041132 052767 000100 136424  BIS      #100,TPS     ;SET THE INTERRUPT ENABLE
11178 041140 005067 136632    CLR      STATUS      ;CLEAR THE PSW
11179 041144 000001          WATE3: WAIT          ;WAIT FOR THE INTERRUPT
11180 041146 012742 001141    MOV      #1141,-(R2)   ;MOVE TO MAILBOX # ***** 1141 *****
11181 041152 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
11182 041154 000000          HALT                    ;WAIT INSTRUCTION DID NOT LOOP
11183 041156 005767 136614    WATE:  TST      STATUS ;IS THE PSW CORRECT?
11184 041162 001404          BEQ      1$
11185                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11186                          ;          CONDITIONAL BRANCH INST. AND <====
11187                          ;          REPLACE THE MOVE INSTRUCTION <====
11188                          ;          WHICH FOLLOWS W/ 725 <====
11189 041164 012742 001142    MOV      #1142,-(R2)   ;MOVE TO MAILBOX # ***** 1142 *****
11190 041170 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
```





```
11247                                     :           REPLACE THE MOVE INSTRUCTION  <====
11248                                     :           WHICH FOLLOWS W/ 764           <====
11249 041330                               RETR2:
11250 041330 012742 001146                 MOV    #1146,-(R2)   ;MOVE TO MAILBOX # ***** 1146 *****
11251 041334 005242                       INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
11252 041336 000000                       HALT                ;LOC 0 DID NOT STORE -1,OR ODD ADDR REFERENCE CAUSE TRAP
11253                                     :           OR SEQUENCE ERROR
11254
11255                                     ;*****
11256                                     ;USING ADDRESS 177700 IN MODE 2, CAUSES BUS ERROR, BUT
11257                                     ;THE REGISTER IN USE WILL BE INCREMENTED.
11258
11259                                     ;*****
11260                                     ;TEST 406      TEST THAT IN MODE 2, BAD ADDRESS REFERENCE CAUSES BUS ERROR.
11261                                     ;*****
11262
11263 041340 005212                               TS406: INC    (R2)           ;UPDATE TEST NUMBER
11264 041342 022712 000406                   CMP    #406,(R2)    ;SEQUENCE ERROR?
11265 041346 001016                           BNE    TS407-10     ;BR TO ERROR HALT ON SEQ ERROR
11266 041350 012737 041376 000004           MOV    #RETR3,@#RTRAPS ;SET TRAP RETURN ADDR
11267 041356 012700 177700                   MOV    #177700,R0   ;STORES BAD MEMORY REFERENCE
11268 041362 012720 001234                   MOV    #1234,(R0)+  ;BAD ADDR REFERENCE, TRAP TO LOC 4
11269 041366 012742 001147                   MOV    #1147,-(R2)  ;MOVE TO MAILBOX # ***** 1147 *****
11270 041372 005242                           INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
11271 041374 000000                           HALT                ;ADDRESSING 177700 DID NOT CAUSE TRAP
11272 041376 022700 177702                   RETR3: CMP    #177702,R0 ;WAS R0 INCREMENTED?
11273 041402 001404                           BEQ    TS407
11274                                     :           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
11275                                     :           ;           CONDITIONAL BRANCH INST. AND  <====
11276                                     :           ;           REPLACE THE MOVE INSTRUCTION  <====
11277                                     :           ;           WHICH FOLLOWS W/ 761           <====
11278 041404 012742 001150                   MOV    #1150,-(R2)  ;MOVE TO MAILBOX # ***** 1150 *****
11279 041410 005242                           INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
11280 041412 000000                           HALT                ;R0 WAS NOT INCREMENTED
11281                                     :           ; OR SEQUENCE ERROR
11282                                     ;*****
11283                                     ;
11284                                     ;AFTER THE FIRST BUS ERROR WAS ENCOUNTERED, AN ATTEMPT WAS MADE
11285                                     ;TO PUSH PC AND PS INTO THE STACK. HOWEVER, IF THE STACK POINTER
11286                                     ;WAS BAD, A DOUBLE BUS ERROR OCCURED. THE STACK POINTER WOULD
11287                                     ;THEN BE SET TO LOCATION 4, OLD PC AND PS WERE PUSHED INTO
11288                                     ;LOCATIONS 0 AND 2. THE PROCESSOR WOULD TRAP TO 4 AND CONTINUE
11289                                     ;EXECUTION.
11290
11291                                     ;*****
11292                                     ;TEST 407      TEST FOR DOUBLE BUS ERROR.
11293                                     ;*****
11294 041414 005212                               TS407: INC    (R2)           ;UPDATE TEST NUMBER
11295 041416 022712 000407                   CMP    #407,(R2)    ;SEQUENCE ERROR?
11296 041422 001054                           BNE    TS410-10     ;BR TO ERROR HALT ON SEQ ERROR
11297 041424 012737 041502 000004           MOV    #DBE1,@#RTRAPS ;SET TRAP RETURN ADDR
11298 041432 012737 000340 000006           MOV    #340,@#6     ;SET UP PS
11299 041440 012737 041472 000010           MOV    #DBE2,@#RTRAP ;SET TRAP RETURN ADDR
11300 041446 012737 000340 000012           MOV    #340,@#12    ;SET UP PS
11301 041454 012706 177700                   MOV    #177700,SP   ;SET ILLEGAL SP
11302 041460 000077                           DBE:   TRAPA        ;ILLEGAL INSTRUCTION
```

```

11303 041462 012742 001151      MOV      #1151,-(R2)      ;MOVE TO MAILBOX # ***** 1151 *****
11304 041466 005242              INC      -(R2)           ;SET MSGTYP TO FATAL ERROR
11305 041470 000000              HALT                    ;DOUBLE BUS ERROR DID NOT CAUSE TRAP
11306 041472              DBE2:
11307 041472 012742 001152      MOV      #1152,-(R2)      ;MOVE TO MAILBOX # ***** 1152 *****
11308 041476 005242              INC      -(R2)           ;SET MSGTYP TO FATAL ERROR
11309 041500 000000              HALT                    ;TRAP TO WRONG LOCATION
11310 041502 022737 041462 000000 DBE1:  CMP      #DBE+2,@#0      ;OLD PC GOT SAVED?
11311 041510 001404              BEQ      DBE3
11312              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11313              ;          CONDITIONAL BRANCH INST. AND <====
11314              ;          REPLACE THE MOVE INSTRUCTION <====
11315              ;          WHICH FOLLOWS W/ 744 <====
11316 041512 012742 001153      MOV      #1153,-(R2)      ;MOVE TO MAILBOX # ***** 1153 *****
11317 041516 005242              INC      -(R2)           ;SET MSGTYP TO FATAL ERROR
11318 041520 000000              HALT                    ;OLD PC DID NOT GET SAVEDD
11319 041522 022737 000340 000002 DBE3:  CMP      #340,@#2       ;CORRECT PS SAVED?
11320 041530 001404              BEQ      DBE4
11321              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11322              ;          CONDITIONAL BRANCH INST. AND <====
11323              ;          REPLACE THE MOVE INSTRUCTION <====
11324              ;          WHICH FOLLOWS W/ 734 <====
11325 041532 012742 001154      MOV      #1154,-(R2)      ;MOVE TO MAILBOX # ***** 1154 *****
11326 041536 005242              INC      -(R2)           ;SET MSGTYP TO FATAL ERROR
11327 041540 000000              HALT                    ;CORRECT PS DID NOT GET SAVE
11328 041542 022706 000000 DBE4:  CMP      #0,SP          ;SP POINTS TO LOC 0?
11329 041546 001404              BEQ      DBE5
11330              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11331              ;          CONDITIONAL BRANCH INST. AND <====
11332              ;          REPLACE THE MOVE INSTRUCTION <====
11333              ;          WHICH FOLLOWS W/ 725 <====
11334 041550 012742 001155      MOV      #1155,-(R2)      ;MOVE TO MAILBOX # ***** 1155 *****
11335 041554 005242              INC      -(R2)           ;SET MSGTYP TO FATAL ERROR
11336 041556 000000              HALT                    ;SP IS NOT POINTING TO LOC 0
11337 041560 012706 001000 DBE5:  MOV      #STBOT,SP      ;RESET SP
11338
11339
11340 ;*****
11341 ;TEST 410      TEST MFPT
11342 ;*****
11343 041564 005212      TS410: INC      (R2)           ;UPDATE TEST NUMBER
11344 041566 022712 000410      CMP      #410,(R2)       ;SEQUENCE ERROR?
11345 041572 001023      BNE      TS411-10        ;BR TO ERROR HALT ON SEQ ERROR
11346 ;THIS TESTS THE MFPT INSTRUCTION- MOVE FROM PROCESSOR TYPE
11347 ;UPON EXECUTION, R0 WILL RECEIVE THE PROCESSOR MODEL CODE
11348 ;WHICH IS '000003' FOR F11.
11349 MFPT=000007
11350 041574 012706 001000      MOV      #STBOT,R6       ;INIT. SP
11351 041600 013746 000010      MOV      @#10,-(SP)      ;SAVE TRAP VECTOR
11352 041604 012737 041634 000010      MOV      #18,@#10       ;SET UP ILLEGAL INSTRUCTION TRAP
11353 041612 010046              MOV      R0,-(SP)       ;SAVE R0
11354 041614 000007      MFPT                    ;GET PROCESSOR MODEL
11355 041616 010037 041644      MOV      R0,@#CPUTYP     ;STORE IT
11356 041622 012600              MOV      (SP)+,R0       ;RESTORE R0
11357 041624 022737 000003 041644      CMP      #3,@#CPUTYP    ;CHECK MODEL TYPE
11358 041632 001405              BEQ      XXI

```

```
11359
11360
11361
11362
11363 041634
11364 041634 012742 001156
11365 041640 005242
11366 041642 000000
11367
11368 041644 000000
11369 041646
11370 041646 012637 000010
11371
11372
11373
11374
11375
11376
11377
11378
11379
11380 041652 005212
11381 041654 022712 000411
11382 041660 001022
11383
11384 041662 012706 001000
11385 041666 012737 041706 000010
11386 041674 012737 000340 000012
11387 041702 075006
11388 041704 000000
11389 041706 012737 041720 000010
11390 041714 076000
11391 041716 000000
11392 041720 012737 041732 000010
11393 041726 076700
11394 041730 000000
11395 041732 012706 001000
11396
11397
11398
11399
11400 041736 005212
11401 041740 022712 000412
11402 041744 001112
11403 041746 042767 000100 135610
11404 041754 012737 042002 000244
11405 041762 013767 000010 000024
11406 041770 012737 042012 000010
11407 041776 170007
11408 042000 000406
11409 042002
11410 042002 013767 042234 000234
11411 042010 000002
11412 042012
11413 042012 000002
11414 042014 000000
```

18:           MOV     #1156,-(R2)     ;MOVE TO MAILBOX # \*\*\*\*\* 1156 \*\*\*\*\*  
              INC     -(R2)       ;SET MSGTYP TO FATAL ERROR  
              HALT                 ;ILLEGAL INSTR TRAP OR WRONG MODEL TYPE

CPUTYP: .WORD 0  
XXT:           MOV     (SP)+,@#10     ;RESTORE TRAP VECTOR

\*\*\*\*\*  
;THIS TEST WILL CHECK THE SERVICE ROUTINE FOR A CONTROL CHIP ERROR.  
;THIS IS DONE BY EXECUTING INSTRUCTIONS WHICH JUMP TO NON-EXISTENT  
;CONTROL-CHIP. ON THE KDF11-A, THIS INCLUDES: FIS(CTL3), CIS AND WCS  
;INSTRUCTIONS. A CTLERR TRAPS TO LOCATION 10.  
;THE RESET LINE IS ALSO ASSERTED FOR 1 CYCLE.  
\*\*\*\*\*

TEST 411     TEST CTLERR SERVICE ROUTINE  
\*\*\*\*\*

TS411:    INC     (R2)             ;UPDATE TEST NUMBER  
          CMP     #411,(R2)       ;SEQUENCE ERROR?  
          BNE     TS412-10       ;BR TO ERROR HALT ON SEQ ERROR

          MOV     #STBOT,R6       ;INIT STACK POINTER  
          MOV     #18,@#10       ;SET UP RETURN ADDR FROM TRAP  
          MOV     #340,@#12       ;SET TRAP PRIORITY=7  
          FADD    R6             ;EXECUTE FIS INSTR..SHOULD CAUSE CTLERR  
          HALT                   ;DID NOT TRAP..CHECK CSEL LINE  
1\$:       MOV     #28,@#10       ;SET UP RETURN ADDR FROM TRAP  
          76000                 ;EXECUTE CIS INSTR..SHOULD TRAP  
          HALT                   ;DID NOT TRAP  
2\$:       MOV     #38,@#10       ;SET UP RETURN ADDR FROM TRAP  
          76700                 ;EXECUTE WCS INSTR..SHOULD CAUSE CTLERR AND TRAP  
          HALT                   ;DID NOT TRAP  
3\$:       MOV     #STBOT,R6       ;RE-INIT STACK POINTER

\*\*\*\*\*  
;TEST 412     TEST THAT ALL RESERVED INSTRUCTIONS TRAP  
\*\*\*\*\*

TS412:    INC     (R2)             ;UPDATE TEST NUMBER  
          CMP     #412,(R2)       ;SEQUENCE ERROR?  
          BNE     RET4            ;BR TO ERROR HALT ON SEQ ERROR  
          BIC     #100,TPS  
          MOV     #TRAP244,@#244   ; SET UP TO SEE IF  
          MOV     @#10,TENSAVE   ; THIS PROCESSOR HAS THE  
          MOV     #TRAP10,@#10   ; FLOATING POINT OPTION  
          .WORD  170007           ; AN ILLEGAL FPP INSTRUCTION  
          BR     AROUND           ; THE FOLLOWING  
                  ; IF FPP IN--  
TRAP244:   MOV     @#FPP,FINISH   ; RESET END OF TABLE POINTER  
          RTI                    ; AND RETURN  
TRAP10:    RTI                    ; LEAVE THE TABLE ALONE  
          AND RETURN             ; AND RETURN  
TENSAVE: .WORD  0                 ; A PLACE TO STORE CONTENTS OF 10

```

11415
11416 042016
11417 042016 012737 000246 000244
11418 042024 016737 177764 000010
11419 042032 012703 042214
11420 042036 012305
11421 042040 012301
11422 042042 020567 000176
11423 042046 001415
11424 042050 010567 000172
11425 042054 005267 000166
11426 042060 012767 042126 135722
11427 042066 012706 001000
11428 042072 005067 135700
11429 042076 000167 000144
11430 042102 012737 062350 000034
11431 042110 012737 000340 000036
11432 042116 012700 000370
11433 042122 000167 000234
11434
11435
11436 042126 020627 000774
11437 042132 001404
11438 042134 012742 001157
11439 042140 005242
11440 042142 000000
11441 042144 026727 136624 042250
11442 042152 001404
11443 042154 012742 001160
11444 042160 005242
11445 042162 000000
11446 042164 005767 136606
11447 042170 001404
11448
11449
11450
11451
11452 042172
11453 042172 012742 001161
11454 042176 005242
11455 042200 000000
11456 042202 026701 000040
11457 042206 001713
11458 042210 000167 177640
11459
11460 042214 000007
11461 042216 000077
11462 042220 000207
11463 042222 000227
11464 042224 006777
11465 042226 007777
11466 042230 075037
11467 042232 076777
11468 042234 167777
11469 042236 177700
11470 042240 177716

AROUND:
MOV #246,@#244 ; CONTINUATION POINT
MOV TENSVE,@#10 ; RESTORE THE TRAP VECTOR
MOV #TABLE,TAB ; RESTORE THE ILLEGAL INST. VECTOR
GIN1: MOV (TAB)+,FIRST ; TABLE POINTER
MOV (TAB)+,LAST ; FIRST OR CURRENT INSTRUCTION
CMP FIRST,FINISH ; LAST INSTRUCTION OR GROUP
BEQ GIN3 ; TESTED ALL
MOV FIRST,INST ; YES BRANCH
GIN2: INC INST ; SET UP INST
MOV #RET,10 ; SET UP RETURN FROM TRAP
MOV #BUFF,SP ; SET UP STACK POINTER
CLR CC ; CLEAR PRIORITY
JMP INST ; EXECUTE RESERVED INSTRUCTION
GIN3: MOV #STRAP,@#34 ; TRAP VECTOR FOR TRAP CALL
MOV #340,@#36 ; LEVEL 7
MOV #370,R0 ; RESET RESERVED AREA 370-402
JMP THRPT ; JUMP TO EIS TEST

;TRAPPING SHOULD SEND YOU HERE
RET: CMP SP,#BUFF-4 ; TEST DECREMENT OF SP
BEQ RET1
MOV #1157,-(R2) ; MOVE TO MAILBOX # ***** 1157 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; WRONG DECREMENT
RET1: CMP BUFF-4,#INST+2 ; LOC OF INST UNINCREMENTED
BEQ RET2
MOV #1160,-(R2) ; MOVE TO MAILBOX # ***** 1160 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; INST INC ON TRAP
RET2: TST BUFF-2
BEQ RET3

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 665 <====

RET4: MOV #1161,-(R2) ; MOVE TO MAILBOX # ***** 1161 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; CONDITION CODES SET ON TRAP OR WRONG $STNM
RET3: CMP INST,LAST
BEQ GIN1 ; SET UP NEW GROUP
JMP GIN2 ; FINISH OLD GROUP
; END OF INSTRUCTION GROUP
; END OF OPERATE

TABLE: 7
77
207 ; RTS,RT1,JMP
227
6777
7777
075037
76777
FPP: 167777 ; START OF THE FPP INSTRUCTIONS
177700
177716

```

11471 042242 177777  
11472 042244 042244  
11473 042246 000000  
11474 042250 000000  
11475 042252 000000  
11476 042254 000000  
11477 042256 000000  
11478  
11479 000000  
11480  
11481  
11482  
11483  
11484  
11485  
11486  
11487  
11488  
11489  
11490  
11491  
11492  
11493  
11494  
11495  
11496  
11497  
11498  
11499  
11500  
11501  
11502  
11503  
11504  
11505  
11506  
11507  
11508  
11509  
11510  
11511  
11512  
11513  
11514  
11515  
11516  
11517  
11518  
11519  
11520

177777  
FINISH: . :END FLAG  
INST: HALT :WILL CONTINUE RESERVED INST  
HALT :SHOULD TRAP TO LOC 10  
HALT :LOC 10 SHOULD SEND YOU TO  
HALT :RET  
HALT  
.SBTTL \*\* STARTING OF EIS TEST \*\*  
.REPT 0

.PAGE

PART THREE: EIS INSTRUCTION TESTS

ABSTRACT

THIS PROGRAM TESTS THE F11 EXTENDED INSTRUCTION SET  
<ASH, ASHC, MUL, AND DIV> USING REGISTERS 0-5 AT-  
LEAST ONCE WITH EACH INSTRUCTION.

SWITCH SETTINGS

IF NO HARDWARE SWITCH REGISTER IS AVAILABLE, THE PROGRAM  
AUTOMATICALLY USES THE CONTENTS OF LOC. 176 AS THE SOFTWARE  
SWITCH REGISTER. THE USER SHOULD SET THIS LOCATION BEFORE  
STARTING THE PROGRAM.

| BIT # | OCTAL VALUE | FUNCTION               |
|-------|-------------|------------------------|
| 15    | 100000..... | HALT ON ERROR          |
| 13    | 020000..... | INHIBIT ERROR PRINTOUT |

AN 8 BIT BYTE \$ENVM [I.E. LOCATION 321] HAS BEEN USED TO DEFINE  
THE OPERATING MODE. ALL TYPEOUTS CAN BE SUPPRESSED BY MAKING  
BIT 5 OF BYTE \$ENVM HIGH, IN OTHER WORDS BY PLACING A 20000 IN  
LOCATION 320

.ENDR

.ABS  
.NLIST MD,MC,CND  
.LIST ME

|       |        |        |               |          |                        |
|-------|--------|--------|---------------|----------|------------------------|
| 11521 |        |        |               |          |                        |
| 11522 |        |        |               |          |                        |
| 11523 | 000000 |        |               | DUMMY=   | 0                      |
| 11524 | 000001 |        |               | ERRNM=   | 1                      |
| 11525 | 000051 |        |               | F=       | 51                     |
| 11526 | 000176 |        |               | N=       | 176                    |
| 11527 | 001000 |        |               | SW09=    | 1000                   |
| 11528 | 002000 |        |               | SW10=    | 2000                   |
| 11529 | 004000 |        |               | SW11=    | 4000                   |
| 11530 | 010000 |        |               | SW12=    | 10000                  |
| 11531 |        |        |               |          |                        |
| 11532 | 042260 |        |               | COUNT:   |                        |
| 11533 |        | 042262 |               | .        | =COUNT+2               |
| 11534 | 042262 |        |               | PSWORD:  |                        |
| 11535 |        | 042264 |               | .        | =PSWORD+2              |
| 11536 | 042264 |        |               | TEMP1:   |                        |
| 11537 |        | 042266 |               | .        | =TEMP1+2               |
| 11538 | 042266 |        |               | TEMP2:   |                        |
| 11539 |        | 042270 |               | .        | =TEMP2+2               |
| 11540 | 042270 |        |               | TEMP3:   |                        |
| 11541 |        | 042272 |               | .        | =TEMP3+2               |
| 11542 | 042272 |        |               | TEMP4:   |                        |
| 11543 |        | 042274 |               | .        | =TEMP4+2               |
| 11544 | 042274 | 000000 |               | TEMP5:   | .WORD                  |
| 11545 | 042276 | 000000 |               | TEMP6:   | .WORD                  |
| 11546 | 042300 | 000    |               | TYPCNT:  | .BYTE                  |
| 11547 | 042301 | 000    |               | \$TPCNT: | .BYTE                  |
| 11548 | 042302 | 000007 |               | S0:      | 7                      |
| 11549 | 042304 | 177771 |               | S1:      | -7                     |
| 11550 | 042306 | 042304 |               | S2:      | S1                     |
| 11551 | 042310 | 177772 |               | S3:      | -6                     |
| 11552 | 042312 | 177777 |               | S4:      | -1                     |
| 11553 | 042314 | 040000 |               | S5:      | 40000                  |
| 11554 | 042316 | 042314 |               | S6:      | S5                     |
| 11555 | 042320 | 040000 |               | S7:      | 40000                  |
| 11556 | 042322 | 177776 |               | S8:      | -2                     |
| 11557 | 042324 | 000002 |               | S9:      | 2                      |
| 11558 | 042326 | 042324 |               | S10:     | S9                     |
| 11559 | 042330 | 000002 |               | S11:     | 2                      |
| 11560 | 042332 | 177570 |               | SWR:     | 177570                 |
| 11561 | 042334 | 177570 |               | DISPLAY: | 177570                 |
| 11562 | 042336 | 000064 |               | TTYOUT:  | 64                     |
| 11563 | 042340 | 177566 |               | \$TPB:   | 177566                 |
| 11564 | 042342 | 177564 |               | \$TPS:   | 177564                 |
| 11565 | 042344 | 005015 | 020040 000040 | \$CRLF:  | .ASCIZ <15><12>/ /     |
| 11566 | 042352 | 006412 | 047520 042527 | POWER:   | .ASCIZ <12><15>/POWER/ |
| 11567 | 042360 | 000122 |               |          |                        |
| 11568 |        |        |               |          |                        |
| 11569 |        |        |               |          |                        |
| 11570 |        |        |               |          |                        |
| 11571 |        |        |               |          |                        |
| 11572 |        |        |               |          |                        |
| 11573 |        |        |               |          |                        |
| 11574 |        |        |               |          |                        |
| 11575 |        |        |               |          |                        |
| 11576 |        |        |               |          |                        |

MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79<sup>J 2</sup> 11:31 PAGE 229  
\*\* STARTING OF EIS TEST \*\*

SEQ 0229

11577

```

11578 042362
11579
11580 042362 012705 000304
11581 042366 005037 042260
11582 042372 012715 000001
11583 042376 012706 001000
11584 042402 013746 000004
11585 042406 013746 000006
11586 042412 012767 042426 135364
11587 042420 005777 177706
11588 042424 000407
11589 042426 012767 000176 177676 1$:
11590 042434 012767 000174 177672
11591 042442 022626
11592 042444 012637 000006 3$:
11593 042450 012637 000004
11594 042454 106427 000000
11595 042460 132737 000001 000320
11596 042466 001403
11597 042470 012767 000322 177634
11598 042476 012737 000001 042264 2$:
11599 042504 005037 042266
11600 042510 012737 000001 042270
11601 042516 005037 042272
11602
11603
  
```

THRPT:

```

MOV #STESTN,R5 ;MAKE R5 POINT TO THE LOCATION $STESTN
CLR @#COUNT ;CLEAR THE COUNTER
MOV #1,(R5) ;INITIALIZE TEST NUMBER
MOV #STBOT,SP ;** STACK AT STBOT **
MOV @#4,-(SP) ;SAVE ERROR VECTOR
MOV @#6,-(SP)
MOV #1$,4 ;SET UP TIMEOUT VECTOR
TST @SWR ;TRY TO REFERENCE HARDWRE SWR
BR 3$ ;BRANCH IF NO TIMEOUT TRAP OCCURS
MOV #SWREG,SWR ;POINT TO SOFTWARE SWR
MOV #DISPREG,DISPLAY ;POINT TO SOFTWARE DISPLAY REG
CMP (SP)+,(SP)+ ;RESTORE STACK
MOV (SP)+,@#6 ;RESTORE ERROR VECTOR
MOV (SP)+,@#4
MTPS #0 ;PLACE #0 IN PSW
BITB #1,@#SENV ;ARE WE UNDER APT ?
BEQ 2$ ;IF NOT THEN GO TO 2$
MOV #SSWREG,SWR ;USE APT SWITCH REGISTER
MOV #1,@#TEMP1 ;TEMP1=1
CLR @#TEMP2 ;TEMP2=0
MOV #1,@#TEMP3 ;TEMP3=1
CLR @#TEMP4 ;TEMP4=0
  
```



11604  
11605  
11606  
11607  
11608  
11609  
11610  
11611  
11612  
11613  
11614  
11615  
11616  
11617  
11618  
11619  
11620  
11621  
11622  
11623  
11624  
11625  
11626  
11627  
11628  
11629  
11630  
11631  
11632  
11633  
11634  
11635  
11636  
11637  
11638  
11639  
11640  
11641  
11642  
11643  
11644  
11645  
11646  
11647  
11648  
11649  
11650  
11651  
11652  
11653  
11654  
11655  
11656  
11657  
11658  
11659

042522 010767 135444  
042526 013700 042264  
042532 032737 000001 000306  
042540 001004  
042542 013701 042266  
042546 072001  
042550 000402  
042552 072067 177510  
042556 106737 042262  
042562 123737 042272 042262  
042570 001403  
042572 004767 016044  
042576 000001  
042600 005237 042260  
042604 023700 042270  
042610 001403  
042612  
042612 004767 016024  
042616 000002  
042620 021537 042260  
042624 001372  
042626 005215  
042630 010767 135336  
042634 021527 000037  
042640 002011  
042642 005237 042266  
042646 006367 177416  
042652 021527 000020  
042656 001004  
042660 000167 001010  
042664 004767 001032  
042670 010767 135276  
042674 013701 042264  
042700 032737 000001 000306

```

:*****
:      ASH INSTRUCTION TESTS
:*****

:*****
:      TESTS 1-36
:*****

ASTART: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV    @#TEMP1,%0    ;LOAD R0 WITH THE CONTENTS OF TEMP1
        BIT    #1,@#SPASS    ;IS IT AN EVEN PASS ?
        BNE    2$           ;IF NOT THEN GO TO 2$
        MOV    @#TEMP2,R1    ;OTHERWISE EXECUTE THE INSTRUCTION
                                ;IN MODE 0 USING R1
        ASH    R1,R0
        BR     4$
2$:     ASH    TEMP2,%0      ;SHIFT R0 BY THE NUMBER SPECIFIED BY TEMP2
4$:     MFPS   @#PSWORD     ;SAVE PS
        CMPB  @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
        BEQ   .+10
        JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;THE PS IS NOT EQUAL TO 0
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)
        1
        INC   @#COUNT      ;INCREMENT THE COUNTER
        CMP   @#TEMP3,%0    ;IS THE RESULT IN R0 EQUAL TO TEMP3?
        BEQ   .+10
6$:     JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;EITHER INCORRECT R0 OR INCORRECT SEQUENCE
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)
        2
        CMP   (R5),@#COUNT ;IS THE TEST NUMBER EQUAL TO THE
                                ;COUNTER?
        BNE   6$           ;IF NOT GO TO THE HLT ABOVE
        INC   (R5)
        MOV   PC,LPADR      ;STORE ERROR LOOP ADDRESS
        CMP   (R5),#37     ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
                                ;BY 14. AND RIGHT BY 14.?
        BGE   8$
        INC   @#TEMP2
        ASL   TEMP3        ;SHIFT TEMP3 LEFT.
        CMP   (R5),#20     ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
        BNE   REGR1
        JMP   NEGAT        ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
        JSR   PC,TST37     ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
8$:     REGR1: MOV    PC,LPADR ;STORE ERROR LOOP ADDRESS
        MOV    @#TEMP1,%1   ;LOAD R1 WITH THE CONTENTS OF TEMP1
        BIT    #1,@#SPASS   ;IS IT AN EVEN PASS ?
    
```

|       |        |        |               |            |                  |  |
|-------|--------|--------|---------------|------------|------------------|--|
| 11660 | 042706 | 001004 |               | BNE        | 2\$              | :IF NOT THEN GO TO 2\$                                   |
| 11661 | 042710 | 013702 | 042266        | MOV        | @#TEMP2,R2       | :OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0             |
| 11662 | 042714 | 072102 |               | ASH        | R2,R1            | :USING R1  |
| 11663 | 042716 | 000402 |               | BR         | 4\$              |  |
| 11664 | 042720 | 072167 | 177342        | 2\$: ASH   | TEMP2,X1         | :SHIFT R1 BY THE NUMBER SPECIFIED BY TEMP2               |
| 11665 | 042724 | 106737 | 042262        | 4\$: MFPS  | @#PSWORD         | :SAVE PS   |
| 11666 | 042730 | 123737 | 042272 042262 | CMPB       | @#TEMP4,@#PSWORD | :IS THE PS = TEMP4 ?                                     |
| 11667 | 042736 | 001403 |               | BEQ        | +.10             |  |
| 11668 | 042740 | 004767 | 015676        | JSR        | PC,\$HLT         | :SEEN AN ERROR, GO TO THE HALT ROUTINE                   |
| 11669 |        |        |               |            |                  | :THE PS IS NOT EQUAL TO 0                                |
| 11670 | 042744 | 000003 |               | 3          |                  | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)               |
| 11671 |        |        |               |            |                  | :BY (013746 000172 000207)                               |
| 11672 |        |        |               |            |                  |  |
| 11673 | 042746 | 005237 | 042260        | INC        | @#COUNT          | :INCREMENT THE COUNTER                                   |
| 11674 | 042752 | 023701 | 042270        | CMP        | @#TEMP3,X1       | :IS THE RESULT IN R1 EQUAL TO TEMP3?                     |
| 11675 | 042756 | 001403 |               | BEQ        | +.10             |  |
| 11676 | 042760 |        |               | 6\$: JSR   | PC,\$HLT         | :SEEN AN ERROR, GO TO THE HALT ROUTINE                   |
| 11677 | 042760 | 004767 | 015656        |            |                  | :EITHER INCORRECT R1 OR INCORRECT SEQUENCE               |
| 11678 |        |        |               |            |                  | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)               |
| 11679 | 042764 | 000004 |               | 4          |                  | :BY (013746 000172 000207)                               |
| 11680 |        |        |               |            |                  |  |
| 11681 |        |        |               |            |                  |  |
| 11682 | 042766 | 021537 | 042260        | CMP        | (R5),@#COUNT     | :IS THE TEST NUMBER EQUAL TO THE COUNTER?                |
| 11683 | 042772 | 001372 |               | BNE        | 6\$              | :IF NOT GO TO THE HLT ABOVE                              |
| 11684 | 042774 | 005215 |               | INC        | (R5)             |  |
| 11685 | 042776 | 010767 | 135170        | MOV        | PC,LPADR         | :STORE ERROR LOOP ADDRESS                                |
| 11686 | 043002 | 021527 | 000037        | CMP        | (R5),#37         | :HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT         |
| 11687 |        |        |               |            |                  | :BY 14. AND RIGHT BY 14.?                                |
| 11688 | 043006 | 002011 |               | BGE        | 8\$              |  |
| 11689 | 043010 | 005237 | 042266        | INC        | @#TEMP2          |  |
| 11690 | 043014 | 006367 | 177250        | ASL        | TEMP3            | :SHIFT TEMP3 LEFT  |
| 11691 | 043020 | 021527 | 000020        | CMP        | (R5),#20         | :HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.? |
| 11692 | 043024 | 001004 |               | BNE        | REGR2            |  |
| 11693 | 043026 | 000167 | 000642        | JMP        | NEGAT            | :IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT              |
| 11694 | 043032 | 004767 | 000664        | 8\$: JSR   | PC,TST37         | :IF SO GO AND CONTINUE THE REST OF THE PROGRAM           |
| 11695 | 043036 | 010767 | 135130        | REGR2: MOV | PC,LPADR         | :STORE ERROR LOOP ADDRESS                                |
| 11696 | 043042 | 013702 | 042264        | MOV        | @#TEMP1,X2       | :LOAD R2 WITH THE CONTENTS OF TEMP1                      |
| 11697 | 043046 | 032737 | 000001 000306 | BIT        | #1,@#SPASS       | :IS IT AN EVEN PASS ?                                    |
| 11698 | 043054 | 001004 |               | BNE        | 2\$              | :IF NOT THEN GO TO 2\$                                   |
| 11699 | 043056 | 013703 | 042266        | MOV        | @#TEMP2,R3       | :OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0             |
| 11700 | 043062 | 072203 |               | ASH        | R3,R2            | :USING R2  |
| 11701 | 043064 | 000402 |               | BR         | 4\$              |  |
| 11702 | 043066 | 072267 | 177174        | 2\$: ASH   | TEMP2,X2         | :SHIFT R2 BY THE NUMBER SPECIFIED BY TEMP2               |
| 11703 | 043072 | 106737 | 042262        | 4\$: MFPS  | @#PSWORD         | :SAVE PS   |
| 11704 | 043076 | 123737 | 042272 042262 | CMPB       | @#TEMP4,@#PSWORD | :IS THE PS = TEMP4 ?                                     |
| 11705 | 043104 | 001403 |               | BEQ        | +.10             |  |
| 11706 | 043106 | 004767 | 015530        | JSR        | PC,\$HLT         | :SEEN AN ERROR, GO TO THE HALT ROUTINE                   |
| 11707 |        |        |               |            |                  | :THE PS IS NOT EQUAL TO 0                                |
| 11708 | 043112 | 000005 |               | 5          |                  | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)               |
| 11709 |        |        |               |            |                  | :BY (013746 000172 000207)                               |
| 11710 |        |        |               |            |                  |  |
| 11711 | 043114 | 005237 | 042260        | INC        | @#COUNT          |  |
| 11712 | 043120 | 023702 | 042270        | CMP        | @#TEMP3,X2       | :IS THE RESULT IN R2 EQUAL TO TEMP3?                     |
| 11713 | 043124 | 001403 |               | BEQ        | +.10             |  |
| 11714 | 043126 |        |               | 6\$: JSR   | PC,\$HLT         | :SEEN AN ERROR, GO TO THE HALT ROUTINE                   |
| 11715 | 043126 | 004767 | 015510        |            |                  |  |

```

11716                                     ;EITHER INCORRECT R2 OR INCORRECT SEQUENCE
11717 043132 000006                       6 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11718                                     ;BY (013746 000172 000207)
11719
11720 043134 021537 042260               CMP (R5),@#COUNT ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11721 043140 001372                       BNE 6$ ;IF NOT GO TO THE HLT ABOVE
11722 043142 005215                       INC (R5)
11723 043144 010767 135022               MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
11724 043150 021527 000037               CMP (R5),#37 ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11725                                     ;LEFT BY 14, AND RIGHT BY 14.?
11726 043154 002011                       BGE 8$
11727 043156 005237 042266               INC @#TEMP2
11728 043162 006367 177102               ASL TEMP3 ;SHIFTED TEMP3 LEFT
11729 043166 021527 000020               CMP (R5),#20 ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11730 043172 001004                       BNE REGR3
11731 043174 000167 000474               JMP NEGAT ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11732 043200 004767 000516               8$: JSR PC,TST37 ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11733 043204 010767 134762               REGR3: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
11734 043210 013703 042264               MOV @#TEMP1,#3 ;LOAD R3 WITH THE CONTENTS OF TEMP1
11735 043214 032737 000001 000306       BIT #1,@#SPASS ;IS IT AN EVEN PASS ?
11736 043222 001004                       BNE 2$ ;IF NOT THEN GO TO 2$
11737 043224 013704 042266               MOV @#TEMP2,R4 ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11738 043230 072304                       ASH R4,R3 ;USING R3
11739 043232 000402                       BR 4$
11740 043234 072367 177026               2$: ASH TEMP2,#3 ;SHIFT R3 BY THE NUMBER SPECIFIED BY TEMP2
11741 043240 106737 042262               4$: MFPS @#PSWORD ;SAVE PS
11742 043244 123737 042272 042262       CMPB @#TEMP4,@#PSWORD;IS THE PS - TEMP4 ?
11743 043252 001403                       BEQ .+10
11744 043254 004767 015362               JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11745                                     ;THE PS IS NOT EQUAL TO 0.
11746 043260 000007                       7 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11747                                     ;BY (013746 000172 000207)
11748
11749 043262 005237 042260               INC @#COUNT
11750 043266 023703 042270               CMP @#TEMP3,#3 ;IS THE RESULT IN R3 EQUAL TO TEMP3?
11751 043272 001403                       BEQ .+10
11752 043274                                     6$:
11753 043274 004767 015342               JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11754                                     ;EITHER INCORRECT R3 OR INCORRECT SEQUENCE
11755 043300 000010                       10 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11756                                     ;BY (013746 000172 000207)
11757
11758 043302 021537 042260               CMP (R5),@#COUNT ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11759 043306 001372                       BNE 6$ ;IF NOT GO TO THE HLT ABOVE
11760 043310 005215                       INC (R5)
11761 043312 010767 134654               MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
11762 043316 021527 000037               CMP (R5),#37 ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11763                                     ;LEFT BY 14, AND RIGHT BY 14.?
11764 043322 002010                       BGE 8$
11765 043324 005237 042266               INC @#TEMP2
11766 043330 006367 176734               ASL TEMP3 ;SHIFT TEMP3 LEFT?
11767 043334 021527 000020               CMP (R5),#20 ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11768 043340 001003                       BNE REGR4
11769 043342 000554                       BR NEGAT ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11770 043344 004767 000352               8$: JSR PC,TST37 ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11771 043350 010767 134616               REGR4: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS

```

|       |        |        |        |        |      |                  |  |
|-------|--------|--------|--------|--------|------|------------------|--|
| 11772 | 043354 | 013704 | 042264 |        | MOV  | @#TEMP1,%4       | ;LOAD R4 WITH THE CONTENTS OF TEMP1                |
| 11773 | 043360 | 010501 |        |        | MOV  | R5,R1            | ;SAVE R5   |
| 11774 | 043362 | 032737 | 000001 | 000306 | BIT  | #1,@#SPASS       | ;IS IT AN EVEN PASS ?                              |
| 11775 | 043370 | 001004 |        |        | BNE  | 2\$              | ;IF NOT THEN GO TO 2\$                             |
| 11776 | 043372 | 013705 | 042266 |        | MOV  | @#TEMP2,R5       | ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0       |
| 11777 | 043376 | 072405 |        |        | ASH  | R5,R4            | ;USING R4  |
| 11778 | 043400 | 000402 |        |        | BR   | 4\$              |  |
| 11779 | 043402 | 072467 | 176660 |        | ASH  | TEMP2,%4         | ;SHIFT R4 BY THE NUMBER SPECIFIED BY TEMP2         |
| 11780 | 043406 | 106737 | 042262 |        | MFPS | @#PSWORD         | ;SAVE PS   |
| 11781 | 043412 | 123737 | 042272 | 042262 | CMPB | @#TEMP4,@#PSWORD | ;IS PS = TEMP4 ?                                   |
| 11782 | 043420 | 001403 |        |        | BEQ  | +.10             |  |
| 11783 | 043422 | 004767 | 015214 |        | JSR  | PC,\$HLT         | ;SEEN AN ERROR, GO TO THE HALT ROUTINE             |
| 11784 |        |        |        |        |      |                  | ;THE PS IS NOT EQUAL TO 0                          |
| 11785 | 043426 | 000011 |        |        | 11   |                  | ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)         |
| 11786 |        |        |        |        |      |                  | ;BY (013746 000172 000207)                         |
| 11787 |        |        |        |        |      |                  |  |
| 11788 | 043430 | 005237 | 042260 |        | INC  | @#COUNT          |  |
| 11789 | 043434 | 023704 | 042270 |        | CMP  | @#TEMP3,%4       | ;IS THE RESULT IN R4 EQUAL TO TEMP3?               |
| 11790 | 043440 | 001403 |        |        | BEQ  | +.10             |  |
| 11791 | 043442 |        |        |        | 6\$: |                  |  |
| 11792 | 043442 | 004767 | 015174 |        | JSR  | PC,\$HLT         | ;SEEN AN ERROR, GO TO THE HALT ROUTINE             |
| 11793 |        |        |        |        |      |                  | ;EITHER INCORRECT R4 OR INCORRECT SEQUENCE         |
| 11794 | 043446 | 000012 |        |        | 12   |                  | ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)         |
| 11795 |        |        |        |        |      |                  | ;BY (013746 000172 000207)                         |
| 11796 |        |        |        |        |      |                  |  |
| 11797 | 043450 | 010105 |        |        | MOV  | R1,R5            | ;RESTORE R5  |
| 11798 | 043452 | 021537 | 042260 |        | CMP  | (R5),@#COUNT     | ;IS THE TEST NUMBER EQUAL TO THE COUNTER?          |
| 11799 | 043456 | 001371 |        |        | BNE  | 6\$              | ;IF NOT GO TO THE HLT ABOVE                        |
| 11800 | 043460 | 005215 |        |        | INC  | (R5)             |  |
| 11801 | 043462 | 010767 | 134504 |        | MOV  | PC,LPADR         | ;STORE ERROR LOOP ADDRESS                          |
| 11802 | 043466 | 021527 | 000037 |        | CMP  | (R5),#37         | ;HAS THE CONTENTS OF REGISTERS BEEN                |
| 11803 |        |        |        |        |      |                  | ;SHIFTED LEFT BY 14. AND RIGHT BY 14.?             |
| 11804 | 043472 | 002010 |        |        | BGE  | 8\$              |  |
| 11805 | 043474 | 005237 | 042266 |        | INC  | @#TEMP2          |  |
| 11806 | 043500 | 006367 | 176564 |        | ASL  | TEMP3            | ;SHIFT TEMP3 LEFT                                  |
| 11807 | 043504 | 021527 | 000020 |        | CMP  | (R5),#20         | ;HAS THE CONTENTS OF REGISTER BEEN SHIFTED BY 14.? |
| 11808 | 043510 | 001003 |        |        | BNE  | REGR5            |  |
| 11809 | 043512 | 000470 |        |        | BR   | NEGAT            | ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT        |
| 11810 | 043514 | 004767 | 000202 |        | JSR  | PC,TST37         | ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM     |
| 11811 | 043520 | 010767 | 134446 |        | MOV  | PC,LPADR         | ;STORE ERROR LOOP ADDRESS                          |
| 11812 | 043524 | 010501 |        |        | MOV  | R5,R1            | ;SAVE R5   |
| 11813 | 043526 | 013705 | 042264 |        | MOV  | @#TEMP1,%5       | ;LOAD R5 WITH THE CONTENTS OF TEMP1                |
| 11814 | 043532 | 032737 | 000001 | 000306 | BIT  | #1,@#SPASS       | ;IS IT AN EVEN PASS ?                              |
| 11815 | 043540 | 001004 |        |        | BNE  | 2\$              | ;IF NOT THEN GO TO 2\$                             |
| 11816 | 043542 | 013700 | 042266 |        | MOV  | @#TEMP2,R0       | ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0       |
| 11817 | 043546 | 072500 |        |        | ASH  | R0,R5            | ;USING R5  |
| 11818 | 043550 | 000402 |        |        | BR   | 4\$              |  |
| 11819 | 043552 | 072567 | 176510 |        | ASH  | TEMP2,%5         | ;SHIFT R5 BY THE NUMBER SPECIFIED BY TEMP2         |
| 11820 | 043556 | 106737 | 042262 |        | MFPS | @#PSWORD         | ;SAVE PS   |
| 11821 | 043562 | 123737 | 042272 | 042262 | CMPB | @#TEMP4,@#PSWORD | ;IS PS = TEMP4 ?                                   |
| 11822 | 043570 | 001403 |        |        | BEQ  | +.10             |  |
| 11823 | 043572 | 004767 | 015044 |        | JSR  | PC,\$HLT         | ;SEEN AN ERROR, GO TO THE HALT ROUTINE             |
| 11824 |        |        |        |        |      |                  | ;THE PS IS NOT EQUAL TO 0.                         |
| 11825 | 043576 | 000013 |        |        | 13   |                  | ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)         |
| 11826 |        |        |        |        |      |                  | ;BY (013746 000172 000207)                         |
| 11827 |        |        |        |        |      |                  |  |

|       |        |        |        |        |        |  |  |
|-------|--------|--------|--------|--------|--------|--|--|
| 11828 | 043600 | 005237 | 042260 |        | INC    | @#COUNT  |  |
| 11829 | 043604 | 023705 | 042270 |        | CMF    | @#TEMP3,25                                     | ;IS THE RESULT IN R5 EQUAL TO TEMP3?                     |
| 11830 | 043610 | 001403 |        |        | BEQ    | .+10   |  |
| 11831 | 043612 |        |        | 6\$:   |        |  |  |
| 11832 | 043612 | 004767 | 015024 |        | JSR    | PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE |  |
| 11833 |        |        |        |        |        |  | ;EITHER INCORRECT R5 OR INCORRECT SEQUENCE               |
| 11834 | 043616 | 000014 |        |        | 14     |  | ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)               |
| 11835 |        |        |        |        |        |  | ;BY (013746 000172 000207)                               |
| 11836 |        |        |        |        |        |  |  |
| 11837 | 043620 | 021137 | 042260 |        | CMF    | (R1),@#COUNT                                   | ;IS THE TEST NUMBER EQUAL TO THE COUNTER?                |
| 11838 | 043624 | 001372 |        |        | BNE    | 6\$  | ;IF NOT GO TO THE HLT ABOVE                              |
| 11839 | 043626 | 010105 |        |        | MOV    | R1,R5  | ;RESTORE R5  |
| 11840 | 043630 | 005215 |        |        | INC    | (R5)   |  |
| 11841 | 043632 | 010767 | 134334 |        | MOV    | PC,LPADR                                       | ;STORE ERROR LOOP ADDRESS                                |
| 11842 | 043636 | 021527 | 000037 |        | CMF    | (R5),#37                                       | ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED              |
| 11843 |        |        |        |        |        |  | ;LEFT BY 14. AND RIGHT BY 14.?                           |
| 11844 | 043642 | 002010 |        |        | BGE    | 8\$  | ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM           |
| 11845 | 043644 | 005237 | 042266 |        | INC    | @#TEMP2  |  |
| 11846 | 043650 | 006367 | 176414 |        | ASL    | TEMP3  | ;SHIFT TEMP3 LEFT  |
| 11847 | 043654 | 021527 | 000020 |        | CMF    | (R5),#20                                       | ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.? |
| 11848 | 043660 | 001405 |        |        | BEQ    | NEGAT  | ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT              |
| 11849 | 043662 | 000402 |        |        | BR     | 10\$   |  |
| 11850 | 043664 | 004767 | 000032 | 8\$:   | JSR    | PC,TST37                                       |  |
| 11851 | 043670 | 000167 | 176626 | 10\$:  | JMP    | ASTART   | ;GO BACK TO START  |
| 11852 | 043674 | 012737 | 040000 | 042264 | NEGAT: | MOV  | #40000,@#TEMP1 ;TEMP1=40000                              |
| 11853 | 043702 | 012737 | 177762 | 042266 |        | MOV  | #177762,@#TEMP2 ;TEMP2=177762                            |
| 11854 | 043710 | 012737 | 000001 | 042270 |        | MOV  | #1,@#TEMP3 ;TEMP3=1                                      |
| 11855 | 043716 | 000167 | 176600 |        | JMP    | ASTART   |  |
| 11856 | 043722 | 021527 | 000037 |        | TST37: | CMF  | (R5),#37 ;IS IT TEST 37?                                 |
| 11857 | 043726 | 001013 |        |        | BNE    | TST40  | ;IF NOT THEN TRY TEST 40                                 |
| 11858 | 043730 | 005037 | 042264 |        | CLR    | @#TEMP1  | ;0   |
| 11859 | 043734 | 012737 | 000020 | 042266 | MOV    | #16,@#TEMP2                                    | ;SHIFTED BY 16   |
| 11860 | 043742 | 005037 | 042270 |        | CLR    | @#TEMP3  | ;IS=0  |
| 11861 | 043746 | 012737 | 000004 | 042272 | MOV    | #4,@#TEMP4                                     | ;AND PS=4  |
| 11862 | 043754 | 000207 |        |        | RTS    | PC   |  |
| 11863 | 043756 | 021527 | 000040 |        | TST40: | CMF  | (R5),#40 ;IS IT TEST 40?                                 |
| 11864 | 043762 | 001003 |        |        | BNE    | TST41  | ;IF NOT THEN TRY TEST 41                                 |
| 11865 | 043764 | 005037 | 042266 |        | CLR    | @#TEMP2  | ;0 SHIFTED BY 0=0 AND PS=4                               |
| 11866 | 043770 | 000207 |        |        | RTS    | PC   |  |
| 11867 | 043772 | 021527 | 000041 |        | TST41: | CMF  | (R5),#41 ;IS IT TEST 41?                                 |
| 11868 | 043776 | 001004 |        |        | BNE    | TST42  | ;IF NOT THEN TRY TEST 42                                 |
| 11869 | 044000 | 012737 | 177760 | 042266 | MOV    | #-16,@#TEMP2                                   | ;0 SHIFTED BY -16.=0 AND PS=4                            |
| 11870 | 044006 | 000207 |        |        | RTS    | PC   |  |
| 11871 | 044010 | 021527 | 000042 |        | TST42: | CMF  | (R5),#42 ;IS IT TEST 42?                                 |
| 11872 | 044014 | 001013 |        |        | BNE    | TST43  | ;IF NOT THEN TRY TEST 43                                 |
| 11873 | 044016 | 012737 | 100000 | 042264 | MOV    | #100000,@#TEMP1                                | ;100000  |
| 11874 | 044024 | 005237 | 042266 |        | INC    | @#TEMP2  | ;SHIFTED BY -15  |
| 11875 | 044030 | 005337 | 042270 |        | DEC    | @#TEMP3  | ;IS=-1   |
| 11876 | 044034 | 012737 | 000010 | 042272 | MOV    | #10,@#TEMP4                                    | ;AND PS=10   |
| 11877 | 044042 | 000207 |        |        | RTS    | PC   |  |
| 11878 | 044044 | 021527 | 000043 |        | TST43: | CMF  | (R5),#43 ;IS IT TEST 43?                                 |
| 11879 | 044050 | 001012 |        |        | BNE    | TST44  | ;IF NOT THEN IF NOT THEN TRY TEST 44                     |
| 11880 | 044052 | 012737 | 125252 | 042264 | MOV    | #125252,@#TEMP1                                | ;125252  |
| 11881 | 044060 | 012737 | 177777 | 042266 | MOV    | #-1,@#TEMP2                                    | ;SHIFTED BY -1   |
| 11882 | 044066 | 012737 | 152525 | 042270 | MOV    | #152525,@#TEMP3                                | ;IS=152525 AND PS=10                                     |
| 11883 | 044074 | 000207 |        |        | RTS    | PC   |  |

```
11884 044076 021527 000044      TST44:  CMP      (R5),#44      ;IS IT TEST 44?
11885 044102 001012                BNE      TST45      ;IF NOT THEN TRY TEST 45
11886 044104 012737 000001 042266      MOV      #1,@TEMP2   ;125252 SHIFTED BY 1
11887 044112 012737 052524 042270      MOV      #52524,@TEMP3 ;IS=52524
11888 044120 012737 000003 042272      MOV      #3,@TEMP4   ;AND PS=3
11889 044126 000207                RTS      PC
11890 044130 021527 000045      TST45:  CMP      (R5),#45      ;IS IT TEST 45?
11891 044134 001012                BNE      TST46      ;IF NOT THEN TRY TEST 46
11892 044136 012737 177776 042266      MOV      #-2,@TEMP2  ;125252 SHIFTED BY -2
11893 044144 012737 165252 042270      MOV      #165252,@TEMP3 ;IS=165252
11894 044152 012737 000011 042272      MOV      #11,@TEMP4  ;AND PS=11
11895 044160 000207                RTS      PC
11896 044162 021527 000046      TST46:  CMP      (R5),#46      ;IS IT TEST 46?
11897 044166 001014                BNE      TST47      ;IF NOT THEN TRY TEST 47
11898 044170 012737 177777 042264      MOV      #-1,@TEMP1  ;-1
11899 044176 012737 000020 042266      MOV      #16,@TEMP2  ;SHIFTED BY 15.
11900 044204 005037 042270      CLR      @TEMP3      ;IS=0
11901 044210 012737 000007 042272      MOV      #7,@TEMP4  ;AND PS=7
11902 044216 000207                RTS      PC
11903 044220 021527 000047      TST47:  CMP      (R5),#47      ;IS IT TEST 47?
11904 044224 001011                BNE      TST50      ;IF NOT THEN TRY TEST 50
11905 044226 005337 042266      DEC      @TEMP2      ;-1 SHIFTED BY 15
11906 044232 012737 100000 042270      MOV      #100000,@TEMP3 ;IS=100000
11907 044240 012737 000011 042272      MOV      #11,@TEMP4  ;AND PS=11
11908 044246 000207                RTS      PC
11909 044250 021527 000050      TST50:  CMP      (R5),#50      ;IS IT TEST 50
11910 044254 001007                BNE      ENT51      ;IF NOT THEN TRY TEST 51
11911 044256 012737 137777 042264      MOV      #137777,@TEMP1 ;137777 SHIFTED BY 15. IS=100000
11912 044264 012737 000013 042272      MOV      #13,@TEMP4  ;AND PS=13
11913 044272 000207                RTS      PC
11914 044274 021527 000051      ENT51:  CMP      (R5),#51      ;IS IT ENTERING TEST 51?
11915 044300 001403                BEQ      .+10
11916 044302 004767 014334      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11917                                ;TEST NUMBER GOOFED
11918 044306 000015                15          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11919                                ;BY (013746 000172 000207)
11920
11921
11922 044310 005726                TST      (SP)+      ;RESTORE STACK POINTER
11923 044312 012704 177771      MOV      #-7,%4
11924 044316 012702 042304      MOV      #51,%2
11925 044322 012703 042306      MOV      #52,%3
```

11926  
11927  
11928  
11929  
11930 044326 010767 133640  
11931 044332 012701 125252  
11932 044336 072127 000005  
11933 044342 106737 042262  
11934 044346 122737 000003 042262  
11935 044354 001403  
11936 044356 004767 014260  
11937  
11938 044362 000016  
11939  
11940  
11941 044364 022701 052500  
11942 044370 001403  
11943 044372  
11944 044372 004767 014244  
11945  
11946 044376 000017  
11947  
11948  
11949 044400 021527 000051  
11950 044404 001372  
11951 044406 005215  
11952  
11953  
11954  
11955  
11956  
11957  
11958  
11959 044410 010767 133556  
11960 044414 012700 125252  
11961 044420 072077 175662  
11962 044424 106737 042262  
11963 044430 122737 000010 042262  
11964 044436 001403  
11965 044440 004767 014176  
11966  
11967 044444 000020  
11968  
11969  
11970 044446 022700 177525  
11971 044452 001403  
11972 044454  
11973 044454 004767 014162  
11974  
11975 044460 000021  
11976  
11977  
11978 044462 021527 000052  
11979 044466 001372  
11980 044470 005215  
11981

```
*****  
:TEST:51 11/34 ASH 125252 SHIFTED BY #5 = 52500 PS = 3  
*****  
TST51: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #125252,%1 ;LOAD R1 WITH 125252  
ASH #5,%1 ;SHIFT R1 BY #5  
MFPS @#PSWORD ;SAVE PS  
CMPB #3,@#PSWORD ;IS THE PS 3?  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;THE PS IS NOT EQUAL TO 3  
16 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #52500,%1 ;IS THE RESULT 52500?  
BEQ .+10  
1$:  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;R1 IS NOT EQUAL TO 52500 OR INCORRECT SEQUENCE  
17 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#51 ;IS $TESTN = #51  
BNE 1$ ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)  
  
*****  
:TEST:52 11/34 ASH 125252 SHIFTED BY @S2 = 177525 PS = 10  
*****  
TST52: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #125252,%0 ;LOAD R0 WITH 125252  
ASH @S2,%0 ;SHIFT R0 BY @S2  
MFPS @#PSWORD ;SAVE PS  
CMPB #10,@#PSWORD ;IS THE PS 10?  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;THE PS IS NOT EQUAL TO 10  
20 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #177525,%0 ;IS THE RESULT 177525?  
BEQ .+10  
1$:  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE  
21 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#52 ;IS $TESTN = #52  
BNE 1$ ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)
```

MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79<sup>F 3</sup> 11:31 PAGE 238  
ASH INSTRUCTION TESTS

SEQ 0238

11982  
11983



11984  
11985  
11986  
11987  
11988 044472 010767 133474  
11989 044476 012700 125252  
11990 044502 072037 042304  
11991 044506 106737 042262  
11992 044512 122737 000010 042262  
11993 044520 001403  
11994 044522 004767 014114  
11995  
11996 044526 000022  
11997  
11998  
11999 044530 022700 177525  
12000 044534 001403  
12001 044536  
12002 044536 004767 014100  
12003  
12004 044542 000023  
12005  
12006  
12007 044544 021527 000053  
12008 044550 001372  
12009 044552 005215  
12010  
12011  
12012  
12013  
12014  
12015  
12016  
12017 044554 010767 133412  
12018 044560 012700 125252  
12019 044564 072012  
12020 044566 106737 042262  
12021 044572 122737 000010 042262  
12022 044600 001403  
12023 044602 004767 014034  
12024  
12025 044606 000024  
12026  
12027  
12028 044610 022700 177525  
12029 044614 001403  
12030 044616  
12031 044616 004767 014020  
12032  
12033 044622 000025  
12034  
12035  
12036 044624 021527 000054  
12037 044630 001372  
12038 044632 005215  
12039

```
*****
:TEST:53 11/34 ASH 125252 SHIFTED BY @#S1 = 177525 PS = 10
*****
TST53: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
        MOV #125252,%0 ;LOAD R0 WITH 125252
        ASH @#S1,%0 ;SHIFT R0 BY @#S1
        MFPS @#PSWORD ;SAVE PS
        CMPB #10,@#PSWORD ;IS THE PS 10?
        BEQ .+10
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;THE PS IS NOT EQUAL TO 10
        22 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP #177525,%0 ;IS THE RESULT 177525?
        BEQ .+10
1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
        23 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP (R5),#53 ;IS $TESTN = #53
        BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
        INC (R5)
```

```
*****
:TEST:54 11/34 ASH 125252 SHIFTED BY (2) = 177525 PS = 10
*****
TST54: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
        MOV #125252,%0 ;LOAD R0 WITH 125252
        ASH (2),%0 ;SHIFT R0 BY (2)
        MFPS @#PSWORD ;SAVE PS
        CMPB #10,@#PSWORD ;IS THE PS 10?
        BEQ .+10
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;THE PS IS NOT EQUAL TO 10
        24 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP #177525,%0 ;IS THE RESULT 177525?
        BEQ .+10
1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
        25 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP (R5),#54 ;IS $TESTN = #54
        BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
        INC (R5)
```

MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79<sup>H 3</sup> 11:31 PAGE 240  
ASH INSTRUCTION TESTS

SEQ 0240

12040  
12041

12042  
12043  
12044  
12045  
12046 044634 010767 133332  
12047 044640 012700 125252  
12048 044644 072022  
12049 044646 106737 042262  
12050 044652 122737 000010 042262  
12051 044660 001403  
12052 044662 004767 013754  
12053  
12054 044666 000026  
12055  
12056  
12057 044670 022700 177525  
12058 044674 001403  
12059 044676  
12060 044676 004767 013740  
12061  
12062 044702 000027  
12063  
12064  
12065 044704 021527 000055  
12066 044710 001372  
12067 044712 005215  
12068  
12069  
12070  
12071  
12072  
12073  
12074  
12075 044714 010767 133252  
12076 044720 012700 125252  
12077 044724 072042  
12078 044726 106737 042262  
12079 044732 122737 000010 042262  
12080 044740 001403  
12081 044742 004767 013674  
12082  
12083 044746 000030  
12084  
12085  
12086 044750 022700 177525  
12087 044754 001403  
12088 044756  
12089 044756 004767 013660  
12090  
12091 044762 000031  
12092  
12093  
12094 044764 021527 000056  
12095 044770 001372  
12096 044772 005215  
12097

```
*****  
:TEST:55 11/34 ASH 125252 SHIFTED BY (2)+ = 177525 PS = 10  
*****  
TST55: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #125252,%0 ;LOAD R0 WITH 125252  
ASH (2)+,%0 ;SHIFT R0 BY (2)+  
MFPS @#PSWORD ;SAVE PS  
CMPB #10,@#PSWORD ;IS THE PS 10?  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;THE PS IS NOT EQUAL TO 10  
26 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #177525,%0 ;IS THE RESULT 177525?  
BEQ .+10  
1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE  
27 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#55 ;IS $TESTN = #55  
BNE 1$ ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)
```

```
*****  
:TEST:56 11/34 ASH 125252 SHIFTED BY -(2) = 177525 PS = 10  
*****  
TST56: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #125252,%0 ;LOAD R0 WITH 125252  
ASH -(2),%0 ;SHIFT R0 BY -(2)  
MFPS @#PSWORD ;SAVE PS  
CMPB #10,@#PSWORD ;IS THE PS 10?  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;THE PS IS NOT EQUAL TO 10  
30 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #177525,%0 ;IS THE RESULT 177525?  
BEQ .+10  
1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE  
31 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#56 ;IS $TESTN = #56  
BNE 1$ ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)
```

MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79<sup>J 3</sup> 11:31 PAGE 242  
ASH INSTRUCTION TESTS

SEQ 0242

12098  
12099

12100  
12101  
12102  
12103  
12104 044774 010767 133172  
12105 045000 012700 125252  
12106 045004 072063 000002  
12107 045010 106737 042262  
12108 045014 122737 000011 042262  
12109 045022 001403  
12110 045024 004767 013612  
12111  
12112 045030 000032  
12113  
12114  
12115 045032 022700 177252  
12116 045036 001403  
12117 045040  
12118 045040 004767 013576  
12119  
12120 045044 000033  
12121  
12122  
12123 045046 021527 000057  
12124 045052 001372  
12125 045054 005215  
12126  
12127  
12128  
12129  
12130  
12131  
12132  
12133 045056 010767 133110  
12134 045062 012700 125252  
12135 045066 072073 000000  
12136 045072 106737 042262  
12137 045076 122737 000010 042262  
12138 045104 001403  
12139 045106 004767 013530  
12140  
12141 045112 000034  
12142  
12143  
12144 045114 022700 177525  
12145 045120 001403  
12146 045122  
12147 045122 004767 013514  
12148  
12149 045126 000035  
12150  
12151  
12152 045130 021527 000060  
12153 045134 001372  
12154 045136 005215  
12155

```
*****
;TEST:57 11/34 ASH 125252 SHIFTED BY 2(3) = 177252 PS = 11
*****

TST57: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
        MOV #125252,%0 ;LOAD R0 WITH 125252
        ASH 2(3),%0 ;SHIFT R0 BY 2(3)
        MFPS @#PSWORD ;SAVE PS
        CMPB #11,@#PSWORD ;IS THE PS 11?
        BEQ .+10
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;THE PS IS NOT EQUAL TO 11
        32 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP #177252,%0 ;IS THE RESULT 177252?
        BEQ .+10
1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;R0 IS NOT EQUAL TO 177252 OR INCORRECT SEQUENCE
        33 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP (R5),#57 ;IS $TESTN = #57
        BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
        INC (R5)

*****
;TEST:60 11/34 ASH 125252 SHIFTED BY @ (3) = 177525 PS = 10
*****

TST60: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
        MOV #125252,%0 ;LOAD R0 WITH 125252
        ASH @(3),%0 ;SHIFT R0 BY @(3)
        MFPS @#PSWORD ;SAVE PS
        CMPB #10,@#PSWORD ;IS THE PS 10?
        BEQ .+10
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;THE PS IS NOT EQUAL TO 10
        34 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP #177525,%0 ;IS THE RESULT 177525?
        BEQ .+10
1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
        35 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP (R5),#60 ;IS $TESTN = #60
        BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
        INC (R5)
```

MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79<sup>L 3</sup> 11:31 PAGE 244  
ASH INSTRUCTION TESTS

SEQ 0244

12156  
i2157

12158  
12159  
12160  
12161  
12162 045140 010767 133026  
12163 045144 012700 125252  
12164 045150 072033  
12165 045152 106737 042262  
12166 045156 122737 000010 042262  
12167 045164 001403  
12168 045166 004767 013450  
12169  
12170 045172 000036  
12171  
12172  
12173 045174 022700 177525  
12174 045200 001403  
12175 045202  
12176 045202 004767 013434  
12177  
12178 045206 000037  
12179  
12180  
12181 045210 021527 000061  
12182 045214 001372  
12183 045216 005215  
12184  
12185  
12186  
12187  
12188  
12189  
12190  
12191 045220 010767 132746  
12192 045224 012700 125252  
12193 045230 072053  
12194 045232 106737 042262  
12195 045236 122737 000010 042262  
12196 045244 001403  
12197 045246 004767 013370  
12198  
12199 045252 000040  
12200  
12201  
12202 045254 022700 177525  
12203 045260 001403  
12204 045262  
12205 045262 004767 013354  
12206  
12207 045266 000041  
12208  
12209  
12210 045270 021527 000062  
12211 045274 001372  
12212 045276 005215  
12213

```
*****  
:TEST:61 11/34 ASH 125252 SHIFTED BY @ (3)+ = 177525 PS = 10  
*****  
TST61: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #125252,%0 ;LOAD R0 WITH 125252  
ASH @ (3)+,%0 ;SHIFT R0 BY @ (3)+  
MFPS @#PSWORD ;SAVE PS  
CMPB #10,@#PSWORD ;IS THE PS 10?  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;THE PS IS NOT EQUAL TO 10  
36 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #177525,%0 ;IS THE RESULT 177525?  
BEQ .+10  
18:  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE  
47 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#61 ;IS $TESTN = #61  
BNE 18 ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)
```

```
*****  
:TEST:62 11/34 ASH 125252 SHIFTED BY @-(3) = 177525 PS = 10  
*****  
TST62: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #125252,%0 ;LOAD R0 WITH 125252  
ASH @-(3),%0 ;SHIFT R0 BY @-(3)  
MFPS @#PSWORD ;SAVE PS  
CMPB #10,@#PSWORD ;IS THE PS 10?  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;THE PS IS NOT EQUAL TO 10  
40 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #177525,%0 ;IS THE RESULT 177525?  
BEQ .+10  
18:  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE  
41 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#62 ;IS $TESTN = #62  
BNE 18 ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)
```

MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79<sup>N 3</sup> 11:31 PAGE 246  
ASH INSTRUCTION TESTS

SEQ 0246

12214  
12215



12216  
12217  
12218  
12219  
12220  
12221  
12222  
12223  
12224  
12225  
12226  
12227  
12228  
12229  
12230  
12231  
12232  
12233  
12234  
12235  
12236  
12237  
12238  
12239  
12240  
12241  
12242  
12243  
12244  
12245  
12246  
12247  
12248  
12249  
12250  
12251  
12252  
12253  
12254  
12255  
12256  
12257  
12258  
12259  
12260  
12261  
12262  
12263  
12264  
12265  
12266  
12267  
12268  
12269  
12270  
12271

045300 010767 132666  
045304 012737 000062 042260  
045312 005037 042264  
045316 012737 000001 042266  
045324 005037 042270  
045330 005037 042272  
045334 012737 000001 042274  
045342 005037 042276  
  
045346 010502  
045350 013700 042264  
045354 013701 042266  
045360 000241  
045362 032737 000001 000306  
045370 001004  
045372 013705 042270  
045376 073005  
045400 000402  
045402 073067 174662  
045406 106737 042262  
045412 123737 042276 042262  
045420 001403  
045422 004767 013214  
  
045426 000042  
  
045430 005237 042260  
045434 023700 042272  
045440 001403  
045442 004767 013174  
  
045446 000043  
  
045450 023701 042274  
045454 001403  
  
045456 004767 013160  
  
045462 000044  
  
045464 010205

REG01:  
  
  
  
  
  
  
2\$:  
4\$:

```

:*****
:      ASHC INSTRUCTION TESTS
:*****

:*****
:TESTS 63-157
:*****

MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
MOV    #62,@#COUNT
CLR    @#TEMP1       ;TEMP1=0
MOV    #1,@#TEMP2   ;TEMP2=1
CLR    @#TEMP3       ;TEMP3=0
CLR    @#TEMP4       ;TEMP4=0
MOV    #1,@#TEMP5   ;TEMP5=1
CLR    @#TEMP6       ;0 1 SHIFTED BY 0=0 1, PS=)

REG01: MOV    R5,R2      ;SAVE R5
        MOV    @#TEMP1,X0 ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 0
        MOV    @#TEMP2,X0!1 ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 1
        CLC
        BIT    #1,@#SPASS ;IS IT AN EVEN PASS ?
        BNE    2$        ;IF NOT THEN GO TO 2$
        MOV    @#TEMP3,R5 ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
        ASHC   R5,R0      ;USING R0
        BR     4$
2$:    ASHC   TEMP3,X0    ;ASHC REGISTER 0 BY THE CONTENTS OF TEMP3
4$:    MFPS   @#PSWORD   ;SAVE PS
        CMPB  @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
        BEQ   .+10
        JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;WRONG PS
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        INC   @#COUNT
        CMP   @#TEMP4,X0 ;IS THE RESULT IN R0 SAME AS TEMP4?
        BEQ   .+10
        JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;WRONG RESULT IN R0
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP   @#TEMP5,X1 ;IS THE RESULT IN R1 SAME AS TEMP5?
        BEQ   .+10
                ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMPS
                ;AND PS=TEMP6
        JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;WRONG RESULT IN R1
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

MOV    R2,R5      ;RESTORE R5
```

```

12272 045466 021537 042260      CMP      (R5),@#COUNT      ;IS TEST NUMBER=COUNTER?
12273 045472 001'03              BEQ      .+10
12274 045474 004767 013142      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12275                                ;NO
12276 045500 000045              45      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12277                                ;BY (013746 000172 000207)
12278
12279 045502 005215              INC      (R5)
12280 045504 021527 000160      CMP      (R5),#160          ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
12281 045510 002014              BGE      6$                ;YES
12282 045512 005237 042270      INC      @#TEMP3
12283 045516 000241              CLC
12284 045520 006137 042274      ROL      @#TEMPS            ;ROTATE TEMPS LEFT BY 1 PLACE
12285 045524 006137 042272      ROL      @#TEMP4            ;INTRODUCE CARRY FROM TEMP4 IN TEMPS
12286 045530 021527 000121      CMP      (R5),#121          ;IS IT TEST 121?
12287 045534 001004              BNE      REGR23
12288 045536 004467 000414      JSR      R4,RITSH           ;IF SO THEN GO AND INITIATE RIGHT SHIFT
12289 045542 004767 000444      6$: JSR      %7,TST160
12290 045546 010767 132420      REGR23: MOV     PC,LPADR      ;STORE ERROR LOOP ADDRESS
12291 045552 013702 042264      MOV      @#TEMP1,%2         ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 2
12292 045556 013703 042266      MOV      @#TEMP2,%2!1       ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 3
12293 045562 000241              CLC
12294 045564 032737 000001 000306  BIT      #1,@#SPASS          ;IS IT AN EVEN PASS ?
12295 045572 001004              BNE      2$                ;IF NOT THEN GO TO 2$
12296 045574 013704 042270      MOV      @#TEMP3,R4         ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12297 045600 073204              ASHC    R4,R2              ;USING R2
12298 045602 000402              BR      4$
12299 045604 073267 174460      2$: ASHC    TEMP3,%2         ;ASHC REGISTER 2 BY THE CONTENTS OF TEMP3
12300 045610 106737 042262      4$: MFPS    @#PSWORD         ;SAVE PS
12301 045614 123737 042276 042262  CMPB    @#TEMP6,@#PSWORD    ;COMPARE PS WITH THE CONTENTS OF TEMP6
12302 045622 001403              BEQ      .+10
12303 045624 004767 013012      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12304                                ;WRONG PS
12305 045630 000046              46      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12306                                ;BY (013746 000172 000207)
12307
12308 045632 005237 042260      INC      @#COUNT
12309 045636 023702 042272      CMP      @#TEMP4,%2         ;IS THE RESULT IN R2 SAME AS TEMP4?
12310 045642 001403              BEQ      .+10
12311 045644 004767 012772      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12312                                ;WRONG RESULT IN R2
12313 045650 000047              47      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12314                                ;BY (013746 000172 000207)
12315
12316 045652 023703 042274      CMP      @#TEMPS,%3         ;IS THE RESULT IN R3 SAME AS TEMPS?
12317 045656 001403              BEQ      .+10              ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMPS
12318                                ;AND PS=TEMP6
12319 045660 004767 012756      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12320                                ;WRONG RESULT IN R1
12321 045664 000050              50      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12322                                ;BY (013746 000172 000207)
12323
12324 045666 021537 042260      CMP      (R5),@#COUNT      ;IS TEST NUMBER=COUNTER?
12325 045672 001403              BEQ      .+10
12326 045674 004767 012742      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12327                                ;NO

```



|       |        |        |        |        |             |                 |  |   |
|-------|--------|--------|--------|--------|-------------|-----------------|--|---|
| 12384 | 046104 | 010105 |        |        | MOV         | R1,R5           |  | ;RESTORE R5                             |
| 12385 | 046106 | 005215 |        |        | INC         | (R5)            |  |   |
| 12386 | 046110 | 021527 | 000160 |        | CMP         | (R5),#160       |  | ;HAVE THE FIRST 159 TEST BEEN EXECUTED? |
| 12387 | 046114 | 002014 |        |        | BGE         | 68              |  | ;YES                                    |
| 12388 | 046116 | 005237 | 042270 |        | INC         | @#TEMP3         |  |   |
| 12389 | 046122 | 000241 |        |        | CLC         |                 |  |   |
| 12390 | 046124 | 006137 | 042274 |        | ROL         | @#TEMP5         |  | ;ROTATE TEMP5 LEFT BY 1 PLACE           |
| 12391 | 046130 | 006137 | 042272 |        | ROL         | @#TEMP4         |  | ;INTRODUCE CARRY FROM TEMP5 IN TEMP4    |
| 12392 | 046134 | 021527 | 000121 |        | CMP         | (R5),#121       |  | ;IS IT TEST 121?                        |
| 12393 | 046140 | 001004 |        |        | BNE         | 88              |  |   |
| 12394 | 046142 | 004467 | 000010 |        | JSR         | R4,RITSH        |  | ;IF SO THEN GO AND INITIATE RIGHT SHIFT |
| 12395 | 046146 | 004767 | 000040 |        | JSR         | %7,TST160       |  |   |
| 12396 | 046152 | 000167 | 177170 |        | JMP         | REG01           |  |   |
| 12397 | 046156 | 022424 |        |        | RITSH: CMP  | (R4)+,(R4)+     |  | ;MAKE R4 POINT TO THE NEXT REG TAG      |
| 12398 | 046160 | 012737 | 040000 | 042264 | MOV         | #40000,@#TEMP1  |  | ;TEMP1=4000                             |
| 12399 | 046166 | 005037 | 042266 |        | CLR         | @#TEMP2         |  | ;TEMP2=0                                |
| 12400 | 046172 | 012737 | 177742 | 042270 | MOV         | #-30,@#TEMP3    |  | ;TEMP3=-30                              |
| 12401 | 046200 | 005037 | 042272 |        | CLR         | @#TEMP4         |  | ;TEMP4=0                                |
| 12402 | 046204 | 005237 | 042274 |        | INC         | @#TEMP5         |  | ;TEMP5=1                                |
| 12403 | 046210 | 000204 |        |        | RTS         | R4              |  |   |
| 12404 | 046212 | 021527 | 000160 |        | TST160: CMP | (R5),#160       |  | ;IS IT TEST 160                         |
| 12405 | 046216 | 001010 |        |        | BNE         | TST161          |  | ;IF NOT THEN TRY TEST 161               |
| 12406 | 046220 | 005037 | 042264 |        | CLR         | @#TEMP1         |  | ;0 0 SHIFTED BY 0                       |
| 12407 | 046224 | 005037 | 042272 |        | CLR         | @#TEMP4         |  | ;IS EQUAL TO 0 0                        |
| 12408 | 046230 | 012737 | 000004 | 042276 | MOV         | #4,@#TEMP6      |  | ;AND PS=4                               |
| 12409 | 046236 | 000207 |        |        | RTS         | %7              |  |   |
| 12410 | 046240 | 021527 | 000161 |        | TST161: CMP | (R5),#161       |  | ;IS IT TEST 161                         |
| 12411 | 046244 | 001004 |        |        | BNE         | TST162          |  |   |
| 12412 | 046246 | 012737 | 177746 | 042270 | MOV         | #-32,@#TEMP3    |  | ;0 0 SHIFTED BY -32=0 0, PS=4           |
| 12413 | 046254 | 000207 |        |        | RTS         | %7              |  |   |
| 12414 | 046256 | 021527 | 000162 |        | TST162: CMP | (R5),#162       |  | ;IS IT TEST 162                         |
| 12415 | 046262 | 001004 |        |        | BNE         | TST163          |  | ;IF NOT THEN TRY TEST 163               |
| 12416 | 046264 | 012737 | 000032 | 042270 | MOV         | #32,@#TEMP3     |  | ;0 0 SHIFTED BY 32=0 0, PS=4            |
| 12417 | 046272 | 000207 |        |        | RTS         | %7              |  |   |
| 12418 | 046274 | 021527 | 000163 |        | TST163: CMP | (R5),#163       |  | ;IS IT TEST 163?                        |
| 12419 | 046300 | 001016 |        |        | BNE         | TST164          |  | ;IF NOT THEN TRY TEST 164               |
| 12420 | 046302 | 012737 | 052525 | 042264 | MOV         | #52525,@#TEMP1  |  | ;52525 0                                |
| 12421 | 046310 | 012737 | 177760 | 042270 | MOV         | #-16,@#TEMP3    |  | ;SHIFTED BY -16.                        |
| 12422 | 046316 | 005037 | 042272 |        | CLR         | @#TEMP4         |  |   |
| 12423 | 046322 | 012737 | 052525 | 042274 | MOV         | #52525,@#TEMP5  |  | ;IS EQUAL TO 0 52525                    |
| 12424 | 046330 | 005037 | 042276 |        | CLR         | @#TEMP6         |  | ;AND PS = 0                             |
| 12425 | 046334 | 000207 |        |        | RTS         | %7              |  |   |
| 12426 | 046336 | 021527 | 000164 |        | TST164: CMP | (R5),#164       |  | ;IS IT TEST 164?                        |
| 12427 | 046342 | 001014 |        |        | BNE         | TST165          |  | ;IF NOT THEN TRY TEST 165               |
| 12428 | 046344 | 012737 | 125252 | 042264 | MOV         | #125252,@#TEMP1 |  | ;125252 0 SHIFTED BY -16.               |
| 12429 | 046352 | 005337 | 042272 |        | DEC         | @#TEMP4         |  |   |
| 12430 | 046356 | 012737 | 125252 | 042274 | MOV         | #125252,@#TEMP5 |  | ;IS EQUAL TO -1 125252                  |
| 12431 | 046364 | 012737 | 000010 | 042276 | MOV         | #10,@#TEMP6     |  | ;AND PS=10                              |
| 12432 | 046372 | 000207 |        |        | RTS         | %7              |  |   |
| 12433 | 046374 | 021527 | 000165 |        | TST165: CMP | (R5),#165       |  | ;IS IT TEST 165?                        |
| 12434 | 046400 | 001007 |        |        | BNE         | TST166          |  | ;IF NOT THEN TRY TEST 166               |
| 12435 | 046402 | 012737 | 177777 | 042264 | MOV         | #-1,@#TEMP1     |  | ; -1 0 SHIFTED BY -16                   |
| 12436 | 046410 | 012737 | 177777 | 042274 | MOV         | #-1,@#TEMP5     |  | ;IS EQUAL TO -1 -1, AND PS=10           |
| 12437 | 046416 | 000207 |        |        | RTS         | %7              |  |   |
| 12438 | 046420 | 021527 | 000166 |        | TST166: CMP | (R5),#166       |  | ;IS IT TEST 166?                        |
| 12439 | 046424 | 001011 |        |        | BNE         | TST167          |  | ;IF NOT THEN TRY TEST 167               |

|       |        |        |        |        |             |  |   |
|-------|--------|--------|--------|--------|-------------|--|---|
| 12440 | 046426 | 012737 | 100000 | 042264 | MOV         | #100000,@#TEMP1                                | :100000 0                                     |
| 12441 | 046434 | 012737 | 177740 | 042270 | MOV         | #-32,@#TEMP3                                   | :SHIFTED BY -32 ;S EQUAL TO -1 -1             |
| 12442 | 046442 | 005237 | 042276 |        | INC         | @#TEMP6  | :AND PS=11                                    |
| 12443 | 046446 | 000207 |        |        | RTS         | %7   |   |
| 12444 | 046450 | 021527 | 000167 |        | TST167: CMP | (R5),#167                                      | :IS IT TEST 167?                              |
| 12445 | 046454 | 001014 |        |        | BNE         | TST170   | :IF NOT THEN TRY TEST 170                     |
| 12446 | 046456 | 005037 | 042264 |        | CLR         | @#TEMP1  |   |
| 12447 | 046462 | 005337 | 042266 |        | DEC         | @#TEMP2  | :0 -1   |
| 12448 | 046466 | 012737 | 000020 | 042270 | MOV         | #16,@#TEMP3                                    | :SHIFTED BY 16.                               |
| 12449 | 046474 | 005037 | 042274 |        | CLR         | @#TEMP5  | :IS EQUAL TO -1 0                             |
| 12450 | 046500 | 005237 | 042276 |        | INC         | @#TEMP6  | :AND PS=12                                    |
| 12451 | 046504 | 000207 |        |        | RTS         | %7   |   |
| 12452 | 046506 | 021527 | 000170 |        | TST170: CMP | (R5),#170                                      | :IS IT TEST 170?                              |
| 12453 | 046512 | 001007 |        |        | BNE         | TST171   | :IF NOT THEN TRY TEST 171                     |
| 12454 | 046514 | 012737 | 125252 | 042266 | MOV         | #125252,@#TEMP2                                | :0 125252 SHIFTED BY 16                       |
| 12455 | 046522 | 012737 | 125252 | 042272 | MOV         | #125252,@#TEMP4                                | :IS EQUAL TO 125252 0, AND PS=12              |
| 12456 | 046530 | 000207 |        |        | RTS         | %7   |   |
| 12457 | 046532 | 021527 | 000171 |        | TST171: CMP | (R5),#171                                      | :IS IT TEST 171?                              |
| 12458 | 046536 | 001010 |        |        | BNE         | TST172   | :IF NOT THEN TRY TEST 172                     |
| 12459 | 046540 | 005337 | 042270 |        | DEC         | @#TEMP3  | :0 125252 SHIFTED BY 15                       |
| 12460 | 046544 | 012737 | 052525 | 042272 | MOV         | #52525,@#TEMP4                                 | :IS EQUAL TO 52525 0                          |
| 12461 | 046552 | 005037 | 042276 |        | CLR         | @#TEMP6  | :AND PS=0                                     |
| 12462 | 046556 | 000207 |        |        | RTS         | %7   |   |
| 12463 | 046560 | 021527 | 000172 |        | TST172: CMP | (R5),#172                                      | :IS IT TEST 172?                              |
| 12464 | 046564 | 001006 |        |        | BNE         | TST173   | :IF NOT THEN TRY TEST 173                     |
| 12465 | 046566 | 012737 | 052525 | 042266 | MOV         | #52525,@#TEMP2                                 | :0 52525                                      |
| 12466 | 046574 | 005237 | 042270 |        | INC         | @#TEMP3  | :SHIFTED BY 16. IS EQUAL TO 52525 0, AND PS=0 |
| 12467 | 046600 | 000207 |        |        | RTS         | %7   |   |
| 12468 | 046602 | 021527 | 000173 |        | TST173: CMP | (R5),#173                                      | :IS IT TEST 173?                              |
| 12469 | 046606 | 001014 |        |        | BNE         | TST174   | :IF NOT THEN TRY TEST 174                     |
| 12470 | 046610 | 012737 | 177777 | 042266 | MOV         | #-1,@#TEMP2                                    | :0 -1   |
| 12471 | 046616 | 005337 | 042270 |        | DEC         | @#TEMP3  | :SHIFTED BY 15.                               |
| 12472 | 046622 | 012737 | 077777 | 042272 | MOV         | #77777,@#TEMP4                                 |   |
| 12473 | 046630 | 012737 | 100000 | 042274 | MOV         | #100000,@#TEMP5                                | :IS EQUAL TO 77777 100000, AND PS=0           |
| 12474 | 046636 | 000207 |        |        | RTS         | %7   |   |
| 12475 | 046640 | 021527 | 000174 |        | TST174: CMP | (R5),#174                                      | :IS IT TEST 174?                              |
| 12476 | 046644 | 001013 |        |        | BNE         | TST175   | :IF NOT THEN TRY TEST 175                     |
| 12477 | 046646 | 012737 | 100000 | 042264 | MOV         | #100000,@#TEMP1                                |   |
| 12478 | 046654 | 005337 | 042266 |        | DEC         | @#TEMP2  | :100000 -2 SHIFTED BY 15.                     |
| 12479 | 046660 | 005037 | 042274 |        | CLR         | @#TEMP5  | :IS EQUAL TO 77777 0                          |
| 12480 | 046664 | 012737 | 000002 | 042276 | MOV         | #2,@#TEMP6                                     | :AND PS=2                                     |
| 12481 | 046672 | 000207 |        |        | RTS         | %7   |   |
| 12482 | 046674 | 021527 | 000175 |        | TST175: CMP | (R5),#175                                      | :IS IT TEST 175?                              |
| 12483 | 046700 | 001015 |        |        | BNE         | ENT176   | :IF NOT THEN TRY TEST 176                     |
| 12484 | 046702 | 012737 | 177777 | 042264 | MOV         | #-1,@#TEMP1                                    |   |
| 12485 | 046710 | 005037 | 042266 |        | CLR         | @#TEMP2  | :-1 0   |
| 12486 | 046714 | 005237 | 042270 |        | INC         | @#TEMP3  | :SHIFTED BY 16.                               |
| 12487 | 046720 | 005037 | 042272 |        | CLR         | @#TEMP4  | :IS EQUAL TO 0 0                              |
| 12488 | 046724 | 012737 | 000007 | 042276 | MOV         | #7,@#TEMP6                                     | :AND PS=7                                     |
| 12489 | 046732 | 000207 |        |        | RTS         | %7   |   |
| 12490 | 046734 | 021527 | 000176 |        | ENT176: CMP | (R5),#176                                      | :IS THE PROGRAM ENTERING TEST 176?            |
| 12491 | 046740 | 001403 |        |        | BEQ         | +.10   |   |
| 12492 | 046742 | 004767 | 011674 |        | JSR         | PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE |   |
| 12493 |        |        |        |        |             | :TEST NUMBER GOOFED                            |   |
| 12494 | 046746 | 000056 |        |        | 56          | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)     |   |
| 12495 |        |        |        |        |             | :BY (013746 000172 000207)                     |   |

MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79<sup>6 4</sup> 11:31 PAGE 252  
ASHC INSTRUCTION TESTS

SEQ 0252

12496  
i2497  
12498 046750 005726  
12499

TST (SP)+

;RESTORE STACK POINTER

12500  
12501  
12502  
12503  
12504 046752 010767 131214  
12505 046756 012701 000000  
12506 046762 012701 000001  
12507 046766 000241  
12508 046770 073127 000010  
12509 046774 106737 042262  
12510 047000 122737 000000 042262  
12511 047006 001403  
12512 047010 004767 011626  
12513  
12514 047014 000057  
12515  
12516  
12517 047016 022701 000400  
12518 047022 001403  
12519 047024 004767 011612  
12520  
12521 047030 000060  
12522  
12523  
12524 047032 021527 000176  
12525 047036 001403  
12526 047040 004767 011576  
12527  
12528 047044 000061  
12529  
12530  
12531 047046 005215  
12532  
12533  
12534  
12535  
12536  
12537  
12538 047050 010767 131116  
12539 047054 012703 000000  
12540 047060 012703 177777  
12541 047064 000241  
12542 047066 073327 000017  
12543 047072 106737 042262  
12544 047076 122737 000011 042262  
12545 047104 001403  
12546 047106 004767 011530  
12547  
12548 047112 000062  
12549  
12550  
12551 047114 022703 100000  
12552 047120 001403  
12553 047122 004767 011514  
12554  
12555 047126 000063

```
*****  
:TEST:176      1 SHIFTED BY 8. = 400 PS = 0  
*****  
TST176: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
          MOV      #DUMMY,%1    ;LOAD R1 WITH DUMMY  
          MOV      #1,%1!1      ;LOAD R1!1 WITH 1  
          CLC  
          ASHC     #8,%1         ;SHIFT R1,R1!1 BY 8.  
          MFPS     @#PSWORD      ;SAVE PS  
          CMPB    #0,@#PSWORD    ;IS THE PS 0?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;THE PS IS NOT EQUAL TO 0  
          57      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
          CMP     #400,%1        ;IS THE RESULT 400?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;R1 IS NOT EQUAL TO 400  
          60      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
          CMP     (R5),#176      ;IS $TESTN = #176?  
          BEQ     .+10           ;IF NOT THEN GO TO HLT  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;TEST IS IN WRONG SEQUENCE  
          61      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
          INC     (R5)  
  
*****  
:TEST:177      -1 SHIFTED BY 15. = 100000 PS = 11  
*****  
TST177: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
          MOV      #DUMMY,%3    ;LOAD R3 WITH DUMMY  
          MOV      #-1,%3!1     ;LOAD R3!1 WITH -1  
          CLC  
          ASHC     #15,%3        ;SHIFT R3,R3!1 BY 15.  
          MFPS     @#PSWORD      ;SAVE PS  
          CMPB    #11,@#PSWORD  ;IS THE PS 11?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;THE PS IS NOT EQUAL TO 11  
          62      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
          CMP     #100000,%3     ;IS THE RESULT 100000?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;R3 IS NOT EQUAL TO 100000  
          63      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
```

```
12556                                     :BY (013740 000172 000207)
12557
12558 047130 021527 000177             CMP      (R5),#177             ;IS $TESTN = #177?
12559 047134 001403                     BEQ      .+10                 ;IF NOT THEN GO TO HLT
12560 047136 004767 011500             JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12561                                     ;TEST IS IN WRONG SEQUENCE
12562 047142 000064                     64      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12563                                     :BY (013746 000172 000207)
12564
12565 047144 005215                     INC      (R5)
12566
12567
```

,



12568  
12569  
12570  
12571  
12572 047146 010767 131020  
12573 047152 010501  
12574 047154 012705 000000  
12575 047160 012705 052525  
12576 047164 000241  
12577 047166 073527 000000  
12578 047172 106737 042262  
12579 047176 122737 000000 042262  
12580 047204 001403  
12581 047206 004767 011430  
12582  
12583 047212 000065  
12584  
12585  
12586 047214 022705 052525  
12587 047220 001403  
12588 047222 004767 011414  
12589  
12590 047226 000066  
12591  
12592  
12593 047230 010105  
12594 047232 021527 000200  
12595 047236 001403  
12596 047240 004767 011376  
12597  
12598 047244 000067  
12599  
12600  
12601 047246 005215  
12602  
12603  
12604  
12605  
12606  
12607  
12608 047250 010767 130716  
12609 047254 012701 000000  
12610 047260 012701 020010  
12611 047264 000241  
12612 047266 073127 177763  
12613 047272 106737 042262  
12614 047276 122737 000000 042262  
12615 047304 001403  
12616 047306 004767 011330  
12617  
12618 047312 000070  
12619  
12620  
12621 047314 022701 000101  
12622 047320 001403  
12623 047322 004767 011314

```
*****
:TEST:200      52525 SHIFTED BY 0 = 52525  PS = 0
*****

TST200: MOV     PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV     R5,R1        ;SAVE R5
        MOV     #DUMMY,%X5   ;LOAD R5 WITH DUMMY
        MOV     #52525,%X5!1 ;LOAD R5!1 WITH 52525
        CLC
        ASHC    #0,%X5       ;SHIFT R5,R5!1 BY 0
        MFPS    @#PSWORD     ;SAVE PS
        CMPB    #0,@#PSWORD  ;IS THE PS 0?
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                          ;THE PS IS NOT EQUAL TO 0
        65      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                          ;BY (013746 000172 000207)

        CMP     #52525,%X5   ;IS THE RESULT 52525?
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                          ;R5 IS NOT EQUAL TO 52525
        66      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                          ;BY (013746 000172 000207)

        MOV     R1,R5        ;RESTORE R5
        CMP     (R5),#200    ;IS $TESTN = #200?
        BEQ     .+10        ;IF NOT THEN GO TO HLT
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                          ;TEST IS IN WRONG SEQUENCE
        67      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                          ;BY (013746 000172 000207)

        INC     (R5)

*****
:TEST:201      20010 SHIFTED BY -13. = 101  PS = 0
*****

TST201: MOV     PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV     #DUMMY,%X1   ;LOAD R1 WITH DUMMY
        MOV     #20010,%X1!1 ;LOAD R1!1 WITH 20010
        CLC
        ASHC    #-13.,%X1    ;SHIFT R1,R1!1 BY -13.
        MFPS    @#PSWORD     ;SAVE PS
        CMPB    #0,@#PSWORD  ;IS THE PS 0?
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                          ;THE PS IS NOT EQUAL TO 0
        70      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                          ;BY (013746 000172 000207)

        CMP     #101,%X1     ;IS THE RESULT 101?
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
```



12638  
12639  
12640  
12641  
12642 047346 010767 130620  
12643 047352 012703 000000  
12644 047356 012703 177777  
12645 047362 000241  
12646 047364 073327 000020  
12647 047370 106737 042262  
12648 047374 122737 000011 042262  
12649 047402 001403  
12650 047404 004767 011232  
12651  
12652 047410 000073  
12653  
12654  
12655 047412 022703 000000  
12656 047416 001403  
12657 047420 004767 011216  
12658  
12659 047424 000074  
12660  
12661  
12662 047426 021527 000202  
12663 047432 001403  
12664 047434 004767 011202  
12665  
12666 047440 000075  
12667  
12668  
12669 047442 005215  
12 )  
126.1  
12672  
12673  
12674  
12675  
12676 047444 010767 130522  
12677 047450 010501  
12678 047452 012705 000000  
12679 047456 012705 000001  
12680 047462 000241  
12681 047464 073527 177777  
12682 047470 106737 042262  
12683 047474 122737 000001 042262  
12684 047502 001403  
12685 047504 004767 011132  
12686  
12687 047510 000076  
12688  
12689  
12690 047512 022705 100000  
12691 047516 001403  
12692 047520 004767 011116  
12693

```
*****
:TEST:202      -1 SHIFTED BY 16. = 0 PS = 11
*****

TST202: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #DUMMY,%3     ;LOAD R3 WITH DUMMY
        MOV      #-1,%3!1     ;LOAD R3!1 WITH -1
        CLC
        ASHC     #16,%3        ;SHIFT R3,R3!1 BY 16.
        MFPS     @#PSWORD      ;SAVE PS
        CMPB     #11,@#PSWORD  ;IS THE PS 11?
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;THE PS IS NOT EQUAL TO 11
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #0,%3         ;IS THE RESULT 0?
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;R3 IS NOT EQUAL TO 0
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      (R5),#202     ;IS $TESTN = #202?
        BEQ      .+10         ;IF NOT THEN GO TO HLT
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;TEST IS IN WRONG SEQUENCE
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        INC      (R5)

*****
:TEST:203      1 SHIFTED BY -1 = 100000 PS = 1
*****

TST203: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      R5,R1         ;SAVE R5
        MOV      #DUMMY,%5     ;LOAD R5 WITH DUMMY
        MOV      #1,%5!1      ;LOAD R5!1 WITH 1
        CLC
        ASHC     #-1,%5        ;SHIFT R5,R5!1 BY -1
        MFPS     @#PSWORD      ;SAVE PS
        CMPB     #1,@#PSWORD  ;IS THE PS 1?
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;THE PS IS NOT EQUAL TO 1
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #100000,%5    ;IS THE RESULT 100000?
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;R5 IS NOT EQUAL TO 100000
```

|       |        |        |        |     |           |  |
|-------|--------|--------|--------|-----|-----------|--|
| 12694 | 047524 | 000077 |        | 77  |           | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403) |
| 12695 |        |        |        |     |           | :BY (013746 000172 000207)                 |
| 12696 |        |        |        |     |           |  |
| 12697 | 047526 | 010105 |        | MOV | R1,R5     | :RESTORE R5                                |
| 12698 | 047530 | 021527 | 000203 | CMP | (R5),#203 | :IS \$TESTN = #203?                        |
| 12699 | 047534 | 001403 |        | BEQ | +.10      | :IF NOT THEN GO TO HLT                     |
| 12700 | 047536 | 004767 | 011100 | JSR | PC,\$HLT  | :SEEN AN ERROR, GO TO THE HALT ROUTINE     |
| 12701 |        |        |        |     |           | :TEST IS IN WRONG SEQUENCE                 |
| 12702 | 047542 | 000100 |        | 100 |           | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403) |
| 12703 |        |        |        |     |           | :BY (013746 000172 000207)                 |
| 12704 |        |        |        |     |           |  |
| 12705 | 047544 | 005215 |        | INC | (R5)      |  |
| 12706 |        |        |        |     |           |  |
| 12707 |        |        |        |     |           |  |

12708  
12709  
12710  
12711  
12712 047546 010767 130420  
12713 047552 012701 000000  
12714 047556 012701 125252  
12715 047562 000241  
12716 047564 073127 177760  
12717 047570 106737 042262  
12718 047574 122737 000011 042262  
12719 047602 001403  
12720 047604 004767 011032  
12721  
12722 047610 000101  
12723  
12724  
12725 047612 022701 125252  
12726 047616 001403  
12727 047620 004767 011016  
12728  
12729 047624 000102  
12730  
12731  
12732 047626 021527 000204  
12733 047632 001403  
12734 047634 004767 011002  
12735  
12736 047640 000103  
12737  
12738  
12739 047642 005215  
12740  
12741  
12742  
12743  
12744  
12745  
12746 047644 010767 130322  
12747 047650 012702 125252  
12748 047654 012703 125252  
12749 047660 000241  
12750 047662 073227 000025  
12751 047666 106737 042262  
12752 047672 122737 000003 042262  
12753 047700 001403  
12754 047702 004767 010734  
12755  
12756 047706 000104  
12757  
12758  
12759 047710 022702 052500  
12760 047714 001403  
12761 047716 004767 010720  
12762  
12763 047722 000105

```
*****  
:TEST:204      125252 SHIFTED BY -16. = 125252 PS = 11  
*****  
TST204: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
          MOV      #DUMMY,%1    ;LOAD R1 WITH DUMMY  
          MOV      #125252,%1:1 ;LOAD R1:1 WITH 125252  
          CLC  
          ASHC     #-16.,%1      ;SHIFT R1,R1:1 BY -16.  
          MFPS     @#PSWORD      ;SAVE PS  
          CMPB    #11,@#PSWORD  ;IS THE PS 11?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                    ;THE PS IS NOT EQUAL TO 11  
          101     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                    ;BY (013746 000172 000207)  
  
          CMP     #125252,%1     ;IS THE RESULT 125252?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                    ;R1 IS NOT EQUAL TO 125252  
          102     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                    ;BY (013746 000172 000207)  
  
          CMP     (R5),#204      ;IS $TESTN = #204?  
          BEQ     .+10           ;IF NOT THEN GO TO HLT  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                    ;TEST IS IN WRONG SEQUENCE  
          103     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                    ;BY (013746 000172 000207)  
  
          INC     (R5)  
  
*****  
:TEST:205      125252 125252 SHIFTED BY 21. = 52500 000000 PS = 3  
*****  
TST205: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
          MOV      #125252,%2    ;LOAD R2 WITH 125252  
          MOV      #125252,%2:1  ;LOAD R2:1 WITH 125252  
          CLC  
          ASHC     #21.,%2       ;SHIFT R2,R2:1 BY 21.  
          MFPS     @#PSWORD      ;SAVE PS  
          CMPB    #3,@#PSWORD   ;IS THE PS 3?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                    ;THE PS IS NOT EQUAL TO 3  
          104     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                    ;BY (013746 000172 000207)  
  
          CMP     #52500,%2      ;IS THE RESULT 52500?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                    ;R2 IS NOT EQUAL TO 52500  
          105     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
```

```

12764                                     :BY (013740 000172 000207)
12765
12766 047724 022703 000000             CMP    #000000,%2!1      ;IS THE RESULT 000000?
12767 047730 001403                     BEQ    .+10
12768 047732 004767 010704             JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12769                                     ;R2!1 IS NOT EQUAL TO 000000
12770 047736 000106                     106    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12771                                     ;BY (013746 000172 000207)
12772
12773 047740 021527 000205             CMP    (R5),#205        ;IS $TESTN = #205?
12774 047744 001403                     BEQ    .+10            ;IF NOT THEN GO TO HLT
12775 047746 004767 010670             JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12776                                     ;TEST IS IN WRONG SEQUENCE
12777 047752 000107                     107    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12778                                     ;BY (013746 000172 000207)
12779
12780 047754 005215                     INC    (R5)
12781
12782
12783
12784 047756 012702 177771             MOV    #-7,%2
12785 047762 012703 042304             MOV    #S1,%3
12786 047766 012704 042306             MOV    #S2,%4
12787
    
```

12788  
12789  
12790  
12791  
12792 047772 010767 130174  
12793 047776 012700 125252  
12794 050002 012701 125252  
12795 050006 000241  
12796 050010 073067 172270  
12797 050014 106737 042262  
12798 050020 122737 000010 042262  
12799 050026 001403  
12800 050030 004767 010606  
12801  
12802 050034 000110  
12803  
12804  
12805 050036 022700 177525  
12806 050042 001403  
12807 050044 004767 010572  
12808  
12809 050050 000111  
12810  
12811  
12812 050052 022701 052525  
12813 050056 001403  
12814 050060  
12815 050060 004767 010556  
12816  
12817 050064 000112  
12818  
12819  
12820 050066 021527 000206  
12821 050072 001372  
12822 050074 005215  
12823  
12824  
12825  
12826  
12827  
12828  
12829 050076 010767 130070  
12830 050102 012700 125252  
12831 050106 012701 125252  
12832 050112 000241  
12833 050114 073077 172166  
12834 050120 106737 042262  
12835 050124 122737 000010 042262  
12836 050132 001403  
12837 050134 004767 010502  
12838  
12839 050140 000113  
12840  
12841  
12842 050142 022701 177525  
12843 050146 001403

```
*****  
:TEST:206      125252 125252 SHIFTED BY S1 = 177525 52525 PS = 10  
*****  
TST206: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
          MOV      #125252,%0    ;LOAD R0 WITH 125252  
          MOV      #125252,%0!1  ;LOAD R0!1 WITH 125252  
          CLC  
          ASHC     S1,%0          ;SHIFT R0,R0!1 BY S1  
          MFPS     @#PSWORD      ;SAVE PS  
          CMPB    #10,@#PSWORD  ;IS THE PS 10?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
          ;THE PS IS NOT EQUAL TO 10  
          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
          ;BY (013746 000172 000207)  
  
          CMP     #177525,%0     ;IS THE RESULT 177525?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
          ;R0 IS NOT EQUAL TO 177525  
          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
          ;BY (013746 000172 000207)  
  
          CMP     #52525,%0!1    ;IS THE RESULT 52525?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
          ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE  
          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
          ;BY (013746 000172 000207)  
  
          CMP     (R5),#206      ;IS THE $TESTN = #206?  
          BNE     1$            ;IF NOT THEN GO TO HLT ABOVE  
          INC     (R5)
```

```
*****  
:TEST:207      125252 125252 SHIFTED BY @S2 = 177525 52525 PS = 10  
*****  
TST207: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
          MOV      #125252,%0    ;LOAD R0 WITH 125252  
          MOV      #125252,%0!1  ;LOAD R0!1 WITH 125252  
          CLC  
          ASHC     @S2,%0        ;SHIFT R0,R0!1 BY @S2  
          MFPS     @#PSWORD      ;SAVE PS  
          CMPB    #10,@#PSWORD  ;IS THE PS 10?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
          ;THE PS IS NOT EQUAL TO 10  
          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
          ;BY (013746 000172 000207)  
  
          CMP     #177525,%0     ;IS THE RESULT 177525?  
          BEQ     .+10
```

|       |        |        |        |      |     |   |
|-------|--------|--------|--------|------|-----|---|
| 12844 | 050150 | 004767 | 010466 |      | JSR | PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE    |
| 12845 |        |        |        |      |     | :R0 IS NOT EQUAL TO 177525                        |
| 12846 | 050154 | 000114 |        |      | 114 | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)        |
| 12847 |        |        |        |      |     | :BY (013746 000172 000207)                        |
| 12848 |        |        |        |      |     |   |
| 12849 | 050156 | 022701 | 052525 |      | CMP | #52525,\$0!1 ;IS THE RESULT 52525?                |
| 12850 | 050162 | 001403 |        |      | BEQ | +.10  |
| 12851 | 050164 |        |        | 1\$: |     |   |
| 12852 | 050164 | 004767 | 010452 |      | JSR | PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE    |
| 12853 |        |        |        |      |     | :R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE |
| 12854 | 050170 | 000115 |        |      | 115 | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)        |
| 12855 |        |        |        |      |     | :BY (013746 000172 000207)                        |
| 12856 |        |        |        |      |     |   |
| 12857 | 050172 | 021527 | 000207 |      | CMP | (R5),#207 ;IS THE \$TESTN = #207?                 |
| 12858 | 050176 | 001372 |        |      | BNE | 1\$ ;IF NOT THEN GO TO HLT ABOVE                  |
| 12859 | 050200 | 005215 |        |      | INC | (R5)  |
| 12860 |        |        |        |      |     |   |
| 12861 |        |        |        |      |     |   |



12862  
12863  
12864  
12865  
12866 050202 010767 127764  
12867 050206 012700 125252  
12868 050212 012701 125252  
12869 050216 000241  
12870 050220 073037 042304  
12871 050224 106737 042262  
12872 050230 122737 000010 042262  
12873 050236 001403  
12874 050240 004767 010376  
12875  
12876 050244 000116  
12877  
12878  
12879 050246 022700 177525  
12880 050252 001403  
12881 050254 004767 010362  
12882  
12883 050260 000117  
12884  
12885  
12886 050262 022701 052525  
12887 050266 001403  
12888 050270  
12889 050270 004767 010346  
12890  
12891 050274 000120  
12892  
12893  
12894 050276 021527 000210  
12895 050302 001372  
12896 050304 005215  
12897  
12898  
12899  
12900  
12901  
12902  
12903 050306 010767 127660  
12904 050312 012700 125252  
12905 050316 012701 125252  
12906 050322 000241  
12907 050324 073013  
12908 050326 106737 042262  
12909 050332 122737 000010 042262  
12910 050340 001403  
12911 050342 004767 010274  
12912  
12913 050346 000121  
12914  
12915  
12916 050350 022700 177525  
12917 050354 001403

```
*****  
:TEST:210      125252 125252 SHIFTED BY @#S1 = 177525 52525 PS = 10  
*****  
TST210: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
          MOV      #125252,%0    ;LOAD R0 WITH 125252  
          MOV      #125252,%0!1  ;LOAD R0!1 WITH 125252  
          CLC  
          ASHC     @#S1,%0        ;SHIFT R0,R0!1 BY @#S1  
          MFPS     @#PSWORD      ;SAVE PS  
          CMPB    #10,@#PSWORD  ;IS THE PS 10?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                    ;THE PS IS NOT EQUAL TO 10  
          116     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                    ;BY (013746 000172 000207)  
  
          CMP     #177525,%0     ;IS THE RESULT 177525?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                    ;R0 IS NOT EQUAL TO 177525  
          117     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                    ;BY (013746 000172 000207)  
  
          CMP     #52525,%0!1    ;IS THE RESULT 52525?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                    ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE  
          120     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                    ;BY (013746 000172 000207)  
  
          CMP     (R5),#210      ;IS THE $TESTN = #210?  
          BNE     1$            ;IF NOT THEN GO TO HLT ABOVE  
          INC     (R5)  
  
*****  
:TEST:211      125252 125252 SHIFTED BY (3) = 177525 52525 PS = 10  
*****  
TST211: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
          MOV      #125252,%0    ;LOAD R0 WITH 125252  
          MOV      #125252,%0!1  ;LOAD R0!1 WITH 125252  
          CLC  
          ASHC     (3),%0        ;SHIFT R0,R0!1 BY (3)  
          MFPS     @#PSWORD      ;SAVE PS  
          CMPB    #10,@#PSWORD  ;IS THE PS 10?  
          BEQ     .+10  
          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                    ;THE PS IS NOT EQUAL TO 10  
          121     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                    ;BY (013746 000172 000207)  
  
          CMP     #177525,%0     ;IS THE RESULT 177525?  
          BEQ     .+10
```

```
12918 050356 004767 010260      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12919                                ;RO IS NOT EQUAL TO 177525
12920 050362 000122      122      ;TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
12921                                ;BY (013746 000172 000207)
12922
12923 050364 022701 052525      CMP      #52525,$0!1      ;IS THE RESULT 52525?
12924 050370 001403      BEQ      .+10
12925 050372      18:
12926 050372 004767 010244      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12927                                ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12928 050376 000123      123      ;TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
12929                                ;BY (013746 000172 000207)
12930
12931 050400 021527 000211      CMP      (R5),#211      ;IS THE $TESTN = #211?
12932 050404 001372      BNE      18      ;IF NOT THEN GO TO HLT ABOVE
12933 050406 005215      INC      (R5)
12934
12935
```

12936  
12937  
12938  
12939  
12940 050410 010767 127556  
12941 050414 012700 125252  
12942 050420 012701 125252  
12943 050424 000241  
12944 050426 073023  
12945 050430 106737 042262  
12946 050434 122737 000010 042262  
12947 050442 001403  
12948 050444 004767 010172  
12949  
12950 050450 000124  
12951  
12952  
12953 050452 022700 177525  
12954 050456 001403  
12955 050460 004767 010156  
12956  
12957 050464 000125  
12958  
12959  
12960 050466 022701 052525  
12961 050472 001403  
12962 050474  
12963 050474 004767 010142  
12964  
12965 050500 000126  
12966  
12967  
12968 050502 021527 000212  
12969 050506 001372  
12970 050510 005215  
12971  
12972  
12973  
12974  
12975  
12976  
12977 050512 010767 127454  
12978 050516 012700 125252  
12979 050522 012701 125252  
12980 050526 000241  
12981 050530 073043  
12982 050532 106737 042262  
12983 050536 122737 000010 042262  
12984 050544 001403  
12985 050546 004767 010070  
12986  
12987 050552 000127  
12988  
12989  
12990 050554 022700 177525  
12991 050560 001403

```
*****
:TEST:212      125252 125252 SHIFTED BY (3)+ = 177525 52525 PS = 10
*****

TST212: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0      ;LOAD R0 WITH 125252
        MOV      #125252,%0!1    ;LOAD R0!1 WITH 125252
        CLC
        ASHC     (3)+,%0          ;SHIFT R0,R0!1 BY (3)+
        MFPS     @#PSWORD        ;SAVE PS
        CMPB    #10,@#PSWORD     ;IS THE PS 10?
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;THE PS IS NOT EQUAL TO 10
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)
        124
        CMP     #177525,%0        ;IS THE RESULT 177525?
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;R0 IS NOT EQUAL TO 177525
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)
        125
        CMP     #52525,%0!1       ;IS THE RESULT 52525?
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)
        126
        CMP     (R5),#212          ;IS THE $TESTN = #212?
        BNE     1$                ;IF NOT THEN GO TO HLT ABOVE
        INC     (R5)
        1$:
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)
        126
        CMP     (R5),#212          ;IS THE $TESTN = #212?
        BNE     1$                ;IF NOT THEN GO TO HLT ABOVE
        INC     (R5)
        1$:
*****
:TEST:213      125252 125252 SHIFTED BY -(3) = 177525 52525 PS = 10
*****

TST213: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0      ;LOAD R0 WITH 125252
        MOV      #125252,%0!1    ;LOAD R0!1 WITH 125252
        CLC
        ASHC     -(3),%0          ;SHIFT R0,R0!1 BY -(3)
        MFPS     @#PSWORD        ;SAVE PS
        CMPB    #10,@#PSWORD     ;IS THE PS 10?
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;THE PS IS NOT EQUAL TO 10
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)
        127
        CMP     #177525,%0        ;IS THE RESULT 177525?
        BEQ     .+10
```

|       |        |        |        |     |     |   |
|-------|--------|--------|--------|-----|-----|---|
| 12992 | 050562 | 004767 | 010054 |     | JSR | PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE    |
| 12993 |        |        |        |     |     | ;RO IS NOT EQUAL TO 177525                        |
| 12994 | 050566 | 000130 |        |     | 130 | ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)        |
| 12995 |        |        |        |     |     | ;BY (013746 000172 000207)                        |
| 12996 |        |        |        |     |     |   |
| 12997 | 050570 | 022701 | 052525 |     | CMP | #52525,\$0!1 ;IS THE RESULT 52525?                |
| 12998 | 050574 | 001403 |        |     | BEQ | +.10  |
| 12999 | 050576 |        |        | 18: |     |   |
| 13000 | 050576 | 004767 | 010040 |     | JSR | PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE    |
| 13001 |        |        |        |     |     | ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE |
| 13002 | 050602 | 000131 |        |     | 131 | ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)        |
| 13003 |        |        |        |     |     | ;BY (013746 000172 000207)                        |
| 13004 |        |        |        |     |     |   |
| 13005 | 050604 | 021527 | 000213 |     | CMP | (R5),#213 ;IS THE \$TESTN = #213?                 |
| 13006 | 050610 | 001372 |        |     | BNE | 18 ;IF NOT THEN GO TO HLT ABOVE                   |
| 13007 | 050612 | 005215 |        |     | INC | (R5)  |
| 13008 |        |        |        |     |     |   |
| 13009 |        |        |        |     |     |   |

```
13010 ;*****
13011 ;TEST:214      125252 125252 SHIFTED BY 2(4) = 177252 125252 PS = 11
13012 ;*****
13013
13014 050614 010767 127352 TST214: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13015 050620 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
13016 050624 012701 125252 MOV #125252,%0!1 ;LOAD R0!1 WITH 125252
13017 050630 000241 CLC
13018 050632 073064 000002 ASHC 2(4),%0 ;SHIFT R0,R0!1 BY 2(4)
13019 050636 106737 042262 MFPS @#PSWORD ;SAVE PS
13020 050642 122737 000011 042262 CMPB #11,@#PSWORD ;IS THE PS 11?
13021 050650 001403 BEQ .+10
13022 050652 004767 007764 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13023 ;THE PS IS NOT EQUAL TO 11
13024 050656 000132 132 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13025 ;BY (013746 000172 000207)
13026
13027 050660 022700 177252 CMP #177252,%0 ;IS THE RESULT 177252?
13028 050664 001403 BEQ .+10
13029 050666 004767 007750 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13030 ;RO IS NOT EQUAL TO 177252
13031 050672 000133 133 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13032 ;BY (013746 000172 000207)
13033
13034 050674 022701 125252 CMP #125252,%0!1 ;IS THE RESULT 125252?
13035 050700 001403 BEQ .+10
13036 050702 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13037 050702 004767 007734 ;R0!1 IS NOT EQUAL TO 125252 OR INCORRECT SEQUENCE
13038 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13039 050706 000134 134 ;BY (013746 000172 000207)
13040
13041
13042 050710 021527 000214 CMP (R5),#214 ;IS THE $TESTN = #214?
13043 050714 001372 BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
13044 050716 005215 INC (R5)
13045
13046
13047 ;*****
13048 ;TEST:215      125252 125252 SHIFTED BY @ (4) = 177525 52525 PS = 10
13049 ;*****
13050
13051 050720 010767 127246 TST215: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13052 050724 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
13053 050730 012701 125252 MOV #125252,%0!1 ;LOAD R0!1 WITH 125252
13054 050734 000241 CLC
13055 050736 073074 000000 ASHC @ (4),%0 ;SHIFT R0,R0!1 BY @ (4)
13056 050742 106737 042262 MFPS @#PSWORD ;SAVE PS
13057 050746 122737 000010 042262 CMPB #10,@#PSWORD ;IS THE PS 10?
13058 050754 001403 BEQ .+10
13059 050756 004767 007660 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13060 ;THE PS IS NOT EQUAL TO 10
13061 050762 000135 135 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13062 ;BY (013746 000172 000207)
13063
13064 050764 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
13065 050770 001403 BEQ .+10
```

|       |        |        |        |      |   |
|-------|--------|--------|--------|------|---|
| 13066 | 050772 | 004767 | 007644 | JSR  | PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE    |
| 13067 |        |        |        |      | :R0 IS NOT EQUAL TO 177525                        |
| 13068 | 050776 | 000136 |        | 136  | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)        |
| 13069 |        |        |        |      | :BY (013746 000172 000207)                        |
| 13070 |        |        |        |      |   |
| 13071 | 051000 | 022701 | 052525 | CMP  | #52525,R0!1 ;IS THE RESULT 52525?                 |
| 13072 | 051004 | 001403 |        | BEQ  | +.10  |
| 13073 | 051006 |        |        | 1\$: |   |
| 13074 | 051006 | 004767 | 007630 | JSR  | PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE    |
| 13075 |        |        |        |      | :R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE |
| 13076 | 051012 | 000137 |        | 137  | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)        |
| 13077 |        |        |        |      | :BY (013746 000172 000207)                        |
| 13078 |        |        |        |      |   |
| 13079 | 051014 | 021527 | 000215 | CMP  | (R5),#215 ;IS THE \$TESTN = #215?                 |
| 13080 | 051020 | 001372 |        | BNE  | 1\$ ;IF NOT THEN GO TO HLT ABOVE                  |
| 13081 | 051022 | 005215 |        | INC  | (R5)  |
| 13082 |        |        |        |      |   |
| 13083 |        |        |        |      |   |

13084  
13085  
13086  
13087  
13088 051024 010767 127142  
13089 051030 012700 125252  
13090 051034 012701 125252  
13091 051040 000241  
13092 051042 073034  
13093 051044 106737 042262  
13094 051050 122737 000010 042262  
13095 051056 001403  
13096 051060 004767 007556  
13097  
13098 051064 000140  
13099  
13100  
13101 051066 022700 177525  
13102 051072 001403  
13103 051074 004767 007542  
13104  
13105 051100 000141  
13106  
13107  
13108 051102 022701 052525  
13109 051106 001403  
13110 051110  
13111 051110 004767 007526  
13112  
13113 051114 000142  
13114  
13115  
13116 051116 021527 000216  
13117 051122 001372  
13118 051124 005215  
13119  
13120  
13121  
13122  
13123  
13124  
13125 051126 010767 127040  
13126 051132 012700 125252  
13127 051136 012701 125252  
13128 051142 000241  
13129 051144 073054  
13130 051146 106737 042262  
13131 051152 122737 000010 042262  
13132 051160 001403  
13133 051162 004767 007454  
13134  
13135 051166 000143  
13136  
13137  
13138 051170 022700 177525  
13139 051174 001403

```
*****
:TEST:216      125252 125252 SHIFTED BY @ (4)+ = 177525 52525 PS = 10
*****

TST216: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0     ;LOAD R0 WITH 125252
        MOV      #125252,%0!1   ;LOAD R0!1 WITH 125252
        CLC
        ASHC     @ (4)+,%0       ;SHIFT R0,R0!1 BY @ (4)+
        MFPS     @#PSWORD       ;SAVE PS
        CMPB     #10,@#PSWORD   ;IS THE PS 10?
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;THE PS IS NOT EQUAL TO 10
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #177525,%0     ;IS THE RESULT 177525?
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;R0 IS NOT EQUAL TO 177525
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #52525,%0!1    ;IS THE RESULT 52525?
        BEQ      .+10
1$:     JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      (R5),#216      ;IS THE $TESTN = #216?
        BNE      1$            ;IF NOT THEN GO TO HLT ABOVE
        INC      (R5)

*****
:TEST:217      125252 125252 SHIFTED BY @-(4) = 177525 52525 PS = 10
*****

TST217: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0     ;LOAD R0 WITH 125252
        MOV      #125252,%0!1   ;LOAD R0!1 WITH 125252
        CLC
        ASHC     @-(4),%0       ;SHIFT R0,R0!1 BY @-(4)
        MFPS     @#PSWORD       ;SAVE PS
        CMPB     #10,@#PSWORD   ;IS THE PS 10?
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;THE PS IS NOT EQUAL TO 10
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #177525,%0     ;IS THE RESULT 177525?
        BEQ      .+10
```

|       |        |        |        |      |     |   |
|-------|--------|--------|--------|------|-----|---|
| 13140 | 051176 | 004767 | 007440 |      | JSR | PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE    |
| 13141 |        |        |        |      |     | :RO IS NOT EQUAL TO 177525                        |
| 13142 | 051202 | 000144 |        |      | 144 | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)        |
| 13143 |        |        |        |      |     | :BY (013746 000172 000207)                        |
| 13144 |        |        |        |      |     |   |
| 13145 | 051204 | 022701 | 052525 |      | CMP | #52525,\$0!1 ;IS THE RESULT 52525?                |
| 13146 | 051210 | 001403 |        |      | BEQ | +.10  |
| 13147 | 051212 |        |        | 1\$: |     |   |
| 13148 | 051212 | 004767 | 007424 |      | JSR | PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE    |
| 13149 |        |        |        |      |     | :RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE |
| 13150 | 051216 | 000145 |        |      | 145 | :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)        |
| 13151 |        |        |        |      |     | :BY (013746 000172 000207)                        |
| 13152 |        |        |        |      |     |   |
| 13153 | 051220 | 021527 | 000217 |      | CMP | (R5),#217 ;IS THE \$TESTN = #217?                 |
| 13154 | 051224 | 001372 |        |      | BNE | 1\$ ;IF NOT THEN GO TO HLT ABOVE                  |
| 13155 | 051226 | 005215 |        |      | INC | (R5)  |
| 13156 |        |        |        |      |     |   |
| 13157 |        |        |        |      |     |   |
| 13158 |        |        |        |      |     |   |
| 13159 |        |        |        |      |     |   |
| 13160 |        |        |        |      |     |   |
| 13161 |        |        |        |      |     |   |
| 13162 |        |        |        |      |     |   |
| 13163 |        |        |        |      |     |   |
| 13164 |        |        |        |      |     |   |



13165  
13166  
13167  
13168  
13169  
13170  
13171  
13172  
13173  
13174  
13175  
13176  
13177  
13178  
13179  
13180  
13181  
13182  
13183  
13184  
13185  
13186  
13187  
13188  
13189  
13190  
13191  
13192  
13193  
13194  
13195  
13196  
13197  
13198  
13199  
13200  
13201  
13202  
13203  
13204  
13205  
13206

\*\*\*\*\*  
MUL INSTRUCTION TESTS  
\*\*\*\*\*

\*\*\*\*\*  
:TEST:220 MUL 1 \* #0 = 0 0 PS = 4  
\*\*\*\*\*

```
TST220: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
        MOV #1,%0 ;LOAD MULTIPLICAND WITH 1
        MUL #0,%0 ;MULTIPLY 1 * #0
        MFPS @#PSWORD ;SAVE PS
        CMPB #4,@#PSWORD ;IS PS = 4
        BEQ .+10
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;PS IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP #0,%0 ;IS HIGH ORDER = 0
        BEQ .+10
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;HIGH ORDER IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP #0,%0!1 ;IS LOW ORDER = 0
        BEQ .+10
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP (R5),#220
        BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        INC (R5)
```

042262

1\$:

13207  
 13208  
 13209  
 13210  
 13211  
 13212  
 13213  
 13214  
 13215  
 13216  
 13217  
 13218  
 13219  
 13220  
 13221  
 13222  
 13223  
 13224  
 13225  
 13226  
 13227  
 13228  
 13229  
 13230  
 13231  
 13232  
 13233  
 13234  
 13235  
 13236  
 13237  
 13238  
 13239  
 13240  
 13241

051326 010767 126640  
 051332 012700 177777  
 051336 070027 000001  
 051342 106737 042262  
 051346 122737 000010 042262  
 051354 001403  
 051356 004767 007260  
 051362 000151  
 051364 022700 177777  
 051370 001403  
 051372 004767 007244  
 051376 000152  
 051400 022701 177777  
 051404 001403  
 051406  
 051406 004767 007230  
 051412 000153  
 051414 021527 000221  
 051420 001372  
 051422 005215

```

:*****
:TEST:221      MUL      -1 * #1 = -1 -1      PS = 10
:*****
TST221: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #-1,%0      ;LOAD MULTIPLICAND WITH -1
        MUL      #1,%0      ;MULTIPLY -1 * #1
        MFPS     @#PSWORD    ;SAVE PS
        CMPB    #10,@#PSWORD ;IS PS = 10
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                        ;PS IS WRONG
        BEQ     .+10 (001403) ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        BY     (013746 000172 000207)
        CMP     #-1,%0      ;IS HIGH ORDER = -1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                        ;HIGH ORDER IS WRONG
        BEQ     .+10 (001403) ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        BY     (013746 000172 000207)
        CMP     #-1,%0!1    ;IS LOW ORDER = -1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
        BEQ     .+10 (001403) ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        BY     (013746 000172 000207)
        CMP     (R5),#221
        BNE     1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        INC     (R5)
1$:
  
```

13242  
13243  
13244  
13245  
13246 051424 010767 126542  
13247 051430 012702 000002  
13248 051434 070227 000002  
13249 051440 106737 042262  
13250 051444 122737 000000 042262  
13251 051452 001403  
13252 051454 004767 007162  
13253  
13254 051460 000154  
13255  
13256  
13257 051462 022702 000000  
13258 051466 001403  
13259 051470 004767 007146  
13260  
13261 051474 000155  
13262  
13263  
13264 051476 022703 000004  
13265 051502 001403  
13266 051504  
13267 051504 004767 007132  
13268  
13269 051510 000156  
13270  
13271  
13272 051512 021527 000222  
13273 051516 001372  
13274 051520 005215  
13275  
13276

```
*****  
:TEST:222      MUL      2 * #2 = 0 4      PS = 0  
*****  
TST222: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #2,%2        ;LOAD MULTIPLICAND WITH 2  
        MUL      #2,%2        ;MULTIPLY 2 * #2  
        MFPS     @#PSWORD     ;SAVE PS  
        CMPB    #0,@#PSWORD   ;IS PS = 0  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;PS IS WRONG  
        154      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     #0,%2        ;IS HIGH ORDER = 0  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;HIGH ORDER IS WRONG  
        155      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     #4,%2!1      ;IS LOW ORDER = 4  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        156      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     (R5),#222  
        BNE     1$           ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```

```

13277
13278
13279
13280
13281 051522 010767 126444
13282 051526 010501
13283 051530 012704 001000
13284 051534 070427 000200
13285 051540 106737 042262
13286 051544 122737 000001 042262
13287 051552 001403
13288 051554 004767 007062
13289
13290 051560 000157
13291
13292
13293 051562 022704 000001
13294 051566 001403
13295 051570 004767 007046
13296
13297 051574 000160
13298
13299
13300 051576 022705 000000
13301 051602 001403
13302 051604
13303 051604 004767 007032
13304
13305 051610 000161
13306
13307
13308 051612 021127 000223
13309 051616 001372
13310 051620 010105
13311 051622 005215
13312
13313
  
```

```

;*****
;TEST:223      MUL      1000 * #200 = 1 0      PS = 1
;*****
TST223: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      R5,R1        ;SAVE R5
        MOV      #1000,R4     ;LOAD MULTIPLICAND WITH 1000
        MUL      #200,R4      ;MULTIPLY 1000 * #200
        MFPS     @#PSWORD     ;SAVE PS
        CMPB     #1,@#PSWORD  ;IS PS = 1
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;PS IS WRONG
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)
        157
        CMP      #1,R4        ;IS HIGH ORDER = 1
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;HIGH ORDER IS WRONG
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)
        160
        CMP      #0,R4!1      ;IS LOW ORDER = 0
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)
        161
        CMP      (R1),#223    ;CHECK THE TEST NUMBER
        BNE      1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        MOV      R1,R5       ;RESTORE R5
        INC      (R5)
  
```

1\$:

13314  
13315  
13316  
13317  
13318 051624 010767 126342  
13319 051630 012700 000002  
13320 051634 070027 077777  
13321 051640 106737 042262  
13322 051644 122737 000001 042262  
13323 051652 001403  
13324 051654 004767 006762  
13325  
13326 051660 000162  
13327  
13328  
13329 051662 022700 000000  
13330 051666 001403  
13331 051670 004767 006746  
13332  
13333 051674 000163  
13334  
13335  
13336 051676 022701 177776  
13337 051702 001403  
13338 051704  
13339 051704 004767 006732  
13340  
13341 051710 000164  
13342  
13343  
13344 051712 021527 000224  
13345 051716 001372  
13346 051720 005215  
13347  
13348

```
*****  
:TEST:224      MUL      2 * #77777 = 0 177776      PS = 1  
*****  
TST224: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #2,%0        ;LOAD MULTIPLICAND WITH 2  
        MUL      #77777,%0     ;MULTIPLY 2 * #77777  
        MFPS     @#PSWORD     ;SAVE PS  
        CMPB    #1,@#PSWORD   ;IS PS = 1  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;PS IS WRONG  
                               ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
        CMP     #0,%0          ;IS HIGH ORDER = 0  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;HIGH ORDER IS WRONG  
                               ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
        CMP     #177776,%0!1   ;IS LOW ORDER = 177776  
        BEQ     .+10  
1$:     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
                               ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
        CMP     (R5),#224  
        BNE     1$             ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```

13349  
13350  
13351  
13352  
13353  
13354  
13355  
13356  
13357  
13358  
13359  
13360  
13361  
13362  
13363  
13364  
13365  
13366  
13367  
13368  
13369  
13370  
13371  
13372  
13373  
13374  
13375  
13376  
13377  
13378  
13379  
13380  
13381  
13382  
13383

051722 010767 126244  
051726 012702 007777  
051732 070227 000010  
051736 106737 042262  
051742 122737 000000 042262  
051750 001403  
051752 004767 006664  
  
051756 000165  
  
051760 022702 000000  
051764 001403  
051766 004767 006650  
  
051772 000166  
  
051774 022703 077770  
052000 001403  
052002 004767 006634  
052006 000167  
  
052010 021527 000225  
052014 001372  
052016 005215

```
*****  
:TEST:225      MUL      7777 * #10 = 0 77770      PS = 0  
*****  
TST225: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #7777,%2      ;LOAD MULTIPLICAND WITH 7777  
        MUL      #10,%2        ;MULTIPLY 7777 * #10  
        MFPS     @#PSWORD      ;SAVE PS  
        CMPB    #0,@#PSWORD    ;IS PS = 0  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;PS IS WRONG  
                               ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     #0,%2          ;IS HIGH ORDER = 0  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;HIGH ORDER IS WRONG  
                               ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     #77770,%2!1    ;IS LOW ORDER = 77770  
        BEQ     .+10  
  
1$:     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
                               ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     (R5),#225  
        BNE     1$             ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```

```

13384
13385
13386
13387
13388 052020 010767 126146
13389 052024 010501
13390 052026 012704 077777
13391 052032 070427 077777
13392 052036 106737 042262
13393 052042 122737 000001 042262
13394 052050 001403
13395 052052 004767 006564
13396
13397 052056 000170
13398
13399
13400 052060 022704 037777
13401 052064 001403
13402 052066 004767 006550
13403
13404 052072 000171
13405
13406
13407 052074 022705 000001
13408 052100 001403
13409 052102
13410 052102 004767 006534
13411
13412 052106 000172
13413
13414
13415 052110 021127 000226
13416 052114 001372
13417 052116 010105
13418 052120 005215
13419
13420
  
```

```

:*****
:TEST:226      MUL      77777 * #77777 = 37777 !      PS = 1
:*****
TST226: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      R5,R1        ;SAVE R5
        MOV      #77777,%4    ;LOAD MULTIPLICAND WITH 77777
        MUL      #77777,%4    ;MULTIPLY 77777 * #77777
        MFPS     @#PSWORD     ;SAVE PS
        CMPB    #1,@#PSWORD   ;IS PS = 1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;PS IS WRONG
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)
        CMP     #37777,%4     ;IS HIGH ORDER = 37777
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;HIGH ORDER IS WRONG
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)
        CMP     #1,%4!1      ;IS LOW ORDER = 1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)
        CMP     (R1),#226     ;CHECK THE TEST NUMBER
        BNE     1$           ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        MOV     R1,R5        ;RESTORE R5
        INC     (R5)
1$:
  
```

13421  
13422  
13423  
13424  
13425 052122 010767 126044  
13426 052126 012702 177777  
13427 052132 070227 077777  
13428 052136 106737 042262  
13429 052142 122737 000010 042262  
13430 052150 001403  
13431 052152 004767 006464  
13432  
13433 052156 000173  
13434  
13435  
13436 052160 022702 177777  
13437 052164 001403  
13438 052166 004767 006450  
13439  
13440 052172 000174  
13441  
13442  
13443 052174 022703 100001  
13444 052200 001403  
13445 052202  
13446 052202 004767 006434  
13447  
13448 052206 000175  
13449  
13450  
13451 052210 021527 000227  
13452 052214 001372  
13453 052216 005215  
13454  
13455

```
*****
:TEST:227      MUL      -1 * #77777 = -1 10001      PS = 10
*****

TST227: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #-1,%2      ;LOAD MULTIPLICAND WITH -1
        MUL      #77777,%2    ;MULTIPLY -1 * #77777
        MFPS     @#PSWORD     ;SAVE PS
        CMPB    #10,@#PSWORD ;IS PS = 10
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                        ;PS IS WRONG
                        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                        ;BY (013746 000172 000207)

        CMP     #-1,%2      ;IS HIGH ORDER = -1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                        ;HIGH ORDER IS WRONG
                        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                        ;BY (013746 000172 000207)

        CMP     #100001,%2!1 ;IS LOW ORDER = 100001
        BEQ     .+10
1$:     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                        ;BY (013746 000172 000207)

        CMP     (R5),#227
        BNE     1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        INC     (R5)
```



13456  
13457  
13458  
13459  
13460 052220 010767 125746  
13461 052224 012700 177776  
13462 052230 070027 077777  
13463 052234 106737 042262  
13464 052240 122737 000011 042262  
13465 052246 001403  
13466 052250 004767 006366  
13467  
13468 052254 000176  
13469  
13470  
13471 052256 022700 177777  
13472 052262 001403  
13473 052264 004767 006352  
13474  
13475 052270 000177  
13476  
13477  
13478 052272 022701 000002  
13479 052276 001403  
13480 052300  
13481 052300 004767 006336  
13482  
13483 052304 000200  
13484  
13485  
13486 052306 021527 000230  
13487 052312 001372  
13488 052314 005215  
13489  
13490

```
*****
;TEST:230      MUL      -2 * #77777 = -1 2      PS = 11
*****

TST230: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #-2,%0        ;LOAD MULTIPLICAND WITH -2
        MUL      #77777,%0     ;MULTIPLY -2 * #77777
        MFPS     @#PSWORD      ;SAVE PS
        CMPB    #11,@#PSWORD   ;IS PS = 11
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;PS IS WRONG
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)

        CMP     #-1,%0         ;IS HIGH ORDER = -1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;HIGH ORDER IS WRONG
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)

        CMP     #2,%0!1        ;IS LOW ORDER = 2
        BEQ     .+10

1$:     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)

        CMP     (R5),#230
        BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        INC     (R5)
```

13491  
13492  
13493  
13494  
13495 052316 010767 125650  
13496 052322 012702 125252  
13497 052326 070227 000002  
13498 052332 106737 042262  
13499 052336 122737 000011 042262  
13500 052344 001403  
13501 052346 004767 006270  
13502  
13503 052352 000201  
13504  
13505  
13506 052354 022702 177777  
13507 052360 001403  
13508 052362 004767 006254  
13509  
13510 052366 000202  
13511  
13512  
13513 052370 022703 052524  
13514 052374 001403  
13515 052376  
13516 052376 004767 006240  
13517  
13518 052402 000203  
13519  
13520  
13521 052404 021527 000231  
13522 052410 001372  
13523 052412 005215  
13524  
13525

```
*****  
:TEST:231      MUL      125252 * #2 = -1 52524      PS = 11  
*****  
TST231: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #125252,#2    ;LOAD MULTIPLICAND WITH 125252  
        MUL      #2,#2        ;MULTIPLY 125252 * #2  
        MFPS     @#PSWORD     ;SAVE PS  
        CMPB    #11,@#PSWORD  ;IS PS = 11  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;PS IS WRONG  
        201     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #-1,#2        ;IS HIGH ORDER = -1  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;HIGH ORDER IS WRONG  
        202     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #52524,#2:1    ;IS LOW ORDER = 52524  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        203     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     (R5),#231  
        BNE     1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```

```
13526
13527
13528
13529
13530 052414 010767 125552
13531 052420 010501
13532 052422 012704 125252
13533 052426 070427 040000
13534 052432 106737 042262
13535 052436 122737 000011 042262
13536 052444 001403
13537 052446 004767 006170
13538
13539 052452 000204 204
13540
13541
13542 052454 022704 165252
13543 052460 001403
13544 052462 004767 006154
13545
13546 052466 000205 205
13547
13548
13549 052470 022705 100000
13550 052474 001403
13551 052476
13552 052476 004767 006140
13553
13554 052502 000206 206
13555
13556
13557 052504 021127 000232
13558 052510 001372
13559 052512 010105
13560 052514 005215
13561
13562
```

\*\*\*\*\*  
:TEST:232 MUL 125252 \* #40000 = 165252 100000 PS = 11  
\*\*\*\*\*

TST232: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV R5,R1 ;SAVE R5  
MOV #125252,%4 ;LOAD MULTIPLICAND WITH 125252  
MUL #40000,%4 ;MULTIPLY 125252 \* #40000  
MFPS @#PSWORD ;SAVE PS  
CMPB #11,@#PSWORD ;IS PS = 11  
BEQ .+10  
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;PS IS WRONG  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

CMP #165252,%4 ;IS HIGH ORDER = 165252  
BEQ .+10  
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;HIGH ORDER IS WRONG  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

CMP #100000,%4!1 ;IS LOW ORDER = 100000  
BEQ .+10

1\$: JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;LOW ORDER IS WRONG OR WRONG SEQUENCE  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

CMP (R1),#232 ;CHECK THE TEST NUMBER  
BNE 1\$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
MOV R1,R5 ;RESTORE R5  
INC (R5)

13563  
 13564  
 13565  
 13566  
 13567  
 13568  
 13569  
 13570  
 13571  
 13572  
 13573  
 13574  
 13575  
 13576  
 13577  
 13578  
 13579  
 13580  
 13581  
 13582  
 13583  
 13584  
 13585  
 13586  
 13587  
 13588  
 13589  
 13590  
 13591  
 13592  
 13593  
 13594  
 13595  
 13596  
 13597

052516 010767 125450  
 052522 012700 107070  
 052526 070027 107070  
 052532 106737 042262  
 052536 122737 000001 042262  
 052544 001403  
 052546 004767 006070  
  
 052552 000207  
  
 052554 022700 031222  
 052560 001403  
 052562 004767 006054  
  
 052566 000210  
  
 052570 022701 026100  
 052574 001403  
 052576  
 052576 004767 006040  
  
 052602 000211  
  
 052604 021527 000233  
 052610 001372  
 052612 005215

```

:*****
:TEST:233      MUL      107070 * #107070 = 31222 26100      PS = 1
:*****
TST233: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #107070,%0    ;LOAD MULTIPLICAND WITH 107070
        MUL      #107070,%0    ;MULTIPLY 107070 * #107070
        MFPS     @#PSWORD      ;SAVE PS
        CMPB    #1,@#PSWORD    ;IS PS = 1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;PS IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     #31222,%0      ;IS HIGH ORDER = 31222
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;HIGH ORDER IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     #26100,%0!1    ;IS LOW ORDER = 26100
        BEQ     .+10
1$:     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     (R5),#233
        BNE     1$             ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        INC     (R5)
  
```

13598  
13599  
13600  
13601  
13602 052614 010767 125352  
13603 052620 012701 177777  
13604 052624 070127 000001  
13605 052630 106737 042262  
13606 052634 122737 000010 042262  
13607 052642 001403  
13608 052644 004767 005772  
13609  
13610 052650 000212  
13611  
13612  
13613 052652 022701 177777  
13614 052656 001403  
13615 052660 004767 005756  
13616  
13617 052664 000213  
13618  
13619  
13620 052666 022701 177777  
13621 052672 001403  
13622 052674  
13623 052674 004767 005742  
13624  
13625 052700 000214  
13626  
13627  
13628 052702 021527 000234  
13629 052706 001372  
13630 052710 005215  
13631  
13632

```
*****  
:TEST:234      MUL      -1 * #1 = -1 -1      PS = 10  
*****  
TST234: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #-1,#1      ;LOAD MULTIPLICAND WITH -1  
        MUL      #1,#1      ;MULTIPLY -1 * #1  
        MFPS     @#PSWORD     ;SAVE PS  
        CMPB    #10,@#PSWORD ;IS PS = 10  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;PS IS WRONG  
        212     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #-1,#1      ;IS HIGH ORDER = -1  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;HIGH ORDER IS WRONG  
        213     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #-1,#1:1    ;IS LOW ORDER = -1  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        214     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     (R5),#234  
        BNE     1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```

13633  
13634  
13635  
13636  
13637  
13638  
13639  
13640  
13641  
13642  
13643  
13644  
13645  
13646  
13647  
13648  
13649  
13650  
13651  
13652  
13653  
13654  
13655  
13656  
13657  
13658  
13659  
13660  
13661  
13662  
13663  
13664  
13665  
13666  
13667

052712 010767 125254  
052716 012703 177777  
052722 070327 000000  
052726 106737 042262  
052732 122737 000004 042262  
052740 001403  
052742 004767 005674  
  
052746 000215  
  
052750 022703 000000  
052754 001403  
052756 004767 005660  
  
052762 000216  
  
052764 022703 000000  
052770 001403  
052772 004767 005644  
052776 000217  
  
053000 021527 000235  
053004 001372  
053006 005215

```
*****
:TEST:235      MUL      -1 * #0 = 0 0      PS = 4
*****
TST235: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #-1,#3      ;LOAD MULTIPLICAND WITH -1
        MUL      #0,#3      ;MULTIPLY -1 * #0
        MFPS     @#PSWORD     ;SAVE PS
        CMPB    #4,@#PSWORD  ;IS PS = 4
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;PS IS WRONG
        215     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     #0,#3      ;IS HIGH ORDER = 0
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;HIGH ORDER IS WRONG
        216     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     #0,#3!1    ;IS LOW ORDER = 0
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
        217     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     (R5),#235
        BNE     1$         ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        INC     (R5)
```

1\$:

```
13668  
13669  
13670  
13671  
13672 053010 010767 125156  
13673 053014 010501  
13674 053016 012705 077777  
13675 053022 070527 100000  
13676 053026 106737 042262  
13677 053032 122737 000011 042262  
13678 053040 001403  
13679 053042 004767 005574  
13680  
13681 053046 000220  
13682  
13683  
13684 053050 022705 100000  
13685 053054 001403  
13686 053056 004767 005560  
13687  
13688 053062 000221  
13689  
13690  
13691 053064 022705 100000  
13692 053070 001403  
13693 053072  
13694 053072 004767 005544  
13695  
13696 053076 000222  
13697  
13698  
13699 053100 021127 000236  
13700 053104 001372  
13701 053106 010105  
13702 053110 005215  
13703  
13704
```

```
*****  
:TEST:236      MUL      77777 * #100000 = 100000 100000      PS = 11  
*****  
TST236: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      R5,R1        ;SAVE R5  
        MOV      #77777,R5     ;LOAD MULTIPLICAND WITH 77777  
        MUL      #100000,R5    ;MULTIPLY 77777 * #100000  
        MFPS     @#PSWORD     ;SAVE PS  
        CMPB    #11,@#PSWORD  ;IS PS = 11  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;PS IS WRONG  
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #100000,R5     ;IS HIGH ORDER = 100000  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;HIGH ORDER IS WRONG  
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #100000,R5:1   ;IS LOW ORDER = 100000  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     (R1),#236      ;CHECK THE TEST NUMBER  
        BNE     1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        MOV     R1,R5        ;RESTORE R5  
        INC     (R5)
```

13705  
13706  
13707  
13708  
13709 053112 010767 125054  
13710 053116 012701 177777  
13711 053122 070127 077777  
13712 053126 106737 042262  
13713 053132 122737 000010 042262  
13714 053140 001403  
13715 053142 004767 005474  
13716  
13717 053146 000223  
13718  
13719  
13720 053150 022701 100001  
13721 053154 001403  
13722 053156 004767 005460  
13723  
13724 053162 000224  
13725  
13726  
13727 053164 022701 100001  
13728 053170 001403  
13729 053172  
13730 053172 004767 005444  
13731  
13732 053176 000225  
13733  
13734  
13735 053200 021527 000237  
13736 053204 001372  
13737 053206 005215  
13738  
13739

```
*****  
:TEST:237      MUL      -1 * #77777 = 100001 100001      PS = 10  
*****  
TST237: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #-1,21      ;LOAD MULTIPLICAND WITH -1  
        MUL      #77777,%1     ;MULTIPLY -1 * #77777  
        MFPS     @#PSWJRD     ;SAVE PS  
        CMPB    #10,@#PSWORD  ;IS PS = 10  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;PS IS WRONG  
        223     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #100001,%1     ;IS HIGH ORDER = 100001  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;HIGH ORDER IS WRONG  
        224     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #100001,%1:1   ;IS LOW ORDER = 100001  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        225     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     (R5),#237  
        BNE     1$           ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```



13740  
13741  
13742  
13743  
13744  
13745  
13746  
13747  
13748  
13749  
13750  
13751  
13752  
13753  
13754  
13755  
13756  
13757  
13758  
13759  
13760  
13761  
13762  
13763  
13764  
13765  
13766  
13767  
13768  
13769  
13770  
13771  
13772  
13773  
13774

053210 010767 124756  
053214 012703 077777  
053220 070327 077777  
053224 106737 042262  
053230 122737 000001 042262  
053236 001403  
053240 004767 005376  
  
053244 000226  
  
053246 022703 000001  
053252 001403  
053254 004767 005362  
  
053260 000227  
  
053262 022703 000001  
053266 001403  
053270  
053270 004767 005346  
053274 000230  
  
053276 021527 000240  
053302 001372  
053304 005215

```
*****  
:TEST:240      MUL      77777 * #77777 = 1 1      PS = 1  
*****  
TST240: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #77777,%3      ;LOAD MULTIPLICAND WITH 77777  
        MUL      #77777,%3      ;MULTIPLY 77777 * #77777  
        MFPS     @#PSWORD      ;SAVE PS  
        CMPB    #1,@#PSWORD    ;IS PS = 1  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;PS IS WRONG  
        226     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     #1,%3          ;IS HIGH ORDER = 1  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;HIGH ORDER IS WRONG  
        227     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     #1,%3!1       ;IS LOW ORDER = 1  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        230     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     (R5),#240  
        BNE     1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```

1\$:

13775  
13776  
13777  
13778  
13779 053306 010767 124660  
13780 053312 010501  
13781 053314 012705 000002  
13782 053320 070527 000002  
13783 053324 106737 042262  
13784 053330 122737 000000 042262  
13785 053336 001403  
13786 053340 004767 005276  
13787  
13788 053344 000231  
13789  
13790  
13791 053346 022705 000004  
13792 053352 001403  
13793 053354 004767 005262  
13794  
13795 053360 000232  
13796  
13797  
13798 053362 022705 000004  
13799 053366 001403  
13800 053370  
13801 053370 004767 005246  
13802  
13803 053374 000233  
13804  
13805  
13806 053376 021127 000241  
13807 053402 001372  
13808 053404 010105  
13809 053406 005215  
13810  
13811

```
*****  
:TEST:241      MUL      2 * #2 = 4 4      PS = 0  
*****  
TST241: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      R5,R1        ;SAVE R5  
        MOV      #2,R5        ;LOAD MULTIPLICAND WITH 2  
        MUL      #2,R5        ;MULTIPLY 2 * #2  
        MFPS     @#PSWORD     ;SAVE PS  
        CMPB    #0,@#PSWORD   ;IS PS = 0  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;PS IS WRONG  
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #4,R5        ;IS HIGH ORDER = 4  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;HIGH ORDER IS WRONG  
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #4,R5!1     ;IS LOW ORDER = 4  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     (R1),#241    ;CHECK THE TEST NUMBER  
        BNE     1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        MOV     R1,R5        ;RESTORE R5  
        INC     (R5)
```

13812 053410 012702 040000  
13813 053414 012703 042314  
13814 053420 012704 042316  
13815  
13816  
13817  
13818  
13819  
13820 053424 010767 124542  
13821 053430 012700 125252  
13822 053434 070067 166654  
13823 053440 106737 042262  
13824 053444 122737 000011 042262  
13825 053452 001403  
13826 053454 004767 005162  
13827  
13828 053460 000234  
13829  
13830  
13831 053462 022700 165252  
13832 053466 001403  
13833 053470 004767 005146  
13834  
13835 053474 000235  
13836  
13837  
13838 053476 022701 100000  
13839 053502 001403  
13840 053504  
13841 053504 004767 005132  
13842  
13843 053510 000236  
13844  
13845  
13846 053512 021527 000242  
13847 053516 001372  
13848 053520 005215  
13849  
13850

```
MOV #40000,X2
MOV #55,X3
MOV #56,X4
:*****
:TEST:242 MUL 125252 * S5 = 165252 100000 PS = 11
:*****
TST242: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #125252,X0 ;LOAD MULTIPLICAND WITH 125252
MUL S5,X0 ;MULTIPLY 125252 * S5
MFPS @#PSWORD ;SAVE PS
CMPB #11,@#PSWORD ;IS PS = 11
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;PS IS WRONG
234 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
CMP #165252,X0 ;IS HIGH ORDER = 165252
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;HIGH ORDER IS WRONG
235 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
CMP #100000,X0!1 ;IS LOW ORDER = 100000
BEQ .+10
1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;LOW ORDER IS WRONG OR WRONG SEQUENCE
236 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
CMP (R5),#242
BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
INC (R5)
```

13851  
13852  
13853  
13854  
13855  
13856  
13857  
13858  
13859  
13860  
13861  
13862  
13863  
13864  
13865  
13866  
13867  
13868  
13869  
13870  
13871  
13872  
13873  
13874  
13875  
13876  
13877  
13878  
13879  
13880  
13881  
13882  
13883  
13884  
13885

053522 010767 124444  
053526 012700 125252  
053532 070077 166560  
053536 106737 042262  
053542 122737 000011 042262  
053550 001403  
053552 004767 005064  
  
053556 000237  
  
053560 022700 165252  
053564 001403  
053566 004767 005050  
  
053572 000240  
  
053574 022701 100000  
053600 001403  
053602  
053602 004767 005034  
053606 000241  
  
053610 021527 000243  
053614 001372  
053616 005215

```
*****  
:TEST:243      MUL      125252 * @S6 = 165252 100000      PS = 11  
*****  
TST243: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #125252,X0    ;LOAD MULTIPLICAND WITH 125252  
        MUL      @S6,X0        ;MULTIPLY 125252 * @S6  
        MFPS     @#PSWORD      ;SAVE PS  
        CMPB    #11,@#PSWORD   ;IS PS = 11  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;PS IS WRONG  
        237      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     #165252,X0     ;IS HIGH ORDER = 165252  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;HIGH ORDER IS WRONG  
        240      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     #100000,X0!1   ;IS LOW ORDER = 100000  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                               ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        241      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                               ;BY (013746 000172 000207)  
  
        CMP     (R5),#243  
        BNE     1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```

```

13886
13887
13888
13889
13890 053620 010767 124346
13891 053624 012700 125252
13892 053630 070037 042314
13893 053634 106737 042262
13894 053640 122737 000011 042262
13895 053646 001403
13896 053650 004767 004766
13897
13898 053654 000242
13899
13900
13901 053656 022700 165252
13902 053662 001403
13903 053664 004767 004752
13904
13905 053670 000243
13906
13907
13908 053672 022701 100000
13909 053676 001403
13910 053700
13911 053700 004767 004736
13912
13913 053704 000244
13914
13915
13916 053706 021527 000244
13917 053712 001372
13918 053714 005215
13919
13920

```

```

:*****
:TEST:244      MUL      125252 * @#S5 = 165252 100000      PS = 11
:*****
TST244: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0    ;LOAD MULTIPLICAND WITH 125252
        MUL      @#S5,%0      ;MULTIPLY 125252 * @#S5
        MFPS     @#PSWORD     ;SAVE PS
        CMPB    #11,@#PSWORD  ;IS PS = 11
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;PS IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #165252,%0    ;IS HIGH ORDER = 165252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;HIGH ORDER IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #100000,%0!1  ;IS LOW ORDER = 100000
        BEQ     .+10
1$:     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      (R5),#244
        BNE     1$             ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        INC     (R5)

```

```
13921
13922
13923
13924
13925 053716 010767 124250
13926 053722 012700 125252
13927 053726 070002
13928 053730 106737 042262
13929 053734 122737 000011 042262
13930 053742 001403
13931 053744 004767 004672
13932
13933 053750 000245
13934
13935
13936 053752 022700 165252
13937 053756 001403
13938 053760 004767 004656
13939
13940 053764 000246
13941
13942
13943 053766 022700 100000
13944 053772 001403
13945 053774
13946 053774 004767 004642
13947
13948 054000 000247
13949
13950
13951 054002 021527 000245
13952 054006 001372
13953 054010 005215
13954
13955
```

```
*****
:TEST:245      MUL      125252 * %2 = 165252 100000      PS = 11
*****
TST245: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0     ;LOAD MULTIPLICAND WITH 125252
        MUL      %2,%0         ;MULTIPLY 125252 * %2
        MFPS     @#PSWORD      ;SAVE PS
        CMPB    #11,@#PSWORD   ;IS PS = 11
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;PS IS WRONG
        245      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)
        CMP     #165252,%0     ;IS HIGH ORDER = 165252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;HIGH ORDER IS WRONG
        246      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)
        CMP     #100000,%0!1   ;IS LOW ORDER = 100000
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
        247      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)
        CMP     (R5),#245
        BNE     1$             ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        INC     (R5)
```

13956  
13957  
13958  
13959  
13960  
13961  
13962  
13963  
13964  
13965  
13966  
13967  
13968  
13969  
13970  
13971  
13972  
13973  
13974  
13975  
13976  
13977  
13978  
13979  
13980  
13981  
13982  
13983  
13984  
13985  
13986  
13987  
13988  
13989  
13990

054012 010767 124154  
054016 012700 125252  
054022 070023  
054024 106737 042262  
054030 122737 000011 042262  
054036 001403  
054040 004767 004576  
  
054044 000250  
  
054046 022700 165252  
054052 001403  
054054 004767 004562  
  
054060 000251  
  
054062 022701 100000  
054066 001403  
054070  
054070 004767 004546  
  
054074 000252  
  
054076 021527 000246  
054102 001372  
054104 005215

```
*****  
:TEST:246      MUL      125252 * (3)+ = 165252 100000      PS = 11  
*****  
TST246: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #125252,%0     ;LOAD MULTIPLICAND WITH 125252  
        MUL      (3)+,%0       ;MULTIPLY 125252 * (3)+  
        MFPS     @#PSWORD      ;SAVE PS  
        CMPB    #11,@#PSWORD   ;IS PS = 11  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;PS IS WRONG  
        250     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #165252,%0     ;IS HIGH ORDER = 165252  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;HIGH ORDER IS WRONG  
        251     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #100000,%0!1   ;IS LOW ORDER = 100000  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        252     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     (R5),#246  
        BNE     1$             ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```

1\$:

13991  
13992  
13993  
13994  
13995 054106 010767 124060  
13996 054112 012700 125252  
13997 054116 070043  
13998 054120 106737 042262  
13999 054124 122737 000011 042262  
14000 054132 001403  
14001 054134 004767 004502  
14002  
14003 054140 000253  
14004  
14005  
14006 054142 022700 165252  
14007 054146 001403  
14008 054150 004767 004466  
14009  
14010 054154 000254  
14011  
14012  
14013 054156 022701 100000  
14014 054162 001403  
14015 054164  
14016 054164 004767 004452  
14017  
14018 054170 000255  
14019  
14020  
14021 054172 021527 000247  
14022 054176 001372  
14023 054200 005215  
14024  
14025

```
*****  
:TEST:247      MUL      125252 * -(3) = 165252 100000      PS = 11  
*****  
TST247: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #125252,%0     ;LOAD MULTIPLICAND WITH 125252  
        MUL      -(3),%0       ;MULTIPLY 125252 * -(3)  
        MFPS     @#PSWORD      ;SAVE PS  
        CMPB    #11,@#PSWORD   ;IS PS = 11  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;PS IS WRONG  
        253     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
        CMP     #165252,%0     ;IS HIGH ORDER = 165252  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;HIGH ORDER IS WRONG  
        254     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
        CMP     #100000,%0!1   ;IS LOW ORDER = 100000  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        255     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
        CMP     (R5),#247  
        BNE     1$             ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```



```
14026
14027
14028
14029
14030 054202 010767 123764
14031 054206 012700 125252
14032 054212 070064 000002
14033 054216 106737 042262
14034 054222 122737 000011 042262
14035 054230 001403
14036 054232 004767 004404
14037
14038 054236 000256
14039
14040
14041 054240 022700 165252
14042 054244 001403
14043 054246 004767 004370
14044
14045 054252 000257
14046
14047
14048 054254 022701 100000
14049 054260 001403
14050 054262
14051 054262 004767 004354
14052
14053 054266 000260
14054
14055
14056 054270 021527 000250
14057 054274 001372
14058 054276 005215
14059
14060
```

```
*****
:TEST:250      MUL      125252 * 2(4) = 165252 100000      PS = 11
*****
TST250: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0     ;LOAD MULTIPLICAND WITH 125252
        MUL      2(4),%0       ;MULTIPLY 125252 * 2(4)
        MFPS     @#PSWORD      ;SAVE PS
        CMPB    #11,@#PSWORD   ;IS PS = 11
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;PS IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     #165252,%0     ;IS HIGH ORDER = 165252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;HIGH ORDER IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     #100000,%0!1   ;IS LOW ORDER = 100000
        BEQ     .+10
1$:     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     (R5),#250
        BNE    1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        INC     (R5)
```

```
14061
14062
14063
14064
14065 054300 010767 123666
14066 054304 012700 125252
14067 054310 070074 000000
14068 054314 106737 042262
14069 054320 122737 000011 042262
14070 054326 001403
14071 054330 004767 004306
14072
14073 054334 000261
14074
14075
14076 054336 022700 165252
14077 054342 001403
14078 054344 004767 004272
14079
14080 054350 000262
14081
14082
14083 054352 022701 100000
14084 054356 001403
14085 054360
14086 054360 004767 004256
14087
14088 054364 000263
14089
14090
14091 054366 021527 000251
14092 054372 001372
14093 054374 005215
14094
14095

:*****
:TEST:251      MUL      125252 * @ (4) = 165252 100000      PS = 11
:*****

TST251: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0     ;LOAD MULTIPLICAND WITH 125252
        MUL      @ (4),%0       ;MULTIPLY 125252 * @ (4)
        MFPS     @#PSWORD      ;SAVE PS
        CMPB    #11,@#PSWORD   ;IS PS = 11
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;PS IS WRONG
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)

        CMP     #165252,%0      ;IS HIGH ORDER = 165252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;HIGH ORDER IS WRONG
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)

        CMP     #100000,%0!1    ;IS LOW ORDER = 100000
        BEQ     .+10
1$:     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)

        CMP     (R5),#251
        BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        INC     (R5)
```

14096  
14097  
14098  
14099  
14100  
14101  
14102  
14103  
14104  
14105  
14106  
14107  
14108  
14109  
14110  
14111  
14112  
14113  
14114  
14115  
14116  
14117  
14118  
14119  
14120  
14121  
14122  
14123  
14124  
14125  
14126  
14127  
14128  
14129  
14130

054376 010767 123570  
054402 012700 125252  
054406 070034  
054410 106737 042262  
054414 122737 000011 042262  
054422 001403  
054424 004767 004212  
054430 000264  
054432 022700 165252  
054436 001403  
054440 004767 004176  
054444 000265  
054446 022701 100000  
054452 001403  
054454  
054454 004767 004162  
054460 000266  
054462 021527 000252  
054466 001372  
054470 005215

```
*****  
:TEST:252      MUL      125252 * @ (4)+ = 165252 100000      PS = 11  
*****  
TST252: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #125252,%0     ;LOAD MULTIPLICAND WITH 125252  
        MUL      @ (4)+,%0      ;MULTIPLY 125252 * @ (4)+  
        MFPS     @#PSWORD      ;SAVE PS  
        CMPB    #11,@#PSWORD   ;IS PS = 11  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;PS IS WRONG  
        264     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #165252,%0     ;IS HIGH ORDER = 165252  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;HIGH ORDER IS WRONG  
        265     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #100000,%0!1    ;IS LOW ORDER = 100000  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;LOW ORDER IS WRONG OR WRONG SEQUENCE  
        266     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     (R5),#252  
        BNE     1$             ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
        INC     (R5)
```

```
14131
14132
14133
14134
14135 054472 010767 123474
14136 054476 012700 125252
14137 054502 070054
14138 054504 106737 042262
14139 054510 122737 000011 042262
14140 054516 001403
14141 054520 004767 004116
14142
14143 054524 000267
14144
14145
14146 054526 022700 165252
14147 054532 001403
14148 054534 004767 004102
14149
14150 054540 000270
14151
14152
14153 054542 022701 100000
14154 054546 001403
14155 054550
14156 054550 004767 004066
14157
14158 054554 000271
14159
14160
14161 054556 021527 000253
14162 054562 001372
14163 054564 005215
14164
14165

;*****
;TEST:253      MUL      125252 * @-(4) = 165252 100000      PS = 11
;*****

TST253: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0    ;LOAD MULTIPLICAND WITH 125252
        MUL      @-(4),%0      ;MULTIPLY 125252 * @-(4)
        MFPS     @#PSWORD      ;SAVE PS
        CMPB    #11,@#PSWORD   ;IS PS = 11
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;PS IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP     #165252,%0      ;IS HIGH ORDER = 165252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;HIGH ORDER IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP     #100000,%0!1    ;IS LOW ORDER = 100000
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP     (R5),#253
        BNE     1$
        INC     (R5)            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE

1$:
```

14166  
14167  
14168  
14169  
14170  
14171  
14172  
14173  
14174  
14175  
14176  
14177  
14178  
14179  
14180  
14181  
14182  
14183  
14184  
14185  
14186  
14187  
14188  
14189  
14190  
14191  
14192  
14193  
14194  
14195  
14196  
14197  
14198  
14199  
14200  
14201  
14202  
14203  
14204  
14205  
14206  
14207  
14208  
14209  
14210  
14211  
14212  
14213

054566 010767 123400  
054572 012700 000000  
054576 012701 000004  
054602 071027 000002  
054606 106737 042262  
054612 122737 000000 042262  
054620 001403  
054622 004767 004014  
054626 000272  
054630 022700 000002  
054634 001403  
054636 004767 004000  
054642 000273  
054644 022701 000000  
054650 001403  
054652 004767 003764  
054656 000274  
054660 021527 000254  
054664 001403  
054666 004767 003750  
054672 000275  
054674 005215

```
*****
:
: DIV INSTRUCTION TESTS
:
*****

:*****
:TEST:254 DIV 0 4 / #2 = 2 REM = 0 PS = 0
:*****

1ST254: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #0,%0 ;LOAD HIGH ORDER WITH 0
MOV #4,%0+1 ;LOAD LOW ORDER WITH 4
DIV #2,%0 ;DIVIDE BY #2
MFPS @#PSWORD ;SAVE PS

CMPB #0,@#PSWORD ;IS PS = 0
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;PS IS WRONG
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #2,%0 ;IS QUOTIENT = 2
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;QUOTIENT IS WRONG
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #0,%0+1 ;IS REMAINDER = 0
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;WRONG REMAINDER
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP (R5),#254
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

INC (R5)
```

14214  
14215  
14216  
14217  
14218 054676 010767 123270  
14219 054702 012702 177777  
14220 054706 012703 177767  
14221 054712 071227 000003  
14222 054716 106737 042262  
14223  
14224 054722 122737 000010 042262  
14225 054730 001403  
14226 054732 004767 003704  
14227  
14228 054736 000276  
14229  
14230  
14231  
14232 054740 022702 177775  
14233 054744 001403  
14234 054746 004767 003670  
14235  
14236 054752 000277  
14237  
14238  
14239  
14240 054754 022703 000000  
14241 054760 001403  
14242 054762 004767 003654  
14243  
14244 054766 000300  
14245  
14246  
14247 054770 021527 000255  
14248 054774 001403  
14249 054776 004767 003640  
14250  
14251 055002 000301  
14252  
14253  
14254 055004 005215  
14255

```
*****  
:TEST:255      DIV      -1 -9. / #3 = -3      REM = 0      PS = 10  
:*****  
TST255: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #-1,%2      ;LOAD HIGH ORDER WITH -1  
        MOV      #-9,%2+1     ;LOAD LOW ORDER WITH -9.  
        DIV      #3,%2      ;DIVIDE BY #3  
        MFPS     @#PSWORD     ;SAVE PS  
  
        CMPB    #10,@#PSWORD ;IS PS = 10  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                ;PS IS WRONG  
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #-3,%2      ;IS QUOTIENT = -3  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                ;QUOTIENT IS WRONG  
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     #0,%2+1     ;IS REMAINDER = 0  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                ;WRONG REMAINDER  
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     (R5),#255  
        BEQ     .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                ;TEST IS IN WRONG SEQUENCE  
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        INC     (R5)
```

14256  
14257  
14258  
14259  
14260 055006 010767 123160 -  
14261 055012 010501  
14262 055014 012704 000000  
14263 055020 012705 000011  
14264 055024 071427 000002  
14265 055030 106737 042262  
14266  
14267 055034 122737 000000 042262  
14268 055042 001403  
14269 055044 004767 003572  
14270  
14271 055050 000302  
14272  
14273  
14274  
14275 055052 022704 000004  
14276 055056 001403  
14277 055060 004767 003556  
14278  
14279 055064 000303  
14280  
14281  
14282  
14283 055066 022705 000001  
14284 055072 001403  
14285 055074 004767 003542  
14286  
14287 055100 000304  
14288  
14289  
14290 055102 010105  
14291 055104 021527 000256  
14292 055110 001403  
14293 055112 004767 003524  
14294  
14295 055116 000305  
14296  
14297  
14298 055120 005215  
14299

```
*****  
:TEST:256      DIV      0 9. / #2 = 4      REM = 1      PS = 0  
*****  
TST256: MOV      PC,LPADR      :STORE ERROR LOOP ADDRESS  
        MOV      R5,R1        :SAVE R5  
        MOV      #0,%4        :LOAD HIGH ORDER WITH 0  
        MOV      #9,%4+1      :LOAD LOW ORDER WITH 9.  
        DIV      #2,%4        :DIVIDE BY #2  
        MFPS     @#PSWORD     :SAVE PS  
  
        CMPB    #0,@#PSWORD   :IS PS = 0  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                :PS IS WRONG  
        302     :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                :BY (013746 000172 000207)  
  
        CMP     #4,%4         :IS QUOTIENT = 4  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                :QUOTIENT IS WRONG  
        303     :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                :BY (013746 000172 000207)  
  
        CMP     #1,%4+1      :IS REMAINDER = 1  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                :WRONG REMAINDER  
        304     :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                :BY (013746 000172 000207)  
  
        MOV     R1,R5         :RESTORE R5  
        CMP     (R5),#256  
        BEQ     .+10         :IF IN WRONG SEQUENCE GO TO THE HLT  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                :TEST IS IN WRONG SEQUENCE  
        305     :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                :BY (013746 000172 000207)  
  
        INC     (R5)
```

14300  
 14301  
 14302  
 14303  
 14304 055122 010767 123044  
 14305 055126 012700 177777  
 14306 055132 012701 177767  
 14307 055136 071027 000002  
 14308 055142 106737 042262  
 14309  
 14310 055146 122737 000010 042262  
 14311 055154 001403  
 14312 055156 004767 003460  
 14313  
 14314 055162 000306  
 14315  
 14316  
 14317  
 14318 055164 022700 177774  
 14319 055170 001403  
 14320 055172 004767 003444  
 14321  
 14322 055176 000307  
 14323  
 14324  
 14325  
 14326 055200 022701 177777  
 14327 055204 001403  
 14328 055206 004767 003430  
 14329  
 14330 055212 000310  
 14331  
 14332  
 14333 055214 021527 000257  
 14334 055220 001403  
 14335 055222 004767 003414  
 14336  
 14337 055226 000311  
 14338  
 14339  
 14340 055230 005215  
 14341

```

:*****
:TEST:257      DIV      -1 -9. / #2 = -4      REM = -1      PS = 10
:*****
TST257: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #-1,%0      ;LOAD HIGH ORDER WITH -1
        MOV      #-9,%0+1    ;LOAD LOW ORDER WITH -9.
        DIV      #2,%0      ;DIVIDE BY #2
        MFPS     @#PSWORD    ;SAVE PS

        CMPB    #10,@#PSWORD ;IS PS = 10
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;PS IS WRONG
        306     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP     #-4,%0      ;IS QUOTIENT = -4
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;QUOTIENT IS WRONG
        307     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP     #-1,%0+1    ;IS REMAINDER = -1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;WRONG REMAINDER
        310     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP     (R5),#257
        BEQ     .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;TEST IS IN WRONG SEQUENCE
        311     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        INC     (R5)
  
```



14342  
14343  
14344  
14345  
14346 055232 010767 122734  
14347 055236 012702 000000  
14348 055242 012703 000002  
14349 055246 071227 177775  
14350 055252 106737 042262  
14351  
14352 055256 122737 000004 042262  
14353 055264 001403  
14354 055266 004767 003350  
14355  
14356 055272 000312  
14357  
14358  
14359  
14360 055274 022702 000000  
14361 055300 001403  
14362 055302 004767 003334  
14363  
14364 055306 000313  
14365  
14366  
14367  
14368 055310 022703 000002  
14369 055314 001403  
14370 055316 004767 003320  
14371  
14372 055322 000314  
14373  
14374  
14375 055324 021527 000260  
14376 055330 001403  
14377 055332 004767 003304  
14378  
14379 055336 000315  
14380  
14381  
14382 055340 005215  
14383

```
*****  
:TEST:260      DIV      0 2 / #-3 = 0      REM = 2      PS = 4  
*****  
TST260: MOV      PC,LPADR      :STORE ERROR LOOP ADDRESS  
        MOV      #0,%2        :LOAD HIGH ORDER WITH 0  
        MOV      #2,%2+1      :LOAD LOW ORDER WITH 2  
        DIV      #-3,%2       :DIVIDE BY #-3  
        MFPS     @#PSWORD     :SAVE PS  
  
        CMPB    #4,@#PSWORD   :IS PS = 4  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                :PS IS WRONG  
        312     :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                :BY (013746 000172 000207)  
  
        CMP     #0,%2        :IS QUOTIENT = 0  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                :QUOTIENT IS WRONG  
        313     :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                :BY (013746 000172 000207)  
  
        CMP     #2,%2+1      :IS REMAINDER = 2  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                :WRONG REMAINDER  
        314     :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                :BY (013746 000172 000207)  
  
        CMP     (R5),#260    :IF IN WRONG SEQUENCE GO TO THE HLT  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                :TEST IS IN WRONG SEQUENCE  
        315     :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                :BY (013746 000172 000207)  
  
        INC     (R5)
```

```

14384
14385
14386
14387
14388 055342 010767 122624
14389 055346 010501
14390 055350 012704 177777
14391 055354 012705 177776
14392 055360 071427 000003
14393 055364 106737 042262
14394
14395 055370 122737 000004 042262
14396 055376 001403
14397 055400 004767 003236
14398
14399 055404 000316
14400
14401
14402
14403 055406 022704 000000
14404 055412 001403
14405 055414 004767 003222
14406
14407 055420 000317
14408
14409
14410
14411 055422 022705 177776
14412 055426 001403
14413 055430 004767 003206
14414
14415 055434 000320
14416
14417
14418 055436 010105
14419 055440 021527 000261
14420 055444 001403
14421 055446 004767 003170
14422
14423 055452 000321
14424
14425
14426 055454 005215
14427
    ;*****
    ;TEST:261 DIV -1 -2 / #3 = 0 REM = -2 PS = 4
    ;*****
TST261: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
        MOV R5,R1 ;SAVE R5
        MOV #-1,%4 ;LOAD HIGH ORDER WITH -1
        MOV #-2,%4+1 ;LOAD LOW ORDER WITH -2
        DIV #3,%4 ;DIVIDE BY #3
        MFPS @#PSWORD ;SAVE PS
        CMPB #4,@#PSWORD ;IS PS = 4
        BEQ .+10
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;PS IS WRONG
        316 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)
        CMP #0,%4 ;IS QUOTIENT = 0
        BEQ .+10
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;QUOTIENT IS WRONG
        317 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)
        CMP #-2,%4+1 ;IS REMAINDER = -2
        BEQ .+10
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;WRONG REMAINDER
        320 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)
        MOV R1,R5 ;RESTORE R5
        CMP (R5),#261
        BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
        JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;TEST IS IN WRONG SEQUENCE
        321 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)
        INC (R5)
    
```

14428  
14429  
14430  
14431  
14432 055456 010767 122510  
14433 055462 012700 177777  
14434 055466 012701 177777  
14435 055472 071027 000001  
14436 055476 106737 042262  
14437  
14438 055502 122737 000010 042262  
14439 055510 001403  
14440 055512 004767 003124  
14441  
14442 055516 000322  
14443  
14444  
14445  
14446 055520 022700 177777  
14447 055524 001403  
14448 055526 004767 003110  
14449  
14450 055532 000323  
14451  
14452  
14453  
14454 055534 022701 000000  
14455 055540 001403  
14456 055542 004767 003074  
14457  
14458 055546 000324  
14459  
14460  
14461 055550 021527 000262  
14462 055554 001403  
14463 055556 004767 003060  
14464  
14465 055562 000325  
14466  
14467  
14468 055564 005215  
14469

```
*****
:TEST:262      DIV      -1 -1 / #1 = -1      REM = 0      PS = 10
*****
TST262: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #-1,%0      ;LOAD HIGH ORDER WITH -1
        MOV      #-1,%0+1    ;LOAD LOW ORDER WITH -1
        DIV      #1,%0      ;DIVIDE BY #1
        MFPS     @#PSWORD    ;SAVE PS

        CMPB     #10,@#PSWORD ;IS PS = 10
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;PS IS WRONG
        322      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #-1,%0      ;IS QUOTIENT = -1
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;QUOTIENT IS WRONG
        323      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #0,%0+1    ;IS REMAINDER = 0
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;WRONG REMAINDER
        324      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      (R5),#262
        BEQ      .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;TEST IS IN WRONG SEQUENCE
        325      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        INC      (R5)
```

14470  
14471  
14472  
14473  
14474  
14475  
14476  
14477  
14478  
14479  
14480  
14481  
14482  
14483  
14484  
14485  
14486  
14487  
14488  
14489  
14490  
14491  
14492  
14493  
14494  
14495  
14496  
14497  
14498  
14499  
14500  
14501  
14502  
14503  
14504  
14505  
14506  
14507  
14508  
14509  
14510  
14511

055566 010767 122400  
055572 012700 000000  
055576 012701 000000  
055602 071027 000001  
055606 106737 042262  
  
055612 122737 000004 042262  
055620 001403  
055622 004767 003014  
  
055626 000326  
  
055630 022700 000000  
055634 001403  
055636 004767 003000  
  
055642 000327  
  
055644 022701 000000  
055650 001403  
055652 004767 002764  
  
055656 000330  
  
055660 021527 000263  
055664 001403  
055666 004767 002750  
  
055672 000331  
  
055674 005215

```

:*****
:TEST:263      DIV      0 0 / #1 = 0      REM = 0      PS = 4
:*****
TST263: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #0,Z0        ;LOAD HIGH ORDER WITH 0
        MOV      #0,Z0+1      ;LOAD LOW ORDER WITH 0
        DIV      #1,Z0        ;DIVIDE BY #1
        MFPS     @#PSWORD     ;SAVE PS

        CMPB     #4,@#PSWORD  ;IS PS = 4
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;PS IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #0,Z0        ;IS QUOTIENT = 0
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;QUOTIENT IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      #0,Z0+1      ;IS REMAINDER = 0
        BEQ      .+10
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;WRONG REMAINDER
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP      (R5),#263
        BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
        ;TEST IS IN WRONG SEQUENCE
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        INC      (R5)
    
```

```
14512
14513
14514
14515
14516 055676 010767 122270
14517 055702 012702 177777
14518 055706 012703 125252
14519 055712 071227 000002
14520 055716 106737 042262
14521
14522 055722 122737 000010 042262
14523 055730 001403
14524 055732 004767 002704
14525
14526 055736 000332
14527
14528
14529
14530 055740 022702 152525
14531 055744 001403
14532 055746 004767 002670
14533
14534 055752 000333
14535
14536
14537
14538 055754 022703 000000
14539 055760 001403
14540 055762 004767 002654
14541
14542 055766 000334
14543
14544
14545 055770 021527 000264
14546 055774 001403
14547 055776 004767 002640
14548
14549 056002 000335
14550
14551
14552 056004 005215
14553
```

```
*****
:TEST:264 DIV -1 125252 / #2 = 152525 REM = 0 PS = 10
*****

TST264: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #-1,#2 ;LOAD HIGH ORDER WITH -1
MOV #125252,#2+1 ;LOAD LOW ORDER WITH 125252
DIV #2,#2 ;DIVIDE BY #2
MFPS @#PSWORD ;SAVE PS

CMPB #10,@#PSWORD ;IS PS = 10
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;PS IS WRONG
332 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #152525,#2 ;IS QUOTIENT = 152525
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;QUOTIENT IS WRONG
333 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #0,#2+1 ;IS REMAINDER = 0
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;WRONG REMAINDER
334 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP (R5),#264
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
335 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

INC (R5)
```

```

14554
14555
14556
14557
14558 056006 010767 122160
14559 056012 010501
14560 056014 012704 177777
14561 056020 012705 177777
14562 056024 071427 177777
14563 056030 106737 042262
14564
14565 056034 122737 000000 042262
14566 056042 001403
14567 056044 004767 002572
14568
14569 056050 000336
14570
14571
14572
14573 056052 022704 000001
14574 056056 001403
14575 056060 004767 002556
14576
14577 056064 000337
14578
14579
14580
14581 056066 022705 000000
14582 056072 001403
14583 056074 004767 002542
14584
14585 056100 000340
14586
14587
14588 056102 010105
14589 056104 021527 000265
14590 056110 001403
14591 056112 004767 002524
14592
14593 056116 000341
14594
14595
14596 056120 005215
14597

```

```

:*****
:TEST:265 DIV -1 -1 / #-1 = 1 REM = 0 PS = 0
:*****
TST265: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV R5,R1 ;SAVE R5
MOV #-1,X4 ;LOAD HIGH ORDER WITH -1
MOV #-1,X4+1 ;LOAD LOW ORDER WITH -1
DIV #-1,X4 ;DIVIDE BY #-1
MFPS @#PSWORD ;SAVE PS
CMPB #0,@#PSWORD ;IS PS = 0
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;PS IS WRONG
336 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
CMP #1,X4 ;IS QUOTIENT = 1
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;QUOTIENT IS WRONG
337 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
CMP #0,X4+1 ;IS REMAINDER = 0
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;WRONG REMAINDER
340 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
MOV R1,R5 ;RESTORE R5
CMP (R5),#265
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
341 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
INC (R5)

```

```
14598
14599
14600
14601
14602 056122 010767 122044
14603 056126 012700 025253
14604 056132 012701 000001
14605 056136 071027 125252
14606 056142 106737 042262
14607
14608 056146 122737 000010 042262
14609 056154 001403
14610 056156 004767 002460
14611
14612 056162 000342
14613
14614
14615
14616 056164 022700 100000
14617 056170 001403
14618 056172 004767 002444
14619
14620 056176 000343
14621
14622
14623
14624 056200 022701 000001
14625 056204 001403
14626 056206 004767 002430
14627
14628 056212 000344
14629
14630
14631 056214 021527 000266
14632 056220 001403
14633 056222 004767 002414
14634
14635 056226 000345
14636
14637
14638 056230 005215
14639

:*****
:TEST:266 DIV 25253 1 / #125252 = 100000 REM = 1 PS = 10
:*****

TST266: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #25253,%0 ;LOAD HIGH ORDER WITH 25253
MOV #1,%0+1 ;LOAD LOW ORDER WITH 1
DIV #125252,%0 ;DIVIDE BY #125252
MFPS @#PSWORD ;SAVE PS

CMPB #10,@#PSWORD ;IS PS = 10
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;PS IS WRONG
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #100000,%0 ;IS QUOTIENT = 100000
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;QUOTIENT IS WRONG
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #1,%0+1 ;IS REMAINDER = 1
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;WRONG REMAINDER
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP (R5),#266
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

INC (R5)
```

14640  
14641  
14642  
14643  
14644  
14645  
14646  
14647  
14648  
14649  
14650  
14651  
14652  
14653  
14654  
14655  
14656  
14657  
14658  
14659  
14660  
14661  
14662  
14663  
14664  
14665  
14666  
14667  
14668  
14669  
14670  
14671  
14672  
14673  
14674  
14675  
14676  
14677  
14678  
14679  
14680  
14681

056232 010767 121734  
056236 012702 037777  
056242 012703 077777  
056246 071227 077777  
056252 106737 042262  
056256 122737 000000 042262  
056264 001403  
056266 004767 002350  
056272 000346  
056274 022702 077777  
056300 001403  
056302 004767 002334  
056306 000347  
056310 022703 077776  
056314 001403  
056316 004767 002320  
056322 000350  
056324 021527 000267  
056330 001403  
056332 004767 002304  
056336 000351  
056340 005215

```
*****  
:TEST:267      DIV      37777 77777 / #77777 = 77777      REM = 77776  
:*****  
TST267: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #37777,%2      ;LOAD HIGH ORDER WITH 37777  
        MOV      #77777,%2+1    ;LOAD LOW ORDER WITH 77777  
        DIV      #77777,%2      ;DIVIDE BY #77777  
        MFPS     @#PSWORD      ;SAVE PS  
  
        CMPB     #0,@#PSWORD    ;IS PS = 0  
        BEQ      .+10  
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                                ;PS IS WRONG  
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                                ;BY (013746 000172 000207)  
  
        CMP      #77777,%2      ;IS QUOTIENT = 77777  
        BEQ      .+10  
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                                ;QUOTIENT IS WRONG  
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                                ;BY (013746 000172 000207)  
  
        CMP      #77776,%2+1    ;IS REMAINDER = 77776  
        BEQ      .+10  
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                                ;WRONG REMAINDER  
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                                ;BY (013746 000172 000207)  
  
        CMP      (R5),#267  
        BEQ      .+10           ;IF IN WRONG SEQUENCE GO TO THE HLT  
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                                ;TEST IS IN WRONG SEQUENCE  
                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                                ;BY (013746 000172 000207)  
  
        INC      (R5)
```

PS = 0



```

14682
14683
14684
14685
14686 056342 010767 121624
14687 056346 010501
14688 056350 012704 000000
14689 056354 012705 100000
14690 056360 071427 000002
14691 056364 106737 042262
14692
14693 056370 122737 000000 042262
14694 056376 001403
14695 056400 004767 002236
14696
14697 056404 000352
14698
14699
14700
14701 056406 022704 040000
14702 056412 001403
14703 056414 004767 002222
14704
14705 056420 000353
14706
14707
14708
14709 056422 022705 000000
14710 056426 001403
14711 056430 004767 002206
14712
14713 056434 000354
14714
14715
14716 056436 010105
14717 056440 021527 000270
14718 056444 001403
14719 056446 004767 002170
14720
14721 056452 000355
14722
14723
14724 056454 005215
14725

```

```

:*****
:TEST:270      DIV      0 100000 / #2 = 40000      REM = 0      PS = 0
:*****
TST270: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      R5,R1        ;SAVE R5
        MOV      #0,%4        ;LOAD HIGH ORDER WITH 0
        MOV      #100000,%4+1 ;LOAD LOW ORDER WITH 100000
        DIV      #2,%4        ;DIVIDE BY #2
        MFPS     @#PSWORD     ;SAVE PS

        CMPB    #0,@#PSWORD   ;IS PS = 0
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;PS IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     #40000,%4     ;IS QUOTIENT = 40000
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;QUOTIENT IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     #0,%4+1      ;IS REMAINDER = 0
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;WRONG REMAINDER
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        MOV     R1,R5        ;RESTORE R5
        CMP     (R5),#270
        BEQ     .+10        ;IF IN WRONG SEQUENCE GO TO THE HLT
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;TEST IS IN WRONG SEQUENCE
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        INC     (R5)

```

14726  
14727  
14728  
14729  
14730  
14731  
14732  
14733  
14734  
14735  
14736  
14737  
14738  
14739  
14740  
14741  
14742  
14743  
14744  
14745  
14746  
14747  
14748  
14749  
14750  
14751  
14752  
14753  
14754  
14755  
14756  
14757  
14758  
14759  
14760  
14761  
14762  
14763  
14764  
14765  
14766  
14767

056456 010767 121510  
056462 012700 177777  
056466 012701 077777  
056472 071027 177776  
056476 106737 042262  
  
056502 122737 000000 042262  
056510 001403  
056512 004767 002124  
  
056516 000356  
  
056520 022700 040000  
056524 001403  
056526 004767 002110  
  
056532 000357  
  
056534 022701 177777  
056540 001403  
056542 004767 002074  
  
056546 000360  
  
056550 021527 000271  
056554 001403  
056556 004767 002060  
  
056562 000361  
  
056564 005215

```
*****  
:TEST:271 DIV 177777 77777 / #177776 = 40000 REM = 177777  
*****  
TST271: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #177777,%0 ;LOAD HIGH ORDER WITH 177777  
MOV #77777,%0+1 ;LOAD LOW ORDER WITH 77777  
DIV #177776,%0 ;DIVIDE BY #177776  
MFPS @#PSWORD ;SAVE PS  
  
CMPB #0,@#PSWORD ;IS PS = 0  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;PS IS WRONG  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #40000,%0 ;IS QUOTIENT = 40000  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;QUOTIENT IS WRONG  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #177777,%0+1 ;IS REMAINDER = 177777  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;WRONG REMAINDER  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#271  
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;TEST IS IN WRONG SEQUENCE  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
INC (R5)
```

PS = 0

```
14768  
14769  
14770  
14771  
14772 056566 010767 121400  
14773 056572 012702 000000  
14774 056576 012703 052525  
14775 056602 071227 052525  
14776 056606 106737 042262  
14777  
14778 056612 122737 000000 042262  
14779 056620 001403  
14780 056622 004767 002014  
14781  
14782 056626 000362  
14783  
14784  
14785  
14786 056630 022702 000001  
14787 056634 001403  
14788 056636 004767 002000  
14789  
14790 056642 000363  
14791  
14792  
14793  
14794 056644 022703 000000  
14795 056650 001403  
14796 056652 004767 001764  
14797  
14798 056656 000364  
14799  
14800  
14801 056660 021527 000272  
14802 056664 001403  
14803 056666 004767 001750  
14804  
14805 056672 000365  
14806  
14807  
14808 056674 005215  
14809
```

```
*****  
:TEST:272 DIV 0 52525 / #52525 = 1 REM = 0 PS = 0  
*****  
TST272: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #0,%2 ;LOAD HIGH ORDER WITH 0  
MOV #52525,%2+1 ;LOAD LOW ORDER WITH 52525  
DIV #52525,%2 ;DIVIDE BY #52525  
MFPS @#PSWORD ;SAVE PS  
  
CMPB #0,@#PSWORD ;IS PS = 0  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;PS IS WRONG  
362 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #1,%2 ;IS QUOTIENT = 1  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;QUOTIENT IS WRONG  
363 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #0,%2+1 ;IS REMAINDER = 0  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;WRONG REMAINDER  
364 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#272  
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;TEST IS IN WRONG SEQUENCE  
365 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
INC (R5)
```

14810  
14811  
14812  
14813  
14814  
14815  
14816  
14817  
14818  
14819  
14820  
14821  
14822  
14823  
14824  
14825  
14826  
14827  
14828  
14829  
14830  
14831  
14832  
14833  
14834  
14835  
14836  
14837  
14838  
14839

056676 010767 121270  
056702 010501  
056704 012704 000000  
056710 012705 077777  
056714 071427 000000  
056720 106737 042262  
056724 042737 000014 042262  
056732 122737 000003 042262  
056740 001403  
056742 004767 001674  
056746 000366  
056750 010105  
056752 021527 000273  
056756 001403  
056760 004767 001656  
056764 000367  
056766 005215

```
*****  
:TEST:273      DIV      0 77777 / #0 = DUMMY      REM = DUMMY      PS = 3  
*****  
TST273: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      R5,R1        ;SAVE R5  
        MOV      #0,%4        ;LOAD HIGH ORDER WITH 0  
        MOV      #77777,%4+1  ;LOAD LOW ORDER WITH 77777  
        DIV      #0,%4        ;DIVIDE BY #0  
        MFPS     @#PSWORD     ;SAVE PS  
        BIC      #14,@#PSWORD  
  
        CMPB     #3,@#PSWORD  ;IS PS = 3  
        BEQ      .+10  
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                        ;PS IS WRONG  
        366      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                        ;BY (013746 000172 000207)  
  
        MOV      R1,R5        ;RESTORE R5  
        CMP      (R5),#273  
        BEQ      .+10        ;IF IN WRONG SEQUENCE GO TO THE HLT  
        JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                        ;TEST IS IN WRONG SEQUENCE  
        367      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                        ;BY (013746 000172 000207)  
  
        INC      (R5)
```

14840  
14841  
14842  
14843  
14844  
14845  
14846  
14847  
14848  
14849  
14850  
14851  
14852  
14853  
14854  
14855  
14856  
14857  
14858  
14859  
14860  
14861  
14862  
14863  
14864  
14865  
14866  
14867

056770 010767 121176  
056774 012700 077777  
057000 012701 177777  
057004 071027 000002  
057010 106737 042262  
057014 042737 000014 042262  
057022 122737 000002 042262  
057030 001403  
057032 004767 001604  
057036 000370  
057040 021527 000274  
057044 001403  
057046 004767 001570  
057052 000371  
057054 005215

```
*****  
:TEST:274      DIV      77777 177777 / #2 = DUMMY      REM = DUMMY      PS = 2  
*****  
TST274: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #77777,%0      ;LOAD HIGH ORDER WITH 77777  
        MOV      #177777,%0+1    ;LOAD LOW ORDER WITH 177777  
        DIV      #2,%0          ;DIVIDE BY #2  
        MFPS     @#PSWORD      ;SAVE PS  
        BIC      #14,@#PSWORD  
  
        CMPB    #2,@#PSWORD     ;IS PS = 2  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                ;PS IS WRONG  
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                ;BY (013746 000172 000207)  
  
        CMP     (R5),#274  
        BEQ     .+10            ;IF IN WRONG SEQUENCE GO TO THE HLT  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                ;TEST IS IN WRONG SEQUENCE  
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                ;BY (013746 000172 000207)  
  
        INC     (R5)
```

14868 057056 012702 000002  
14869 057062 012703 042324  
14870 057066 012704 042326  
14871  
14872  
14873  
14874  
14875  
14876 057072 010767 121074  
14877 057076 012700 000000  
14878 057102 012701 052525  
14879 057106 071067 163212  
14880 057112 106737 042262  
14881  
14882 057116 122737 000000 042262  
14883 057124 001403  
14884 057126 004767 001510  
14885  
14886 057132 000372  
14887  
14888  
14889  
14890 057134 022700 025252  
14891 057140 001403  
14892 057142 004767 001474  
14893  
14894 057146 000373  
14895  
14896  
14897  
14898 057150 022701 000001  
14899 057154 001403  
14900 057156 004767 001460  
14901  
14902 057162 000374  
14903  
14904  
14905 057164 021527 000275  
14906 057170 001403  
14907 057172 004767 001444  
14908  
14909 057176 000375  
14910  
14911  
14912 057200 005215  
14913

```

MOV #2,%2
MOV #S9,%3
MOV #S10,%4
:*****
:TEST:275 DIV 0 52525 / S9 = 25252 REM = 1 PS = 0
:*****
TST275: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #0,%0 ;LOAD HIGH ORDER WITH 0
MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
DIV S9,%0 ;DIVIDE BY S9
MFPS @#PSWORD ;SAVE PS

CMPB #0,@#PSWORD ;IS PS = 0
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;PS IS WRONG
372 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #25252,%0 ;IS QUOTIENT = 25252
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;QUOTIENT IS WRONG
373 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #1,%0+1 ;IS REMAINDER = 1
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;WRONG REMAINDER
374 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP (R5),#275
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
375 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

INC (R5)

```

14914  
14915  
14916  
14917  
14918 057202 010767 120764  
14919 057206 012700 000000  
14920 057212 012701 052525  
14921 057216 071077 163104  
14922 057222 106737 042262  
14923  
14924 057226 122737 000000 042262  
14925 057234 001403  
14926 057236 004767 001400  
14927  
14928 057242 000376  
14929  
14930  
14931  
14932 057244 022700 025252  
14933 057250 001403  
14934 057252 004767 001364  
14935  
14936 057256 000377  
14937  
14938  
14939  
14940 057260 022701 000001  
14941 057264 001403  
14942 057266 004767 001350  
14943  
14944 057272 000400  
14945  
14946  
14947 057274 021527 000276  
14948 057300 001403  
14949 057302 004767 001334  
14950  
14951 057306 000401  
14952  
14953  
14954 057310 005215  
14955

```
*****  
:TEST:276      DIV      0 52525 / @S10 = 25252      REM = 1      PS = 0  
*****  
TST276: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #0,%0          ;LOAD HIGH ORDER WITH 0  
        MOV      #52525,%0+1    ;LOAD LOW ORDER WITH 52525  
        DIV      @S10,%0        ;DIVIDE BY @S10  
        MFPS     @#PSWORD       ;SAVE PS  
  
        CMPB    #0,@#PSWORD     ;IS PS = 0  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                ;PS IS WRONG  
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                ;BY (013746 000172 000207)  
  
        CMP     #25252,%0       ;IS QUOTIENT = 25252  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                ;QUOTIENT IS WRONG  
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                ;BY (013746 000172 000207)  
  
        CMP     #1,%0+1        ;IS REMAINDER = 1  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                ;WRONG REMAINDER  
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                ;BY (013746 000172 000207)  
  
        CMP     (R5),#276      ;IF IN WRONG SEQUENCE GO TO THE HLT  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
                ;TEST IS IN WRONG SEQUENCE  
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
                ;BY (013746 000172 000207)  
  
        INC     (R5)
```

```

14956
14957
14958
14959
14960 057312 010767 120654
14961 057316 012700 000000
14962 057322 012701 052525
14963 057326 071037 042324
14964 057332 106737 042262
14965
14966 057336 122737 000000 042262
14967 057344 001403
14968 057346 004767 001270
14969
14970 057352 000402
14971
14972
14973
14974 057354 022700 025252
14975 057360 001403
14976 057362 004767 001254
14977
14978 057366 000403
14979
14980
14981
14982 057370 022701 000001
14983 057374 001403
14984 057376 004767 001240
14985
14986 057402 000404
14987
14988
14989 057404 021527 000277
14990 057410 001403
14991 057412 004767 001224
14992
14993 057416 000405
14994
14995
14996 057420 005215
14997

:*****
:TEST:277      DIV      0 52525 / @#S9 = 25252      REM = 1      PS = 0
:*****

TST277: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #0,%0        ;LOAD HIGH ORDER WITH 0
        MOV      #52525,%0+1  ;LOAD LOW ORDER WITH 52525
        DIV      @#S9,%0      ;DIVIDE BY @#S9
        MFPS     @#PSWORD     ;SAVE PS

        CMPB    #0,@#PSWORD   ;IS PS = 0
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;PS IS WRONG
        402     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)

        CMP     #25252,%0     ;IS QUOTIENT = 25252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;QUOTIENT IS WRONG
        403     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)

        CMP     #1,%0+1      ;IS REMAINDER = 1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;WRONG REMAINDER
        404     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)

        CMP     (R5),#277
        BEQ     .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                                ;TEST IS IN WRONG SEQUENCE
        405     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                                ;BY (013746 000172 000207)

        INC     (R5)
  
```



```

14998
14999
15000
15001
15002 057422 010767 120544
15003 057426 012700 000000
15004 057432 012701 052525
15005 057436 071002
15006 057440 106737 042262
15007
15008 057444 122737 000000 042262
15009 057452 001403
15010 057454 004767 001162
15011
15012 057460 000406
15013
15014
15015
15016 057462 022700 025252
15017 057466 001403
15018 057470 004767 001146
15019
15020 057474 000407
15021
15022
15023
15024 057476 022701 000001
15025 057502 001403
15026 057504 004767 001132
15027
15028 057510 000410
15029
15030
15031 057512 021527 000300
15032 057516 001403
15033 057520 004767 001116
15034
15035 057524 000411
15036
15037
15038 057526 005215
15039
  
```

```

*****
:TEST:300      DIV      0 52525 / %2 = 25252      REM = 1      PS = 0
*****
TST300: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #0,%0        ;LOAD HIGH ORDER WITH 0
        MOV      #52525,%0+1  ;LOAD LOW ORDER WITH 52525
        DIV      %2,%0        ;DIVIDE BY %2
        MFPS     @#PSWORD     ;SAVE PS

        CMPB    #0,@#PSWORD   ;IS PS = 0
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                          ;PS IS WRONG
        406     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                          ;BY (013746 000172 000207)

        CMP     #25252,%0     ;IS QUOTIENT = 25252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                          ;QUOTIENT IS WRONG
        407     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                          ;BY (013746 000172 000207)

        CMP     #1,%0+1      ;IS REMAINDER = 1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                          ;WRONG REMAINDER
        410     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                          ;BY (013746 000172 000207)

        CMP     (R5),#300
        BEQ     .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                          ;TEST IS IN WRONG SEQUENCE
        411     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                          ;BY (013746 000172 000207)

        INC     (R5)
  
```

```
15040  
15041  
15042  
15043  
15044 057530 010767 120436  
15045 057534 012700 000000  
15046 057540 012701 052525  
15047 057544 071023  
15048 057546 106737 042262  
15049  
15050 057552 122737 000000 042262  
15051 057560 001403  
15052 057562 004767 001054  
15053  
15054 057566 000412  
15055  
15056  
15057  
15058 057570 022700 025252  
15059 057574 001403  
15060 057576 004767 001040  
15061  
15062 057602 000413  
15063  
15064  
15065  
15066 057604 022701 000001  
15067 057610 001403  
15068 057612 004767 001024  
15069  
15070 057616 000414  
15071  
15072  
15073 057620 021527 000301  
15074 057624 001403  
15075 057626 004767 001010  
15076  
15077 057632 000415  
15078  
15079  
15080 057634 005215  
15081
```

```
*****  
:TEST:301 DIV 0 52525 / (3)+ = 25252 REM = 1 PS = 0  
*****  
TST301: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #0,%0 ;LOAD HIGH ORDER WITH 0  
MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525  
DIV (3)+,%0 ;DIVIDE BY (3)+  
MFPS @#PSWORD ;SAVE PS  
  
CMPB #0,@#PSWORD ;IS PS = 0  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;PS IS WRONG  
412 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #25252,%0 ;IS QUOTIENT = 25252  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;QUOTIENT IS WRONG  
413 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #1,%0+1 ;IS REMAINDER = 1  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;WRONG REMAINDER  
414 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#301  
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;TEST IS IN WRONG SEQUENCE  
415 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
INC (R5)
```

```

15082
15083
15084
15085
15086 057636 010767 120330
15087 057642 012700 000000
15088 057646 012701 052525
15089 057652 071043
15090 057654 106737 042262
15091
15092 057660 122737 000000 042262
15093 057666 001403
15094 057670 004767 000746
15095
15096 057674 000416
15097
15098
15099
15100 057676 022700 025252
15101 057702 001403
15102 057704 004767 000732
15103
15104 057710 000417
15105
15106
15107
15108 057712 022701 000001
15109 057716 001403
15110 057720 004767 000716
15111
15112 057724 000420
15113
15114
15115 057726 021527 000302
15116 057732 001403
15117 057734 004767 000702
15118
15119 057740 000421
15120
15121
15122 057742 005215
15123

```

```

*****
;TEST:302      DIV      0 52525 / -(3) = 25252      REM = 1      PS = 0
*****
TST302: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #0,%0        ;LOAD HIGH ORDER WITH 0
        MOV      #52525,%0+1  ;LOAD LOW ORDER WITH 52525
        DIV      -(3),%0      ;DIVIDE BY -(3)
        MFPS     @#PSWORD     ;SAVE PS

        CMPB    #0,@#PSWORD   ;IS PS = 0
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                        ;PS IS WRONG
                        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                        ;BY (013746 000172 000207)

        CMP     #25252,%0     ;IS QUOTIENT = 25252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                        ;QUOTIENT IS WRONG
                        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                        ;BY (013746 000172 000207)

        CMP     #1,%0+1      ;IS REMAINDER = 1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                        ;WRONG REMAINDER
                        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                        ;BY (013746 000172 000207)

        CMP     (R5),#302
        BEQ     .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                        ;TEST IS IN WRONG SEQUENCE
                        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                        ;BY (013746 000172 000207)

        INC     (R5)

```

```
15124  
15125  
15126  
15127  
15128 057744 010767 120222  
15129 057750 012700 000000  
15130 057754 012701 052525  
15131 057760 071064 000002  
15132 057764 106737 042262  
15133  
15134 057770 122737 000000 042262  
15135 057776 001403  
15136 060000 004767 000636  
15137  
15138 060004 000422  
15139  
15140  
15141  
15142 060006 022700 025252  
15143 060012 001403  
15144 060014 004767 000622  
15145  
15146 060020 000423  
15147  
15148  
15149  
15150 060022 022701 000001  
15151 060026 001403  
15152 060030 004767 000606  
15153  
15154 060034 000424  
15155  
15156  
15157 060036 021527 000303  
15158 060042 001403  
15159 060044 004767 000572  
15160  
15161 060050 000425  
15162  
15163  
15164 060052 005215  
15165
```

```
*****  
:TEST:303 DIV 0 52525 / 2(4) = 25252 REM = 1 PS = 0  
*****  
TST303: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #0,%0 ;LOAD HIGH ORDER WITH 0  
MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525  
DIV 2(4),%0 ;DIVIDE BY 2(4)  
MFPS @#PSWORD ;SAVE PS  
  
CMPB #0,@#PSWORD ;IS PS = 0  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;PS IS WRONG  
422 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #25252,%0 ;IS QUOTIENT = 25252  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;QUOTIENT IS WRONG  
423 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #1,%0+1 ;IS REMAINDER = 1  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;WRONG REMAINDER  
424 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#303  
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;TEST IS IN WRONG SEQUENCE  
425 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
INC (R5)
```

```

15166
15167
15168
15169
15170 060054 010767 120112
15171 060060 012700 000000
15172 060064 012701 052525
15173 060070 071074 000000
15174 060074 106737 042262
15175
15176 060100 122737 000000 042262
15177 060106 001403
15178 060110 004767 000526
15179
15180 060114 000426
15181
15182
15183
15184 060116 022700 025252
15185 060122 001403
15186 060124 004767 000512
15187
15188 060130 000427
15189
15190
15191
15192 060132 022701 000001
15193 060136 001403
15194 060140 004767 000476
15195
15196 060144 000430
15197
15198
15199 060146 021527 000304
15200 060152 001403
15201 060154 004767 000462
15202
15203 060160 000431
15204
15205
15206 060162 005215
15207

```

```

;*****
;TEST:304 DIV 0 52525 / @ (4) = 25252 REM = 1 PS = 0
;*****
TST304: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #0,%0 ;LOAD HIGH ORDER WITH 0
MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
DIV @ (4),%0 ;DIVIDE BY @ (4)
MFPS @#PSWORD ;SAVE PS

CMPB #0,@#PSWORD ;IS PS = 0
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;PS IS WRONG
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #25252,%0 ;IS QUOTIENT = 25252
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;QUOTIENT IS WRONG
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #1,%0+1 ;IS REMAINDER = 1
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;WRONG REMAINDER
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP (R5),#304
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

INC (R5)

```

15208  
 15209  
 15210  
 15211  
 15212 060164 010767 120002  
 15213 060170 012700 000000  
 15214 060174 012701 052525  
 15215 060200 071034  
 15216 060202 106737 042262  
 15217  
 15218 060206 122737 000000 042262  
 15219 060214 001403  
 15220 060216 004767 000420  
 15221  
 15222 060222 000432  
 15223  
 15224  
 15225  
 15226 060224 022700 025252  
 15227 060230 001403  
 15228 060232 004767 000404  
 15229  
 15230 060236 000433  
 15231  
 15232  
 15233  
 15234 060240 022701 000001  
 15235 060244 001403  
 15236 060246 004767 000370  
 15237  
 15238 060252 000434  
 15239  
 15240  
 15241 060254 021527 000305  
 15242 060260 001403  
 15243 060262 004767 000354  
 15244  
 15245 060266 000435  
 15246  
 15247  
 15248 060270 005215  
 15249

```

:*****
:TEST:305      DIV      0 52525 / @ (4)+ = 25252      REM = 1      PS = 0
:*****
:ST305: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #0,%0         ;LOAD HIGH ORDER WITH 0
        MOV      #52525,%0+1   ;LOAD LOW ORDER WITH 52525
        DIV      @ (4)+,%0     ;DIVIDE BY @ (4)+
        MFPS     @#PSWORD      ;SAVE PS

        CMPB    #0,@#PSWORD    ;IS PS = 0
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;PS IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     #25252,%0      ;IS QUOTIENT = 25252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;QUOTIENT IS WRONG
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     #1,%0+1       ;IS REMAINDER = 1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;WRONG REMAINDER
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        CMP     (R5),#305     ;IF IN WRONG SEQUENCE GO TO THE HLT
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;TEST IS IN WRONG SEQUENCE
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

        INC     (R5)
  
```

```
15250
15251
15252
15253
15254 060272 010767 117674
15255 060276 012700 000000
15256 060302 012701 052525
15257 060306 071054
15258 060310 106737 042262
15259
15260 060314 122737 000000 042262
15261 060322 001403
15262 060324 004767 000312
15263
15264 060330 000436
15265
15266
15267
15268 060332 022700 025252
15269 060336 001403
15270 060340 004767 000276
15271
15272 060344 000437
15273
15274
15275
15276 060346 022701 000001
15277 060352 001403
15278 060354 004767 000262
15279
15280 060360 000440
15281
15282
15283 060362 021527 000306
15284 060366 001403
15285 060370 004767 000246
15286
15287 060374 000441
15288
15289
15290 060376 005215
15291
```

\*\*\*\*\*  
:TEST:306 DIV 0 52525 / @-(4) = 25252 REM = 1 PS = 0  
\*\*\*\*\*

TST306: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #0,%0 ;LOAD HIGH ORDER WITH 0  
MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525  
DIV @-(4),%0 ;DIVIDE BY @-(4)  
MFPS @#PSWORD ;SAVE PS

CMPB #0,@#PSWORD ;IS PS = 0  
BEQ .+10  
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;PS IS WRONG  
436 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

CMP #25252,%0 ;IS QUOTIENT = 25252  
BEQ .+10  
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;QUOTIENT IS WRONG  
437 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

CMP #1,%0+1 ;IS REMAINDER = 1  
BEQ .+10  
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;WRONG REMAINDER  
440 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

CMP (R5),#306  
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT  
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;TEST IS IN WRONG SEQUENCE  
441 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

INC (R5)

```
15292 060400 013746 000004          MOV    @#4,-(SP)          ;SAVE LOCATION 4 IN STACK
15293 060404 012737 060420 000004    MOV    #18,@#4          ;SET UP TIME OUT TRAP VECTOR
15294 060412 012737 000001 164000    MOV    #1,@#164000      ;TURN OFF MULTI-TESTER
15295 060420 012637 000004          MOV    (SP)+,@#4        ;RESTORE LOCATION 4
15296 060424 005227 177777          INC    #-1              ;TYPE ID. THE FIRST PASS
15297 060430 001002                    BNE    SKPMSG           ;BRANCH AROUND AFTER THE FIRST PASS
15298 060432 104401 060456          TYPE   ,MSG1
15299 060436 005267 001756          SKPMSG: INC  PASSPT      ;SHOULD PRINT THIS PASS?
15300 060442 022767 000017 001750    CMP    #17,PASSPT       ;DID 17-OCTAL PASSES YET?
15301 060450 001424                    BEQ    BREOP            ;IF YES, BRANCH TO EOP
15302 060452 000137 001024          JMP    @#RESTRT         ;JUMP TO RESTART
15303 060456 005015 042115 030455    MSG1:  .ASCIZ  <15><12>*MD-11 CJKDBA-A F-11 CPU DIAG.*<15><12>
15304 060464 020061 045103 042113
15305 060472 040502 040455 020040
15306 060500 026506 030461 041440
15307 060506 052520 042040 040511
15308 060514 027107 005015 000
15309          060522
15310          .EVEN
15311          .SBTTL ** ROUTINES LIST **
15312 060522 012767 177777 001670    BREOP: MOV    #-1,PASSPT ;RESET PASSPT
15313
15314          .SBTTL END OF PASS ROUTINE
15315
15316          ;*****
15317          ;*INCREMENT THE PASS NUMBER ($PASS)
15318          ;*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
15319          ;*IF THERES A MONITOR GO TO IT
15320          ;*IF THERE ISN'T JUMP TO RESTRT
15321
15322          $EOP:
15323          SCOPE
15324 060530 000240          INC    $PASS           ;;INCREMENT THE PASS NUMBER
15325 060532 005267 117550          BIC    #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
15326 060536 042767 100000 117542    DEC    (PC)+           ;;LOOP?
15327 060544 005327
15328 060546 000001          $EOPCT: .WORD 1
15329 060550 003022          BGT    $DOAGN          ;;YES
15330 060552 012737          MOV    (PC)+,@(PC)+   ;;RESTORE COUNTER
15331 060554 000001          $ENDCT: .WORD 1
15332 060556 060546          $EOPCT
15333 060560 104401 060625          TYPE   ,SENDMG        ;;TYPE 'END PASS #'
15334 060564 016746 117516          MOV    $PASS,-(SP)    ;;SAVE $PASS FOR TYPEOUT
15335 060570 104405          TYPDS          ;;GO TYPE--DECIMAL ASCII WITH SIGN
15336 060572 104401 060622          TYPE   ,SENULL        ;;TYPE A NULL CHARACTER
15337 060576 013700 000042          $GET42: MOV  @#42,R0    ;;GET MONITOR ADDRESS
15338 060602 001405          BEQ    $DOAGN          ;;BRANCH IF NO MONITOR
15339 060604 000005          RESET          ;;CLEAR THE WORLD
15340 060606 004710          $ENDAD: JSR  PC,(R0)   ;;GO TO MONITOR
15341 060610 000240          NOP          ;;SAVE ROOM
15342 060612 000240          NOP          ;;FOR
15343 060614 000240          NOP          ;;ACT11
15344 060616 000137          $DOAGN: JMP  @(PC)+           ;;RETURN
15345 060620 001024          $RTNAD: .WORD  RESTRT
15346 060622 0377 377 000          $ENULL: .BYTE  -1,-1,0 ;:NULL CHARACTER STRING
15347 060625 015 042412 042116          $ENDMG: .ASCIZ  <15><12>/END PASS #/
```



MAINDEC-11-CJKDBA-A F-11 CPU DIAG.  
CJKDBA.P11 30-JAN-79 11:07

MACY11 30A(1052) 30-JAN-79<sup>D 10</sup> 11:31 PAGE 327  
END OF PASS ROUTINE

SEQ 0327

15348 060632 050040 051501 020123  
15349 060640 000043  
15350  
15351  
15352

.SBTTL HALT ROUTINE

```

15353
15354
15355      :*      HALT ROUTINE
15356      :*      -----
15357      :*
15358      :*      PROGRAM COMES HERE ON ENCOUNTERING ANY ERROR
15359      :*
15360
15361 060642 017637 000000 000302 $HLT.  MOV    @($P),@#$FATAL ;PLACE THE ERROR NUMBER AT LOCATION $FATAL
15362 060650 011637 061030          MOV    ($P),@#CONTIN ;SAVE ERROR NUMBER ADDRESS
15363 060654 032777 020000 161450  BIT    #20000,@$SWR ;HAS THE OPERATOR ASKED TO SUPRESS ERROR TYPE OUTS
15364 060662 001021          BNE    6$
15365 060664 104401 042344          TYPE   , $CR LF ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15366 060670 104401 060770          TYPE   ,MSGERR ;INFORM ERROR
15367 060674 104401 042344          TYPE   , $CR LF ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15368 060700 013746 061030          MOV    @#CONTIN,-($P) ;RETREIVE ERROR NUMBER ADDR
15369 060704 162716 000004          SUB    #4,($P) ;CALCULATE ERROR PC
15370 060710 104402          TYPOC ;TYPE PC
15371 060712 104401 042344          TYPE   , $CR LF ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15372 060716 013746 000302          MOV    @#$FATAL,-($P) ;RETREIVE ERROR NUMBER
15373 060722 104403          TYPOS ;TYPE ERROR NUMBER
15374 060724          .BYTE 3
15375 060725          .BYTE 0
15376 060726 105767 117366          6$:  TSTB  $ENV ;IF WE ARE NOT UNDER APT. THEN GO TO
15377 060732 001403          BEQ   8$ ;8$
15378 060734 005237 000300          INC   @#$MSGTY ;OTHERWISE INFORM APT. ABOUT SEEING THE ERROR
15379 060740 000777          BR    ;AND LOOP
15380 060742 104401 042344          8$:  TYPE   , $CR LF ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15381 060746 005777 161360          TST   @$SWR ;IS IT REQUIRED TO HALT ON ERROR ?
15382 060752 100001          BPL   10$ ;IF NOT THEN GO TO 10$
15383 060754 000000          HALT
15384 060756 013746 061030          10$: MOV    @#CONTIN,-($P) ;
15385 060762 062716 000002          ADD   #2,($P) ;CALCULATE RETURN ADDRESS
15386 060766 000207          RTS   PC ;RETURN
15387 060770 042440 051122 051117 MSGERR: .ASCIZ / ERROR! PC, AND ERROR # ARE: /
15388 060776 020041 020040 041520
15389 061004 020054 047101 020104
15390 061012 051105 047522 020122
15391 061020 020043 051101 035105
15392 061026 000040
15393          .EVEN
15394 061030 000000          CONTIN: .WORD 0
15395
15396          .SBTTL POWER FAIL ROUTINE
15397 061032 012767 061042 116764 PWRDN: MOV    #PWRUP,24
15398 061040 000000          HALT
15399
15400 061042 012767 061032 116754 PWRUP: MOV    #PWRDN,24
15401 061050 012706 001000          MOV    #BUFF,$P
15402 061054 132767 000040 117237 BIT    #40,$ENVM ;WILL APT ALLOW PRINTING?
15403 061062 001013          BNE    PFRES ;NO
15404 061064 012700 061116          MOV    #MSGPWF,$R0 ;GET MSG ADDR.
15405 061070 105737 177564          PWAIT: TSTB @#TPS ;TTY READY
15406 061074 100375          BPL   PWAIT ;NO WAIT
15407 061076 112037 177566          MOVB  (R0)+,@#TPB ;PRINT CHARACTER
15408 061102 001372          BNE   PWAIT ;NEXT IF NOT DONE.

```

15409 061104 105737 177564  
15410 061110 100375  
15411 061112 000167 117706  
15412 061116 005015 047520 042527  
15413 061124 020122 040506 046111  
15414 061132 042105 000041  
15415  
15416  
15417  
15418  
15419  
15420  
15421  
15422  
15423  
15424  
15425  
15426  
15427  
15428  
15429  
15430  
15431  
15432  
15433 061136 105767 000261  
15434 061142 100002  
15435 061144 000000  
15436 061146 000430  
15437 061150 010046  
15438 061152 017600 000002  
15439 061156 122767 000001 117134  
15440 061164 001011  
15441 061166 132767 000100 117125  
15442 061174 001405  
15443 061176 010067 000004  
15444 061202 004767 000230  
15445 061206 000000  
15446 061210 132767 000040 117103  
15447 061216 001003  
15448 061220 112046  
15449 061222 001005  
15450 061224 005726  
15451 061226 012600  
15452 061230 062716 000002  
15453 061234 000002  
15454 061236 122716 000011  
15455 061242 001430  
15456 061244 122716 000200  
15457 061250 001006  
15458 061252 005726  
15459 061254 104401  
15460 061256 042344  
15461 061260 105067 000130  
15462 061264 000755  
15463 061266 004767 000056  
15464 061272 126726 000124

PWAIT1: TSTB @#TPS  
BPL PWAIT1  
PFRES: JMP RESTR  
MSGPWF: .ASCIZ <15><12>.POWER FAILED!.

.EVEN  
.SBTTL TYPE ROUTINE

\*\*\*\*\*  
\*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.  
\*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.  
\*NOTE1: \$NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.  
\*NOTE2: \$FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.  
\*NOTE3: \$FILLC CONTAINS THE CHARACTER TO FILL AFTER.

\*CALL:  
\*1) USING A TRAP INSTRUCTION  
\* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING  
\*OR  
\* TYPE  
\* MESADR

\$TYPE: TSTB \$TPFLG ;;IS THERE A TERMINAL?  
BPL 1\$ ;;BR IF YES  
HALT ;;HALT HERE IF NO TERMINAL  
BR 3\$ ;;LEAVE  
1\$: MOV R0,-(SP) ;;SAVE R0  
MOV @2(SP),R0 ;;GET ADDRESS OF ASCIZ STRING  
CMPB #APTENV,\$ENV ;;RUNNING IN APT MODE  
BNE 62\$ ;;NO,GO CHECK FOR APT CONSOLE  
BITB #APTSPOOL,\$ENVM ;;SPOOL MESSAGE TO APT  
BEQ 62\$ ;;NO,GO CHECK FOR CONSOLE  
MOV R0,61\$ ;;SETUP MESSAGE ADDRESS FOR APT  
JSR PC,\$ATY3 ;;SPOOL MESSAGE TO APT  
61\$: .WORD 0 ;;MESSAGE ADDRESS  
62\$: BITB #APTCSUP,\$ENVM ;;APT CONSOLE SUPPRESSED  
BNE 60\$ ;;YES,SKIP TYPE OUT  
2\$: MOVB (R0)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK  
BNE 4\$ ;;BR IF IT ISN'T THE TERMINATOR  
TST (SP)+ ;;IF TERMINATOR POP IT OFF THE STACK  
60\$: MOV (SP)+,R0 ;;RESTORE R0  
3\$: ADD #2,(SP) ;;ADJUST RETURN PC  
RTI ;;RETURN  
4\$: CMPB #HT,(SP) ;;BRANCH IF <HT>  
BEQ 8\$  
CMPB #CRLF,(SP) ;;BRANCH IF NOT <CRLF>  
BNE 5\$  
TST (SP)+ ;;POP <CR><LF> EQUIV  
TYPE ;;TYPE A CR AND LF  
\$CRLF  
CLRB \$CHARCNT ;;CLEAR CHARACTER COUNT  
BR 2\$ ;;GET NEXT CHARACTER  
5\$: JSR PC,\$TYPEC ;;GO TYPE THIS CHARACTER  
6\$: CMPB \$FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?

```
15465 061276 001350          BNE      2$          ;;IF NO GO GET NEXT CHAR.
15466 061300 016746 000114    MOV      $NULL,-(SP) ;;GET # OF FILLER CHARS. NEEDED
15467                                ;;AND THE NULL CHAR.
15468 061304 105366 000001    7$:     DECB     1(SP)  ;;DOES A NULL NEED TO BE TYPED?
15469 061310 002770          BLT      6$          ;;BR IF NO--GO POP THE NULL OFF OF STACK
15470 061312 004767 000032    JSR      PC,$TYPEC  ;;GO TYPE A NULL
15471 061316 105367 000072    DECB     $CHARCNT   ;;DO NOT COUNT AS A COUNT
15472 061322 000770          BR       7$          ;;LOOP
15473
15474                                ;HORIZONTAL TAB PROCESSOR
15475
15476 061324 112716 000040    8$:     MOVVB   #' ,(SP) ;;REPLACE TAB WITH SPACE
15477 061330 004767 000014    9$:     JSR      PC,$TYPEC ;;TYPE A SPACE
15478 061334 132767 000007 000052    BITB    #7,$CHARCNT ;;BRANCH IF NOT AT
15479 061342 001372          BNE      9$          ;;TAB STOP
15480 061344 005726          TST      (SP)+      ;;POP SPACE OFF STACK
15481 061346 000724          BR       2$          ;;GET NEXT CHARACTER
15482 061350 105777 160766    $TYPEC: TSTB    @STPS  ;;WAIT UNTIL PRINTER IS READY
15483 061354 100375          BPL      $TYPEC
15484 061356 116677 000002 160754    MOVVB   2(SP),@STPB ;;LOAD CHAR TO BE TYPED INTO DATA REG.
15485 061364 122766 000015 000002    CMPB    #CR,2(SP)  ;;IS CHARACTER A CARRIAGE RETURN?
15486 061372 001003          BNE      1$          ;;BRANCH IF NO
15487 061374 105067 000014    CLRB    $CHARCNT   ;;YES--CLEAR CHARACTER COUNT
15488 061400 000406          BR       $TYPEX     ;;EXIT
15489 061402 122766 000012 000002    1$:     CMPB    #LF,2(SP) ;;IS CHARACTER A LINE FEED?
15490 061410 001402          BEQ      $TYPEX     ;;BRANCH IF YES
15491 061412 105227          INCB    (PC)+      ;;COUNT THE CHARACTER
15492 061414 000000          $CHARCNT: .WORD 0  ;;CHARACTER COUNT STORAGE
15493 061416 000207          $TYPEX: RTS      PC
15494
15495 061420      000          $NULL:  .BYTE 0    ;;CONTAINS NULL CHARACTER FOR FILLS
15496 061421      002          $FILLS: .BYTE 2    ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
15497 061422      012          $FILLC: .BYTE 12   ;;INSERT FILL CHARS. AFTER A 'LINE FEED'
15498 061423      000          $TPFLG: .BYTE 0    ;;'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
15499 061424      077          $QUES:  .ASCII "'?" ;;QUESTION MARK
15500 061425      012      000          $LF:    .ASCIZ <12> ;;LINEFEED
15501                                .EVEN
15502                                .SBTTL  APT COMMUNICATIONS ROUTINE
15503
15504                                ;*****
15505 061430 112767 000001 000236    $ATY1: MOVVB   #1,$FFLG ;;TO REPORT FATAL ERROR
15506 061436 112767 000001 000226    $ATY3: MOVVB   #1,$MFLG ;;TO TYPE A MESSAGE
15507 061444 000403          BR       $ATYC
15508 061446 112767 000001 000220    $ATY4: MOVVB   #1,$FFLG ;;TO ONLY REPORT FATAL ERROR
15509 061454          $ATYC:
15510 061454 010046          MOV      R0,-(SP)  ;;PUSH R0 ON STACK
15511 061456 010146          MOV      R1,-(SP)  ;;PUSH R1 ON STACK
15512 061460 105767 000206          TSTB    $MFLG     ;;SHOULD TYPE A MESSAGE?
15513 061464 001450          BEQ      5$          ;;IF NOT: BR
15514 061466 122767 000001 116624    CMPB    #APTENV,$ENV ;;OPERATING UNDER APT?
15515 061474 001031          BNE      3$          ;;IF NOT: BR
15516 061476 132767 000100 116615    BITB    #APTPOOL,$ENVM ;;SHOULD SPOOL MESSAGES?
15517 061504 001425          BEQ      3$          ;;IF NOT: BR
15518 061506 017600 000004          MOV      @4(SP),R0  ;;GET MESSAGE ADDR.
15519 061512 062766 000002 000004    ADD     #2,4(SP)    ;;BUMP RETURN ADDR.
15520 061520 005767 116554    1$:     TST      $MSGTYPE ;;SEE IF DONE W/ LAST XMISSION?
```

```

15521 061524 001375          BNE      1$          ;;IF NOT: WAIT
15522 061526 010067 116562  MOV      RO,$MSGAD  ;;PUT ADDR IN MAILBOX
15523 061532 105720          2$:      TSTB     (RO)+  ;;FIND END OF MESSAGE
15524 061534 001376          BNE      2$
15525 061536 166700 116552  SUB      $MSGAD,RO  ;;SUB START OF MESSAGE
15526 061542 006200          ASR      RO         ;;GET MESSAGE LNGLTH IN WORDS
15527 061544 010067 116546  MOV      RO,$MSGGLT ;;PUT LENGTH IN MAILBOX
15528 061550 012767 000004 116522  MOV      #4,$MSGTYPE ;;TELL APT TO TAKE MSG.
15529 061556 000413          BR       5$
15530 061560 017667 000004 000016 3$:      MOV      @4(SP),4$  ;;PUT MSG ADDR IN JSR LINKAGE
15531 061566 062766 000002 000004  ADD      #2,4(SP)    ;;BUMP RETURN ADDRESS
15532 061574 016746 116176  MOV      177776,-(SP) ;;PUSH 177776 ON STACK
15533 061600 004767 177332  JSR     PC,$TYPE    ;;CALL TYPE MACRO
15534 061604 000000          .WORD   0
15535 061606          5$:
15536 061606 105767 000062 10$:     TSTB     $FFLG    ;;SHOULD REPORT FATAL ERROR?
15537 061612 001416          BEQ     12$        ;;IF NOT: BR
15538 061614 005767 116500  TST     $ENV       ;;RUNNING UNDER APT?
15539 061620 001413          BEQ     12$        ;;IF NOT: BR
15540 061622 005767 116452 11$:     TST     $MSGTYPE  ;;FINISHED LAST MESSAGE?
15541 061626 001375          BNE     11$        ;;IF NOT: WAIT
15542 061630 017667 000004 116444  MOV      @4(SP),$FATAL ;;GET ERROR #
15543 061636 062766 000002 000004  ADD      #2,4(SP)    ;;BUMP RETURN ADDR.
15544 061644 005267 116430  INC     $MSGTYPE    ;;TELL APT TO TAKE ERROR
15545 061650 105067 000020 12$:     CLR     $FFLG     ;;CLEAR FATAL FLAG
15546 061654 105067 000013  CLR     $LFLG     ;;CLEAR LOG FLAG
15547 061660 105067 000006  CLR     $MFLG     ;;CLEAR MESSAGE FLAG
15548 061664 012601  MOV     (SP)+,R1   ;;POP STACK INTO R1
15549 061666 012600  MOV     (SP)+,R0   ;;POP STACK INTO R0
15550 061670 000207  RTS     PC         ;;RETURN
15551 061672 000          $MFLG: .BYTE 0    ;;MESSG. FLAG
15552 061673 000          $LFLG: .BYTE 0    ;;LOG FLAG
15553 061674 000          $FFLG: .BYTE 0    ;;FATAL FLAG
15554 061676          .EVEN
15555 000200  APTSIZE=200
15556 000001  APTENV=001
15557 000100  APTSPool=100
15558 000040  APTCSUP=040
15559          .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
15560
15561          ;*****
15562          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
15563          ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
15564          ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
15565          ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
15566          ;*REPLACED WITH SPACES.
15567          ;*CALL:
15568          ;*      MOV      NUM,-(SP)  ;;PUT THE BINARY NUMBER ON THE STACK
15569          ;*      TYPDS  ;;GO TO THE ROUTINE
15570
15571          $TYPDS:
15572          MOV      R0,-(SP)  ;;PUSH R0 ON STACK
15573          MOV      R1,-(SP)  ;;PUSH R1 ON STACK
15574          MOV      R2,-(SP)  ;;PUSH R2 ON STACK
15575          MOV      R3,-(SP)  ;;PUSH R3 ON STACK
15576          MOV      R5,-(SP)  ;;PUSH R5 ON STACK

```

```
15577 061710 012746 020200      MOV      #20200,-(SP)      ;;SET BLANK SWITCH AND SIGN
15578 061714 016605 000020      MOV      20(SP),R5        ;;GET THE INPUT NUMBER
15579 061720 100004          BPL      1$              ;;BR IF INPUT IS POS.
15580 061722 005405          NEG      R5              ;;MAKE THE BINARY NUMBER POS.
15581 061724 112766 000055 000001  MOVB     #'-,1(SP)        ;;MAKE THE ASCII NUMBER NEG.
15582 061732 005000          CLR      R0              ;;ZERO THE CONSTANTS INDEX
15583 061734 012703 062112      MOV      #$DBLK,R3        ;;SETUP THE OUTPUT POINTER
15584 061740 112723 000040      MOVB     #' ,(R3)+        ;;SET THE FIRST CHARACTER TO A BLANK
15585 061744 005002          CLR      R2              ;;CLEAR THE BCD NUMBER
15586 061746 016001 062102      MOV      $DTBL(R0),R1     ;;GET THE CONSTANT
15587 061752 160105          SUB      R1,R5           ;;FORM THIS BCD DIGIT
15588 061754 002402          BLT     4$              ;;BR IF DONE
15589 061756 005202          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
15590 061760 000774          BR      3$              ;;
15591 061762 060105          4$:     ADD     R1,R5     ;;ADD BACK THE CONSTANT
15592 061764 005702          TST     R2              ;;CHECK IF BCD DIGIT=0
15593 061766 001002          BNE     5$              ;;FALL THROUGH IF 0
15594 061770 105716          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
15595 061772 100407          BMI     7$              ;;BR IF YES
15596 061774 106316          5$:     ASLB    (SP)      ;;MSD?
15597 061776 103003          BCC     6$              ;;BR IF NO
15598 062000 116663 000001 177777  MOVB     1(SP),-1(R3)     ;;YES--SET THE SIGN
15599 062006 052702 000060      BIS     #'0,R2           ;;MAKE THE BCD DIGIT ASCII
15600 062012 052702 000040      6$:     BIS     #' ,R2     ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
15601 062016 110223          MOVB     R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
15602 062020 005720          TST     (R0)+           ;;JUST INCREMENTING
15603 062022 020027 000010      CMP     R0,#10          ;;CHECK THE TABLE INDEX
15604 062026 002746          BLT     2$              ;;GO DO THE NEXT DIGIT
15605 062030 003002          BGT     8$              ;;GO TO EXIT
15606 062032 010502          MOV     R5,R2           ;;GET THE LSD
15607 062034 000764          BR      6$              ;;GO CHANGE TO ASCII
15608 062036 105726          8$:     TSTB    (SP)+     ;;WAS THE LSD THE FIRST NON-ZERO?
15609 062040 100003          BPL     9$              ;;BR IF NO
15610 062042 116663 177777 177776  MOVB     -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
15611 062050 105013          9$:     CLRB    (R3)      ;;SET THE TERMINATOR
15612 062052 012605          MOV     (SP)+,R5        ;;POP STACK INTO R5
15613 062054 012603          MOV     (SP)+,R3        ;;POP STACK INTO R3
15614 062056 012602          MOV     (SP)+,R2        ;;POP STACK INTO R2
15615 062060 012601          MOV     (SP)+,R1        ;;POP STACK INTO R1
15616 062062 012600          MOV     (SP)+,R0        ;;POP STACK INTO R0
15617 062064 104401 062112      TYPE    , $DBLK         ;;NOW TYPE THE NUMBER
15618 062070 016666 000002 000004  MOV      2(SP),4(SP)     ;;ADJUST THE STACK
15619 062076 012616          MOV     (SP)+,(SP)
15620 062100 000002          RTI                          ;;RETURN TO USER
15621 062102 023420      $DTBL: 1000.
15622 062104 001750          1000.
15623 062106 000144          100.
15624 062110 000012          10.
15625 062112 000004      $DBLK: .BLKW 4
15626          .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
15627
15628          ;*****
15629          ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
15630          ;*OCTAL (ASCII) NUMBER AND TYPE IT.
15631          ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
15632          ;*CALL:
```

```

15633      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
15634      ;*      TYPOS      ;;CALL FOR TYPEOUT
15635      ;*      .BYTE      N      ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
15636      ;*      .BYTE      M      ;;M=1 OR 0
15637      ;*      ;;1=TYPE LEADING ZEROS
15638      ;*      ;;0=SUPPRESS LEADING ZEROS
15639
15640      ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
15641      ;*$TYPOS OR $TYPOC
15642      ;*CALL:
15643      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
15644      ;*      TYPON      ;;CALL FOR TYPEOUT
15645
15646      ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
15647      ;*CALL:
15648      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
15649      ;*      TYPOC      ;;CALL FOR TYPEOUT
15650
15651      062122 017646 000000      $TYPOS: MOV      @(SP),-(SP)      ;;PICKUP THE MODE
15652      062126 116667 000001 000211      MOV      1(SP),%OFILL      ;;LOAD ZERO FILL SWITCH
15653      062134 112667 000207      MOV      (SP)+,%OMODE+1      ;;NUMBER OF DIGITS TO TYPE
15654      062140 062716 000002      ADD      #2,(SP)      ;;ADJUST RETURN ADDRESS
15655      062144 000406      BR      $TYPON
15656      062146 112767 000001 000171      $TYPOC: MOV      #1,%OFILL      ;;SET THE ZERO FILL SWITCH
15657      062154 112767 000006 000165      MOV      #6,%OMODE+1      ;;SET FOR SIX(6) DIGITS
15658      062162 112767 000005 000154      $TYPON: MOV      #5,%OCNT      ;;SET THE ITERATION COUNT
15659      062170 010346      MOV      R3,-(SP)      ;;SAVE R3
15660      062172 010446      MOV      R4,-(SP)      ;;SAVE R4
15661      062174 010546      MOV      R5,-(SP)      ;;SAVE R5
15662      062176 116704 000145      MOV      %OMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
15663      062202 005404      NEG      R4
15664      062204 062704 000006      ADD      #6,R4      ;;SUBTRACT IT FOR MAX. ALLOWED
15665      062210 110467 000132      MOV      R4,%OMODE      ;;SAVE IT FOR USE
15666      062214 116704 000125      MOV      %OFILL,R4      ;;GET THE ZERO FILL SWITCH
15667      062220 016605 000012      MOV      12(SP),R5      ;;PICKUP THE INPUT NUMBER
15668      062224 005003      CLR      R3      ;;CLEAR THE OUTPUT WORD
15669      062226 006105      1$:      ROL      R5      ;;ROTATE MSB INTO 'C'
15670      062230 000404      BR      3$      ;;GO DO MSB
15671      062232 006105      2$:      ROL      R5      ;;FORM THIS DIGIT
15672      062234 006105      ROL      R5
15673      062236 006105      ROL      R5
15674      062240 010503      MOV      R5,R3
15675      062242 006103      3$:      ROL      R3      ;;GET LSB OF THIS DIGIT
15676      062244 105367 000076      DECB     %OMODE      ;;TYPE THIS DIGIT?
15677      062250 100016      BPL      7$      ;;BR IF NO
15678      062252 042703 177770      BIC      #177770,R3      ;;GET RID OF JUNK
15679      062256 001002      BNE      4$      ;;TEST FOR 0
15680      062260 005704      TST      R4      ;;SUPPRESS THIS 0?
15681      062262 001403      BEQ      5$      ;;BR IF YES
15682      062264 005204      4$:      INC      R4      ;;DON'T SUPPRESS ANYMORE 0'S
15683      062266 052703 000060      BIS      #'0,R3      ;;MAKE THIS DIGIT ASCII
15684      062272 052703 000040      5$:      BIS      #' ,R3      ;;MAKE ASCII IF NOT ALREADY
15685      062276 110367 000040      MOV      R3,8$      ;;SAVE FOR TYPING
15686      062302 104401 062342      TYPE     ,8$      ;;GO TYPE THIS DIGIT
15687      062306 105367 000032      7$:      DECB     %OCNT      ;;COUNT BY 1
15688      062312 003347      BGT      2$      ;;BR IF MORE TO DO

```

```

15689 062314 002402          BLT      6$          ;;BK IF DONE
15690 062316 005204          INC      R4          ;;INSURE LAST DIGIT ISN'T A RANK
15691 062320 000744          BR       2$          ;;GO DO THE LAST DIGIT
15692 062322 012605          6$:     MOV      (SP)+,R5      ;;RESTORE R5
15693 062324 012604          MOV      (SP)+,R4      ;;RESTORE R4
15694 062326 012603          MOV      (SP)+,R3      ;;RESTORE R3
15695 062330 016666 000002 000004  MOV      2(SP),4(SP)    ;;SET THE STACK FOR RETURNING
15696 062336 012616          MOV      (SP)+,(SP)
15697 062340 000002          RTI          ;;RETURN
15698 062342 000          8$:     .BYTE   0          ;;STORAGE FOR ASCII DIGIT
15699 062343 000          .BYTE   0          ;;TERMINATOR FOR TYPE ROUTINE
15700 062344 000          $OCNT:  .BYTE   0          ;;OCTAL DIGIT COUNTER
15701 062345 000          $OFILL: .BYTE   0          ;;ZERO FILL SWITCH
15702 062346 000000          $OMODE: .WORD   0          ;;NUMBER OF DIGITS TO TYPE
15703
15704
15705
15706
15707
15708
15709
15710
15711 062350 010046          $TRAP:  MOV      R0,-(SP)    ;;SAVE R0
15712 062352 016600 000002  MOV      2(SP),R0        ;;GET TRAP ADDRESS
15713 062356 005740          TST     -(R0)          ;;BACKUP BY 2
15714 062360 111000          MOVB   (R0),R0        ;;GET RIGHT BYTE OF TRAP
15715 062362 006300          ASL    R0             ;;POSITION FOR INDEXING
15716 062364 016000 062404  MOV     $TRPAD(R0),R0    ;;INDEX TO TABLE
15717 062370 000200          RTS     R0            ;;GO TO ROUTINE
15718
15719
15720
15721
15722 062372 011646          ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
15723 062374 016666 000004 000002  $TRAP2: MOV     (SP),-(SP)  ;;MOVE THE PC DOWN
15724 062402 000002          MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
15725
15726          RTI          ;;RESTORE THE PSW
15727
15728          .SBTTL  TRAP TABLE
15729
15730          ;;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
15731          ;;*BY THE "TRAP" INSTRUCTION.
15732
15733          :          ROUTINE
15734          :          -----
15735          $TRPAD: .WORD   $TRAP2
15736          $TYPE  ;;CALL=TYPE   TRAP+1(104401)  TTY TYPEOUT ROUTINE
15737          $TYPOC ;;CALL=TYPOC  TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
15738          $TYPOS ;;CALL=TYPOS  TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
15739          $TYPON ;;CALL=TYPON  TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
15740          $TYPDS ;;CALL=TYPDS  TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
15741
15742          PASSPT: -1
          .END

```



|         |        |         |        |        |        |         |        |         |        |
|---------|--------|---------|--------|--------|--------|---------|--------|---------|--------|
| A       | 026126 | AMTYP3= | 000000 | BRC1   | 003376 | BUFF =  | 001000 | DNMB3B  | 011510 |
| ABASE = | 000000 | AMTYP4= | 000000 | BRC2   | 003406 | CC =    | 177776 | DNMB3C  | 011526 |
| ACDW1 = | 000000 | APASS = | 000000 | BRC3   | 003416 | CCERR   | 026072 | DNMB3D  | 011544 |
| ACDW2 = | 000000 | APRIOR= | 000000 | BREOP  | 060522 | CC1     | 025330 | DNMB3E  | 011554 |
| ACPUOP= | 000000 | APTCSU= | 000040 | BRH    | 025334 | CC2     | 025774 | DNMB4A  | 011744 |
| ADC1    | 020430 | APTENV= | 000001 | BRMFPD | 027146 | CC3     | 025760 | DNMB4B  | 011754 |
| ADC2    | 020440 | APTSIZ= | 000200 | BRN1   | 003256 | CLRCD   | 025756 | DNMB4C  | 011772 |
| ADC3    | 020460 | APTSPO= | 000100 | BRN2   | 003266 | CLR1    | 020032 | DNMB4D  | 012002 |
| ADC4    | 020470 | AROUN   | 025324 | BRN3   | 003276 | CMP1    | 020674 | DNMB4E  | 012012 |
| ADC5    | 020506 | AROUND  | 042016 | BRTAB  | 027242 | CMP2    | 020704 | DNMB4F  | 012026 |
| ADDW0 = | 000000 | ASL1    | 021744 | BRV1   | 003326 | CMP3    | 020726 | DNM03A  | 010164 |
| ADDW1 = | 000000 | ASL2    | 021754 | BRV2   | 003336 | CMP4    | 020736 | DNM03B  | 010174 |
| ADDW10= | 000000 | ASL3    | 021772 | BRV3   | 003346 | CMP5    | 020762 | DNM03C  | 010204 |
| ADDW11= | 000000 | ASL4    | 022002 | BRZ1   | 003206 | CMP6    | 020772 | DNM1    | 010036 |
| ADDW12= | 000000 | ASL5    | 022016 | BRZ2   | 003216 | CMP7    | 021012 | DNM1A   | 011150 |
| ADDW13= | 000000 | ASL6    | 022026 | BRZ3   | 003226 | COM1    | 021052 | DNM1B   | 011160 |
| ADDW14= | 000000 | ASL7    | 022052 | BR1    | 027536 | CONT    | 025364 | DNM2    | 010052 |
| ADDW15= | 000000 | ASR1    | 022114 | BR10   | 027752 | CONT IN | 061030 | DNM2A   | 011226 |
| ADDW2 = | 000000 | ASR2    | 022124 | BR11   | 030056 | CON1    | 026010 | DNM2B   | 011236 |
| ADDW3 = | 000000 | ASR3    | 022146 | BR12   | 030124 | CON2    | 026102 | DNM2C   | 011244 |
| ADDW4 = | 000000 | ASR4    | 022156 | BR13   | 030172 | CORH    | 037614 | DNM2D   | 011254 |
| ADDW5 = | 000000 | ASR5    | 022172 | BR14   | 030240 | COUNT   | 042260 | DNM3    | 010070 |
| ADDW6 = | 000000 | ASR6    | 022202 | BR15   | 030336 | CPUTYP  | 041644 | DNM4    | 010112 |
| ADDW7 = | 000000 | ASR7    | 022232 | BR16   | 030356 | CR =    | 000015 | DNM4A   | 011636 |
| ADDW8 = | 000000 | ASTART  | 042522 | BR17   | 030376 | CRLF =  | 000200 | DNM4B   | 011646 |
| ADDW9 = | 000000 | ASWREG= | 000000 | BR2    | 027564 | CTRAP   | 037572 | DNM4C   | 011662 |
| ADD1    | 020244 | ATESTN= | 000000 | BR20   | 030416 | DAERR   | 025566 | DNM5A   | 012106 |
| ADD2    | 020254 | ATRAP   | 037554 | BR21   | 030436 | DBE     | 041460 | DNM5B   | 012116 |
| ADD3    | 020270 | AUNIT = | 000000 | PR22   | 030456 | DBE1    | 041502 | DNM5C   | 012134 |
| ADD4    | 020300 | AUSWR = | 000000 | BR23   | 030476 | DBE2    | 041472 | DNM6A   | 012214 |
| ADD5    | 020316 | AUTO1   | 037662 | BR3    | 027606 | DBE3    | 041522 | DNM6B   | 012224 |
| ADD6    | 020326 | AVECT1= | 000000 | BR33   | 030546 | DBE4    | 041542 | DNM6C   | 012242 |
| ADD7    | 020340 | AVECT2= | 000000 | BR34   | 030560 | DBE5    | 041560 | DNM7A   | 012324 |
| ADD8    | 020350 | BELL =  | 000240 | BR35   | 030572 | DEC1    | 017672 | DNM7B   | 012334 |
| ADD9    | 020370 | BIC1    | 017374 | BR36   | 030604 | DEC2    | 017702 | DNM7C   | 012352 |
| ADEVCT= | 000000 | BIC2    | 017404 | BR37   | 030626 | DEC3    | 017716 | DOPB2A  | 010576 |
| ADEVN = | 000000 | BIC3    | 017422 | BR4    | 027632 | DEC4    | 017726 | DOPB2B  | 010654 |
| AENV =  | 000000 | BIS1    | 017464 | BR40   | 030640 | DEC5    | 017742 | DOP0A   | 007600 |
| AENVN = | 000000 | BIS2    | 017474 | BR41   | 030652 | DEC6    | 017752 | DOP0B   | 007624 |
| AFATAL= | 000000 | BIS3    | 017514 | BR45   | 032200 | DEC7    | 017774 | DOP0C   | 007644 |
| AMADR1= | 000000 | BITCHK  | 025514 | BR46   | 032746 | DISPLA  | 042334 | DOP0D   | 007674 |
| AMADR2= | 000000 | BITCLR  | 025442 | BR46A  | 032756 | DISPRE  | 000174 | DOP03A  | 007752 |
| AMADR3= | 000000 | BITCON  | 025576 | BR.7   | 033536 | DL11W   | 037732 | DOP03B  | 007762 |
| AMADR4= | 000000 | BITSET  | 025460 | BR5    | 027656 | DL11W1  | 037736 | DOP1    | 010410 |
| AMAMS1= | 000000 | BIT1    | 017304 | BR51   | 034304 | DNMBOA  | 011072 | DOP2    | 010522 |
| AMAMS2= | 000000 | BIT2    | 017314 | BR51A  | 034314 | DNMBOB  | 011102 | DOP4    | 014026 |
| AMAMS3= | 000000 | BIT3    | 017332 | BR6    | 027702 | DNMB2A  | 011330 | DOP5    | 014112 |
| AMAMS4= | 000000 | BRA1    | 001130 | BR7    | 027726 | DNMB2B  | 011340 | DTRAP1  | 037612 |
| AMSGAD= | 000000 | BRA2    | 001140 | BR70   | 037272 | DNMB2C  | 011354 | DUMMY = | 000000 |
| AMSGLG= | 000000 | BRA3    | 001152 | BR71   | 040222 | DNMB2D  | 011370 | END1    | 027232 |
| AMSGTY= | 000000 | BRA4    | 001160 | BTCON  | 025652 | DNMB2E  | 011400 | ENT176  | 046734 |
| AMTYP1= | 000000 | BRA5    | 001170 | BTERR  | 025636 | DNMB2F  | 011416 | ENT51   | 044274 |
| AMTYP2= | 000000 | BRCT    | 025362 | BTRAP  | 037630 | DNMB3A  | 011500 | ER      | 025346 |

|                |               |               |               |                |
|----------------|---------------|---------------|---------------|----------------|
| ERRNM = 000442 | JSR2A 016614  | MDM4A 013106  | MRK2 022642   | NODL 040244    |
| F = 000063     | JSR2B 016624  | MDM4B 013116  | MRK3 022652   | NODL1 040430   |
| FINISH 042244  | JSR3 016504   | MDM4C 013132  | MRK4 022674   | NODL2 040542   |
| FIRST =X000005 | JSR3A 016544  | MDM5A 013336  | MRK5 022706   | NODL3 041030   |
| FOVER 036520   | JSR3B 016554  | MDM5B 013346  | MRK6 022722   | NOP = 000240   |
| FPP 042234     | JSR4 016646   | MDM5C 013364  | MSGERR 060770 | NOR 037460     |
| GIN1 042036    | JSR4A 016664  | MDM5D 013402  | MSGPWF 061116 | NOSUB 037576   |
| GIN2 042054    | JSR4B 016674  | MDM5E 013430  | MSG1 060456   | PASSPT 062420  |
| GIN3 042102    | JSR5 016752   | MDM6A 013500  | MTP10 025150  | PCN01 026570   |
| HERE = 000000  | JSR5A 016776  | MDM6B 013510  | MTPS1 022772  | PCN1 041236    |
| MICORE 037464  | JSR5B 017006  | MDM6C 013526  | MTPS1A 023012 | PCN2 026624    |
| MLT = 000000   | JSR6 016706   | MDM6D 013546  | MTPS2 023066  | PCN3 026672    |
| MT = 000011    | JSR6A 016732  | MDM6E 013576  | MTPS3 023156  | PCN4 026734    |
| IJMP 016410    | JSR6AD 017056 | MDM7A 013652  | MTPS4 023244  | PCN5 026774    |
| IJMP4 016162   | JSR6B 016742  | MDM7B 013662  | MTPS5 023324  | PFRES 061112   |
| IJMP5 016352   | JSR7 017020   | MDM7C 013700  | MTPS6 023414  | POWER 042352   |
| ILLA = 004700  | JSR7A 017036  | MDM7D 013720  | MTPS7 023504  | PROFTE 037730  |
| ILLB = 000100  | JSR7B 017046  | MDM7E 013744  | N = 000307    | PS = 177776    |
| INC1 017554    | K1 027456     | MFP10 025036  | NBR 025340    | PSWORD 042262  |
| INC2 017564    | K10 027474    | MFP10A 025064 | NEGAT 043674  | PUSRM = 030000 |
| INC3 017606    | K11 027476    | MFPS1 023560  | NEG00 004346  | PWAIT 061070   |
| INC4 017616    | K12 027500    | MFPS2A 023650 | NEG01 004356  | PWAIT1 061104  |
| INC5 017632    | K2 027460     | MFPS2B 023660 | NEG02 004372  | PWRDN 061032   |
| INST 042246    | K3 027462     | MFPS2C 023700 | NEG03 004406  | PWRUP 061042   |
| INSTC 031004   | K4 027464     | MFPS3A 023756 | NEG04 004416  | RA 033546      |
| INSTK 035162   | K5 027466     | MFPS3B 023766 | NEG1 020550   | RA1 032210     |
| ITRAPS= 000004 | K6 027470     | MFPS3C 024006 | NEG10 004462  | RB 033534      |
| JMPCK 016412   | K7 027472     | MFPS4A 024064 | NEG11 004472  | RB1 032176     |
| JMPERR 025706  | LAST =X000001 | MFPS4B 024074 | NEG12 004510  | RC 033530      |
| JMPSEQ 016432  | LF = 000012   | MFPS4C 024114 | NEG13 004524  | RC1 032172     |
| JMPT 025676    | LPADR 000172  | MFPS5A 024172 | NEG14 004534  | REGR1 042670   |
| JMP2 016164    | MBDM2A 012570 | MFPS5B 024202 | NEG2 020560   | REGR2 043036   |
| JMP2A 016202   | MBDM2B 012600 | MFPS5C 024222 | NEG20 004602  | REGR23 045546  |
| JMP3 016112    | MBDM2C 012616 | MFPS6A 024302 | NEG21 004612  | REGR3 043204   |
| JMP3A 016130   | MBDM2D 012630 | MFPS6B 024312 | NEG22 004640  | REGR4 043350   |
| JMP3B 016150   | MBDM2E 012640 | MFPS6C 024332 | NEG3 020602   | REGR5 043520   |
| JMP4 016214    | MBDM2F 012656 | MFPS7A 024412 | NEG30 005162  | REG01 045346   |
| JMP4A 016232   | MBDM4A 013212 | MFPS7B 024422 | NEG31 005172  | REG1 002066    |
| JMP4B 016252   | MBDM4B 013230 | MFPS7C 024442 | NEG32 005206  | REG1A 002132   |
| JMP5 016320    | MBDM4C 013242 | MFP1 = 000007 | NEG33 005232  | REG1E 002100   |
| JMP5A 016340   | MBDM4D 013252 | MOR0 026134   | NEG34 005246  | REG2 002202    |
| JMP6 016264    | MBDM4E 013266 | MOR1 026174   | NEG4 020612   | REG2A 002230   |
| JMP6A 016304   | MDM1A 012422  | MOR2 026254   | NEG40 005640  | REG2B 002256   |
| JMP7 016354    | MDM1B 012432  | MOR3 026346   | NEG41 005650  | REG2C 002310   |
| JMP7A 016374   | MDM2A 012476  | MOR4 026366   | NEG42 005664  | REG3 002336    |
| JSRCK 017064   | MDM2B 012506  | MOR5 026406   | NEG5 020632   | REG3A 002402   |
| JSRCKA 017060  | MDM2C 012514  | MOR6 026476   | NEG50 005740  | REG3E 002350   |
| JSRCK1 017102  | MDM2D 012524  | MOR7 026516   | NEG51 005750  | REG4 002452    |
| JSRSEQ 017062  | MDM3A 012732  | MOR8 026536   | NEG52 005764  | REG4A 002516   |
| JSRO 016446    | MDM3B 012742  | MOV1 017214   | NEG60 006042  | REG4E 002464   |
| JSR1 016452    | MDM3C 012760  | MOV2 017224   | NEG61 006052  | REG45 045746   |
| JSR1A 016474   | MDM3D 013000  | MOV3 017242   | NEG70 006122  | REG5 002566    |
| JSR2 016564    | MDM3E 013026  | MRK1 022620   | NEG71 006132  | REG5A 002632   |

|        |        |        |        |         |          |        |        |         |        |
|--------|--------|--------|--------|---------|----------|--------|--------|---------|--------|
| REG5E  | 002600 | RETG4  | 034216 | ROT6    | 015260   | SETUP  | 025202 | SOPB3B  | 005022 |
| REG6   | 002702 | RETG5  | 034750 | ROT7    | 015334   | SET2BR | 025310 | SOPB3C  | 005070 |
| REG6A  | 002746 | RETH5  | 035076 | RT11    | 037160   | SHL    | 001536 | SOPB3D  | 005112 |
| REG6E  | 002714 | RETJ   | 035122 | RT12    | 037172   | SHLE   | 001552 | SOPX    | 006170 |
| RESET2 | 040630 | RETK   | 035164 | RTRAP = | 000010   | SHR    | 001652 | SOPXAD  | 006234 |
| RESET3 | 040620 | RETL   | 035236 | RTRAP1= | 000034   | SHRE   | 001666 | SOPZA   | 004106 |
| REST   | 024536 | RETM   | 035302 | RTRAP2= | 000020   | SKPMSG | 060436 | SOP0A   | 003452 |
| RESTR1 | 001024 | RETN   | 035352 | RTRAP3= | 000030   | SKP104 | 037744 | SOP0B   | 003472 |
| RET    | 042126 | RETO   | 035462 | RTRAP4= | 000014   | SKTST2 | 040646 | SOP0C   | 003534 |
| RETA   | 030710 | RETR1  | 041252 | RTRAP5= | 000004   | SNMBOA | 006326 | SOP0D   | 003560 |
| RETAH  | 030720 | RETR2  | 041330 | RTS1    | 017150   | SNMB1A | 006432 | SOP1A   | 003662 |
| RETAI  | 036610 | RETR3  | 041376 | RT11    | 037006   | SNMB1B | 006442 | SOP1B   | 003674 |
| RETA1  | 031464 | RET1   | 042144 | RT12    | 037022   | SNMB1C | 006464 | SOP2B   | 004126 |
| RETA2  | 032272 | RET2   | 042164 | RT13    | 037064   | SNMB2A | 006606 | SOP3A   | 004712 |
| RETA3  | 033024 | RET3   | 042202 | RT14    | 037102   | SNMB2B | 006616 | SOP3B   | 004726 |
| RETA4  | 033630 | RET4   | 042172 | RT15    | 037036   | SNMB2C | 006632 | SOP4A   | 005324 |
| RETA5  | 034362 | R1TSM  | 046156 | RT16    | 037122   | SNMB2D | 006652 | SOP4B   | 005344 |
| RETB   | 030744 | ROL1   | 021434 | ROTAP   | 037512   | SNMB2E | 006662 | SOP5A   | 005422 |
| RETB1  | 036656 | ROL2   | 021444 | R1ERR   | 002150   | SNMB3A | 007024 | SOP5B   | 005436 |
| RETB1  | 031510 | ROL3   | 021462 | R2ERR   | 002300   | SNMB3B | 007034 | SOP6A   | 005502 |
| RETB2  | 032316 | ROL4   | 021472 | R3ERR   | 002420   | SNMB3C | 007052 | SOP6B   | 005516 |
| RETB3  | 033050 | ROL5   | 021506 | R4ERR   | 002534   | SNMB3D | 007062 | SOP7A   | 005564 |
| RETB4  | 033654 | ROL6   | 021516 | R5ERR   | 002650   | SNM0A  | 006266 | SOP7B   | 005600 |
| RETB5  | 034406 | ROL7   | 021540 | R6      | =%000006 | SNM1A  | 006370 | START   | 001002 |
| RETC   | 031006 | ROR1   | 021602 | R6ERR   | 002764   | SNM2A  | 006526 | STATUS= | 177776 |
| RETC1  | 036730 | ROR2   | 021612 | R7      | =%000007 | SNM2B  | 006536 | STBOT   | 001000 |
| RETC1  | 031552 | ROR3   | 021630 | R7TRX   | 040100   | SNM3A  | 006736 | STP     | 036530 |
| RETC2  | 032360 | ROR4   | 021640 | SBC1    | 021266   | SNM3B  | 006746 | STPP    | 031374 |
| RETC3  | 033112 | ROR5   | 021656 | SBC2    | 021276   | SNM4A  | 007132 | STPPA   | 031404 |
| RETC4  | 033716 | ROR6   | 021666 | SBC3    | 021314   | SNM4B  | 007142 | STP3    | 037366 |
| RETC5  | 034450 | ROR7   | 021702 | SBC4    | 021324   | SNM5A  | 007214 | STP3D   | 037376 |
| RETD   | 031060 | ROTX   | 015220 | SBC5    | 021342   | SNM5B  | 007224 | STP4    | 041204 |
| RETD1  | 031624 | ROTXAD | 015344 | SBC6    | 021352   | SNM6A  | 007300 | STP4E   | 041214 |
| RETD2  | 032432 | ROTOA  | 014336 | SBC7    | 021372   | SNM6B  | 007310 | SUB0    | 007526 |
| RETD3  | 033164 | ROTOB  | 014346 | SBO     | 015376   | SNM7A  | 007362 | SUB0A   | 007536 |
| RETD4  | 033770 | ROTOC  | 014370 | SB2     | 015520   | SNM7B  | 007372 | SUB1    | 021114 |
| RETD5  | 034522 | ROT1A  | 014436 | SB4     | 015644   | SOB1   | 022502 | SUB2    | 021124 |
| RETE   | 031124 | ROT1B  | 014446 | SB5     | 015732   | SOB2   | 022510 | SUB3    | 021146 |
| RETE1  | 031666 | ROT1C  | 014472 | SB5A    | 015724   | SOB3   | 022520 | SUB4    | 021156 |
| RETE2  | 032476 | ROT1D  | 014502 | SB5X    | 015742   | SOB4   | 022540 | SUB5    | 021174 |
| RETE3  | 033230 | ROT1E  | 014532 | SB5XAD  | 015744   | SOPA   | 006204 | SUB6    | 021204 |
| RETE4  | 034034 | ROT2A  | 014604 | SB6     | 016004   | SOPB   | 006224 | SUB7    | 021224 |
| RETE5  | 034566 | ROT2B  | 014614 | SB6X    | 016014   | SOPBOA | 003614 | SWB1    | 020156 |
| RETF   | 031174 | ROT2C  | 014644 | SB7     | 016054   | SOPBOB | 003624 | SWB2    | 020166 |
| RETF1  | 031736 | ROT2D  | 014654 | SB7X    | 016064   | SOPB1A | 003736 | SWB3    | 020204 |
| RETF2  | 032546 | ROT2E  | 014710 | SB7XAD  | 016066   | SOPB1B | 003754 | SWR     | 042332 |
| RETF3  | 033300 | ROT3A  | 014756 | SCOPE = | 000240   | SOPB1C | 004020 | SWREG   | 000176 |
| RETF4  | 034104 | ROT3B  | 014766 | SC3     | 026052   | SOPB1D | 004040 | SW09 =  | 001000 |
| RETF5  | 034636 | ROT3C  | 015014 | SC4     | 026066   | SOPB2A | 004174 | SW10 =  | 002000 |
| RETG   | 031306 | ROT3D  | 015024 | SECPRT  | 027452   | SOPB2B | 004214 | SW11 =  | 004000 |
| RETG1  | 032050 | ROT3E  | 015052 | SETBR   | 025220   | SOPB2C | 004264 | SW12 =  | 010000 |
| RETG2  | 032660 | ROT4   | 015126 | SETCC   | 025240   | SOPB2D | 004310 | SXT0    | 022300 |
| RETG3  | 033412 | ROT5   | 015210 | SETCD   | 026050   | SOPB3A | 004776 | SXT1    | 022310 |

|         |         |         |        |        |        |       |        |       |        |
|---------|---------|---------|--------|--------|--------|-------|--------|-------|--------|
| SXT2    | 022340  | TRCSR = | 177560 | TST234 | 052614 | TST51 | 044326 | TS145 | 012446 |
| S0      | 042302  | TRC1    | 037356 | TST235 | 052712 | TST52 | 044410 | TS146 | 012542 |
| S1      | 042304  | TRPADR  | 037502 | TST236 | 053010 | TST53 | 044472 | TS147 | 012676 |
| S10     | 042326  | TRT =   | 000003 | TST237 | 053112 | TST54 | 044554 | TS15  | 001760 |
| S11     | 042330  | TRO     | 040212 | TST240 | 053210 | TST55 | 044634 | TS150 | 013054 |
| S2      | 042306  | TR2     | 040232 | TST241 | 053306 | TST56 | 044714 | TS151 | 013150 |
| S3      | 042310  | TR3     | 040360 | TST242 | 053424 | TST57 | 044774 | TS152 | 013304 |
| S4      | 042312  | TR4     | 040372 | TST243 | 053522 | TST60 | 045056 | TS153 | 013446 |
| S5      | 042314  | TR5     | 040362 | TST244 | 053620 | TST61 | 045140 | TS154 | 013616 |
| S6      | 042316  | TST160  | 046212 | TST245 | 053716 | TST62 | 045220 | TS155 | 013764 |
| S7      | 042320  | TST161  | 046240 | TST246 | 054012 | TS1   | 001112 | TS156 | 014050 |
| S8      | 042322  | TST162  | 046256 | TST247 | 054106 | TS10  | 001512 | TS157 | 014134 |
| S9      | 042324  | TST163  | 046274 | TST250 | 054202 | TS100 | 006474 | TS16  | 002012 |
| TAB =   | X000003 | TST164  | 046336 | TST251 | 054300 | TS101 | 006554 | TS160 | 014220 |
| TABLE   | 042214  | TST165  | 046374 | TST252 | 054376 | TS102 | 006700 | TS161 | 014304 |
| TBL1    | 014046  | TST166  | 046420 | TST253 | 054472 | TS103 | 006764 | TS162 | 014400 |
| TBL2    | 014132  | TST167  | 046450 | TST254 | 054566 | TS104 | 007100 | TS163 | 014542 |
| TDEC1   | 035610  | TST170  | 046506 | TST255 | 054676 | TS105 | 007156 | TS164 | 014720 |
| TDEC2   | 035634  | TST171  | 046532 | TST256 | 055006 | TS106 | 007242 | TS165 | 015062 |
| TDEC3   | 035702  | TST172  | 046560 | TST257 | 055122 | TS107 | 007324 | TS166 | 015136 |
| TDEC4   | 035750  | TST173  | 046602 | TST260 | 055232 | TS11  | 001562 | TS167 | 015222 |
| TDEC6   | 035764  | TST174  | 046640 | TST261 | 055342 | TS110 | 007406 | TS17  | 002044 |
| TDEC7   | 035774  | TST175  | 046674 | TST262 | 055456 | TS111 | 007442 | TS170 | 015270 |
| TDEC77  | 040062  | TST176  | 046752 | TST263 | 055566 | TS112 | 007476 | TS171 | 015346 |
| TDEC8   | 040052  | TST177  | 047050 | TST264 | 055676 | TS113 | 007552 | TS172 | 015414 |
| TEMP1   | 042264  | TST200  | 047146 | TST265 | 056006 | TS114 | 007714 | TS173 | 015456 |
| TEMP2   | 042266  | TST201  | 047250 | TST266 | 056122 | TS115 | 010004 | TS174 | 015536 |
| TEMP3   | 042270  | TST202  | 047346 | TST267 | 056232 | TS116 | 010130 | TS175 | 015600 |
| TEMP4   | 042272  | TST203  | 047444 | TST270 | 056342 | TS117 | 010214 | TS176 | 015660 |
| TEMP5   | 042274  | TST204  | 047546 | TST271 | 056456 | TS12  | 001626 | TS177 | 015746 |
| TEMP6   | 042276  | TST205  | 047644 | TST272 | 056566 | TS120 | 010252 | TS2   | 001202 |
| TENSAV  | 042014  | TST206  | 047772 | TST273 | 056676 | TS121 | 010310 | TS20  | 002110 |
| TESTN1  | 027502  | TST207  | 050076 | TST274 | 056770 | TS122 | 010346 | TS200 | 016016 |
| TEST1   | 020070  | TST210  | 050202 | TST275 | 057072 | TS123 | 010426 | TS201 | 016070 |
| TEST2   | 020100  | TST211  | 050306 | TST276 | 057202 | TS124 | 010466 | TS202 | 016434 |
| TEST3   | 020116  | TST212  | 050410 | TST277 | 057312 | TS125 | 010540 | TS203 | 017112 |
| THRPR1  | 042362  | TST213  | 050512 | TST300 | 057422 | TS126 | 010614 | TS204 | 017166 |
| TONT1   | 037302  | TST214  | 050614 | TST301 | 057530 | TS127 | 010672 | TS205 | 017252 |
| TO10    | 027362  | TST215  | 050720 | TST302 | 057636 | TS13  | 001676 | TS206 | 017342 |
| TO114   | 027422  | TST216  | 051024 | TST303 | 057744 | TS130 | 010734 | TS207 | 017432 |
| TO14    | 027372  | TST217  | 051126 | TST304 | 060054 | TS131 | 010776 | TS21  | 002160 |
| TO244   | 027432  | TST220  | 051230 | TST305 | 060164 | TS132 | 011040 | TS210 | 017524 |
| TO250   | 027442  | TST221  | 051326 | TST306 | 060272 | TS133 | 011116 | TS211 | 017642 |
| TO30    | 027402  | TST222  | 051424 | TST37  | 043722 | TS134 | 011174 | TS212 | 020004 |
| TO34    | 027412  | TST223  | 051522 | TST40  | 043756 | TS135 | 011272 | TS213 | 020042 |
| TO4     | 027352  | TST224  | 051624 | TST41  | 043772 | TS136 | 011434 | TS214 | 020126 |
| TPB =   | 177566  | TST225  | 051722 | TST42  | 044010 | TS137 | 011574 | TS215 | 020214 |
| TPS =   | 177564  | TST226  | 052020 | TST43  | 044044 | TS14  | 001726 | TS216 | 020400 |
| TRACE   | 037262  | TST227  | 052122 | TST44  | 044076 | TS140 | 011702 | TS217 | 020516 |
| TRAPA = | 000077  | TST230  | 052220 | TST45  | 044130 | TS141 | 012044 | TS22  | 002234 |
| TRAPB   | 037672  | TST231  | 052316 | TST46  | 044162 | TS142 | 012154 | TS220 | 020642 |
| TRAP10  | 042012  | TST232  | 052414 | TST47  | 044220 | TS143 | 012262 | TS221 | 021022 |
| TRAP24  | 042002  | TST233  | 052516 | TST50  | 044250 | TS144 | 012372 | TS222 | 021062 |

|       |        |       |        |       |        |         |        |          |        |
|-------|--------|-------|--------|-------|--------|---------|--------|----------|--------|
| TS223 | 021234 | TS301 | 030000 | TS360 | 036234 | TS66    | 005526 | YNTAB    | 027300 |
| TS224 | 021402 | TS302 | 030306 | TS361 | 036304 | TS67    | 005610 | \$APTHD  | 000330 |
| TS225 | 021550 | TS303 | 030516 | TS362 | 036372 | TS7     | 001434 | \$ATYC   | 061454 |
| TS226 | 021712 | TS304 | 030664 | TS363 | 036442 | TS70    | 005700 | \$ATY1   | 061430 |
| TS227 | 022062 | TS305 | 030720 | TS364 | 036542 | TS71    | 006002 | \$ATY3   | 061436 |
| TS23  | 002314 | TS306 | 030762 | TS365 | 036610 | TS72    | 006066 | \$ATY4   | 061446 |
| TS230 | 022242 | TS307 | 031026 | TS366 | 036674 | TS73    | 006156 | \$CHARC  | 061414 |
| TS231 | 022350 | TS31  | 002660 | TS367 | 036750 | TS74    | 006236 | \$CPUOP  | 000326 |
| TS232 | 022462 | TS310 | 031144 | TS37  | 003166 | TS75    | 006276 | \$CRLF   | 042344 |
| TS233 | 022550 | TS311 | 031416 | TS370 | 037022 | TS76    | 006336 | \$DBLK   | 062112 |
| TS234 | 022732 | TS312 | 031464 | TS371 | 037122 | TS77    | 006400 | \$DEVCT  | 000310 |
| TS235 | 023022 | TS313 | 031526 | TS372 | 037206 | TTCSR = | 177564 | \$DOAGN  | 060616 |
| TS236 | 023104 | TS314 | 031572 | TS373 | 037316 | TTT37   | 041024 | \$DTBL   | 062102 |
| TS237 | 023174 | TS315 | 031706 | TS374 | 037404 | TTYOUT  | 042336 | \$ENDAD  | 060606 |
| TS24  | 002360 | TS316 | 032146 | TS375 | 037744 | TTY11   | 041014 | \$ENDCT  | 060554 |
| TS240 | 023260 | TS317 | 032236 | TS376 | 040100 | TTY3    | 040734 | \$ENDMG  | 060625 |
| TS241 | 023342 | TS32  | 002724 | TS377 | 040244 | TTY4    | 041006 | \$ENULL  | 060622 |
| TS242 | 023432 | TS320 | 032272 | TS4   | 001274 | TYPCNT  | 042300 | \$ENV    | 000320 |
| TS243 | 023522 | TS321 | 032334 | TS40  | 003236 | TYPDS = | 104405 | \$ENVM   | 000321 |
| TS244 | 023614 | TS322 | 032400 | TS400 | 040430 | TYPE =  | 104401 | \$EOP    | 060530 |
| TS245 | 023716 | TS323 | 032516 | TS401 | 040542 | TYPOC = | 104402 | \$EOPCT  | 060546 |
| TS246 | 024024 | TS324 | 032770 | TS402 | 040646 | TYPON = | 104404 | \$ERN =  | 001162 |
| TS247 | 024132 | TS325 | 033024 | TS403 | 041030 | TYPOS = | 104403 | \$ERROR= | 000302 |
| TS25  | 002430 | TS326 | 033066 | TS404 | 041214 | USP1    | 024562 | \$ETABL  | 000320 |
| TS250 | 024240 | TS327 | 033132 | TS405 | 041272 | USP1A   | 024606 | \$ETEND  | 000330 |
| TS251 | 024350 | TS33  | 002774 | TS406 | 041340 | USP2    | 024662 | \$FATAL  | 000302 |
| TS252 | 024460 | TS330 | 033250 | TS407 | 041414 | USP3    | 024706 | \$FFLG   | 061674 |
| TS253 | 024536 | TS331 | 033504 | TS41  | 003306 | USP4    | 024742 | \$FILLC  | 061422 |
| TS254 | 024614 | TS332 | 033574 | TS410 | 041564 | USRM =  | 140000 | \$FILLS  | 061421 |
| TS255 | 024760 | TS333 | 033630 | TS411 | 041652 | USTBOT  | 027352 | \$GET42  | 060576 |
| TS256 | 025064 | TS334 | 033672 | TS412 | 041736 | VDEC1   | 036036 | \$HIBTS  | 000330 |
| TS257 | 025172 | TS335 | 033736 | TS42  | 003356 | VDEC10  | 036266 | \$HLT    | 060642 |
| TS26  | 002474 | TS336 | 034054 | TS43  | 003426 | VDEC11  | 036336 | \$LF     | 061425 |
| TS260 | 025432 | TS337 | 034326 | TS44  | 003506 | VDEC12  | 036346 | \$LFLG   | 061673 |
| TS261 | 025602 | TS34  | 003034 | TS45  | 003570 | VDEC13  | 036424 | \$MAIL   | 000300 |
| TS262 | 025656 | TS340 | 034362 | TS46  | 003634 | VDEC14  | 036434 | \$MBADR  | 000332 |
| TS263 | 025716 | TS341 | 034424 | TS47  | 003704 | VDEC2   | 036026 | \$MFLG   | 061672 |
| TS264 | 026134 | TS342 | 034470 | TS5   | 001332 | VDEC3   | 036106 | \$MSGAD  | 000314 |
| TS265 | 026214 | TS343 | 034606 | TS50  | 003764 | VDEC4   | 036076 | \$MSGLG  | 000316 |
| TS266 | 026274 | TS344 | 035042 | TS51  | 004050 | VDEC5   | 036156 | \$MSGTY  | 000300 |
| TS267 | 026424 | TS345 | 035076 | TS52  | 004136 | VDEC6   | 036146 | \$NULL   | 061420 |
| TS27  | 002544 | TS346 | 035140 | TS53  | 004224 | VDEC7   | 036226 | \$OCNT   | 062344 |
| TS270 | 026554 | TS347 | 035204 | TS54  | 004320 | VDEC8   | 036216 | \$OMODE  | 062346 |
| TS271 | 026610 | TS35  | 003072 | TS55  | 004432 | VDEC9   | 036276 | \$PASS   | 000306 |
| TS272 | 026650 | TS350 | 035322 | TS56  | 004552 | WATE    | 041156 | \$PASTM  | 000336 |
| TS273 | 026716 | TS351 | 035554 | TS57  | 004656 | WATE1   | 041102 | \$QUES   | 061424 |
| TS274 | 026760 | TS352 | 035610 | TS6   | 001370 | WATE2   | 041116 | \$RTNAD  | 060620 |
| TS275 | 027020 | TS353 | 035652 | TS60  | 004736 | WATE3   | 041144 | \$SETUP= | 000020 |
| TS276 | 027066 | TS354 | 035774 | TS61  | 005032 | XOR1    | 022414 | \$STUP = | 177777 |
| TS277 | 027146 | TS355 | 036044 | TS62  | 005122 | XOR2    | 022424 | \$SVPC = | 000400 |
| TS3   | 001236 | TS356 | 036114 | TS63  | 005272 | XOR3    | 022452 | \$SWR =  | 000000 |
| TS30  | 002610 | TS357 | 036164 | TS64  | 005354 | XXT     | 041646 | \$SWREG  | 000322 |
| TS300 | 027502 | TS36  | 003130 | TS65  | 005446 | YBR     | 025344 | \$TESTN  | 000304 |

|                |                |                |                |                  |
|----------------|----------------|----------------|----------------|------------------|
| \$TM = 000413  | \$TRAP2 062372 | \$TYPE 061136  | \$UNIT 000312  | \$\$GET4= 000000 |
| \$TPB 042340   | \$TRP = 000006 | \$TYPEC 061350 | \$UNITM 000340 | \$OFILL 062345   |
| \$TPCNT 042301 | \$TRPAD 062404 | \$TYPEX 061416 | \$USWR 000324  | . = 062422       |
| \$TPFLG 061423 | \$STM 000334   | \$TYPOC 062146 | \$X = 041746   | .\$X = 000330    |
| \$TPS 042342   | \$STNM= 000304 | \$TYPON 062162 | \$XX = 177666  |                  |
| \$TRAP 062350  | \$TYPDS 061676 | \$TYPOS 062122 | \$XXX = 000665 |                  |

. ABS. 062422 000

ERRORS DETECTED: 0

CJKDBA,CJKDBA/SOL/NL:TOC=CJKDBA.P11  
RUN-TIME: 49 62 1 SECONDS  
RUN-TIME RATIO: 260/113=2.2  
CORE USED: 19K (37 PAGES)