

000000

.REPT 0

IDENTIFICATION

PRODUCT CODE: AC-F141D-MC

PRODUCT NAME: CJKDBD0 DCF11-AA CPU DIAG

DATE: JAN-81

MAINTAINER: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1979,1981 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.

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

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 CHECKS 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.

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 PROCESSOR WITH DCF11-AA CHIP SET, 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 HAL (00000) (THIS LOCATION IS ALSO THE STATUS

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



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

```
*STA: 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

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. IF THE PROGRAM IS BEING RUN IN APT MODE (BIT 0 OF \$ENV SET TO A ONE) THEN THE LOCATION \$SWREG IS USED AS THE SWITCH REGISTER.

BIT #	OCTAL VALUE	FUNCTION
15	100000.....	HALT ON ERROR
13	020000.....	INHIBIT ERROR PRINTOUT
2	000004.....	PROGRAM RESERVED -- PROGRAM WILL SET IF CIS OPTION IS AVAILABLE
1	000002.....	30K SYSTEM DO NOT CHECK FOR TRAPS BETWEEN 28K-30K
0	000001.....	SKIP TRAPS TEST

NOTE: SWITCHES '15' AND '13' ONLY EFFECT PART THREE OF THIS DIAGNOSTIC. PARTS 1 AND 2 ALWAYS HALT ON ERROR.

ALSO, WITHIN THE APT TABLE, AN 8 BIT BYTE \$ENVM [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.

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.

296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312

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  
\$ENVM IS HIGH.  
WHILE RUNNING UNDER APT THE DIAGNOSTIC WILL NOT  
SUPPORT SPOOLING OF CONSOLE OUTPUTS.

9.2 POWER FAIL ROUTINE  
-----

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

```
313
314
315
316 .TITLE CJKDB-D DCF11-AA CPU DIAG.
317 .ENABLE ABS
318 .NLIST CND,MC,MD
319 .LIST ME
320 000240 SCOPE=NOP
321 000007 R7=X7
322 000006 R6=X6
323 177776 PS=177776
324 177560 TKS=177560
325 177562 TKB=177562
326 177564 TPS=177564
327 177566 TPB=177566
328 140000 USRM=140000
329 030000 PUSRM=30000
330 .SBTTL ACT11 HOOKS
331
332 *****
333 ;HOOKS REQUIRED BY ACT11
334 000400 $SVPC= ;SAVE PC
335 000046 -.46
336 000046 $ENDAD ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
337 000052 -.52
338 000052 .WORD 0 ;;2)SET LOC.52 TO ZERO
339 000400 .= $SVPC ;; RESTORE PC
340 000300 .=300
341 .SBTTL APT MAILBOX-E TABLE
342
343 *****
344 .EVEN
345 000300 $MAIL: ;; APT MAILBOX
346 000300 000000 $MSGTY: .WORD AMSGTY ;; MESSAGE TYPE CODE
347 000302 000000 $FATAL: .WORD AFATAL ;; FA AL ERROR NUMBER
348 000304 000000 $TESTN: .WORD ATESTN ;; TEST NUMBER
349 000306 000000 $PASS: .WORD APASS ;; PASS COUNT
350 000310 000000 $DEVCT: .WORD ADEVCT ;; DEVICE COUNT
351 000312 000000 $UNIT: .WORD AUNIT ;; I/O UNIT NUMBER
352 000314 000000 $MSGAD: .WORD AMSGAD ;; MESSAGE ADDRESS
353 000316 000000 $MSGLG: .WORD AMSGLG ;; MESSAGE LENGTH
354 000320 $ETABLE: ;; APT ENVIRONMENT TABLE
355 000320 000 $ENV: .BYTE AENV ;; ENVIRONMENT BYTE
356 000321 000 $ENVM: .BYTE AENVM ;; ENVIRONMENT MODE BITS
357 000322 000000 $SWREG: .WORD ASWREG ;; APT SWITCH REGISTER
358 000324 000000 $USWR: .WORD AUSWR ;; USER SWITCHES
359 000326 000000 $CPUOP: .WORD ACPUOP ;; CPU TYPE,OPTIONS
360 * BIT 15-11=CPU TYPE
361 * 11/04=01,11/05=02,11/20=03,11/40=04,11/45 05
362 * 11/70=06,PDQ=07,Q=10
363 * BIT 10=REAL TIME CLOCK
364 * BIT 9=FLOATING POINT PROCESSOR
365 * BIT 8=MEMORY MANAGEMENT
366 000330 $ETEND:
367 .MEXIT
368 .SBTTL APT PARAMETER BLOCK
```

```
369
370
371
372
373      000330
374      000024
375 000024 000200
376      000044
377 000044 000330
378      000330
379
380
381
382
383 000330
384 000330 000000
385 000332 000300
386 000334 000013
387 000336 000020
388 000340 000005
389 000342 000014
390
391
392
393      000004
394 000004 027324
395 000006 000000
396 000010 027334
397 000012 000000
398 000014 027344
399      000030
400 000030 027354
401 000032 000000
402 000034 027364
403 000036 000000
404      000114
405 000114 027374
406 000116 000000
407      000244
408 000244 027404
409 000246 000000
410 000250 027414
411 000252 000000
412
413
414 000172 000172
415 000174 000000
416 000176 000000
417
418
419
420
421
422 000370 000000 000000 000000
423 000376 000000 000000 000000
424 000404 000001 000001 177777

:*****
:SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
:*****
.$X=      ;;SAVE CURRENT LOCATION
-24      ;;SET POWER FAIL TO POINT TO START OF PROGRAM
200      ;;FOR APT START UP
-44      ;;POINT TO APT INDIRECT ADDRESS PNTR.
$APTHDR  ;;POINT TO APT HEADER BLOCK
-.$X     ;;RESET LOCATION COUNTER
:*****
:SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
:INTERFACE SPEC.

$APTHD:
$HIBTS: .WORD 0      ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$MADR:  .WORD $MAIL  ;;ADDRESS OF APT MAILBOX (BITS 0-15)
$STIM:  .WORD 13     ;;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
```

```
425 ;*****
426 ;SET UP STARTING ADDRESS
427 000302 $ERROR=$FATAL
428 000304 $STSTM=$TESTN
429 001000 .=1000
430 001000 000000 STBOT: .WORD 0 ;STACK POINTER
431
432 000200 .=200
433 000200 000167 000576 JMP START
434 000204 012706 001000 MOV #STBOT,R6 ;SET STACK POINTER
435 000210 012702 000304 MOV #STESTN,R2 ;SET MAILBOX POINTER
436 000214 000137 JMP @PC+ ;JUMP TO SUBTEST
437 000216 000000 0 ;ADDR. OF SUBTEST GOES HERE
438
439 001002 .=1002
440 .SBTTL **STARTING OF CPU TEST **
441 001002 012737 061074 000024 START: MOV #PWRDN,@#24 ;SET UP FOR POWER FAIL
442 001010 012737 000000 000306 MOV #0,@$SPASS ;CLEAR PASS COUNT
443 001016 012737 000016 062540 MOV #16,@$PASSPT ;SET PRINT COUNTER
444 001024 012706 001000 RESTRT: MOV #STBOT,R6 ;INITIALIZE STACK POINTER
445 001030 012702 000304 MOV #STESTN,R2 ;SET UP POINTER TO MESSAGE TYPE
446 001034 012737 000000 000304 MOV #0,@$STSTM ;CLEAR TEST NUMBER
447 001042 012737 000000 000302 MOV #0,@$ERROR ;CLEAR ERROR NUMBER
448 001050 012737 000000 000300 MOV #0,@$MSGTY ;CLEAR MESSAGE TYPE (FOR APT)
449 ;*****
450 ;TEST 1 CHECK BRANCHES ON Z BIT
451 ;*****
452 001056 005212 TS1: INC (R2) ;UPDATE TEST NUMBER
453 001060 022712 000001 CMP #1,(R2) ;SEQUENCE ERROR?
454 001064 001024 BNE TS2-10 ;BR TO ERROR HALT ON SEQ ERROR
455 001066 000257 CCC ;CLEAR ALL CONDITION CODES
456 001070 001401 BEQ BRA1 ;SHOULD BRANCH
457 001072 000404 BR BRA2 ;BAD BRANCH OF Z-BIT
458 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
459 ; BRANCH INSTRUCTION AND <
460 ; REPLACE THE MOVE INSTRUCTION <
461 ; FOLLOWING W/ 774 <
462 001074 BRA1:
463 001074 012742 000001 MOV #1,-(R2) ;MOVE TO MAILBOX # ***** 1 *****
464 001100 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
465 001102 000000 HALT ;SHOULD HAVE BRANCHED: Z-0
466 001104 BRA2:
467 001104 001004 BNE BRA3
468 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
469 ; CONDITIONAL BRANCH INST. AND <
470 ; REPLACE THE MOVE INSTRUCTION <
471 ; WHICH FOLLOWS W/ 767 <
472 001106 012742 000002 MOV #2,-(R2) ;MOVE TO MAILBOX # ***** 2 *****
473 001112 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
474 001114 000000 HALT ;
475 001116 000264 BRA3: SEZ
476 001120 001001 BNE BRA4
477 001122 000404 BR BRA5
478 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--
479 ; BRANCH INSTRUCTION AND <-
480 ; REPLACE THE MOVE INSTRUCTION <
```

```
481  
482 001124  
483 001124 012742 000003  
484 001130 005242  
485 001132 000000  
486 001134  
487 001134 001404  
488  
489  
490  
491  
492 001136 012742 000004  
493 001142 005242  
494 001144 000000  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511 001146 005212  
512 001150 022712 000002  
513 001154 001006  
514 001156 012737 000000 000000  
515  
516 001164 005737 000000  
517 001170 001404  
518  
519  
520  
521  
522 001172 012742 000005  
523 001176 005242  
524 001200 000000  
525  
526  
527  
528  
529  
530 001202 005212  
531 001204 022712 000003  
532 001210 001007  
533 001212 012737 125252 000000  
534  
535 001220 022737 125252 000000  
536 001226 001404
```

```
;  
: FOLLOWING W/ 760  
: < =  
BRA4: MOV #3, -(R2) ; MOVE TO MAILBOX # ***** 3 *****  
: INC -(R2) ; SET MSGTYP TO FATAL ERROR  
: HALT ; SHOULD NOT HAVE BRANCHED HERE ON Z-1  
BRA5: BEQ TS2  
: ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==--=  
: ; CONDITIONAL BRANCH INST. AND <==--=  
: ; REPLACE THE MOVE INSTRUCTION <----=  
: ; WHICH FOLLOWS W/ 753 <=====  
MOV #4, -(R2) ; MOVE TO MAILBOX # ***** 4 *****  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; SHOULD HAVE BRANCHED ON Z=1  
; 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  
\*\*\*\*\*

```
TS2: INC (R2) ; UPDATE TEST NUMBER  
: CMP #2, (R2) ; SEQUENCE ERROR?  
: BNE TS3-10 ; BR TO ERROR HALT ON SEQ ERROR  
: MOV #0, @#0 ; MOVE ZEROES THRU ADDRESS LINES, DATA  
: ; LINES AND INTERNAL PATHS  
: TST @#0 ; SUCCESSFUL?  
: BEQ TS3
```

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

```
MOV #5, -(R2) ; MOVE TO MAILBOX # ***** 5 *****  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; DATA INCORRECT  
; OR SEQUENCE ERROR
```

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

```
TS3: INC (R2) ; UPDATE TEST NUMBER  
: CMP #3, (R2) ; SEQUENCE ERROR?  
: BNE TS4-10 ; BR TO ERROR HALT ON SEQ ERROR  
: MOV #125252, @#0 ; MOVE ALTERNATING ONES AND ZEROES  
: ; THRU DATA PATHS  
: CMP #125252, @#0 ; SUCCESSFUL  
: BEQ TS4
```

```

537 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
538 ; CONDITIONAL BRANCH INST. AND <===
539 ; REPLACE THE MOVE INSTRUCTION <===
540 ; WHICH FOLLOWS W/ 770 <===
541 001230 012742 000006 MOV #6,-(R2) ;MOVE TO MAILBOX # ***** 6 *****
542 001234 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
543 001236 000000 HALT ;DATA INCORRECT
544 ; OR SEQUENCE ERROR
545

```

```

546 :*****
547 :TEST 4 TEST OF PATTERN 052525 IN DATA PATH
548 :*****

```

```

549 001240 005212 TS4: INC (R2) ;UPDATE TEST NUMBER
550 001242 022712 000004 CMP #4,(R2) ;SEQUENCE ERROR?
551 001246 001007 BNE TS5-10 ;BR TO ERROR HALT ON SEQ ERROR
552 001250 012737 052525 000000 MOV #052525,@#0 ;MOVE ALTERNATING ZEROES AND ONES
553 ;THRU DATA PATH
554 001256 022737 052525 000000 CMP #052525,@#0 ;SUCCESSFUL?
555 001264 001404 BEQ TS5

```

```

556 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
557 ; CONDITIONAL BRANCH INST. AND <===
558 ; REPLACE THE MOVE INSTRUCTION <===
559 ; WHICH FOLLOWS W/ 770 <===
560 001266 012742 000007 MOV #7,-(R2) ;MOVE TO MAILBOX # ***** 7 *****
561 001272 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
562 001274 000000 HALT ;DATA INCORRECT
563 ; OR SEQUENCE ERROR
564

```

```

565 :*****
566 :TEST 5 TEST OF ALL ONES IN DATA PATH
567 :*****

```

```

568 001276 005212 TS5: INC (R2) ;UPDATE TEST NUMBER
569 001300 022712 000005 CMP #5,(R2) ;SEQUENCE ERROR?
570 001304 001007 BNE TS6-10 ;BR TO ERROR HALT ON SEQ ERROR
571 001306 012737 177777 000000 MOV #177777,@#0 ;MOVE ONES THRU DATA PATH
572 001314 022737 177777 000000 CMP #177777,@#0 ;SUCCESSFUL
573 001322 001404 BEQ TS6

```

```

574 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
575 ; CONDITIONAL BRANCH INST. AND <- -
576 ; REPLACE THE MOVE INSTRUCTION <- -
577 ; WHICH FOLLOWS W/ 770 <- -
578 001324 012742 000010 MOV #10,-(R2) ;MOVE TO MAILBOX # ***** 10 *****
579 001330 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
580 001332 000000 HALT ;DATA INCORRECT
581 ; OR SEQUENCE ERROR
582

```

```

583 :*****
584 :SBTTL B-REGISTER TEST
585 :

```

```

586 : THE B-REGISTER (LOCATION 0) SHIFTING LOGIC TESTS ARE USED
587 : TO TEST THAT THE B-REGISTER CAN HOLD VARIOUS DATA PATTERNS AND THAT
588 : THE ASSOCIATED LOGIC SUPPORTS THE SHIFTING FUNCTIONS WITHIN THE
589 : B-REGISTER AND C-BIT.
590 : A ONE IS SHIFTED THROUGH EVERY BIT IN THE B-REGISTER AND C-BIT IN
591 : BOTH DIRECTIONS.
592 : THE B-REGISTER ITSELF IS TESTED IN ITS ABILITY AS A BUFFER AND AS

```



```
593 ;A SHIFT REGISTER. DATA IS ALSO PASSED THROUGH THE DATA PATH AND ALU.  
594 ; IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0) TO SEE  
595 ; WHICH BITS OF THE B-REGISTER MAY BE FAILING.  
596 ;  
597 ;*****  
598 ;TEST 6 SHIFT BIT 0 TO BIT 1  
599 ;*****  
600 TS6: INC (R2) ;UPDATE TEST NUMBER  
601 CMP #6,(R2) ;SEQUENCE ERROR?  
602 BNE TS7-10 ;BR TO ERROR HALT ON SEQ ERROR  
603 CLC ;CLEAR CARRY BIT  
604 MOV #1,@#0 ;LOAD A 1  
605 ROL @#0 ;SHIFT LEFT  
606 CMP #2,@#0 ;SUCCESSFUL  
607 BEQ TS7  
608 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -  
609 ; CONDITIONAL BRANCH INST. AND < -  
610 ; REPLACE THE MOVE INSTRUCTION < -  
611 ; WHICH FOLLOWS W/ 765 < -  
612 C01370 012742 000011 MOV #11,-(R2) ;MOVE TO MAILBOX # ***** 11 *****  
613 001374 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
614 001376 000000 HALT ;BIT 1 NOT SET  
615 ; OR SEQUENCE ERROR  
616 ;  
617 ;*****  
618 ;TEST 7 SHIFT CARRY INTO BIT 0  
619 ;*****  
620 TS7: INC (R2) ;UPDATE TEST NUMBER  
621 CMP #7,(R2) ;SEQUENCE ERROR?  
622 BNE TS10-10 ;BR TO ERROR HALT ON SEQ ERROR  
623 MOV #0,@#0 ;CLEAR LOCATION  
624 SEC ;SET CARRY  
625 ROL @#0 ;ROTATE CARRY BIT TO BIT 0  
626 BCC TS10  
627 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <  
628 ; CONDITIONAL BRANCH INST. AND <  
629 ; REPLACE THE MOVE INSTRUCTION <  
630 ; WHICH FOLLOWS W/ 770 <  
631 001426 012742 000012 MOV #12,-(R2) ;MOVE TO MAILBOX # ***** 12 *****  
632 001432 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
633 001434 000000 HALT ;CARRY CLEAR  
634 ; OR SEQUENCE ERROR  
635 001436 022737 000001 000000 CMP #1,@#0 ;BIT 0 SET  
636 001444 001404 BEQ TS10  
637 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-  
638 ; CONDITIONAL BRANCH INST. AND <  
639 ; REPLACE THE MOVE INSTRUCTION <  
640 ; WHICH FOLLOWS W/ 760 <  
641 001446 012742 000013 MOV #13,-(R2) ;MOVE TO MAILBOX # ***** 13 *****  
642 001452 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
643 001454 000000 HALT ;BIT 0 NOT SET  
644 ; OR SEQUENCE ERROR  
645 ;  
646 ;*****  
647 ;TEST 10 LEFT SHIFT FROM BIT 0 TO C-BIT  
648 ;*****
```

```

649 001456 005212          TS10:  INC      (R2)          ;UPDATE TEST NUMBER
650 001460 022712 000010    CMP      #10,(R2)         ;SEQUENCE ERROR?
651 001464 001014          BNE     TS11-10         ;BR TO ERROR HALT ON SEQ ERROR
652 001466 012737 000001 000000 MOV     #1,@#0         ;SET BIT 0
653 001474 012700 177757    MOV     #-21,R0        ;SET BIT COUNTER
654 001500 000241          CLC          ;CLEAR C-BIT
655 001502 005200          SHL:   INC      R0          ;INCREMENT BIT COUNTER
656 001504 001404          BEQ     SHLE          ;BR TO ERROR HALT IF BIT IS LOST
657 001506 006137 000000    ROL     @#0           ;SHIFT LEFT ONE POSITION
658 001512 103373          BCC     SHL          ;BRANCH IF C-BIT NOT SET
659 001514 001404          BEQ     TS11
660                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
661                                     ; CONDITIONAL BRANCH INST. AND <====
662                                     ; REPLACE THE MOVE INSTRUCTION <====
663                                     ; WHICH FOLLOWS W/ 763 <====
664 001516                                     SHLE:
665 001516 012742 000014    MOV     #14,-(R2)      ;MOVE TO MAILBOX # ***** 14 *****
666 001522 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
667 001524 000000          HALT          ;LEFT SHIFTING LOGIC FAILED
668                                     ; OR SEQUENCE ERROR
669
670 :*****
671 :TEST 11      SHIFT BIT 15 TO BIT 14
672 :*****
673 001526 005212          TS11:  INC      (R2)          ;UPDATE TEST NUMBER
674 001530 022712 000011    CMP     #11,(R2)       ;SEQUENCE ERROR?
675 001534 001012          BNE     TS12-10         ;BR TO ERROR HALT ON SEQ ERROR
676 001536 012737 100000 000000 MOV     #100000,@#0    ;SET BIT 15
677 001544 000241          CLC          ;CLEAR CARRY
678 001546 006037 000000    ROR     @#0           ;SHIFT BIT 15 TO BIT 14
679 001552 022737 040000 000000 CMP     #40000,@#0     ;SUCCESSFUL
680 001560 001404          BEQ     TS12
681                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
682                                     ; CONDITIONAL BRANCH INST. AND <
683                                     ; REPLACE THE MOVE INSTRUCTION <
684                                     ; WHICH FOLLOWS W/ 765 <
685 001562 012742 000015    MOV     #15,-(R2)      ;MOVE TO MAILBOX # ***** 15 *****
686 001566 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
687 001570 000000          HALT          ;BIT 14 NOT SET
688                                     ; OR SEQUENCE ERROR
689
690 :*****
691 :TEST 12      RIGHT SHIFT FROM BIT 15 TO C-BIT
692 :*****
693 001572 005212          TS12:  INC      (R2)          ;UPDATE TEST NUMBER
694 001574 022712 000012    CMP     #12,(R2)       ;SEQUENCE ERROR?
695 001600 001014          BNE     TS13-10         ;BR TO ERROR HALT ON SEQ ERROR
696 001602 012737 100000 000000 MOV     #100000,@#0    ;SET BIT 15
697 001610 012700 177757    MOV     #-21,R0        ;SET BIT COUNTER
698 001614 000241          CLC          ;CLEAR C-BIT
699 001616 005200          SHR:   INC      R0          ;INCREMENT BIT COUNTER
700 001620 001404          BEQ     SHRE          ;BR TO ERROR HALT IF BIT IS LOST
701 001622 006037 000000    ROR     @#0           ;ROTATE RIGHT ONE POSITION
702 001626 103373          BCC     SHR          ;BRANCH IF C-BIT CLEAR
703 001630 001404          BEQ     TS13
704                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS

```

705  
706  
707  
708 001632  
709 001632 012742 000016  
710 001636 005242  
711 001640 000000  
712  
713

```
          ;          CONDITIONAL BRANCH INST. AND <- - -  
          ;          REPLACE THE MOVE INSTRUCTION <- - -  
          ;          WHICH FOLLOWS W/ 763 <- - -  
SHRE:    MOV      #16,-(R2) ;MOVE TO MAILBOX # ***** 16 *****  
          INC      -(R2)   ;SET MSGTYP TO FATAL ERROR  
          HALT    ;RIGHT SHIFT LOGIC FAILED  
          ; OR SEQUENCE ERROR
```

714  
715  
716  
717  
718  
719  
720  
721  
722  
723  
724  
725  
726  
727  
728  
729  
730  
731  
732  
733  
734  
735  
736

```
*****  
:SBTTL SCRATCH AD TESTS  
:  
: THE SCRATCH PAD TESTS ARE USED TO VERIFY THAT VARIOUS  
: DATA PATTERNS CAN BE SUCCESSFULLY HELD IN THE SCRATCH PAD  
: CIRCUITRY. MOVE AND COMPARE INSTRUCTIONS ARE USED TO TEST THAT  
: R0 CAN HOLD VARIOUS DATA PATTERNS. EACH DATA PATTERN IS  
: MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR SCOPING. THE  
: SUCCESSFUL COMPLETION OF THESE TESTS SHOULD VERIFY THE CIRCUITRY EXTERNAL  
: TO THE SCRATCH PAD ITSELF.  
: THE REMAINDER OF THE GENERAL REGISTERS ARE TESTED BY MOVING  
: A BIT INTO BIT 0 OF THE REGISTER AND SHIFTING IT LEFT ONE  
: BIT AT A TIME INTO THE CARRY BIT. THE RESULT IS THEN CHECKED TO INSURE THAT  
: NO BITS WERE PICKED. THE PROCEDURE IS THEN REPEATED UNDER OPPOSITE  
: CONDITIONS. THE GENERAL REGISTER AND THE CARRY BIT ARE SET TO  
: ALL ONES, AND A ZERO IS SHIFTED LEFT FROM BIT 0 INTO THE CARRY BIT.  
: THE RESULT IS THEN CHECKED TO INSURE THAT NO ZEROES WERE PICKED.  
: AT THIS POINT ALL OF THE GENERAL REGISTERS HAVE BEEN EXERCISED  
: AS WELL AS REGISTER 11.  
*****
```

737 001642 005212  
738 001644 022712 000013  
739 001650 001004  
740  
741 001652 012700 000000  
742 001656 005700  
743 001660 001404  
744  
745  
746  
747

```
*****  
:TEST 13 TEST IF R0 CAN HOLD ALL ZEROES  
*****  
TS13:    INC      (R2) ;UPDATE TEST NUMBER  
          CMP      #15,(R2) ;SEQUENCE ERROR?  
          BNE     TS14-10 ;BR TO ERROR HALT ON SEQ ERROR  
  
          MOV      #0,R0 ;MOVE ZEROES TO R0  
          TST     R0 ;SUCCESSFUL?  
          BEQ     TS14  
  
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <  
          ;          CONDITIONAL BRANCH INST. AND <  
          ;          REPLACE THE MOVE INSTRUCTION <=-  
          ;          WHICH FOLLOWS W/ 773 <  
748 001662 012742 000017  
749 001666 005242  
750 001670 000000  
751  
752  
753
```

754  
755  
756 001672 005212  
757 001674 022712 000014  
758 001700 001005  
759 001702 012700 125252  
760 001706 020027 125252

```
*****  
:TEST 14 TEST IF R0 CAN HOLD ONES AND ZEROES  
*****  
TS14:    INC      (R2) ;UPDATE TEST NUMBER  
          CMP      #14,(R2) ;SEQUENCE ERROR?  
          BNE     TS15-10 ;BR TO ERROR HALT ON SEQ ERROR  
          MOV      #125252,R0 ;MOVE ALTERNATING ONES AND ZEROES TO R0  
          CMP     R0,#125252 ;SUCCESSFUL?
```

```
761 001712 001404          BEQ      TS15          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -
762                                     ;          CONDITIONAL BRANCH INST. AND <== -
763                                     ;          REPLACE THE MOVE INSTRUCTION <== -
764                                     ;          WHICH FOLLOWS W/ 772 <
765                                     ;
766 001714 012742 000020    MOV      #20,-(R2)      ;MOVE TO MAILBOX # ***** 20 *****
767 001720 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
768 001722 000000          HALT                    ;RO NOT 125252
769                                     ; OR SEQUENCE ERROR
```

```
770
771
772 :*****
773 :TEST 15 TEST IF R0 CAN HOLD ZEROS AND ONES
774 :*****
```

```
774 001724 005212          TS15: INC      (R2)          ;UPDATE TEST NUMBER
775 001726 022712 000015    CMP      #15,(R2)      ;SEQUENCE ERROR?
776 001732 001005          BNE     TS16-10        ;BR TO ERROR HALT ON SEQ ERROR
777 001734 012700 052525    MOV      #052525,R0    ;MOVE ALTERNATING ZEROS AND ONES TO R0
778 001740 020027 052525    CMP      R0,#052525    ;SUCCESSFUL?
779 001744 001404          BEQ     TS16
```

```
780                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
781                                     ;          CONDITIONAL BRANCH INST. AND <
782                                     ;          REPLACE THE MOVE INSTRUCTION <
783                                     ;          WHICH FOLLOWS W/ 772 <
784 001746 012742 000021    MOV      #21,-(R2)      ;MOVE TO MAILBOX # ***** 21 *****
785 001752 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
786 001754 000000          HALT                    ;RO NOT 52525
787                                     ; OR SEQUENCE ERROR
```

```
788
789 :*****
790 :TEST 16 TEST IF R0 CAN HOLD ALL ONES
791 :*****
```

```
792 001756 005212          TS16: INC      (R2)          ;UPDATE TEST NUMBER
793 001760 022712 000016    CMP      #16,(R2)      ;SEQUENCE ERROR?
794 001764 001005          BNE     TS17-10        ;BR TO ERROR HALT ON SEQ ERROR
795 001766 012700 177777    MOV      #177777,R0    ;MOVE ALL ONES TO R0
796 001772 020027 177777    CMP      R0,#177777    ;SUCCESSFUL?
797 001776 001404          BEQ     TS17
```

```
798                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
799                                     ;          CONDITIONAL BRANCH INST. AND <
800                                     ;          REPLACE THE MOVE INSTRUCTION <
801                                     ;          WHICH FOLLOWS W/ 772 <
802 002000 012742 000022    MOV      #22,-(R2)      ;MOVE TO MAILBOX # ***** 22 *****
803 002004 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
804 002006 000000          HALT                    ;RO NOT 177777
805                                     ; OR SEQUENCE ERROR
```

```
806
807 :*****
808 :TEST 17 TEST IF R1 CAN HOLD A ONE IN ALL BITS
809 :*****
```

```
810 002010 005212          TS17: INC      (R2)          ;UPDATE TEST NUMBER
811 002012 022712 000017    CMP      #17,(R2)      ;SEQUENCE ERROR?
812 002016 001012          BNE     TS20-10        ;BR TO ERROR HALT ON SEQ ERROR
813 002020 012701 000001    MOV      #1,R1         ;SET BIT 0
814 002024 012700 177757    MOV      #-21,R0       ;SET BIT COUNTER
815 002030 000241          CLC                    ;CLEAR C-BIT
816 002032 005200          REG1: INC      R0      ;INCREMENT BIT COUNTER
```

```
817 002034 001403 BEQ REG1E ;BR TO ERROR HALT IF BIT IS LOST
818 002036 006101 ROL R1 ;ROTATE 1 POSITION
819 002040 103374 BCC REG1 ;ALL DONE
820 002042 001404 BEQ TS20
821 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < - =
822 : CONDITIONAL BRANCH INST. AND < == =
823 : REPLACE THE MOVE INSTRUCTION < == =
824 : WHICH FOLLOWS W/ 765 < == =
825 002044 REG1E:
826 002044 012742 000025 MOV #23,-(R2) ;MOVE TO MAILBOX # ***** 23 *****
827 002050 INC -(R2) ;SET MSGTYP TO FATAL ERROR
828 002052 000000 HALT ;FAILURE WITH R1
829 ; OR SEQUENCE ERROR
830
831 :*****
832 :TEST 20 TEST IF R1 CAN HOLD A ZERO IN ALL BITS
833 :*****
834 002054 005212 TS20: INC (R2) ;UPDATE TEST NUMBER
835 002056 022712 000020 CMP #20,(R2) ;SEQUENCE ERROR?
836 002062 001014 BNE TS21-10 ;BR TO ERROR HALT ON SEQ ERROR
837 002064 012701 177776 MOV #-2,R1 ;SET ALL ONES IN R1 EXCEPT FOR BIT 0
838 002070 012700 177757 MOV #-21,R0 ;SET BIT COUNTER
839 002074 000261 SEC ;SET C-BIT
840 002076 005200 REG1A: INC R0 ;INCREMENT COUNTER
841 002100 001405 BEQ R1ERR ;BR TO ERROR HALT IF COUNTER=0
842 002102 006101 ROL R1 ;ROTATE 1 POSITION
843 002104 103774 BCS REG1A ;CONTINUE UNTIL C-BIT IS CLEAR
844 002106 022701 177777 CMP #-1,R1 ;CHECK DATA IN R1
845 002112 001404 BEQ TS21
846 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < - =
847 : CONDITIONAL BRANCH INST. AND < == =
848 : REPLACE THE MOVE INSTRUCTION < == =
849 : WHICH FOLLOWS W/ 763 < == =
850 002114 R1ERR:
851 002114 012742 000024 MOV #24,-(R2) ;MOVE TO MAILBOX # ***** 24 *****
852 002120 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
853 002122 000000 HALT ;FAILURE WITH R1
854 ; OR SEQUENCE ERROR
855 :*****
856 :TEST 21 TEST IF R2 CAN HOLD A ONE IN ALL BITS
857 :*****
858 002124 005212 TS21: INC (R2) ;UPDATE TEST NUMBER
859 002126 022712 000021 CMP #21,(R2) ;SEQUENCE ERROR?
860 002132 001012 BNE REG2A-14 ;BR TO ERROR HALT ON SEQ ERROR
861 002134 012702 000001 MOV #1,R2 ;SET BIT 0
862 002140 012700 177757 MOV #-21,R0 ;SET BIT COUNTER
863 002144 000241 CLC ;CLEAR C-BIT
864 002146 005200 REG2: INC R0 ;INCREMENT BIT COUNTER
865 002150 001403 BEQ REG2A-14 ;BR TO ERROR HALT IF BIT IS LOST
866 002152 006102 ROL R2 ;ROTATE 1 POSITION
867 002154 103374 BCC REG2 ;ALL DONE
868 002156 001404 BEQ REG2A
869 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < - =
870 : BRANCH INSTRUCTION AND < == =
871 : REPLACE THE MOVE INSTRUCTION < -
872 : FOLLOWING W/ 771 <
```

```

873 002160 012702 000304      MOV    #$TESTN,R2      ;RESTORE POINTER
874 002164 012742 000025      MOV    #25,-(R2)      ;MOVE TO MAILBOX # ***** 25 *****
875 002170 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
876 002172 000000              HALT                   ;FAILURE WITH R2
877 002174 012702 000304      REG2A: MOV   #$TESTN,R2      ;RESTORE POINTER

```

```

878
879
880 ;*****
881 ;TEST 22      TEST IF R2 CAN HOLD A ZERO IN ALL BITS
882 ;*****

```

```

882 002200 005212              TS22: INC    (R2)          ;UPDATE TEST NUMBER
883 002202 022712 000022      CMP    #22,(R2)      ;SEQUENCE ERROR?
884 002206 001020              BNE   TS23-10        ;BR TO ERROR HALT ON SEQ ERROR
885 002210 012702 177776      MOV    #-2,R2        ;SET ALL ONES IN R2 EXCEPT FOR BIT 0
886 002214 012700 177757      MOV    #-21,R0       ;SET BIT COUNTER
887 002220 000261              SEC                   ;SET C-BIT
888 002222 005200      REG2B: INC    R0          ;INCREMENT BIT COUNTER
889 002224 001407              BEQ   R2ERR         ;BR TO ERROR HALT IF COUNTER=0
890 002226 006102              ROL   R2             ;ROTATE 1 POSITION
891 002230 103774              BCS   REG2B         ;CONTINUE UNTIL C-BIT IS CLEAR
892 002232 022702 177777      CMP    #-1,R2        ;CHECK DATA IN R2
893 002236 001406              BEQ   REG2C
894 002240 012702 000304      MOV    #$TESTN,R2    ;RESTORE POINTER
895 002244
896 002244 012742 000026      R2ERR: MOV   #26,-(R2)   ;MOVE TO MAILBOX # ***** 26 *****
897 002250 005242              INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
898 002252 000000              HALT                   ;FAILURE WITH R2
899 002254 012702 000304      REG2C: MOV   #$TESTN,R2    ;RESTORE POINTER

```

```

900
901 ;*****
902 ;TEST 23      TEST IF R3 CAN HOLD A ONE IN ALL BITS
903 ;*****

```

```

904 002260 005212              TS23: INC    (R2)          ;UPDATE TEST NUMBER
905 002262 022712 000023      CMP    #23,(R2)      ;SEQUENCE ERROR?
906 002266 001012              BNE   TS24-10        ;BR TO ERROR HALT ON SEQ ERPOP
907 002270 012703 000001      MOV    #1,R3         ;SET BIT 0
908 002274 012700 177757      MOV    #-21,R0       ;SET BIT COUNTER
909 002300 000241              CLC                   ;CLEAR C-BIT
910 002302 005200      REG3: INC    R0          ;INCREMENT BIT COUNTER
911 002304 001403              BEQ   REG3E         ;BR TO ERROR HALT IF BIT IS LOST
912 002306 006103              ROL   R3             ;ROTATE 1 POSITION
913 002310 103374              BCC   REG3
914 002312 001404              BEQ   TS24

```

```

915 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
916 ; CONDITIONAL BRANCH INST. AND
917 ; REPLACE THE MOVE INSTRUCTION
918 ; WHICH FOLLOWS W/ 765

```

```

919 002314      REG3E:
920 002314 012742 000027      MOV    #27,-(R2)   ;MOVE TO MAILBOX # ***** 27 *****
921 002320 005242              INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
922 002322 000000              HALT                   ;FAILURE WITH R3
923 ; OR SEQUENCE ERROR

```

```

924
925 ;*****
926 ;TEST 24      TEST IF R3 CAN HOLD A ZERO IN ALL BITS
927 ;*****

```

```

928 002324 005212              TS24: INC    (R2)          ;UPDATE TEST NUMBER

```

929 002326 022712 000024  
 930 002332 001014  
 931 002334 012703 177776  
 932 002340 012700 177757  
 933 002344 000261  
 934 002346 005200  
 935 002350 001405  
 936 002352 006103  
 937 002354 103774  
 938 002356 022703 177777  
 939 002362 001404

REG3A: 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  
 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 <

940  
 941  
 942  
 943  
 944 002364  
 945 002364 012742 000030  
 946 002370 005242  
 947 002372 000000  
 948  
 949  
 950  
 951  
 952

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  
 \*\*\*\*\*

953 002374 005212  
 954 002376 022712 000025  
 955 002402 001012  
 956 002404 012704 000001  
 957 002410 012700 177757  
 958 002414 000241  
 959 002416 005200  
 960 002420 001403  
 961 002422 006104  
 962 002424 103374  
 963 002426 001404  
 964  
 965  
 966  
 967  
 968 002430  
 969 002430 012742 000031  
 970 002434 005242  
 971 002436 000000  
 972  
 973  
 974  
 975  
 976

TS25: INC (R2) ;UPDATE TEST NUMBER  
 CMP #25,(R2) ;SEQUENCE ERROR?  
 BNE TS26-10 ;BR TO ERROR HALT ON SEQ FRROR  
 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 <

964  
 965  
 966  
 967  
 968 002430  
 969 002430 012742 000031  
 970 002434 005242  
 971 002436 000000  
 972  
 973  
 974  
 975  
 976

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  
 \*\*\*\*\*

977 002440 005212  
 978 002442 022712 000026  
 979 002446 001014  
 980 002450 012704 177776  
 981 002454 012700 177757  
 982 002460 000261  
 983 002462 005200  
 984 002464 001405

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  
 MOV #-21,R0 ;SET BIT COUNTER  
 SEC ;SET C-BIT  
 REG4A: INC R0 ;INCREMENT BIT COUNTER  
 BEQ R4ERR ;BR TO ERROR HALT IF COUNTER 0

```
985 002466 006104 ROL R4 ;ROTATE 1 POSITION
986 002470 103774 BCS REG4A ;CONTINUE UNTIL C-BIT IS CLEAR
987 002472 022704 177777 CMP #-1,R4 ;CHECK DATA
988 002476 001404 BEQ TS27
989 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
990 ; CONDITIONAL BRANCH INST. AND <-
991 ; REPLACE THE MOVE INSTRUCTION <-
992 ; WHICH FOLLOWS W/ 763 <
993 002500 R4ERR:
994 002500 012742 000032 MOV #32,-(R2) ;MOVE TO MAILBOX # ***** 32 *****
995 002504 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
996 002506 000000 HALT ;FAILURE WITH R4
997 ; OR SEQUENCE ERROR
998
999
```

```
1000 ;*****
1001 ;TEST 27 TEST IF R5 CAN HOLD A ONE IN ALL BITS
1002 ;*****
1003 TS27: INC (R2) ;UPDATE TEST NUMBER
1004 002512 022712 000027 CMP #27,(R2) ;SEQUENCE ERROR?
1005 002516 001012 BNE TS30-10 ;BR TO ERROR HALT ON SEG FRROR
1006 002520 012705 000001 MOV #1,R5 ;SET BIT 0
1007 002524 012700 177757 MOV #-21,R0 ;SET BIT COUNTER
1008 002530 000241 CLC ;CLEAR C-BIT
1009 002532 005200 REG5: INC R0 ;INCREMENT BIT COUNTER
1010 002534 001403 BEQ REG5E ;BR TO ERROR HALT IF BIT IS LOST
1011 002536 006105 ROL R5 ;ROTATE 1 POSITION
1012 002540 103374 BCC REG5 ;ALL DONE
1013 002542 001404 BEQ TS30
```

```
1014 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
1015 ; CONDITIONAL BRANCH INST. AND <
1016 ; REPLACE THE MOVE INSTRUCTION <
1017 ; WHICH FOLLOWS W/ 765 <
1018 REG5E:
1019 002544 012742 000033 MOV #33,-(R2) ;MOVE TO MAILBOX # ***** 33 *****
1020 002550 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1021 002552 000000 HALT ;FAILURE WITH R5
1022 ; OR SEQUENCE ERROR
1023
1024
```

```
1025 ;*****
1026 ;TEST 30 TEST IF R5 CAN HOLD A ZERO IN ALL BITS
1027 ;*****
1027 002554 005212 TS30: INC (R2) ;UPDATE TEST NUMBER
1028 002556 022712 000030 CMP #30,(R2) ;SEQUENCE ERROR?
1029 002562 001014 BNE TS31-10 ;BR TO ERROR HALT ON SEQ ERROR
1030 002564 012705 177776 MOV #-2,R5 ;SET ALL ONES IN R5 EXCEPT FOR BIT 0
1031 002570 012700 177757 MOV #-21,R0 ;SET BIT COUNTER
1032 002574 000261 SEC ;SET C-BIT
1033 002576 005200 REG5A: INC R0 ;INCREMENT BIT COUNTER
1034 002600 001405 BEQ R5ERR ;BR TO ERROR HALT IF COUNTER 0
1035 002602 006105 ROL R5 ;ROTATE 1 POSITION
1036 002604 103774 BCS REG5A ;CONTINUE UNTIL C-BIT IS CLEAR
1037 002606 022705 177777 CMP #-1,R5 ;CHECK DATA
1038 002612 001404 BEQ TS31
```

```
1039 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
1040 ; CONDITIONAL BRANCH INST. AND <
```



```
1041 : REPLACE THE MOVE INSTRUCTION <===  
1042 : WHICH FOLLOWS W/ 763 <===  
1043 002614 R5ERR: :  
1044 002614 012742 000034 MOV #34, -(R2) ;MOVE TO MAILBOX # ***** 34 *****  
1045 002620 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1046 002627 000000 HALT ;FAILURE WITH R5  
1047 : OR SEQUENCE ERROR  
1048 :  
1049 :  
1050 :*****  
1051 :TEST 31 TEST IF R6 CAN HOLD A ONE IN ALL BITS  
1052 :*****  
1053 002624 005212 TS31: INC (R2) ;UPDATE TEST NUMBER  
1054 002626 022712 000031 CMP #31, (R2) ;SEQUENCE ERROR?  
1055 002632 001015 BNE TS32-10 ;BR TO ERROR HALT ON SEQ ERROR  
1056 002634 012767 000340 175134 MOV #340, PS ;LOCK OUT INTERRUPTS WHILE PLAYING WITH R6  
1057 002642 012706 000001 MOV #1, R6 ;SET BIT 0  
1058 002646 012700 177757 MOV #-21, R0 ;SET BIT COUNTER  
1059 002652 000241 CLC ;CLEAR C-BIT  
1060 002654 005200 REG6: INC R0 ;INCREMENT BIT COUNTER  
1061 002656 001403 BEQ REG6E ;BR TO ERROR HALT IF BIT IS LOST  
1062 002660 006106 ROL R6 ;ROTATE 1 POSITION  
1063 002662 103374 BCC REG6 ;ALL DONE  
1064 002664 001404 BEQ TS32  
1065 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <  
1066 : CONDITIONAL BRANCH INST. AND <-  
1067 : REPLACE THE MOVE INSTRUCTION <  
1068 : WHICH FOLLOWS W/ 762 <-  
1069 002666 REG6E: MOV #35, -(R2) ;MOVE TO MAILBOX # ***** 35 *****  
1070 002672 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1071 002674 000000 HALT ;FAILURE WITH R6  
1072 : OR SEQUENCE ERROR  
1073 :  
1074 :  
1075 :*****  
1076 :TEST 32 TEST IF R6 CAN HOLD A ZERO IN ALL BITS  
1077 :*****  
1078 002676 005212 TS32: INC (R2) ;UPDATE TEST NUMBER  
1079 002700 022712 000032 CMP #32, (R2) ;SEQUENCE ERROR?  
1080 002704 001014 BNE TS33-10 ;BR TO ERROR HALT ON SEQ ERROR  
1081 002706 012706 177776 MOV #-2, R6 ;SET ALL ONES IN R6 EXCEPT FOR BIT 0  
1082 002712 012700 177757 MOV #-21, R0 ;SET BIT COUNTER  
1083 002716 000261 SEC ;SET C-BIT  
1084 002720 005200 REG6A: INC R0 ;INCREMENT BIT COUNT  
1085 002722 001405 BEQ R6ERR ;BR TO ERROR HALT IF COUNTER 0  
1086 002724 006106 ROL R6 ;ROTATE 1 POSITION  
1087 002726 103774 BCS REG6A ;CONTINUE UNTIL C-BIT IS CLEAR  
1088 002730 022706 177777 CMP #-1, R6 ;CHECK DATA  
1089 002734 001404 BEQ TS33  
1090 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-  
1091 : CONDITIONAL BRANCH INST. AND <-  
1092 : REPLACE THE MOVE INSTRUCTION <  
1093 : WHICH FOLLOWS W/ 763 <-  
1094 002736 R6ERR: MOV #36, -(R2) ;MOVE TO MAILBOX # ***** 36 *****  
1095 002742 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1096 002744 000000 HALT ;FAILURE WITH R6
```

: OR SEQUENCE ERROR

1097  
1098  
1099  
1100  
1101  
1102  
1103  
1104  
1105  
1106  
1107  
1108  
1109  
1110  
1111  
1112  
1113  
1114  
1115  
1116  
1117  
1118  
1119  
1120  
1121  
1122  
1123  
1124  
1125  
1126  
1127  
1128  
1129  
1130  
1131  
1132  
1133  
1134  
1135  
1136  
1137  
1138  
1139  
1140  
1141  
1142  
1143  
1144  
1145  
1146  
1147  
1148  
1149  
1150  
1151  
1152

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

: 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.

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

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

:\*\*\*\*\*  
:TEST 35 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES  
:\*\*\*\*\*

177776  
001000  
000000  
177776  
000037  
000034  
000252  
000252  
000040

```
1153 003044 005212 TS35: INC (R2) ;UPDATE TEST NUMBER
1154 003046 022712 000035 CMP #35,(R2) ;SEQUENCE ERROR?
1155 003052 001007 BNE TS36-10 ;BR TO ERROR HALT ON SEQ ERROR
1156 003054 012737 000105 177776 MOV #105,@#PS ;MOVE ALT. ONES AND ZEROES TO PSW
1157 003062 023727 177776 000105 CMP @#PS,#105 ;SUCCESSFUL?
1158 003070 001404 BEQ TS36 ;
1159 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=
1160 ; CONDITIONAL BRANCH INST. AND <=
1161 ; REPLACE THE MOVE INSTRUCTION <
1162 ; WHICH FOLLOWS W/ 770 <=
1163 003072 012742 000041 MOV #41,-(R2) ;MOVE TO MAILBOX # ***** 41 *****
1164 003076 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1165 003100 000000 HALT ;PSW NOT 105
1166 ; OR SEQUENCE ERROR
```

```
*****
;TEST 36 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES
*****
```

```
1171 003102 005212 TS36: INC (R2) ;UPDATE TEST NUMBER
1172 003104 022712 000036 CMP #36,(R2) ;SEQUENCE ERROR?
1173 003110 001007 BNE TS37-10 ;BR TO ERROR HALT ON SEQ ERROR
1174 003112 012737 000357 177776 MOV #357,@#PS ;MOVE ONES TO PSW
1175 003120 023727 177776 000357 CMP @#PS,#357 ;SUCCESSFUL
1176 003126 001404 BEQ TS37 ;
1177 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=
1178 ; CONDITIONAL BRANCH INST. AND <=
1179 ; REPLACE THE MOVE INSTRUCTION <
1180 ; WHICH FOLLOWS W/ 770 <
1181 003130 012742 000042 MOV #42,-(R2) ;MOVE TO MAILBOX # ***** 42 *****
1182 003134 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1183 003136 000000 HALT ;PSW NOT 357
1184 ; 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
*****
```

```
1203 003140 005212 TS37: INC (R2) ;UPDATE TEST NUMBER
1204 003142 022712 000037 CMP #37,(R2) ;SEQUENCE ERROR?
1205 003146 001014 BNE TS40-10 ;BR TO ERROR HALT ON SEQ ERROR
1206 ;FIRST WITH Z-BIT ON
1207 003150 000257 CCC ;CC=0100: JUST Z-BIT
1208 003152 000264 SEZ
```

```

1209 003154 001001          BNE    BRZ1          ;CHECK OPPOSITE CONDITION
1210 003156 001404          BEQ    BRZ2          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < ---
1211                                     ;          CONDITIONAL BRANCH INST. AND <=---
1212                                     ;          REPLACE THE MOVE INSTRUCTION < ---
1213                                     ;          WHICH FOLLOWS W/ 773 < -
1214
1215 003160          BRZ1:
1216 003160 012742 000043    MOV    #43,-(R2)      ;MOVE TO MAILBOX # ***** 43 *****
1217 003164 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
1218 003166 000000          HALT                ;IMPROPER BR W/ Z-1
1219                                     ;CHECK WITH Z-BIT OFF
1220 003170 000277          BRZ2:
1221 003172 000244          SCC                    ;CC=1011: ALL BUT Z-BIT
1222 003174 001401          CLZ                    ;
1223 003176 001004          BEQ    BRZ3          ;
1224                                     ;          TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=---
1225                                     ;          CONDITIONAL BRANCH INST. AND <-
1226                                     ;          REPLACE THE MOVE INSTRUCTION <=---
1227                                     ;          WHICH FOLLOWS W/ 763 <----
1228 003200          BRZ3:
1229 003200 012742 000044    MOV    #44,-(R2)      ;MOVE TO MAILBOX # ***** 44 *****
1230 003204 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
1231 003206 000000          HALT                ;IMPROPER BR W/ Z=0
1232                                     ; OR SEQUENCE ERROR
1233
1234 :*****
1235 :
1236 :          THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE N-BIT.
1237 :THE N-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1238 :BMI AND BPL ARE TESTED FOR PROPER EXECUTION. THEN THE N-BIT IS
1239 :SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1240 :AGAIN FOR PROPER OPERATION.
1241 :          THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
1242 :CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
1243 :BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
1244 :LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY
1245 :USED IN THE TEST ARE VERIFIED HERE.
1246 :
1247 :*****
1248 :TEST 40          TEST BRANCHES AROUND N-BIT
1249 :*****
1250 003210 005212          TS40:  INC    (R2)          ;UPDATE TEST NUMBER
1251 003212 022712 000040    CMP    #40,(R2)      ;SEQUENCE ERROR?
1252 003216 001014          BNE    TS41-10       ;BR TO ERROR HALT ON SEQ ERROR
1253                                     ;FIRST WITH N-BIT ON
1254 003220 000257          CCC                    ;CC=1000: JUST N-BIT
1255 003222 000270          SEN                    ;
1256 003224 100001          BPL    BRN1          ;CHECK OPPOSITE CONDITION
1257 003226 100404          BMI    BRN2          ;
1258                                     ;          TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
1259                                     ;          CONDITIONAL BRANCH INST. AND <-
1260                                     ;          REPLACE THE MOVE INSTRUCTION <-
1261                                     ;          WHICH FOLLOWS W/ 773 <- -
1262 003230          BRN1:
1263 003230 012742 000045    MOV    #45,-(R2)      ;MOVE TO MAILBOX # ***** 45 *****
1264 003234 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR

```

```
1265 003236 000000          HALT          ;IMPROPER BR W/ N-1
1266          ;CHECK WITH N-BIT OFF
1267 003240 000277 BRN2:  SCC          ;CC=0111
1268 003242 000250          CLN
1269 003244 100401          BMI          BRN3
1270 003246 100004          BPL          TS41          ;CHECK OPPOSITE CONDITION
1271          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
1272          ;                               CONDITIONAL BRANCH INST. AND <===
1273          ;                               REPLACE THE MOVE INSTRUCTION <===
1274          ;                               WHICH FOLLOWS W/ 763 <===
1275 003250
1276 003250 012742 000046 BRN3:  MOV          #46,-(R2) ;MOVE TO MAILBOX # ***** 46 *****
1277 003254 005242          INC          -(R2)      ;SET MSGTYP TO FATAL ERROR
1278 003256 000000          HALT          ;IMPROPER BR W/ N=0
1279          ; OR SEQUENCE ERROR
1280
1281 ;*****
1282 ;
1283 ; THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE V-BIT.
1284 ; THE V-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1285 ; BVS AND BVC ARE TESTED FOR PROPER EXECUTION. THEN THE V-BIT IS
1286 ; SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1287 ; AGAIN FOR PROPER OPERATION.
1288 ;
1289 ;*****
1290 ;TEST 41 TEST BRANCHES AROUND V-BIT
1291 ;*****
1292 003260 005212          TS41:  INC          (R2)      ;UPDATE TEST NUMBER
1293 003262 022712 000041          CMP          #41,(R2)    ;SEQUENCE ERROR?
1294 003266 001014          BNE          TS42-10    ;BR TO ERROR HALT ON SEQ ERROR
1295          ;FIRST WITH V-BIT ON
1296 003270 000257          CCC          ;CC=0010: JUST V-BIT
1297 003272 000262          SEV
1298 003274 102001          BVC          BRV1        ;CHECK OPPOSITE CONDITION
1299 003276 102404          BVS          BRV2
1300          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
1301          ;                               CONDITIONAL BRANCH INST. AND <==
1302          ;                               REPLACE THE MOVE INSTRUCTION <==
1303          ;                               WHICH FOLLOWS W/ 773 <==
1304 003300
1305 003300 012742 000047 BRV1:  MOV          #47,-(R2) ;MOVE TO MAILBOX # ***** 47 *****
1306 003304 005242          INC          -(R2)      ;SET MSGTYP TO FATAL ERROR
1307 003306 000000          HALT          ;IMPROPER BR W/ V=1
1308          ;CHECK WITH V-BIT OFF
1309 003310 000277 BRV2:  SCC          ;CC=1101: ALL BVT V-BIT
1310 003312 000242          CLV
1311 003314 102401          BVS          BRV3        ;CHECK OPPOSITE CONDITION
1312 003316 102004          BVC          TS42
1313          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
1314          ;                               CONDITIONAL BRANCH INST. AND <===
1315          ;                               REPLACE THE MOVE INSTRUCTION <===
1316          ;                               WHICH FOLLOWS W/ 763 <==
1317 003320
1318 003320 012742 000050 BRV3:  MOV          #50,-(R2) ;MOVE TO MAILBOX # ***** 50 *****
1319 003324 005242          INC          -(R2)      ;SET MSGTYP TO FATAL ERROR
1320 003326 000000          HALT          ;IMPROPER BR W/ V 0
```

1321  
1322  
1323  
1324  
1325  
1326  
1327  
1328  
1329  
1330  
1331  
1332  
1333  
1334 003330 005212  
1335 003332 022712 000042  
1336 003336 001014  
1337  
1338 003340 000257  
1339 003342 000261  
1340 003344 103001  
1341 003346 103404  
1342  
1343  
1344  
1345  
1346 003350  
1347 003350 012742 000051  
1348 003354 005242  
1349 003356 000000  
1350  
1351 003360 000277  
1352 003362 000241  
1353 003364 103401  
1354 003366 100404  
1355  
1356  
1357  
1358  
1359 003370  
1360 003370 012742 000052  
1361 003374 005242  
1362 003376 000000  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373  
1374  
1375  
1376

```

; OR SEQUENCE ERROR
;*****
;
;   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.
;*****
;TEST 42      TEST BRANCHES AROUND C-BIT
;*****
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
      CCC                      ;CC=0001: JUST C-BIT
      SEC
      BCC      BRC1         ;CHECK OPPOSITE CONDITION
      BCS      BRC2
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
      ;          CONDITIONAL BRANCH INST. AND <=
      ;          REPLACE THE MOVE INSTRUCTION <-
      ;          WHICH FOLLOWS W/ 773 <=
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
BRC2:  SCC                      ;CC=1110
      CLC
      BCS      BRC3         ;CHECK OPPOSITE CONDITION
      BMI      TS43
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
      ;          CONDITIONAL BRANCH INST. AND <
      ;          REPLACE THE MOVE INSTRUCTION <-
      ;          WHICH FOLLOWS W/ 763 <-
BRC3:  MOV      #52,-(R2)    ;MOVE TO MAILBOX # ***** 52 *****
      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
      HALT                    ;IMPROPER BR W/ C=0
      ; OR SEQUENCE ERROR
;*****
;SBTTL MICROCODE TESTS
;
;   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
;A SMALL LOOP CONVENIENT FOR SCOPING. THE TEST IS SET UP USING
;ONLY INSTRUCTIONS AND ADDRESSING MODES WHICH HAVE BEEN PREVIOUSLY
;VERIFIED.
```

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  
1423  
1424  
1425  
1426  
1427  
1428  
1429  
1430  
1431  
1432

```

: 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 SOP0A
:
: 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
      HALT ;CUMMULATIVE RESULT OF CLR,INC,NEG AND COM INSTS. FAILED
: OR SEQUENCE ERROR

```

003400 005212  
003402 022712 000043

003406 001020  
003410 005000  
003412 001404

003414 012742 000053

003420 005242  
003422 000000  
003424 005200  
003426 005100  
003430 005200  
003432 100404

003434 012742 000054

003440 005242  
003442 000000  
003444 005100  
003446 001404

003450 012742 000055

003454 005242  
003456 000000

1433  
1434  
1435  
1436  
1437  
1438  
1439  
1440  
1441  
1442  
1443  
1444  
1445  
1446 003460 005212  
1447 003462 022712 000044  
1448 003466 001021  
1449 003470 005000  
1450 003472 005300  
1451 003474 100404  
1452  
1453  
1454  
1455  
1456 003476 012742 000056  
1457 003502 005242  
1458 003504 000000  
1459 003506 000261  
1460 003510 005500  
1461 003512 001007  
1462 003514 000261  
1463 003516 005600  
1464 003520 100004  
1465 003522 005100  
1466 003524 005200  
1467 003526 005300  
1468 003530 001404  
1469  
1470  
1471  
1472  
1473 003532  
1474 003532 012742 000057  
1475 003536 005242  
1476 003540 000000  
1477  
1478  
1479  
1480  
1481  
1482  
1483  
1484  
1485  
1486  
1487  
1488 003542 005212

```
.....  
: 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  
HALT ;N-BIT NOT SET ON DEC  
SOPOC: SEC ;INITIALIZE CARRY  
ADC R0 ;TRY ADD CARRY INST  
BNE SOPOD  
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 ;CUMULATIVE 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.  
:.....  
: TEST 45 TEST MODE 0 EVEN BYTE USING SOP INST  
:.....  
TS45: INC (R2) ;UPDATE TEST NUMBER
```



```
1489 003544 022712 000045      CMP      #45,(R2)      ;SEQUENCE ERROR?
1490 003550 001012              BNE      TS46-10      ;BR TO ERROR HALT ON SEQ ERROR
1491 003552 105000              CLR      R0           ;TRY CLEARING EVEN BYTE OF REGISTER
1492 003554 001404              BEQ      SOPB0A
1493                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1494                          ;          CONDITIONAL BRANCH INST. AND <====
1495                          ;          REPLACE THE MOVE INSTRUCTION <====
1496                          ;          WHICH FOLLOWS W/ 775 <====
1497 003556 012742 000060      MOV      #60,-(R2)    ;MOVE TO MAILBOX # ***** 60 *****
1498 003562 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
1499 003564 000000              HALT                    ;CLR DID NOT SET Z-BIT
1500 003566 105100      SOPB0A: COMB      R0   ;TRY SETTING EVEN BYTE OF REGISTER
1501 003570 100002              BPL      SOPB0B
1502 003572 105200              INCB     R0           ;TRY INCREMENTING EVEN BYTE OF REGISTER>>
1503 003574 001404              BEQ      TS46
1504                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1505                          ;          CONDITIONAL BRANCH INST. AND <====
1506                          ;          REPLACE THE MOVE INSTRUCTION <====
1507                          ;          WHICH FOLLOWS W/ 765 <====
1508 003576 012742 000061      SOPB0B: MOV      #61,-(R2) ;MOVE TO MAILBOX # ***** 61 *****
1509 003576 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
1510 003602 000000              HALT                    ;TEST CUMULATIVE RESULT OF ABOVE BYTE INST.
1511 003604 000000              ; OR SEQUENCE ERROR
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525 003606 005212 000046      TS46:   INC      (R2)      ;UPDATE TEST NUMBER
1526 003610 022712              CMP      #46,(R2)    ;SEQUENCE ERROR?
1527 003614 001014              BNE      TS47-10      ;BR TO ERROR HALT ON SEQ ERROR
1528 003616 005000              CLR      R0           ;INITIALIZE R0
1529 003620 005010              CLR      (R0)        ;TRY CLEAR INST W/MODE 1
1530 003622 001404              BEQ      SOP1A
1531                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
1532                          ;          CONDITIONAL BRANCH INST. AND <
1533                          ;          REPLACE THE MOVE INSTRUCTION <
1534                          ;          WHICH FOLLOWS W/ 774 <
1535 003624 012742 000062      MOV      #62,-(R2)    ;MOVE TO MAILBOX # ***** 62 *****
1536 003630 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
1537 003632 000000              HALT                    ;CLR DID NOT SET Z-BIT
1538 003634 005310      SOP1A: DEC      (R0)   ;TRY DECREMENT INST W/MODE 1
1539 003636 100003              BPL      SOP1B
1540 003640 000261              SEC                    ;INITIALIZE CARRY
1541 003642 005510              ADC      (R0)        ;TRY ADD-CARRY W/MODE 1
1542 003644 001404              BEQ      TS47
1543                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
1544                          ;          CONDITIONAL BRANCH INST. AND <
```

```
1545 : REPLACE THE MOVE INSTRUCTION <-
1546 : WHICH FOLLOWS W/ 763 <=
1547 003646 SOP1B: MOV #63,-(R2) ;MOVE TO MAILBOX # ***** 63 *****
1548 003646 012742 000063 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1549 003652 005242 ;TEST CUMMULATIVE RESULT OF ABOVE INST
1550 003654 000000 HALT ; OR SEQUENCE ERROR
1551
1552
1553 :*****
1554 :
1555 : THIS TEST VERIFIES THE BYTE INSTRUCTION MICROCODE FOR MODE 1
1556 : SINGLE OPERAND INSTRUCTIONS.
1557 : THIS IS THE FIRST PLACE THE SIGN EXTEND LOGIC IS EXERCISED
1558 : AND VERIFIED.
1559 :
1560 :*****
1561 : TEST 47 TEST MODE 1 EVEN BYTE USING SOP INST
1562 :*****
1563 003656 005212 TS47: INC (R2) ;UPDATE TEST NUMBER
1564 003660 022712 000047 CMP #47,(R2) ;SEQUENCE ERROR?
1565 003664 001020 BNE TS50-10 ;BR TO ERROR HALT ON SEQ ERROR
1566 003666 005000 CLR R0 ;INITIALIZE R0
1567 003670 005010 CLR (R0) ;INITIALIZE LCL. 0
1568 003672 005110 COM (R0)
1569 003674 105010 CLRB (R0) ;TRY TO CLEAR BYTE 0
1570 003676 001404 BEQ SOPB1A
1571 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
1572 : CONDITIONAL BRANCH INST. AND <-
1573 : REPLACE THE MOVE INSTRUCTION <
1574 : WHICH FOLLOWS W/ 772 <=
1575 003700 012742 000064 MOV #64,-(R2) ;MOVE TO MAILBOX # ***** 64 *****
1576 003704 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1577 003706 000000 HALT ;CLRB DID NOT SET Z-BIT
1578 003710 005210 SOPB1A: INC (R0) ;INCREMENT TO TEST WORD
1579 003712 100005 BPL SOPB1B
1580 003714 105110 COMB (R0) ;COMPLEMENT: ODD BYTE 376
1581 003716 105210 INCB (R0) ;INC: ODD BYTE = 377
1582 003720 100002 BPL SOPB1B
1583 003722 105210 INCB (R0) ;INCREMENT ODD BYTE=0
1584 003724 001404 BEQ TS50
1585 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
1586 : CONDITIONAL BRANCH INST. AND <-
1587 : REPLACE THE MOVE INSTRUCTION <
1588 : WHICH FOLLOWS W/ 757 <
1589 003726 SOPB1B: MOV #65,-(R2) ;MOVE TO MAILBOX # ***** 65 *****
1590 003726 012742 000065 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1591 003732 005242 HALT ;CHECK CUMMULATIVE RESULT OF ABOVE INST
1592 003734 000000 ; OR SEQUENCE ERROR
1593
1594
1595
1596 :*****
1597 :
1598 : THIS TEST VERIFIES THAT SINGLE OPERAND BYTE INSTRUCTIONS WILL
1599 : FUNCTION CORRECTLY FOR ODD BYTES.
1600 : THIS IS THE FIRST TIME THAT ADDRESS LINE 0 HAS BEEN
```

1601  
1602  
1603  
1604  
1605  
1606  
1607  
1608 003736 005212  
1609 003740 022712 000050  
1610 003744 001022  
1611 003746 005000  
1612 003750 005010  
1613 003752 005110  
1614 003754 005200  
1615 003756 105010  
1616 003760 001404  
1617  
1618  
1619  
1620  
1621 003762 012742 000066  
1622 003766 005242  
1623 003770 000000  
1624 003772 005300  
1625 003774 005210  
1626 003776 005200  
1627 004000 105110  
1628 004002 105210  
1629 004004 100002  
1630 004006 105210  
1631 004010 001404  
1632  
1633  
1634  
1635  
1636 004012  
1637 004012 012742 000067  
1638 004016 005242  
1639 004020 000000  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655 004022 005212  
1656 004024 022712 000051

;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 TS51-0 ;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 ;RO=WORD ADDR.  
INC (R0) ;INCREMENT TO TEST WORD  
INC R0 ;RO=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 CUMMULATIVE 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.  
;\*\*\*\*\*  
TS51: INC (R2) ;UPDATE TEST NUMBER  
CMP #51,(R2) ;SEQUENCE ERROR?

```

1657 004030 001023      BNE      T552-10      ;BR TO ERROR HALT ON SEQ ERROR
1658 004032 005000      CLR      R0           ;SET R0=400
1659 004034 105100      COMB     R0
1660 004036 005200      INC      R0
1661 004040 005010      CLR      (R0)        ;CLEAR 400
1662 004042 005110      COM      (R0)        ;INITIALIZE: 400=-1
1663 004044 005020      CLR      (R0)+       ;TRY CLEARING WITH MODE 2
1664 004046 001404      BEQ      SOPZA
1665                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== =
1666                                     ;          CONDITIONAL BRANCH INST. AND <==--
1667                                     ;          REPLACE THE MOVE INSTRUCTION <====
1668                                     ;          WHICH FOLLOWS W/ 770 <====
1669 004050 012742 000070  MOV      #70,-(R2)    ;MOVE TO MAILBOX # ***** 70 *****
1670 004054 005242      INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
1671 004056 000000      HALT
1672 004060 005300      SOPZA: DEC      R0   ;CLR INST DID NOT SET Z-BIT
1673 004062 005300      DEC      R0         ;RESET R0
1674 004064 005120      COM      (R0)+      ;TRY COMPLEMENTING WITH MODE 2
1675 004066 100004      BPL      SOP2B
1676 004070 005300      DEC      R0         ;RESET R0
1677 004072 005300      DEC      R0
1678 004074 005220      INC      (R0)+     ;TRY INCREMENTING WITH MODE 2
1679 004076 001404      BEQ      T552
1680                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
1681                                     ;          CONDITIONAL BRANCH INST. AND <-
1682                                     ;          REPLACE THE MOVE INSTRUCTION <
1683                                     ;          WHICH FOLLOWS W/ 754 <
1684 004100      SOP2B:
1685 004100 012742 000071  MOV      #71,-(R2)    ;MOVE TO MAILBOX # ***** 71 *****
1686 004104 005242      INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
1687 004106 000000      HALT                ;CHECK CUMULATIVE RESULT OF ABOVE INST
1688                                     ; OR SEQUENCE ERROR
1689
1690 .....
1691 .....
1692 .....
1693 .....
1694 .....
1695 .....
1696 .....
1697 .....
1698 .....
1699 .....
1700 .....
1701 .....
1702 .....
1703 004110 005212      TS52: INC      (R2)    ;UPDATE TEST NUMBER
1704 004112 022712 000052  CMP      #52,(R2)    ;SEQUENCE ERROR?
1705 004116 001023      BNE      T553-10     ;BR TO ERROR HALT ON SEQ ERROR
1706 004120 005000      CLR      R0         ;SET R0=400
1707 004122 105100      COMB     R0
1708 004124 005200      INC      R0
1709 004126 005010      CLR      (R0)        ;CLEAR 400
1710 004130 005110      COM      (R0)        ;INITIALIZE: 400=-1
1711 004132 105020      CLRB    (R0)+       ;TRY TO CLEAT 400 W/MODE 2
1712 004134 001404      BEQ      SOPB2A

```

```

.....
THIS TEST VERIFIES MODE 2 SINGLE OPERAND INSTRUCTIONS WHICH
ADDRESS EVEN BYTES. R0 IS SET TO 400 AND USED TO INITIALIZE LOCATION
400 TO -1. CLRB INSTRUCTION IS THEN EXECUTED ON BYTE 400 WITH
MODE 2.
R0 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.
.....

```

```
1713 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < = - =  
1714 ; CONDITIONAL BRANCH INST. AND < = - - =  
1715 ; REPLACE THE MOVE INSTRUCTION < = - =  
1716 ; WHICH FOLLOWS W/ 770 < = - -  
1717 004136 012742 000072      MOV #72,-(R2) ;MOVE TO MAILBOX # ***** 72 *****  
1718 004142 005242      INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1719 004144 000000      HALT ;CLR DID NOT SET Z-BIT  
SOPB2A: 1720 004146 005300      DEC R0 ;RESULT R0=400  
1721 004150 005210      INC (R0) ;INC 400 TO TEST WORD  
1722 004152 105110      COMB (R0)  
1723 004154 105220      INCB (R0)+ ;TRY TO INC EVEN BYTE  
1724 004156 100003      BPL SOPB2B  
1725 004160 005300      DEC R0 ;RESET R0=400  
1726 004162 105220      INCB (R0)+ ;TRY INCREMENT OF EVEN BYTE  
1727 004164 001404      BEQ TS53
```

```
1728 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <  
1729 ; CONDITIONAL BRANCH INST. AND <  
1730 ; REPLACE THE MOVE INSTRUCTION <  
1731 ; WHICH FOLLOWS W/ 754 <
```

```
SOPB2B: 1732 004166 012742 000073      MOV #73,-(R2) ;MOVE TO MAILBOX # ***** 73 *****  
1733 004166 012742 000073      INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1734 004172 005242      HALT ;TEST CUMMULATIVE RESULT OF ABOVE INST.  
1735 004174 000000 ; OR SEQUENCE ERROR
```

```
1736  
1737  
1738 :*****  
1739 :  
1740 : THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS  
1741 : TEST. HERE, THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.  
1742 :  
1743 :*****
```

```
1744 :TEST 53 TEST MODE 2 ODD BYTE USING SOP INST.  
1745 :*****
```

```
TS53: 1746 004176 005212      INC (R2) ;UPDATE TEST NUMBER  
1747 004200 022712 000053      CMP #53,(R2) ;SEQUENCE ERROR?  
1748 004204 001026      BNE TS54-10 ;BR TO ERROR HALT ON SEQ ERROR  
1749 004206 005000      CLR R0 ;SET R0=400  
1750 004210 105100      COMB R0  
1751 004212 005200      INC R0  
1752 004214 005010      CLR (R0) ;CLEAR LOC 400  
1753 004216 005110      COM (R0) ;INITIALIZE: 400 -1  
1754 004220 005200      INC R0 ;R0=ODD BYTE  
1755 004222 105020      CLRB (R0)+ ;TRY TO CLEAR ODD BYTE  
1756 004224 001404      BEQ SOPB2C
```

```
1757 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -  
1758 ; CONDITIONAL BRANCH INST. AND < = - - =  
1759 ; REPLACE THE MOVE INSTRUCTION < -  
1760 ; WHICH FOLLOWS W/ 767 < - -
```

```
SOPB2C: 1761 004226 012742 000074      MOV #74,-(R2) ;MOVE TO MAILBOX # ***** 74 *****  
1762 004232 005242      INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1763 004234 000000      HALT ;CLRB DID NOT SET Z-BIT  
1764 004236 005300      DEC R0 ;R0=WORD ADDR.  
1765 004240 005300      DEC R0  
1766 004242 005220      INC (R0)+ ;INCREMENT WORD  
1767 004244 005300      DEC R0 ;POINT TO ODD BYTE  
1768 004246 105110      COMB (R0) ;COMPLEMENT ODD BYTE
```

```
1769 004250 105220 INCB (R0)+ ;TRY TO INCREMENT ODD BYTE
1770 004252 100003 BPL SOPB2D
1771 004254 005300 DEC RO ;RESET RO TO ODD BYTE
1772 004256 105220 INCB (R0)+ ;TRY TO INCREMENT ODD BYTE
1773 004260 001404 BEQ T554
1774 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
1775 ; CONDITIONAL BRANCH INST. AND <====
1776 ; REPLACE THE MOVE INSTRUCTION <====
1777 ; WHICH FOLLOWS W/ 751 <--
1778 004262 SOPB2D:
1779 004262 012742 000075 MOV #75,-(R2) ;MOVE TO MAILBOX # ***** 75 *****
1780 004266 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1781 004270 000000 HALT ;TEST CUMMULATIVE RESULT OF ABOVE INST.
1782 ; OR SEQUENCE ERROR
1783
1784 :*****
1785 :
1786 : THESE TESTS CHECK THE NEGATE INSTRUCTION IN ALL MODES. PREVIOUSLY
1787 : TESTED SINGLE-OPERAND INSTRUCTIONS ARE USED TO TEST THE NEGATE INSTRUCTION.
1788 :
1789 :*****
1790 :TEST 54 TEST MODE 0 USING NEGATE INSTRUCTION
1791 :*****
1792 004272 005212 T554: INC (R2) ;UPDATE TEST NUMBER
1793 004274 022712 000054 CMP #54,(R2) ;SEQUENCE ERROR?
1794 004300 001035 BNE TS55-10 ;BR TO ERROR HALT ON SEQ ERROR
1795 004302 005000 CLR RO ;SET RO=0
1796 004304 005200 INC RO ; RO=1
1797 004306 005400 NEG RO ;TRY NEGATE MODE 0: RO--1
1798 004310 100003 BPL NEG00 ;CC=1001?
1799 004312 001402 BEQ NEG00
1800 004314 102401 BVS NEG00
1801 004316 103404 BCS NEG01
1802 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
1803 ; CONDITIONAL BRANCH INST. AND <-
1804 ; REPLACE THE MOVE INSTRUCTION <
1805 ; WHICH FOLLOWS W/ 770 <
1806 004320 NEG00:
1807 004320 012742 000076 MOV #76,-(R2) ;MOVE TO MAILBOX # ***** 76 *****
1808 004324 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1809 004326 000000 HALT ;NEGATE DID NOT SET CC'S CORRECTLY
1810
1811 004330 NEG01: INC RO ;TEST DATA RESULT
1812 004332 001404 BEQ NEG02
1813 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
1814 ; CONDITIONAL BRANCH INST. AND <-
1815 ; REPLACE THE MOVE INSTRUCTION <
1816 ; WHICH FOLLOWS W/ 762 <
1817 004334 012742 000077 MOV #77,-(R2) ;MOVE TO MAILBOX # ***** 77 *****
1818 004340 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1819 004342 000000 HALT ;DATA RESULT OF NEGATE INCORRECT
1820
1821 004344 105100 NEG02: COMB RO ;RO=377
1822 004346 105400 NEGB RO ;RO=1
1823 004350 100403 BMI NEG03 ;CC=0001?
1824 004352 001402 BEQ NEG03
```

```
1825 004354 102401      BVS    NEG03
1826 004356 103404      BCS    NEG04
1827
1828
1829
1830
1831 004360
1832 004360 012742 000100      NEG03: MOV    #100,-(R2)      ;MOVE TO MAILBOX # ***** 100 *****
1833 004364 005242      INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
1834 004366 000000      HALT
1835 004370 005300      NEG04: DEC    R0          ;NEGB DID NOT SET CC'S CORRECTLY
1836 004372 001404      BEQ    TS55             ;TEST DATA RESULT
1837
1838
1839
1840
1841 004374 012742 000101      MOV    #101,-(R2)      ;MOVE TO MAILBOX # ***** 101 *****
1842 004400 005242      INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
1843 004402 000000      HALT
1844
1845
1846
1847
1848 004404 005212
1849 004406 022712 000055      TS55:  INC    (R2)         ;UPDATE TEST NUMBER
1850 004412 001040      CMP    #55,(R2)        ;SEQUENCE ERROR?
1851 004414 005000      BNE    TS56-10         ;BR TO ERROR HALT ON SEQ ERROR
1852 004416 005010      CLR    R0              ;POINT TO LOC. 0
1853 004420 005210      CLR    (R0)            ;CLEAR LOC. 0
1854 004422 005410      INC    (R0)            ;LOC. 0=1
1855 004424 100003      NEG    (R0)            ;TRY NEG. LOC. 0=-1
1856 004426 001402      BPL    NEG10           ;CC=1001
1857 004430 102401      BEQ    NEG10
1858 004432 103404      BVS    NEG10
1859
1860
1861
1862
1863 004434
1864 004434 012742 000102      NEG10: MOV    #102,-(R2)      ;MOVE TO MAILBOX # ***** 102 *****
1865 004440 005242      INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
1866 004442 000000      HALT
1867
1868 004444 005237 000000      NEG11: INC    #0         ;TEST DATA RESULT
1869 004450 001404      BEQ    NEG12
1870
1871
1872
1873
1874 004452 012742 000103      MOV    #103,-(R2)      ;MOVE TO MAILBOX # ***** 103 *****
1875 004456 005242      INC    -(R2)             ;SET MSGTYP TO FATAL ERROR
1876 004460 000000      HALT
1877 004462 105110      NEG12: COMB   (R0)        ;DATA RESULT OF NEGATE INCORRECT
1878 004464 105410      NEGB  (R0)            ;LOC. 0=377
1879 004466 100403      BMI    NEG13           ;TRY NEGB LOC. 0=1
1880 004470 001402      BEQ    NEG13           ;CC=0001?
```

```

1881 004472 102401      BVS      NEG13
1882 004474 103404      BCS      NEG14
1883                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
1884                                     ;          CONDITIONAL BRANCH INST. AND <===
1885                                     ;          REPLACE THE MOVE INSTRUCTION <===
1886                                     ;          WHICH FOLLOWS W/ 746 <===
1887 004476
1888 004476 012742 000104  NEG13:  MOV      #104,-(R2)  ;MOVE TO MAILBOX # ***** 104 *****
1889 004502 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
1890 004504 000000      HALT
1891 004506 005337 000000  NEG14:  DEC      @#0      ;NEGB DID NOT SET CC'S CORRECTLY
1892 004512 001404      BEQ      TS56          ;TEST DATA RESULT
1893
1894                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
1895                                     ;          CONDITIONAL BRANCH INST. AND <===
1896                                     ;          REPLACE THE MOVE INSTRUCTION <===
1897 004514 012742 000105  NEG13:  MOV      #105,-(R2)  ;MOVE TO MAILBOX # ***** 105 *****
1898 004520 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
1899 004522 000000      HALT          ;DATA RESULT OF NEGB INCORRECT
1900                                     ; OR SEQUENCE ERROR
1901
1902 *****
1903 :TEST 56          TEST MODE 2 USING NEGATE INSTRUCTION
1904 *****
1904 004524 005212      TS56:  INC      (R2)          ;UPDATE TEST NUMBER
1905 004526 022712 000056  CMP      #56,(R2)      ;SEQUENCE ERROR?
1906 004532 001032      BNE      TS57-10      ;BR TO ERROR HALT ON SEQ ERROR
1907 004534 005000      CLR      R0           ;POINT TO LOC. 0
1908 004536 005010      CLR      (R0)         ;CLEAR LOC. 0
1909 004540 005210      INC      (R0)         ;LOC. 0-1
1910 004542 005420      NEG      (R0)+        ;TRY NEG.: LOC. 0=-1
1911 004544 100003      BPL      NEG20        ;CC=1001?
1912 004546 001402      BEQ      NEG20
1913 004550 102401      BVS      NEG20
1914 004552 103404      BCS      NEG21
1915
1916                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
1917                                     ;          CONDITIONAL BRANCH INST. AND <==
1918                                     ;          REPLACE THE MOVE INSTRUCTION <==
1919                                     ;          WHICH FOLLOWS W/ 767 <-
1919 004554
1920 004554 012742 000106  NEG20:  MOV      #106,-(R2)  ;MOVE TO MAILBOX # ***** 106 *****
1921 004560 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
1922 004562 000000      HALT          ;NEGATE DID NOT SET CC'S CORRECTLY
1923 004564 105300      NEG21:  DECB     R0           ;R0=LOC. 0
1924 004566 105300      DECB     R0
1925 004570 105420      NEGB     (R0)+        ;BYTE 0-1  R0=1
1926 004572 105420      NEGB     (R0)+        ;BYTE 1=1  R0=2
1927 004574 105340      DECB     -(R0)        ;R0=1 LOC. 0=01
1928 004576 005300      DEC      R0           ;R0=0
1929 004600 001404      BEQ      NEG22
1930
1931                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--
1932                                     ;          CONDITIONAL BRANCH INST. AND <==
1933                                     ;          REPLACE THE MOVE INSTRUCTION <--
1934                                     ;          WHICH FOLLOWS W/ 754 <
1934 004602 012742 000107  NEG20:  MOV      #107,-(R2)  ;MOVE TO MAILBOX # ***** 107 *****
1935 004606 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
1936 004610 000000      HALT          ;REGISTER NOT INCREMENTED CORRECTLY

```



```
1937 004612 005337 000000      NEG22: DEC      @#0      ;LOC. 0=0
1938 004616 001404              BEQ      TS57
1939                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1940                               ;          CONDITIONAL BRANCH INST. AND <====
1941                               ;          REPLACE THE MOVE INSTRUCTION <----=
1942                               ;          WHICH FOLLOWS W/ 745 <===
1943 004620 012742 000110      MOV      #110,-(R2) ;MOVE TO MAILBOX # ***** 110 *****
1944 004624 005242              INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
1945 004626 000000              HALT                    ;NEG BYTE INSTRUCTIONS FAILED
1946                               ; OR SEQUENCE ERROR
```

```
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
```

```
*****
:
: 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.
: RO IS SET TO 400, THE START OF THE ADDRESS TABLE, AND A CLR
: INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR LOC. 0. THEN RO
: 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).
*****
```

```
1964
1965 004630 005212 000057      TS57: INC      (R2)      ;UPDATE TEST NUMBER
1966 004632 022712              CMP      #57,(R2)    ;SEQUENCE ERROR?
1967 004636 001020              BNE     TS60-10     ;BR TO ERROR HALT ON SEQ ERROR
1968 004640 005000              CLR     RO          ;SET RO=400
1969 004642 105100              COMB   RO
1970 004644 005200              INC     RO
1971 004646 005010              CLR     (RO)        ;CLEAR LOC 400
1972 004650 005030              CLR     @(RO)+      ;TRY TO CLEAR LOC 0 USING MODE 3 ;RO 402
1973 004652 001404              BEQ     SOP3A
```

```
1974
1975
1976
1977
1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
```

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

```
1978 004654 012742 000111      MOV      #111,-(R2) ;MOVE TO MAILBOX # ***** 111 *****
1979 004660 005242              INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
1980 004662 000000              HALT                    ;CLR DID NOT SET Z-BIT
1981 004664 005300      SOP3A: DEC     RO          ;RESET RO=400
1982 004666 005300              DEC     RO
1983 004670 005130              COM     @(RO)+      ;TRY TO COMPLEMENT LOC 0 OF MODE 3 ;RO=402
1984 004672 100002              BPL     SOP3B
1985 004674 005230              INC     @(RO)+      ;TRY TO INCREMENT LOC 0 W/MODE 3 ;RO=404
1986 004676 001404              BEQ     TS60
```

```
1987
1988
1989
1990
1991
1992
```

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

```
1991 004700      SOP3B: MOV      #112,-(R2) ;MOVE TO MAILBOX # ***** 112 *****
1992 004700 012742 000112
```

1993 004704 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1994 004706 000000 HALT ;CUMMULATIVE RFSULT OF ABOVE INST FAILED  
1995 ; OR SEQUENCE ERROR  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011

\*\*\*\*\*  
: 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).  
\*\*\*\*\*

2012 004710 005212 000060  
2013 004712 022712  
2014 004716 001026  
2015 004720 005004  
2016 004722 105104  
2017 004724 005204  
2018 004726 005000  
2019 004730 005010  
2020 004732 005110  
2021 004734 105034  
2022 004736 001404

\*\*\*\*\*  
: 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

2023  
2024  
2025  
2026

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

2027 004740 012742 000113  
2028 004744 005242  
2029 004746 000000  
2030 004750 005304  
2031 004752 005304  
2032 004754 005234  
2033 004756 100006  
2034 004760 105434  
2035 004762 100004  
2036 004764 005304  
2037 004766 005304  
2038 004770 105234  
2039 004772 001404

MOV #113,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 113 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CLRB DID NOT SET Z-BIT  
SOPB3A: 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

2040  
2041  
2042  
2043

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

2044 004774  
2045 004774 012742 000114  
2046 005000 005242  
2047 005002 000000  
2048

SOPB3B: MOV #114,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 114 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CUMMULATIVE RESULT OF ABOVE INST FAILED  
; OR SEQUENCE ERROR

2049  
2050  
2051  
2052  
2053  
2054  
2055  
2056  
2057  
2058  
2059  
2060  
2061  
2062  
2063  
2064  
2065  
2066  
2067  
2068  
2069  
2070  
2071  
2072  
2073  
2074  
2075  
2076  
2077  
2078  
2079  
2080  
2081  
2082  
2083  
2084  
2085  
2086  
2087  
2088  
2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104

005004 005212  
005006 022712 000061  
005012 001024  
005014 005000  
005016 105100  
005020 005200  
005022 005030  
005024 005130  
005026 105030  
005030 001404  
  
005032 012742 000115  
005036 005242  
005040 000000  
005042 005300  
005044 005300  
005046 005300  
005050 005300  
005052 005230  
005054 105430  
005056 100002  
005060 105230  
005062 001404  
  
005064 012742 000116  
005070 005242  
005072 000000

```
*****  
: 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. RO SERVES AS THE TABLE POINTER.  
: RO 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. RO 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 RO ;SET RO=400  
COMB RO  
INC RO  
CLR @ (RO)+ ;INITIALIZE  
COM @ (RO)+ ;LOC 0=-1 RO=404  
CLRB @ (RO)+ ;TRY TO CLEAR ODD BYTE LOC. 0=377 RO=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  
HALT ;CLRB DID NOT SET Z-BIT  
SOPB3C: DEC RO ;RESET RO=402  
DEC RO  
DEC RO ;POINT TO EVEN BYTE ADDR.  
DEC RO  
INC @ (RO)+ ;INCREMENT WORD LOC. 0=400 RO=404  
NEGB @ (RO)+ ;TRY TO NEGATE ODD BYTE LOC. 0-177400 RO=406  
BPL SOPB3D  
INCB @ (RO)+ ;TRY TO INCREMENT ODD BYTE LOC.0=0 RO=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 ;CUMMULATIVE RESULT OF ABOVE INSTS FAILED  
: OR SEQUENCE ERROR  
*****  
: TEST 62 TEST MODE 3 USING NEGATE INSTRUCTION  
*****  
TS62: INC (R2) ;UPDATE TEST NUMBER
```

```

2105 005076 022712 000062      CMP      #62,(R2)      ;SEQUENCE ERROR?
2106 005102 001054      BNE     TS63-10      ;BR TO ERROR HALT ON SEQ ERROR
2107 005104 005000      CLR     R0           ;R0=400
2108 005106 105100      COMB    R0
2109 005110 005200      INC     R0
2110 005112 005010      CLR     (R0)         ;LOC. 400=0
2111 005114 005004      CLR     R4           ;R4=0
2112 005116 005014      CLR     (R4)         ;LOC. 0=0
2113 005120 005214      INC     (R4)         ;LOC. 0=1
2114 005122 005430      NEG     @ (R0)+      ;TRY NEGATE LOC. 0=-1 R0-402
2115 005124 100003      BPL     NEG30        ;CC=1001?
2116 005126 001402      BEQ     NEG30
2117 005130 102401      BVS     NEG30
2118 005132 103404      BCS     NEG31
2119
2120
2121
2122
2123 005134
2124 005134 012742 0C0117      NEG30:  MOV     #117,-(R2) ;MOVE TO MAILBOX # ***** 117 *****
2125 005140 005242      INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
2126 005142 000000      HALT
2127 005144 005214      NEG31:  INC     (R4)        ;NEG DID NOT SET CC'S CORRECTLY
2128 005146 001404      BEQ     NEG32        ;LOC. 0=0
2129
2130
2131
2132
2133 005150 012742 000120      MOV     #120,-(R2) ;MOVE TO MAILBOX # ***** 120 *****
2134 005154 005242      INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
2135 005156 000000      HALT
2136 005160 105137 000001      NEG32:  COMB    @#1      ;DATA RESULT OF NEG INCORRECT
2137 005164 005237 000000      INC     @#0          ;LOC 0=177400
2138 005170 105430      NEGB   @ (R0)+      ;LOC. 0=177401
2139 005172 100404      BMI     NEG33        ;TRY NEGB LOC. 0=177777 R0-404
2140
2141
2142
2143
2144 005174 012742 000121      MOV     #121,-(R2) ;MOVE TO MAILBOX # ***** 121 *****
2145 005200 005242      INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
2146 005202 000000      HALT
2147 005204 105430      NEG33:  NEGB   @ (R0)+      ;NEGB FAILED WITH EVEN BYTE
2148 005206 100004      BPL     NEG34        ;TRY NEGB LOC.0=777 R0=406
2149
2150
2151
2152
2153 005210 012742 000122      MOV     #122,-(R2) ;MOVE TO MAILBOX # ***** 122 *****
2154 005214 005242      INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
2155 005216 000000      HALT
2156 005220 105137 000001      NEG34:  COMB    @#1      ;NEGB FAILED WITH ODD BYTE
2157 005224 105237 000001      INC     @#1          ;LOC. 0=177377
2158 005230 005214      INC     (R4)         ;LOC. 0=177777
2159 005232 001404      BEQ     TS63         ;LOC. 0=0
2160

```

KDB-D DCF11-AA CPU DIAG.  
KDBD.P11 24-NOV-80 11:07

MACY11 30A(1052) 14-JAN-81 11:46 PAGE 41  
T62 TEST MODE 3 USING NEGATE INSTRUCTION

SEQ 0041

2161  
2162  
2163  
2164 005234 012742 000123  
2165 005240 005242  
2166 005242 000000  
2167  
2168  
2169  
2170  
2171  
2172

MOV #123,-(R2)  
INC -(R2)  
HALT

: CONDITIONAL BRANCH INST. AND <===  
: REPLACE THE MOVE INSTRUCTION <===  
: WHICH FOLLOWS W/ 723 <===  
: MOVE TO MAILBOX # \*\*\*\*\* 123 \*\*\*\*\*  
: SET MSGTYP TO FATAL ERROR  
: DATA RESULT OF NEGB'S INCORRECT  
: OR SEQUENCE ERROR

.....  
: THIS TEST VERIFIES MODE 4 SINGLE OPERAND INSTRUCTIONS.  
: RO IS SET TO 400. A CLR INSTRUCTION IS EXECUTED IN MODE 4 TO CLEAR

```

2173 ;LOC. 376. R0 IS RESET TO 400 AND A COM INSTRUCTION USING MODE 4
2174 ;COMPLEMENTS LOC.376.
2175 ; TWO INC INSTRUCTIONS AND A MODE 4 INSTRUCTION ARE EXECUTED
2176 ; TO COMPLETE THE TEST.
2177 *****
2178 ;TEST 63 TEST MODE 4 USING SOP INSTS
2179 *****
2180 005244 005212 TS63: INC (R2) ;UPDATE TEST NUMBER
2181 005246 022712 000063 CMP #63,(R2) ;SEQUENCE ERROR?
2182 005252 001021 BNE TS64-10 ;BR TO ERROR HALT ON SEQ ERROR
2183 005254 005000 CLR R0 ;SET R0-400
2184 005256 105100 COMB R0
2185 005260 005200 INC R0
2186 005262 005040 CLR -(R0) ;TRY TO CLEAR USING MODE 4
2187 005264 001404 BEQ SOP4A
2188 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
2189 ; CONDITIONAL BRANCH INST. AND <-
2190 ; REPLACE THE MOVE INSTRUCTION <-
2191 ; WHICH FOLLOWS W/ 772 <-
2192 005266 012742 000124 MOV #124,-(R2) ;MOVE TO MAILBOX # ***** 124 *****
2193 005272 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2194 005274 000000 HALT ;CLR DID NOT SET Z-BIT
2195 005276 005200 SOP4A: INC R0 ;RESET R0
2196 005300 005200 INC R0
2197 005302 005140 COM -(R0) ;TRY TO COMPLEMENT USING MODE 4
2198 005304 100004 BPL SOP4B
2199 005306 005200 INC R0 ;MOVE POINTER
2200 005310 005200 INC R0
2201 005312 005240 INC -(R0)
2202 005314 001404 BEQ TS64
2203 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < = -
2204 ; CONDITIONAL BRANCH INST. AND <-
2205 ; REPLACE THE MOVE INSTRUCTION <-
2206 ; WHICH FOLLOWS W/ 756 <-
2207 005316 SOP4B: MOV #125,-(R2) ;MOVE TO MAILBOX # ***** 125 *****
2208 005316 012742 000125 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2209 005322 005242 HALT ;CHECK CUMMULATIVE RESULT OF ABOVE INST.
2210 005324 000000 ; OR SEQUENCE ERROR
2211
2212 *****
2213 ;
2214 ; THIS TEST VERIFIES MODE 5 SINGLE OPERAND INSTRUCTIONS. IT
2215 ; USES LOCATION 0 AS ITS TARGET DATA. A TABLE LOCATED AT LOC. 372
2216 ; THRU 374 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
2217 ; INSTRUCTIONS UNDER TEST.
2218 ; R0 IS SET TO 376, (THE START OF THE ADDRESS TABLE) +2,
2219 ; AND A CLR INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR
2220 ; LOC. 0. THEN R0 IS INCREMENTED BY TWO AND TWO OTHER MODE 3
2221 ; INSTRUCTIONS OPERATE ON LOC. 0 TO VERIFY THE DATA RESULTS OF
2222 ; THE TEST. THE PROPER DECREMENTING OF THE REGISTER IS ALSO
2223 ; VERIFIED IN THIS MANNER.
2224 ; IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
2225 ; (LOC. 372 THRU 374) HAS THE PROPER VALUES (0).
2226
2227 *****
2228

```

2229  
2230  
2231 005326 005212  
2232 005330 022712 000064  
2233 005334 001025  
2234 005336 012700 000370  
2235 005342 005020  
2236 005344 005020  
2237 005346 005020  
2238 005350 005010  
2239 005352 005000  
2240 005354 005020  
2241 005356 105400  
2242 005360 005050  
2243 005362 001404  
2244  
2245  
2246  
2247  
2248 005364 012742 000126  
2249 005370 005242  
2250 005372 000000  
2251 005374 005200  
2252 005376 005200  
2253 005400 005150  
2254 005402 100002  
2255 005404 005250  
2256 005406 001404  
2257  
2258  
2259  
2260  
2261 005410  
2262 005410 012742 000127  
2263 005414 005242  
2264 005416 000000  
2265  
2266  
2267  
2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278 005420 005212  
2279 005422 022712 000065  
2280 005426 001020  
2281 005430 005000  
2282 005432 105100  
2283 005434 005200  
2284 005436 005060 177400

```

;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
      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)
      HALT
SOP5A:  INC     R0          ;SET MSGTYP TO FATAL ERROR
      INC     R0          ;CLR DID NOT SET Z-BIT
      COM     @-(R0)       ;RESET R0
      BPL     SOP5B
      INC     @-(R0)       ;TRY TO COMPLEMENT 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)
      HALT
      ;SET MSGTYP TO FATAL ERROR
      ;TEST CUMULATIVE RESULT OF ABOVE INSTS
      ; OR SEQUENCE ERROR

```

```

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

```

```
2285 005442 001404          BEQ      SOP6A          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2286                                     ;          CONDITIONAL BRANCH INST. AND <====
2287                                     ;          REPLACE THE MOVE INSTRUCTION <====
2288                                     ;          WHICH FOLLOWS W/ 771 <====
2289                                     ;
2290 005444 012742 000130      MOV      #130,-(R2)      ;MOVE TO MAILBOX # ***** 130 *****
2291 005450 005242             INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2292 005452 000000             HALT                    ;CLR DID NOT SET Z-BIT
2293 005454 005160 177400      SOP6A:  COM      -400(R0) ;TRY TO COMPLEMENT LOCATION 0 W/MODE 6
2294 005460 100003             BPL      SOP6B          ;
2295 005462 005260 177400      INC      -400(R0)      ;TRY TO INCREMENT LOCATION 0 W/MODE 6
2296 005466 001404          BEQ      TS66          ;
2297                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2298                                     ;          CONDITIONAL BRANCH INST. AND <====
2299                                     ;          REPLACE THE MOVE INSTRUCTION <====
2300                                     ;          WHICH FOLLOWS W/ 757 <====
2301 005470             SOP6B:
2302 005470 012742 000131      MOV      #131,-(R2)      ;MOVE TO MAILBOX # ***** 131 *****
2303 005474 005242             INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2304 005476 000000             HALT                    ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
2305                                     ; OR SEQUENCE ERROR
2306
2307 :*****
2308 :
2309 : THIS TEST VERIFIES MODE 7 SINGLE OPERAND INSTRUCTIONS. IT USES
2310 : THE POINTER TO LOC. 0 WHICH IS STORED AT LOC. 402.
2311 : RO IS SET TO 400 AND A MODE 7 CLR INSTRUCTION IS
2312 : EXECUTED WITH A +2 OFFSET TO CLEAR LOC. 0.
2313 : SEVERAL OTHER MODE 7 INSTRUCTIONS ARE THEN USED ON THE COMMON
2314 : LOCATION TO VERIFY THE DATA RESULTS.
2315 :
2316 :*****
2317 :TEST 66          TEST MODE 7 USING SOP INST.
2318 :*****
2319 005500 005212             TS66:  INC      (R2)          ;UPDATE TEST NUMBER
2320 005502 022712 000066      CMP      #66,(R2)        ;SEQUENCE ERROR?
2321 005506 001021             BNE      TS67-10        ;BR TO ERROR HALT ON SEQ ERROR
2322 005510 005000             CLR      R0            ;SET R0=400
2323 005512 105100             COMB     R0
2324 005514 005200             INC      R0
2325 005516 005210             INC      (R0)          ;R0=1
2326 005520 005070 000002      CLR      @2(R0)        ;TRY TO CLEAR LOC. 0 W/MODE 7
2327 005524 001404          BEQ      SOP7A          ;
2328                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
2329                                     ;          CONDITIONAL BRANCH INST. AND < -
2330                                     ;          REPLACE THE MOVE INSTRUCTION < -
2331                                     ;          WHICH FOLLOWS W/ 770 < -
2332 005526 012742 000132      MOV      #132,-(R2)      ;MOVE TO MAILBOX # ***** 132 *****
2333 005532 005242             INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2334 005534 000000             HALT                    ;CLR DID NOT SET Z-BIT
2335 005536 005170 000002      SOP7A:  COM      @2(R0)   ;TRY TO COMPLEMENT LOC. 0 W/MODE 7
2336 005542 100003             BPL      SOP7B          ;
2337 005544 005270 000002      INC      @2(R0)        ;TRY TO INCREMENT LOC. 0 W/MODE 7
2338 005550 001404          BEQ      TS67          ;
2339                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
2340                                     ;          CONDITIONAL BRANCH INST. AND <
```





```

2397 005664 005010 CLR (R0) :LOC. 0=0
2398 005666 105100 COMB R0 :R0=377
2399 005670 005200 INC R0 :R0=400
2400 005672 005010 CLR (R0) :SET 400 = 0
2401 005674 005004 CLR R4 :R4=0
2402 005676 005314 DEC (R4) :LOC. 0=177777
2403 005700 005450 NEG @-(R0) :TRY NEGATE: LOC. 0=1
2404 005702 100403 BMI NEG50 :CC=0001?
2405 005704 001402 BEQ NEG50
2406 005706 102401 BVS NEG50
2407 005710 103404 BCS NEG51
2408 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2409 : CONDITIONAL BRANCH INST. AND <====
2410 : REPLACE THE MOVE INSTRUCTION <====
2411 : WHICH FOLLOWS W/ 763 <====
2412 005712 NEG50:
2413 005712 012742 000137 MOV #137, -(R2) :MOVE TO MAILBOX # ***** 137 *****
2414 005716 005242 INC -(R2) :SET MSGTYP TO FATAL ERROR
2415 005720 000000 HALT :NEG DID NOT SET CC'S CORRECTLY
2416 005722 005314 NEG51: DEC (R4)
2417 005724 001404 BEQ NEG52
2418 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
2419 : CONDITIONAL BRANCH INST. AND <- -
2420 : REPLACE THE MOVE INSTRUCTION <- -
2421 : WHICH FOLLOWS W/ 755 <- -
2422 005726 012742 000140 MOV #140, -(R2) :MOVE TO MAILBOX # ***** 140 *****
2423 005732 005242 INC -(R2) :SET MSGTYP TO FATAL ERROR
2424 005734 000000 HALT :DATA RESULT OF NEG INCORRECT
2425 005736 105100 NEG52: COMB R0
2426 005740 005300 DEC R0
2427 005742 001404 BEQ TS71
2428 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
2429 : CONDITIONAL BRANCH INST. AND <= -
2430 : REPLACE THE MOVE INSTRUCTION <
2431 : WHICH FOLLOWS W/ 746 <
2432 005744 012742 000141 MOV #141, -(R2) :MOVE TO MAILBOX # ***** 141 *****
2433 005750 005242 INC -(R2) :SET MSGTYP TO FATAL ERROR
2434 005752 000000 HALT :REGISTER NOT DECREMENTED PROPERLY
2435 : OR SEQUENCE ERROR
2436 : *****
2437 : TEST 71 TEST MODE 6 WITH NEGATE *****
2438 : *****
2439 005754 005212 TS71: INC (R2) :UPDATE TEST NUMBER
2440 005756 022712 000071 CMP #71, (R2) :SEQUENCE ERROR?
2441 005762 001022 BNE TS72-10 :BR TO ERROR HALT ON SEQ ERROR
2442 005764 005000 CLR R0 :R0=0
2443 005766 005004 CLR R4 :R4=0
2444 005770 105100 COMB R0 :R0=377
2445 005772 005014 CLR (R4) :LOC. 0=0
2446 005774 105024 CLRB (R4)+ :LOC. 0=177777, R4 '
2447 005776 105114 COMB (R4) :LOC. 0=177400
2448 006000 005460 177401 NEG -377(R0) :LOC. 0=400
2449 006004 100403 BMI NEG60 :CC=0001
2450 006006 001402 BEQ NEG60
2451 006010 102401 BVS NEG60
2452 006012 103404 BCS NEG61

```

```
2453 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2454 ; CONDITIONAL BRANCH INST. AND <====
2455 ; REPLACE THE MOVE INSTRUCTION <====
2456 ; WHICH FOLLOWS W/ 763 <====
2457 006014
2458 006014 012742 00G142 NEG60: MOV #142,-(R2) ;MOVE TO MAILBOX # ***** 142 *****
2459 006020 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2460 006022 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
2461 006024 105314 NEG61: DECB (R4)
2462 006026 001404 BEQ TS72
2463 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2464 ; CONDITIONAL BRANCH INST. AND <====
2465 ; REPLACE THE MOVE INSTRUCTION <====
2466 ; WHICH FOLLOWS W/ 755 <====
2467 006030 012742 000143 MOV #143,-(R2) ;MOVE TO MAILBOX # ***** 143 *****
2468 006034 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2469 006036 000000 HALT ;DATA RESULT OF NEG INCORRECT
2470 ; OR SEQUENCE ERROR
2471 ;*****
2472 ;TEST 72 TEST MODE 7 W/ NEGATE
2473 ;*****
2474 006040 005212 TS72: INC (R2) ;UPDATE TEST NUMBER
2475 006042 022712 000072 CMP #72,(R2) ;SEQUENCE ERROR?
2476 006046 001024 BNE TS73-10 ;BR TO ERROR HALT ON SEQ ERROR
2477 006050 005000 CLR R0 ;R0=0
2478 006052 005010 CLR (R0) ;LOC. 0=0
2479 006054 005110 COM (R0) ;LOC. 0=177777
2480 006056 105100 COMB R0 ;R0=377
2481 006060 105470 000005 NEGB @5(R0) ;R0+5=404, 404-1, LOC. 0-777
2482 006064 100403 BMI NEG70 ;CC=0001?
2483 006066 001402 BEQ NEG70
2484 006070 102401 BVS NEG70
2485 006072 103404 BCS NEG71
2486 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
2487 ; CONDITIONAL BRANCH INST. AND <
2488 ; REPLACE THE MOVE INSTRUCTION <
2489 ; WHICH FOLLOWS W/ 765 <
2490 006074
2491 006074 012742 000144 NEG70: MOV #144,-(R2) ;MOVE TO MAILBOX # ***** 144 *****
2492 006100 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2493 006102 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
2494 006104 105100 NEG71: COMB R0 ;R0=0
2495 006106 105120 COMB (R0)+ ;LOC. 0=400, R0-1
2496 006110 105310 DECB (R0) ;LOC. 0=0
2497 006112 005467 171662 NEG 0 ;USE NEG MODE 67 TO TST FOR ZERO
2498 006116 001404 BEQ TS73
2499 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
2500 ; CONDITIONAL BRANCH INST. AND <=-
2501 ; REPLACE THE MOVE INSTRUCTION <
2502 ; WHICH FOLLOWS W/ 753 < -
2503 006120 012742 000145 MOV #145,-(R2) ;MOVE TO MAILBOX # ***** 145 *****
2504 006124 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2505 006126 000000 HALT ;DATA RESULT OF NEG WAS INCORRECT
2506 ; OR SEQUENCE ERROR
2507
2508 ;*****
```

2509  
 2510  
 2511  
 2512  
 2513  
 2514  
 2515  
 2516  
 2517  
 2518  
 2519 006130 005212  
 2520 006132 022712 000073  
 2521 006136 001017  
 2522 006140 005027  
 2523 006142 177777  
 2524 006144 001404  
 2525  
 2526  
 2527  
 2528  
 2529 006146 012742 000146  
 2530 006152 005242  
 2531 006154 000000  
 2532 006156 005237 006142  
 2533 006162 005467 177754  
 2534 006166 100003  
 2535 006170 005277 000012  
 2536 006174 001405  
 2537  
 2538  
 2539  
 2540  
 2541 006176  
 2542 006176 012742 000147  
 2543 006202 005242  
 2544 006204 000000  
 2545  
 2546 006206 006142  
 2547  
 2548  
 2549  
 2550  
 2551  
 2552  
 2553  
 2554  
 2555  
 2556  
 2557  
 2558  
 2559 006210 005212  
 2560 006212 022712 000074  
 2561 006216 001010  
 2562 006220 005000  
 2563 006222 000277  
 2564 006224 000244

```

:
:   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
:
:*****
:
:   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

```

2565 006226 005700  
2566 006230 02403  
2567 006232 100402  
2568 006234 103401  
2569 006236 001404  
2570  
2571  
2572  
2573  
2574 006240  
2575 006240 012742 000150  
2576 006244 005242  
2577 006246 000000  
2578  
2579  
2580  
2581  
2582  
2583  
2584  
2585  
2586  
2587  
2588  
2589  
2590  
2591 006250 005212  
2592 006252 022712 000075  
2593 006256 001010  
2594 006260 005000  
2595 006262 105100  
2596 006264 000277  
2597 006266 000250  
2598 006270 105700  
2599 006272 102402  
2600 006274 101401  
2601 006276 100404  
2602  
2603  
2604  
2605  
2606 006300  
2607 006300 012742 000151  
2608 006304 005242  
2609 006306 000000  
2610  
2611  
2612  
2613  
2614  
2615  
2616  
2617  
2618  
2619  
2620

TST R0 ;TRY TST W/ MODE 0  
BVS SNMOA ;CHECK THAT CR=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.  
: RO 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 ODE 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 <

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

\*\*\*\*\*  
: 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.  
\*\*\*\*\*

2621  
2622  
2623 006310 005212  
2624 006312 022712 000076  
2625 006316 001011  
2626 006320 005000  
2627 006322 005010  
2628 006324 000277  
2629 006326 000244  
2630 006330 005710  
2631 006332 102403  
2632 006334 103402  
2633 006336 100401  
2634 006340 001404  
2635  
2636  
2637  
2638  
2639 006342  
2640 006342 012742 000152  
2641 006346 005242  
2642 006350 000000  
2643  
2644  
2645  
2646  
2647  
2648  
2649  
2650  
2651  
2652  
2653  
2654  
2655 006352 005212  
2656 006354 022712 000077  
2657 006360 001026  
2658 006362 005000  
2659 006364 005010  
2660 006366 105110  
2661 006370 000277  
2662 006372 000250  
2663 006374 105710  
2664 006376 102402  
2665 006400 101401  
2666 006402 100404  
2667  
2668  
2669  
2670  
2671 006404  
2672 006404 012742 000153  
2673 006410 005242  
2674 006412 000000  
2675 006414 005000  
2676 006416 005200

```
:TEST 76          TEST MODE 1 SOP NON-MODIFYING
:*****
TS76:  INC      (R2)          ;UPDATE TEST NUMBER
      CMP      #76,(R2)      ;SEQUENCE ERROR?
      BNE     TS77-10       ;BR TO ERROR HALT ON SEQ ERROR
      CLR     R0            ;POINT TO LOC 0
      CLR     (R0)         ;CLEAR LOC 0
      SCC     ;INITIALIZE
      CLZ     ;CC=1011
      TST     (R0)         ;TRY TST W/ MODE 1
      BVS     SNM1A        ;CHECK CC=0100
      BCS     SNM1A
      BMI     SNM1A
      BEQ     TS77

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

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

:*****
:          THIS TEST SETS LOCATION 0 TO 377 AND THEN USES R0 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.
:*****
:TEST 77          TEST MODE 1 BYTE INST. NON-MODIFYING
:*****
TS77:  INC      (R2)          ;UPDATE TEST NUMBER
      CMP      #77,(R2)      ;SEQUENCE ERROR?
      BNE     TS100-10      ;BR TO ERROR HALT ON SEQ ERROR
      CLR     R0            ;POINT TO LOC 0
      CLR     (R0)         ;CLEAR LOC 0
      COMB    (R0)         ;COMPLEMENT BYTE 0
      SCC     ;SET CC=0111
      CLN     ;
      TSTB    (R0)         ;TRY TST ON EVEN BYTE
      BVS     SNMB1A
      BLOS    SNMB1A
      BMI     SNMB1B

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

SNMB1A: MOV      #153,-(R2)    ;MOVE TO MAILBOX # ***** 153 *****
      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
      HALT     ;CC'S NOT CORRECT

SNMB1B: CLR     R0
      INC     R0
```

```
2677 006420 000277          SCC          ;SET CC=1011
2678 006422 000244          CLZ
2679 006424 105710          TSTB      (R0)      ;TRY TO TST AN ODD BYTE
2680 006426 102403          BVS      SNMB1C    ;CHECK CC=0100
2681 006430 103402          BCS      SNMB1C
2682 006432 100401          BMI      SNMB1C
2683 006434 001404          BEQ      TS100
2684          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
2685          ;          CONDITIONAL BRANCH INST. AND <===
2686          ;          REPLACE THE MOVE INSTRUCTION <===
2687          ;          WHICH FOLLOWS W/ 751 <===
2688 006436          SNMB1C:
2689 006436 012742 000154      MOV      #154,-(R2) ;MOVE TO MAILBOX # ***** 154 *****
2690 006442 005242          INC      -(R2)     ;SET MSGTYP TO FATAL ERROR
2691 006444 000000          HALT          ;CC'S NOT CORRECT
2692          ; OR SEQUENCE ERROR
2693
2694          :*****
2695          :
2696          :      THIS TEST VERIFIES THE SINGLE-OPERAND NON-MODIFYING INSTRUCTIONS
2697          :USING MODE 2. IT USES THE IDENTICAL PROCEDURE EMPLOYED IN THE
2698          :MODE 1 TESTS. ADDITIONALLY, THE REGISTER IS CHECKED TO ASSURE THAT
2699          :IT IS INCREMENTED PROPERLY.
2700          :
2701          :*****
2702          :TEST 100      TEST MODE 2 WITH SOP NON-MODIFYING
2703          :*****
2704 006446 005212          TS100: INC      (R2)      ;UPDATE TEST NUMBER
2705 006450 022712 000100      CMP      #100,(R2)  ;SEQUENCE ERROR?
2706 006454 001020          BNE      TS101-10 ;BR TO ERROR HALT ON SEQ ERROR
2707 006456 005000          CLR      R0        ;INITIALIZE R0-0
2708 006460 005010          CLR      (R0)      ;CLEAR LOC 0
2709 006462 000277          SCC          ;SET CC=1011
2710 006464 000244          CLZ
2711 006466 005720          TST      (R0)+     ;TRY TST W/ MODE 2
2712 006470 102403          BVS      SNM2A     ;CHECK CC=0100
2713 006472 103402          BCS      SNM2A
2714 006474 100401          BMI      SNM2A
2715 006476 001404          BEQ      SNM2B
2716          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
2717          ;          CONDITIONAL BRANCH INST. AND <
2718          ;          REPLACE THE MOVE INSTRUCTION <-
2719          ;          WHICH FOLLOWS W/ 766 <
2720 006500          SNM2A:
2721 006500 012742 000155      MOV      #155,-(R2) ;MOVE TO MAILBOX # ***** 155 *****
2722 006504 005242          INC      -(R2)     ;SET MSGTYP TO FATAL ERROR
2723 006506 000000          HALT          ;CC'S NOT CORRECT
2724 006510 005300          SNM2B: DEC      R0        ;RESET R0
2725 006512 005300          DEC      R0
2726 006514 001404          BEQ      TS101
2727          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
2728          ;          CONDITIONAL BRANCH INST. AND <-
2729          ;          REPLACE THE MOVE INSTRUCTION <
2730          ;          WHICH FOLLOWS W/ 757 <
2731 006516 012742 000156      MOV      #156,-(R2) ;MOVE TO MAILBOX # ***** 156 *****
2732 006522 005242          INC      -(R2)     ;SET MSGTYP TO FATAL ERROR
```

2733 006524 000000

HALT

;MODE 2 DID NOT INC REG CORRECTLY  
; OR SEQUENCE ERROR

2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 2 SINGLE OPERAND NON-MODIFYING BYTE  
: INSTRUCTIONS IT USES R0 TO POINT TO LOC. 0. WITH LOCATION 0  
: SET TO 377, THE EVEN AND ODD BYTE IS TESTED WITH TSTB INSTRUCTIONS  
: TO VERIFY THE CORRECT CC ARE SET. THE REGISTER IS CHFKED FOR  
: PROPER INCREMENTING.  
\*\*\*\*\*

2744  
2745  
2746

\*\*\*\*\*  
: TEST 101 TEST MODE 2 - BYTE W/ SOP NON-MODIFYING  
\*\*\*\*\*

2747 006526 005212  
2748 006530 022712 000101  
2749 006534 001042  
2750 006536 005000  
2751 006540 005010  
2752 006542 105110  
2753 006544 000277  
2754 006546 000250  
2755 006550 105720  
2756 006552 102402  
2757 006554 101401  
2758 006556 100404

TS101: INC (R2) ;UPDATE TEST NUMBER  
CMP #101,(R2) ;SEQUENCE ERROR?  
BNE TS102-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;CLEAR R0  
CLR (R0) ;CLEAR LOC 0  
COMB (R0) ;SET LUC 0=377  
SCC ;SET CC=0111  
CLN  
TSTB (R0)+ ;TRY TST OF EVEN BYTE  
BVS SNMB2A  
BLOS SNMB2A  
BMI SNMB2B

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

2763 006560  
2764 006560 012742 000157  
2765 006564 005242  
2766 006566 000000  
2767 006570 005300  
2768 006572 001404

SNMB2A: MOV #157,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 157 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S NOT SET CORRECTLY  
SNMB2B: DEC R0 ;DECREMENT R0  
BEQ SNMB2C

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

2772  
2773 006574 012742 000160  
2774 006600 005242  
2775 006602 000000  
2776 006604 005200  
2777 006606 000277  
2778 006610 000244  
2779 006612 105720  
2780 006614 102403  
2781 006616 103402  
2782 006620 100401  
2783 006622 001404

MOV #160,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 160 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;MODE 2 DID NOT INC REG CORRECTLY  
SNMB2C: INC R0 ;POINT TO ODD BYTE  
SCC ;SET CC=1011  
CLZ  
TSTB (R0)+ ;TRY TST OF ODD BYTE  
BVS SNMB2D  
BCS SNMB2D  
BMI SNMB2D  
BEQ SNMB2E

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

2788 006624

SNMB2D:



```

2789 006624 012742 000161      MOV    #161,-(R2)      ;MOVE TO MAILBOX # ***** 161 *****
2790 006630 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
2791 006632 000000              HALT                  ;CC'S NOT CORRECT
2792 006634 005300      SNMB2E: DEC    R0
2793 006636 005300              DEC    R0
2794 006640 001404              BEQ    TS102
2795                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <= -
2796                                ;          CONDITIONAL BRANCH INST. AND  <= ==
2797                                ;          REPLACE THE MOVE INSTRUCTION  <= ---
2798                                ;          WHICH FOLLOWS W/ 735          <= --
2799 006642 012742 000162      MOV    #162,-(R2)      ;MOVE TO MAILBOX # ***** 162 *****
2800 006646 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
2801 006650 000000              HALT                  ;R0 DID NOT INCREMENT PROPERLY
2802                                ; OR SEQUENCE ERROR
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813

```

```

:*****
: 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.
:*****

```

```

2814 006652 005212 000102      TST102: INC    (R2)          ;UPDATE TEST NUMBER
2815 006654 022712              CMP    #102,(R2)      ;SEQUENCE ERROR?
2816 006660 001022              BNE    TS103-10       ;BR TO ERROR HALT ON SEQ ERROR
2817 006662 005000              CLR    R0             ;R0=0
2818 006664 005010              CLR    (R0)           ;CLEAR LOC 0
2819 006666 105100              COMB   R0             ;R0=376
2820 006670 005300              DEC    R0
2821 006672 000277              SCC
2822 006674 000244              CLZ
2823 006676 005730              TST    @(R0)+         ;TRY TST W/ MODE 3
2824 006700 102403              BVS    SNM3A          ;CHECK CC=0100
2825 006702 103402              BCS    SNM3A
2826 006704 100401              BMI    SNM3A
2827 006706 001404              BEQ    SNM3B
2828                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <
2829                                ;          CONDITIONAL BRANCH INST. AND  <
2830                                ;          REPLACE THE MOVE INSTRUCTION  <
2831                                ;          WHICH FOLLOWS W/ 764          <
2832
2833 006710 012742 000163      SNM3A: MOV    #163,-(R2)      ;MOVE TO MAILBOX # ***** 163 *****
2834 006714 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
2835 006716 000000              HALT                  ;CC'S NOT CORRECT
2836 006720 005300      SNM3B: DEC    R0
2837 006722 105100              COMB   R0             ;R0=0
2838 006724 001404              BEQ    TS103
2839                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <
2840                                ;          CONDITIONAL BRANCH INST. AND  <
2841                                ;          REPLACE THE MOVE INSTRUCTION  <
2842                                ;          WHICH FOLLOWS W/ 755          <
2843 006726 012742 000164      MOV    #164,-(R2)      ;MOVE TO MAILBOX # ***** 164 *****
2844 006732 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR

```

2845 006734 000000

HALT

;MODE 3 DID NOT INC REG CORRECTLY  
; OR SEQUENCE ERROR

2846  
2847  
2848  
2849  
2850  
2851  
2852  
2853  
2854  
2855  
2856  
2857

\*\*\*\*\*  
: 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.  
\*\*\*\*\*

2858  
2859

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

2860 006736 005212  
2861 006740 022712 000103  
2862 006744 001036  
2863 006746 005000  
2864 006750 005010  
2865 006752 105110  
2866 006754 105100  
2867 006756 005200  
2868 006760 005720  
2869 006762 000277  
2870 006764 000250  
2871 006766 105730  
2872 006770 102402  
2873 006772 101401  
2874 006774 100404

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

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

2875  
2876  
2877  
2878

2879 006776  
2880 006776 012742 000165  
2881 007002 005242  
2882 007004 000000  
2883 007006 000277  
2884 007010 000244  
2885 007012 105730  
2886 007014 102403  
2887 007016 103402  
2888 007020 100401  
2889 007022 001404

SNMB3A: MOV #165,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 165 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S NOT CORRECT  
SNMB3B: SCC ;SET CC=1011  
CLZ  
TSTB @ (R0)+ ;TRY TST OF ODD BYTE  
BVS SNMB3C ;CHECK CC=0100  
BCS SNMB3C  
BMI SNMB3C  
BEQ SNMB3D

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

2890  
2891  
2892  
2893

2894 007024  
2895 007024 012742 000166  
2896 007030 005242  
2897 007032 000000  
2898 007034 005720  
2899 007036 005710  
2900 007040 100404

SNMB3C: MOV #166,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 166 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S NOT CORRECT  
SNMB3D: TST (R0)+ ;R0=410  
TST (R0)  
BMI TS104

2901  
2902  
2903  
2904  
2905 007042 C12742 000167  
2906 007046 005242  
2907 007050 000000  
2908  
2909  
2910

MOV #167,-(R2) ; MOVE TO MAILBOX # \*\*\*\*\* 167 \*\*\*\*\*  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; TSTB DID NOT INCREMENT RO CORRECTLY  
; OR SEQUENCE ERROR

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 4 SOP NON-MODIFYING INSTRUCTIONS.  
: LOC. 0 IS SET TO -1 AND THE CC'S ARE SET TO THE COMPLEMENT OF THE  
: EXPECTED RESULTS. RO AND SET TO 2 AND A TST MODE 4 IS EXECUTED.  
: THE CC'S ARE CHECKED WITH CONDITIONAL BRANCH INSTRUCTIONS AND THE REGISTER  
: IS CHECKED FOR PROPER DECREMENTING.  
\*\*\*\*\*

2911  
2912  
2913  
2914  
2915  
2916  
2917  
2918  
2919

TEST 104 TEST MODE 4 W/ SOP NON-MODIFYING INSTS

2920 C07052 005212  
2921 007054 022712 000104  
2922 007060 001017  
2923 007062 005000  
2924 007064 005010  
2925 007066 005120  
2926 007070 000277  
2927 007072 000244  
2928 007074 005740  
2929 007076 102402  
2930 007100 101401  
2931 007102 100404  
2932

TS104: INC (R2) ; UPDATE TEST NUMBER  
CMP #104,(R2) ; SEQUENCE ERROR?  
BNE TS105-10 ; BR TO ERROR HALT ON SEQ ERROR  
CLR RO ; RO=0  
CLR (RO) ; LOC 0=0  
COM (RO)+ ; LOC 0=-1  
SCC ; SET CC=1011  
CLZ  
TST -(RO) ; TRY TST W/ MODE 4  
BVS SNM4A ; CHECK CC=0100  
BLOS SNM4A  
BMI SNM4B

2933  
2934  
2935  
2936 007104  
2937 007104 012742 000170  
2938 007110 005242  
2939 007112 000000  
2940 007114 005700  
2941 007116 001404  
2942

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

SNM4A: MOV #170,-(R2) ; MOVE TO MAILBOX # \*\*\*\*\* 170 \*\*\*\*\*  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; CC'S NOT CORRECT

SNM4B: TST RO  
BEQ TS105

2943  
2944  
2945  
2946 007120 012742 000171  
2947 007124 005242  
2948 007126 000000  
2949  
2950

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

MOV #171,-(R2) ; MOVE TO MAILBOX # \*\*\*\*\* 171 \*\*\*\*\*  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; TST MODE 4 DID NOT DEC RO CORRECTLY  
; OR SEQUENCE ERROR

2951  
2952  
2953  
2954  
2955  
2956

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

2957  
2958  
2959  
2960  
2961 007130 005212  
2962 007132 022712 000105  
2963 007136 001022  
2964 007140 005000  
2965 007142 005010  
2966 007144 005110  
2967 007146 105100  
2968 007150 005200  
2969 007152 000277  
2970 007154 000250  
2971 007156 005750  
2972 007160 102402  
2973 007162 101401  
2974 007164 100404  
2975  
2976  
2977  
2978  
2979 007166  
2980 007166 012742 000172  
2981 007172 005242  
2982 007174 000000  
2983 007176 005200  
2984 007200 105100  
2985 007202 001404  
2986  
2987  
2988  
2989  
2990 007204 012742 000173  
2991 007210 005242  
2992 007212 000000  
2993  
2994  
2995  
2996  
2997  
2998  
2999  
3000  
3001  
3002  
3003  
3004  
3005 007214 005212  
3006 007216 022712 000106  
3007 007222 001021  
3008 007224 005000  
3009 007226 005010  
3010 007230 005110  
3011 007232 105100  
3012 007234 000277

```
*****
:
:TEST 105      TEST MODE 5 W/ SOP NON-MODIFYING INSTS
:*****
TS105:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #105,(R2)    ;SEQUENCE ERROR?
        BNE     TS106-10     ;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
        INC     RO          ;RO=400
        SCC     ;SET CC=0111
        CLN
        TST     @-(RO)      ;TRY TST W/ MODE 5
        BVS     SNM5A       ;CHECK CC=1000
        BLOS    SNM5A
        BMI     SNM5B
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
:           CONDITIONAL BRANCH INST. AND <-
:           REPLACE THE MOVE INSTRUCTION <-
:           WHICH FOLLOWS W/ 764 <
:
SNM5A:  MOV      #172,-(R2)   ;MOVE TO MAILBOX # ***** 172 *****
        INC     -(R2)
        HALT
SNM5B:  INC     RO          ;RO=377
        COMB    RO          ;RO=0
        BEQ     TS106
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
:           CONDITIONAL BRANCH INST. AND <
:           REPLACE THE MOVE INSTRUCTION <
:           WHICH FOLLOWS W/ 755 <
:
        MOV     #173,-(R2)   ;MOVE TO MAILBOX # ***** 173 *****
        INC     -(R2)
        HALT
        ;MODE 5 DID NOT DEC RO CORRECTLY
        ; OR SEQUENCE ERROR
:
:*****
:
: THIS TEST VERIFIES MODE 6 SOP NON-MODIFYING INSTRUCTIONS.
: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
```

3013	007236	000250		CLN					
3014	007240	005760	177401	TST	-377(R0)			; TRY TST W/ MODE 6	
3015	007244	102402		BVS	SNM6A			; CHECK CC=1000	
3016	007246	101401		BLOS	SNM6A				
3017	007250	100404		BMI	SNM6B				
3018								; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
3019								CONDITIONAL BRANCH INST. AND	<====
3020								REPLACE THE MOVE INSTRUCTION	<====
3021								WHICH FOLLOWS W/ 764	<====
3022	007252			SNM6A:					
3023	007252	012742	000174	MOV	#174,-(R2)			; MOVE TO MAILBOX # ***** 174 *****	
3024	007256	005242		INC	-(R2)			; SET MSGTYP TO FATAL ERROR	
3025	007260	000000		HALT				; CC'S INCORRECT	
3026	007262	105100		SNM6B:	COMB	R0		; R0-0	
3027	007264	001404		BEQ	TS107				
3028								; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
3029								CONDITIONAL BRANCH INST. AND	<====
3030								REPLACE THE MOVE INSTRUCTION	<====
3031								WHICH FOLLOWS W/ 756	<====
3032	007266	012742	000175	MOV	#175,-(R2)			; MOVE TO MAILBOX # ***** 175 *****	
3033	007272	005242		INC	-(R2)			; SET MSGTYP TO FATAL ERROR	
3034	007274	000000		HALT				; TST MODE 6 INCORRECTLY CHANGED R0	
3035								; OR SEQUENCE ERROR	

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  
3071  
3072  
3073  
3074  
3075  
3076  
3077

007276 005212  
007300 022712 000107  
007304 001021  
007306 005000  
007310 005010  
007312 005110  
007314 105100  
007316 000277  
007320 000250  
007322 005770 000001  
007326 102402  
007330 101401  
007332 100404  
  
007334 012742 000176  
007340 005242  
007342 000000  
007344 105100  
007346 001404  
  
007350 012742 000177  
007354 005242  
007356 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 (R0) ;LOC 0=0  
COM (R0) ;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 ;CC'S NOT CORRECT  
SNM7B: COMB RO ;RO=0  
BEQ TS110  
  
: 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
```

3078  
3079  
3080  
3081  
3082  
3083  
3084  
3085  
3086  
3087  
3088 007360 005212  
3089 007362 022712 000110  
3090 007366 001006  
3091 007370 005000  
3092 007372 005100  
3093 007374 005004  
3094 007376 060004  
3095 007400 005204  
3096 007402 001404  
3097  
3098  
3099  
3100  
3101 007404 012742 000200  
3102 007410 005242  
3103 007412 000000  
3104  
3105  
3106  
3107  
3108  
3109  
3110  
3111  
3112  
3113 007414 005212  
3114 007416 022712 000111  
3115 007422 001006  
3116 007424 005000  
3117 007426 005004  
3118 007430 005100  
3119 007432 010004  
3120 007434 005204  
3121 007436 001404  
3122  
3123  
3124  
3125  
3126 007440 012742 000201  
3127 007444 005242  
3128 007446 000000  
3129  
3130  
3131  
3132  
3133

.....  
: 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.  
:.....

```

3134
3135
3136
3137
3138 007450 005212
3139 007452 022712 000112
3140 007456 001016
3141 007460 005000
3142 007462 005004
3143 007464 005204
3144 007466 160400
3145 007470 100003
3146 007472 001402
3147 007474 102401
3148 007476 103404
3149
3150
3151
3152
3153 007500
3154 007500 012742 000202
3155 007504 005242
3156 007506 000000
3157 007510 005200
3158 007512 001404
3159
3160
3161
3162
3163 007514 012742 000203
3164 007520 005242
3165 007522 000000
3166

```

```

:
:.....
: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

```



3167  
3168  
3169  
3170  
3171  
3172  
3173  
3174  
3175  
3176  
3177  
3178  
3179 007524 005212  
3180 007526 022712 000113  
3181 007532 001051  
3182 007534 005000  
3183 007536 010004  
3184 007540 001404  
3185  
3186  
3187  
3188  
3189 007542 012742 000204  
3190 007546 005242  
3191 007550 000000  
3192 007552 005200  
3193 007554 005100  
3194 007556 005104  
3195 007560 040004  
3196 007562 005304  
3197 007564 001404  
3198  
3199  
3200  
3201  
3202 007566 012742 000205  
3203 007572 005242  
3204 007574 000000  
3205 007576 050004  
3206 007600 005204  
3207 007602 005204  
3208 007604 001404  
3209  
3210  
3211  
3212  
3213 007606 012742 000206  
3214 007612 005242  
3215 007614 000000  
3216 007616 005000  
3217 007620 105100  
3218 007622 005004  
3219 007624 005104  
3220 007626 040004  
3221 007630 060004  
3222 007632 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     DOP0A
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <----
:          CONDITIONAL BRANCH INST. AND <----
:          REPLACE THE MOVE INSTRUCTION <----
:          WHICH FOLLOWS W/ 774 <----
        MOV     #204,-(R2)    ;MOVE TO MAILBOX # ***** 204 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT
DOP0A:  INC     R0           ;Z-BIT NOT SET
        COM     R0           ;R0=1
        COM     R0           ;R0=177776
        COM     R4           ;R4=177777
        BIC     R0,R4        ;TRY BIC: R4-1
        DEC     R4           ;R4=0
        BEQ     DOP0B
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
:          CONDITIONAL BRANCH INST. AND <---
:          REPLACE THE MOVE INSTRUCTION <-
:          WHICH FOLLOWS W/ 762 <=
        MOV     #205,-(R2)    ;MOVE TO MAILBOX # ***** 205 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT
DOP0B:  BIS     R0,R4        ;BIC CLEAR RESULT INCORRECT
        INC     R4           ;TRY BIS: R4=177777
        INC     R4           ;R4=0
        BEQ     DOP0C
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
:          CONDITIONAL BRANCH INST. AND <--
:          REPLACE THE MOVE INSTRUCTION <- -
:          WHICH FOLLOWS W/ 752 <-
        MOV     #206,-(R2)    ;MOVE TO MAILBOX # ***** 206 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT
DOP0C:  CLR     R0           ;RESULT OF BIS INCORRECT
        COM     R0           ;R0=0
        COM     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

```



3244  
3245  
3246  
3247  
3248  
3249  
3250  
3251  
3252 007666 005212  
3253 007670 022712 000114  
3254 007674 001024  
3255 007676 005000  
3256 007700 005010  
3257 007702 105110  
3258 007704 005220  
3259 007706 005400  
3260 007710 060037 000000  
3261 007714 100403  
3262 007716 001402  
3263 007720 102401  
3264 007722 103404  
3265  
3266  
3267  
3268  
3269 007724  
3270 007724 012742 000211  
3271 007730 005242  
3272 007732 000000  
3273 007734 105137 000000  
3274 007740 005337 000000  
3275 007744 001404  
3276  
3277  
3278  
3279  
3280 007746 012742 000212  
3281 007752 005242  
3282 007754 000000  
3283

```
.....  
: THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND INSTRUCTIONS. IT SETS  
: DATA IN R0 AND LOCATION 0 AND OF-RATES 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
```

3284  
3285  
3286  
3287  
3288  
3289  
3290  
3291  
3292  
3293  
3294  
3295  
3296  
3297  
3298  
3299  
3300  
3301  
3302  
3303  
3304  
3305  
3306  
3307  
3308  
3309  
3310  
3311  
3312  
3313  
3314  
3315  
3316  
3317  
3318  
3319  
3320  
3321  
3322  
3323  
3324  
3325  
3326  
3327  
3328  
3329  
3330  
3331  
3332  
3333  
3334  
3335  
3336  
3337  
3338  
3339

007756 005212  
007760 022712 000115  
007764 001042  
007766 005000  
007770 005004  
007772 005204  
007774 020400  
007776 003004  
  
010000 012742 000213  
010004 005242  
010006 000000  
010010 020004  
010012 002404  
  
010014 012742 000214  
010020 005242  
010022 000000  
010024 005200  
010026 020400  
010030 001404  
  
010032 012742 000215  
010036 005242  
010040 000000  
010042 005000  
010044 005100  
010046 005004  
010050 030004  
010052 001404  
  
010054 012742 000216  
010060 005242  
010062 000000  
010064 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
*****
```

```
3340 010066 030004 BIT R0,R4 ;TRY BIT AGAIN
3341 010070 100404 BMI TS116
3342 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
3343 ; CONDITIONAL BRANCH INST. AND <===
3344 ; REPLACE THE MOVE INSTRUCTION <-
3345 ; WHICH FOLLOWS W/ 735 <---
3346 010072 012742 000217 MOV #217,-(R2) ;MOVE TO MAILBOX # ***** 217 *****
3347 010076 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3348 010100 000000 HALT ;CC'S NOT CORRECT FOR BIT
3349 ; OR SEQUENCE ERROR
```

\*\*\*\*\*

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.

\*\*\*\*\*

```
3358 010102 005212 TS116: INC (R2) ;UPDATE TEST NUMBER
3359 010104 022712 000116 CMP #116,(R2) ;SEQUENCE ERROR?
3360 010110 001022 BNE TS117-10 ;BR TO ERROR HALT ON SEQ ERROR
3361 010112 005000 CLR R0 ;R0=0
3362 010114 005010 CLR (R0) ;LOC. 0=0
3363 010116 005110 COM (R0) ;LOC. 0=177777
3364 010120 005200 INC R0 ;R0=1
3365 010122 020037 000000 CMP R0,#0 ;TRY CMP MODE 0,3
3366 010126 100403 BMI DNM03A ;CC=0001
3367 010130 001402 BEQ DNM03A
3368 010132 102401 BVS DNM03A
3369 010134 103404 BCS DNM03B
```

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

```
3374 010136 DNM03A:
3375 010136 012742 000220 MOV #220,-(R2) ;MOVE TO MAILBOX # ***** 220 *****
3376 010142 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3377 010144 000000 HALT ;CC'S NOT SET CORRECTLY
3378 010146 005300 DNM03B: DEC R0
3379 010150 001002 BNE DNM03C
3380 010152 005210 INC (R0)
3381 010154 001404 BEQ TS117
```

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

```
3386 010156 DNM03C:
3387 010156 012742 000221 MOV #221,-(R2) ;MOVE TO MAILBOX # ***** 221 *****
3388 010162 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3389 010164 000000 HALT ;DATA INCORRECTLY MODIFIED BY CMP
3390 ; OR SEQUENCE ERROR
```

3391  
3392  
3393  
3394  
3395  
3396  
3397  
3398  
3399  
3400  
3401 010166 005212  
3402 010170 022712 000117  
3403 010174 001007  
3404 010176 005000  
3405 010200 005100  
3406 010202 005004  
3407 010204 005014  
3408 010206 005214  
3409 010210 061400  
3410 C10212 001404  
3411  
3412  
3413  
3414  
3415 010214 C12742 000222  
3416 010220 005242  
3417 010222 000000  
3418

```
.....  
: 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
```

```

3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429 010224 005212
3430 010226 022712 000120
3431 010232 001007
3432 010234 005000
3433 010236 005010
3434 010240 005110
3435 010242 005004
3436 010244 151004
3437 010246 105104
3438 010250 001404
3439
3440
3441
3442
3443 010252 012742 000223
3444 010256 005242
3445 010260 000000
3446

```

```

:*****
:
:      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

```

```

3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458 010262 005212
3459 010264 022712 000121
3460 010270 001007
3461 010272 005000
3462 010274 005010
3463 010276 005110
3464 010300 005004
3465 010302 105104
3466 010304 121004
3467 010306 001404
3468
3469
3470
3471
3472 010310 012742 000224
3473 010314 005242
3474 010316 000000
3475

```

```

:*****
:
: 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

```



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  
3514  
3515  
3516  
3517  
3518  
3519  
3520

010320 005212  
010322 022712 000122  
010326 001020  
010330 005000  
010332 005010  
010334 105110  
010336 005110  
010340 005004  
010342 005104  
010344 111004  
010346 005704  
010350 001404  
  
010352 012742 000225  
010356 005242  
010360 000000  
010362 005110  
010364 111004  
010366 100404  
  
010370 012742 000226  
010374 005242  
010376 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
      MOV B (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 <===
: MOVE TO MAILBOX # ***** 225 *****
: SET MSGTYP TO FATAL ERROR
: MOV B SHOULD SIGN X-TEND
: LOC 0=177777
: DO MOV B W/ EVEN BYTE
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
: CONDITIONAL BRANCH INST. AND < ---
: REPLACE THE MOVE INSTRUCTION < -
: WHICH FOLLOWS W/ 757 <
: MOVE TO MAILBOX # ***** 226 *****
: SET MSGTYP TO FATAL ERROR
: MOV B SHOULD SIGN X-TEND
: OR SEQUENCE ERROR
:
: *****
```

```

3521
3522
3523
3524
3525
3526
3527
3528
3529
3530
3531
3532 010400 005212
3533 010402 022712 000123
3534 010406 001010
3535 010410 005000
3536 010412 005010
3537 010414 005004
3538 010416 005204
3539 010420 105114
3540 010422 151410
3541 010424 005210
3542 010426 001404
3543
3544
3545
3546
3547 010430 012742 000227
3548 010434 005242
3549 010436 000000
3550

```

```

:*****
:
: 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

```

```

3551
3552
3553
3554
3555
3556
3557
3558
3559
3560
3561
3562 010440 005212
3563 010442 022712 000124
3564 010446 001015
3565 010450 005000
3566 010452 005010
3567 010454 005110
3568 010456 012004
3569 010460 005204
3570 010462 001404
3571
3572
3573
3574
3575 010464 012742 000230
3576 010470 005242
3577 010472 000000
3578 010474 005300
3579 010476 005300
3580 010500 001404
3581
3582
3583
3584
3585 010502 012742 000231
3586 010506 005242
3587 010510 000000
3588

```

```

*****
: 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
HALT ;RESULT OF MOV INST INCORRECT
DOP2: 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

```

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  
3621  
3622  
3623  
3624  
3625  
3626  
3627

```
.....  
: 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  
HALT ;BICB DESTINATION INCORRECT  
DOPB2A: COMB @#0 ;CHECK BICB SOURCE  
BEQ TS126  
: 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 *****  
INL -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;BICB SOURCE INCORRECTLY CHANGED  
: OR SEQUENCE ERROR
```

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  
3660  
3661  
3662  
3663  
3664  
3665  
3666

010566 005212  
010570 022712 000126  
010574 001017  
010576 005000  
010600 005004  
010602 005010  
010604 005110  
010606 105120  
010610 112004  
010612 005204  
010614 001404  
  
010616 012742 000234  
010622 005242  
010624 000000  
010626 005740  
010630 005700  
010632 001404  
  
010634 012742 000235  
010640 005242  
010642 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 '0 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  
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
```

3667  
3668  
3669  
3670  
3671  
3672  
3673  
3674  
3675  
3676  
3677  
3678 010644 005212  
3679 010646 022712 000127  
3680 010652 001011  
3681 010654 012737 052525 000000  
3682 010662 012700 125252  
3683 010666 053700 000000  
3684 010672 005200  
3685 010674 001404  
3686  
3687  
3688  
3689  
3690 010676 012742 000236  
3691 010702 005242  
3692 010704 000000  
3693  
3694  
3695  
3696  
3697  
3698  
3699  
3700  
3701  
3702  
3703  
3704 010706 005212  
3705 010710 022712 000130  
3706 010714 001011  
3707 010716 012737 052652 000000  
3708 010724 005000  
3709 010726 153700 000000  
3710 010732 022700 000252  
3711 010736 001404  
3712  
3713  
3714  
3715  
3716 010740 012742 000237  
3717 010744 005242  
3718 010746 000000  
3719

\*\*\*\*\*  
: 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 ZERO.  
\*\*\*\*\*

: 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  
\*\*\*\*\*

```

3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730 010750 005212
3731 010752 022712 000131
3732 010756 001011
3733 010760 012737 052652 000000
3734 010766 005000
3735 010770 153700 000001
3736 010774 022700 000125
3737 011000 001404
3738
3739
3740
3741
3742 011002 012742 000240
3743 011006 005242
3744 011010 000000
3745
3746
3747
3748
3749
3750 011012 005212
3751 011014 022712 000132
3752 011020 001017
3753 011022 005000
3754 011024 105100
3755 011026 000263
3756 011030 132700 000200
3757 011034 001403
3758 011036 102402
3759 011040 103001
3760 011042 100404
3761
3762
3763
3764
3765 011044
3766 011044 012742 000241
3767 011050 005242
3768 011052 000000
3769 011054 105100
3770 011056 001404
3771
3772
3773
3774
3775 011060 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/ 760 <
:
MOV #242,-(R2) ;MOVE TO MAILBOX # ***** 242 *****

```

```
3776 011064 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
3777 011066 000000          HALT                    ;DEST. DATA MODIFIED
3778                                     ; OR SEQUENCE ERROR
3779
3780 :*****
3781 :TEST 133      TEST DEST. MODE 1 W/ DOP NON-MODIFYING INST
3782 :*****
3783 011070 005212          TS133: INC      (R2)          ;UPDATE TEST NUMBER
3784 011072 022712 000133  CMP      #133,(R2)      ;SEQUENCE ERROR?
3785 011076 001017          BNE     TS134-10       ;BR TO ERROR HALT ON SEQ ERROR
3786 011100 005000          CLR     R0            ;R0=0
3787 011102 005010          CLR     (R0)          ;LOC. 0=0
3788 011104 000241          CLC                    ;CLEAR C BIT
3789 011106 032710 177777  BIT     #177777,(R0)   ;TRY DOPNM DEST. MODE 1
3790 011112 100403          BMI     DNM1A         ;BR TO ERROR IF N BIT SET
3791 011114 102402          BVS     DNM1A         ;BR TO ERROR IF V BIT SET
3792 011116 103401          BCS     DNM1A         ;BR TO ERROR IF C BIT SET
3793 011120 001404          BEQ     DNM1B
3794                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
3795                                     ; CONDITIONAL BRANCH INST. AND <
3796                                     ; REPLACE THE MOVE INSTRUCTION <-
3797                                     ; WHICH FOLLOWS W/ 766 <-
3798 011122
3799 011122 012742 000243  DNM1A: MOV     #243,-(R2)     ;MOVE TO MAILBOX # ***** 243 *****
3800 011126 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
3801 011130 000000          HALT                    ;COND. CODES INCORRECT
3802 011132 005710          DNM1B: TST     (R0)     ;CHECK TEST DATA
3803 011134 001404          BEQ     TS134
3804                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
3805                                     ; CONDITIONAL BRANCH INST. AND <-
3806                                     ; REPLACE THE MOVE INSTRUCTION <
3807                                     ; WHICH FOLLOWS W/ 760 <
3808 011136 012742 000244          MOV     #244,-(R2)     ;MOVE TO MAILBOX # ***** 244 *****
3809 011142 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
3810 011144 000000          HALT                    ;DESTINATION DATA MODIFIED
3811                                     ; OR SEQUENCE ERROR
3812
3813 :*****
3814 :TEST 134      TEST DEST. MODE 2 W/ DOP NON-MODIFYING INST.
3815 :*****
3816 011146 005212          TS134: INC     (R2)     ;UPDATE TEST NUMBER
3817 011150 022712 000134  CMP     #134,(R2)     ;SEQUENCE ERROR?
3818 011154 001027          BNE     TS135-i0     ;BR TO ERROR HALT ON SEQ ERROR
3819 011156 005000          CLR     R0            ;R0=0
3820 011160 005010          CLR     (R0)          ;LOC. 0=0
3821 011162 052710 125252  BIS     #125252,(R0)   ;LOC. 0=125252
3822 011166 032720 077777  BIT     #77777,(R0)+  ;TRY DOPNM INST W/ MODE 2
3823 011172 102402          BVS     DNM2A         ;BR TO ERROR IF V BIT SET
3824 011174 001401          BEQ     DNM2A         ;BR TO ERROR IF Z-BIT SET
3825 011176 100004          BPL     DNM2B
3826                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
3827                                     ; CONDITIONAL BRANCH INST. AND <
3828                                     ; REPLACE THE MOVE INSTRUCTION <
3829                                     ; WHICH FOLLOWS W/ 766 <-
3830 011200
3831 011200 012742 000245  DNM2A: MOV     #245,-(R2)     ;MOVE TO MAILBOX # ***** 245 *****
```





```
3888 011326 005200
3889 011330 132720 000201
3890 011334 001402
3891 011336 102401
3892 011340 100004
3893
3894
3895
3896
3897 011342
3898 011342 012742 000252
3899 011346 005242
3900 011350 000000
3901 011352 005300
3902 011354 005300
3903 011356 001404
3904
3905
3906
3907
3908 011360 012742 000253
3909 011364 005242
3910 011366 000000
3911 011370 022710 052652
3912 011374 001404
3913
3914
3915
3916
3917 011376 012742 000254
3918 011402 005242
3919 011404 000000
3920
3921
3922
3923
3924
3925
3926 011406 005212
3927 011410 022712 000136
3928 011414 001050
3929 011416 005000
3930 011420 005010
3931 011422 052710 125125
3932 011426 105100
3933 011430 005200
3934 011432 005010
3935 011434 000263
3936 011436 132730 000201
3937 011442 001403
3938 011444 102402
3939 011446 103001
3940 011450 100004
3941
3942
3943

DNMB2C: INC RO ;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 RO ;DEC RO TO CHECK IT.
DEC RO
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 RO ;RO=0
CLR (R0) ;LOC. 0=0
BIS #125125,(R0) ;LOC. 0=125125
COMB RO ;RO=377
INC RO ;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 < -
```

```

3944                                     ;           WHICH FOLLOWS W/ 761           <-- =
3945 011452                               DNMB3A:                               ;
3946 011452 012742 000255                 MOV    #255,-(R2)                   ;MOVE TO MAILBOX # ***** 255 *****
3947 011456 005242                         INC    -(R2)                       ;SET MSGTYP TO FATAL ERROR
3948 011460 000000                         HALT                                     ;COND. CODES INCORRECT
3949 011462 022700 000402                 DNMB3B: CMP    #402,R0              ;CHECK DEST. REGISTER INC. BY 2 AND INC BY 2 AGAIN
3950 011466 001404                         BEQ    DNMB3C                       ;
3951                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3952                                     ;          CONDITIONAL BRANCH INST. AND <====
3953                                     ;          REPLACE THE MOVE INSTRUCTION <====
3954                                     ;          WHICH FOLLOWS W/ 752           <====
3955 011470 012742 000256                 MOV    #256,-(R2)                   ;MOVE TO MAILBOX # ***** 256 *****
3956 011474 005242                         INC    -(R2)                       ;SET MSGTYP TO FATAL ERROR
3957 011476 000000                         HALT                                     ;DEST. REGISTER NOT INCREMENTED BY 2
3958 011500 005200                         DNMB3C: INC    RC                    ;RO=404
3959 011502 005200                         INC    RO
3960 011504 132730 000201                 BITB   #201,@(R0)+                  ;TRY DOPNM DEST MODE 3-BYTE(ODD)
3961 011510 001402                         BEQ    DNMB3D                       ;BR TO ERROR IF Z BIT SET
3962 011512 102401                         BVS   DNMB3D                       ;BR TO ERROR IF V BIT SET
3963 011514 100404                         BMI   DNMB3E                       ;
3964                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3965                                     ;          CONDITIONAL BRANCH INST. AND <====
3966                                     ;          REPLACE THE MOVE INSTRUCTION <====
3967                                     ;          WHICH FOLLOWS W/ 737           <====
3968 011516                               DNMB3D:                               ;
3969 011516 012742 000257                 MOV    #257,-(R2)                   ;MOVE TO MAILBOX # ***** 257 *****
3970 011522 005242                         INC    -(R2)                       ;SET MSGTYP TO FATAL ERROR
3971 011524 000000                         HALT                                     ;COND. CODES INCORRECT
3972 011526 005004                         DNMB3E: CLR    R4                    ;R4=0
3973 011530 022714 125125                 CMP    #125125,(R4)                 ;CHECK DEST. DATA
3974 011534 001404                         BEQ    TS137                         ;
3975                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3976                                     ;          CONDITIONAL BRANCH INST. AND <====
3977                                     ;          REPLACE THE MOVE INSTRUCTION <====
3978                                     ;          WHICH FOLLOWS W/ 727           <====
3979 011536 012742 000260                 MOV    #260,-(R2)                   ;MOVE TO MAILBOX # ***** 260 *****
3980 011542 005242                         INC    -(R2)                       ;SET MSGTYP TO FATAL ERROR
3981 011544 000000                         HALT                                     ;DEST. DATA MODIFIED
3982                                     ; OR SEQUENCE ERROR
3983
3984 ;*****
3985 ;TEST 137 TEST DEST. MODE 4 W/DOP NON-MODIFYING INST.
3986 ;*****
3987 011546 005212                               TS137: INC    (R2)                   ;UPDATE TEST NUMBER
3988 011550 022712 000137                 CMP    #137,(R2)                   ;SEQUENCE ERROR?
3989 011554 001033                         BNE   TS140-10                      ;BR TO ERROR HALT ON SEQ ERROR
3990 011556 005000                         CLR    RO                           ;RO=0
3991 011560 005010                         CLR    (R0)                          ;LOC. 0=0
3992 011562 052710 125252                 BIS    #125252,(R0)                 ;LOC. 0=125125
3993 011566 052700 000002                 BIS    #2,R0                        ;RO=2
3994 011572 000277                         SCC                                     ;SET ALL COND. CODE BITS
3995 011574 032740 020000                 BIT    #2000,-(R0)                  ;TRY DOPNM W/ MODE 4
3996 011600 100403                         BMI   DNMB4A                        ;BR TO ERROR IF N-BIT SET
3997 011602 102402                         BVS   DNMB4A                        ;BR TO ERROR IF V-BIT SET
3998 011604 103001                         BCC   DNMB4A                        ;BR TO ERROR IF C-BIT CHAR
3999 011606 001004                         BNE   DNMB4B

```

C 7

```
4000 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4001 ; CONDITIONAL BRANCH INST. AND <====
4002 ; REPLACE THE MOVE INSTRUCTION <====
4003 ; WHICH FOLLOWS W/ 762 <====
4004 011610 DNM4A:
4005 011610 012742 000261 MOV #261,-(R2) ;MOVE TO MAILBOX # ***** 261 *****
4006 011614 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4007 011616 000000 HALT ;COND. CODES INCORRECT
4008 011620 005700 DNM4B: TST R0 ;CHECK DEST. REGISTER
4009 011622 001404 BEQ DNM4C
4010 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4011 ; CONDITIONAL BRANCH INST. AND <====
4012 ; REPLACE THE MOVE INSTRUCTION <====
4013 ; WHICH FOLLOWS W/ 754 <====
4014 011624 012742 000262 MOV #262,-(R2) ;MOVE TO MAILBOX # ***** 262 *****
4015 011630 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4016 011632 000000 HALT ;DEST. REGISTER NOT DECREMENTED BY 2
4017 011634 022737 125252 000000 DNM4C: CMP #125252,@#0 ;CHECK DEST. DATA
4018 011642 001404 BEQ TS140
4019 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4020 ; CONDITIONAL BRANCH INST. AND <====
4021 ; REPLACE THE MOVE INSTRUCTION <====
4022 ; WHICH FOLLOWS W/ 744 <====
4023 011644 012742 000263 MOV #263,-(R2) ;MOVE TO MAILBOX # ***** 263 *****
4024 011650 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4025 011652 000000 HALT ;DEST. DATA MODIFIED
4026 ; OR SEQUENCE ERROR
4027
4028 ;*****
4029 ;TEST 140 TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
4030 ;*****
4031 011654 005212 TS140: INC (R2) ;UPDATE TEST NUMBER
4032 011656 022712 000140 CMP #140,(R2) ;SEQUENCE ERROR?
4033 011662 001051 BNE TS141-10 ;BR TO ERROR HALT ON SEQ ERROR
4034 011664 005000 CLR R0 ;R0=0
4035 011666 005010 CLR (R0) ;LOC. 0=0
4036 011670 052710 052652 BIS #52652,(R0) ;LOC. 0=52652
4037 011674 052700 000002 BIS #2,R0 ;R0=2
4038 011700 000257 CCC ;COND. CODES=0
4039 011702 132740 000201 BITB #201,-(R0) ;TRY DOPNM INST W/MODE 4 ODD BYTE
4040 011706 102403 BVS DNM4A ;BR TO ERROR IF V BIT SET
4041 011710 001402 BEQ DNM4A ;BR TO ERROR IF Z BIT SET
4042 011712 103401 BCS DNM4A ;BR TO ERROR IF C BIT SET
4043 011714 001004 BNE DNM4B
4044 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4045 ; CONDITIONAL BRANCH INST. AND <====
4046 ; REPLACE THE MOVE INSTRUCTION <====
4047 ; WHICH FOLLOWS W/ 762 <====
4048 011716 DNM4A:
4049 011716 012742 000264 MOV #264,-(R2) ;MOVE TO MAILBOX # ***** 264 *****
4050 011722 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4051 011724 000000 HALT ;COND. CODES INCORRECT
4052 011726 022700 000001 DNM4B: CMP #1,R0 ;CHECK DEST. REGISTER
4053 011732 001404 BEQ DNM4C
4054 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < - -
4055 ; CONDITIONAL BRANCH INST. AND < - -
```

```
4056                                     ;          REPLACE THE MOVE INSTRUCTION <====  
4057                                     ;          WHICH FOLLOWS W/ 753          <====  
4058 011734 012742 000265                MOV    #265,-(R2)                ;MOVE TO MAILBOX # ***** 265 *****  
4059 011740 005242                        INC    -(R2)                    ;SET MSGTYP TO FATAL ERROR  
4060 011742 000000                        HALT                               ;DEST REG. NOT DECREMENTED BY 1  
4061 011744 132740 000201                DNMB3C: BITB  #201,-(R0)         ;TRY DOPNM INST. W/MODE 4 EVEN BYTE  
4062 011750 001401                        BEQ    DNMB4D                   ;BR TO ERROR IF 7-BIT SET  
4063 011752 100404                        BMI    DNMB4E                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4064                                     ;                                     <====  
4065                                     ;          CONDITIONAL BRANCH INST. AND <====  
4066                                     ;          REPLACE THE MOVE INSTRUCTION <====  
4067                                     ;          WHICH FOLLOWS W/ 743          <====  
4068 011754                                     DNMB4D:                          ;  
4069 011754 012742 000266                MOV    #266,-(R2)                ;MOVE TO MAILBOX # ***** 266 *****  
4070 011760 005242                        INC    -(R2)                    ;SET MSGTYP TO FATAL ERROR  
4071 011762 000000                        HALT                               ;COND. CODES INCORRECT  
4072 011764 005700                        DNMB4E: TST  R0                  ;CHECK DEST. REGISTER  
4073 011766 001404                        BEQ    DNMB4F                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
4074                                     ;                                     <====  
4075                                     ;          CONDITIONAL BRANCH INST. AND <====  
4076                                     ;          REFLACE THE MOVE INSTRUCTION <====  
4077                                     ;          WHICH FOLLOWS W/ 735          <====  
4078 011770 012742 000267                MOV    #267,-(R2)                ;MOVE TO MAILBOX # ***** 267 *****  
4079 011774 005242                        INC    -(R2)                    ;SET MSGTYP TO FATAL ERROR  
4080 011776 000000                        HALT                               ;DEST. REG. NOT DECREMENTED BY 1  
4081 012000 022710 052652                DNMB4F: CMP  #52652,(R0)         ;CHECK DESTINATION DATA  
4082 012004 001404                        BEQ    TS141                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < - -  
4083                                     ;                                     < - -  
4084                                     ;          CONDITIONAL BRANCH INST. AND < - -  
4085                                     ;          REPLACE THE MOVE INSTRUCTION < - -  
4086                                     ;          WHICH FOLLOWS W/ 726          < - -  
4087 012006 012742 000270                MOV    #270,-(R2)                ;MOVE TO MAILBOX # ***** 270 *****  
4088 012012 005242                        INC    -(R2)                    ;SET MSGTYP TO FATAL ERROR  
4089 012014 000000                        HALT                               ;DEST. DATA MODIFIED  
4090                                     ; OR SEQUENCE ERROR  
4091  
4092  
4093 ;*****  
4094 ;TEST 141 TEST DEST MODE 5 W/DOP NON-MODIFYING INST.  
4095 ;*****  
4095 012016 005212                                     TS141: INC  (R2)                ;UPDATE TEST NUMBER  
4096 012020 022712 000141                CMP    #141,(R2)                ;SEQUENCE ERROR?  
4097 012024 001034                        BNE    TS142-10                 ;BR TO ERROR HALT ON SEQ ERROR  
4098 012026 005000                        CLR    R0                        ;R0=0  
4099 012030 005010                        CLR    (R0)                     ;LOC 0=0  
4100 012032 052710 100000                BIS    #100000,(R0)             ;LOC. 0=100000  
4101 012036 052700 000402                BIS    #402,R0                  ;R0=2  
4102 012042 000277                        SCC                               ;SET ALL COND. CODE BITS  
4103 012044 032750 100000                BIT    #100000,@-(R0)           ;TRY DOPNM W/MODE 5  
4104 012050 102403                        BVS    DNMB5A                   ;BR TO ERROR IF V-BIT SET  
4105 012052 103002                        BCC    DNMB5A                   ;BR TO ERROR IF C-BIT CLEAR  
4106 012054 001401                        BEQ    DNMB5A                   ;BR TO ERROR IF Z-BIT SET  
4107 012056 100404                        BMI    DNMB5B                   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <----  
4108                                     ;                                     <----  
4109                                     ;          CONDITIONAL BRANCH INST. AND < - -  
4110                                     ;          REPLACE THE MOVE INSTRUCTION < - -  
4111                                     ;          WHICH FOLLOWS W/ 762          < - -
```

CJKDB-D DCF--AA (PU DIAG.  
CJKDBD.P1 24-NOV-80 11:07

MACY11 30A(1052) 14-JAN-81 11:46 PAGE 82  
T141 TEST DEST MODE 5 W/DOP NON-MODIFYING INST.

SEQ 0082

```

4112 012060
4113 012060 012742 000271
4114 012064 005242
4115 012066 000000
4116 012070 022700 000400
4117 012074 001404
4118
4119
4120
4121
4122 012076 012742 000272
4123 012102 005242
4124 012104 000000
4125 012106 022737 100000 000000
4126 012114 001404
4127
4128
4129
4130
4131 012116 012742 000273
4132 012122 005242
4133 012124 000000
4134
4135
4136
4137
4138
4139 012126 005212
4140 012130 022712 000142
4141 012134 001033
4142 012136 005000
4143 012140 005010
4144 012142 052710 000001
4145 012146 005100
4146 012150 032760 000001 000001
4147 012156 001403
4148 012160 102402
4149 012162 103001
4150 012164 100004
4151
4152
4153
4154
4155 012166
4156 012166 012742 000274
4157 012172 005242
4158 012174 000000
4159 012176 022700 177777
4160 012202 001404
4161
4162
4163
4164
4165 012204 012742 000275
4166 012210 005242
4167 012212 000000

```

```

DNM5A:
MOV #271,-(R2) ;MOVE TO MAILBOX # ***** 271 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;COND. CODES INCORRECT
DNM5B: 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 <===
MOV #272,-(R2) ;MOVE TO MAILBOX # ***** 272 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DEST. REGISTER NOT DECREMENTED BY 2
DNM5C: CMP #100000,a#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 <-
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,(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 <-
MOV #275,-(R2) ;MOVE TO MAILBOX # ***** 275 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DEST. REGISTER MODIFIED

```

```

4168 012214 022737 000001 000000 DNM6C: CMP #1,@#0 ;CHECK DEST. DATA
4169 012222 001404 BEQ TS143 ;
4170 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -
4171 ; CONDITIONAL BRANCH INST. AND <==--
4172 ; REPLACE THE MOVE INSTRUCTION <==--
4173 ; WHICH FOLLOWS W/ 744 <==--
4174 012224 012742 000276 MOV #276,-(R2) ;MOVE TO MAILBOX # ***** 276 *****
4175 012230 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4176 012232 000000 HALT ;DEST. DATA MODIFIED
4177 ; OR SEQUENCE ERROR
4178
4179
4180
4181

```

```

:*****
:TEST 143 TEST DEST MODE 7 W/DOP NON-MODIFYING INST.
:*****

```

```

4182 012234 005212 TS143: INC (R2) ;UPDATE TEST NUMBER
4183 012236 022712 000143 CMP #143,(R2) ;SEQUENCE ERROR?
4184 012242 001034 BNE TS144-10 ;BR TO ERROR HALT ON SEQ ERROR
4185 012244 005000 CLR R0 ;R0=0
4186 012246 005010 CLR (R0) ;LOC. 0=0 C-BIT=0
4187 012250 052710 125125 BIS #125125,(R0) ;LOC. 0=125125
4188 012254 052700 000001 BIS #1,R0 ;R0=1
4189 012260 132770 000125 000403 BITB #125,@403(R0) ;TRY DOPNM W/MODE 7
4190 012266 102403 BVS DNM7A ;BR TO ERROR IF V-BIT SET
4191 012270 100402 BMI DNM7A ;BR TO ERROR IF N-BIT SET
4192 012272 103401 BCS DNM7A ;BR TO ERROR IF C-BIT SET
4193 012274 001404 BEQ DNM7B
4194
4195
4196
4197

```

```

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

```

```

4198 012276 DNM7A: MOV #277,-(R2) ;MOVE TO MAILBOX # ***** 277 *****
4199 012276 012742 000277 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4200 012302 005242 HALT ;COND. CODES INCORRECT
4201 012304 000000 DNM7B: CMP #1,R0 ;CHECK DEST. REGISTER
4202 012306 022700 000001 BEQ DNM7C
4203 012312 001404
4204
4205
4206
4207

```

```

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

```

```

4208 012314 012742 000300 MOV #300,-(R2) ;MOVE TO MAILBOX # ***** 300 *****
4209 012320 005242 INC -(R2) ;SET MSGTYP TO FATAL ERRGR
4210 012322 000000 HALT ;DESTINATION REGISTER MODIFIED
4211 012324 022737 125125 000000 DNM7C: CMP #125125,@#0 ;CHECK DEST. DATA
4212 012332 001404 BEQ TS144
4213
4214
4215
4216

```

```

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

```

```

4217 012334 012742 000301 MOV #301,-(R2) ;MOVE TO MAILBOX # ***** 301 *****
4218 012340 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4219 012342 000000 HALT ;DEST. DATA INCORRECT
4220 ; OR SEQUENCE ERROR
4221
4222
4223

```

```

:*****
:
:

```

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

4231 012344 005212  
4232 012346 022712 000144  
4233 012352 001016  
4234 012354 005000  
4235 012356 005010  
4236 012360 005100  
4237 012362 005004  
4238 012364 010014  
4239 012366 102402  
4240 012370 001401  
4241 012372 100404

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 < -

4246 012374  
4247 012374 012742 000302  
4248 012400 005242  
4249 012402 000000  
4250 012404 005704  
4251 012406 001404

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 <

4256 012410 012742 000303  
4257 012414 005242  
4258 012416 000000

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

4270 012420 005212  
4271 012422 022712 000145  
4272 012426 001026  
4273 012430 005000  
4274 012432 005001  
4275 012434 005010  
4276 012436 005110  
4277 012440 010120  
4278 012442 100402  
4279 012444 102401

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



```

4280 012446 001474          BEQ      MDM2B
4281
4282
4283
4284
4285 012450          MDM2A:
4286 012450 012742 00C304    MOV      #304,-(R2)      ;MOVE TO MAILBOX # ***** 304 *****
4287 012454 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4288 012456 000000          HALT
4289 012460 005300          MDM2B: DEC      R0      ;CC'S INCORRECT
4290 012462 005300          DEC      R0
4291 012464 001404          BEQ      MDM2D
4292
4293
4294
4295
4296 012466          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <= -
4297 012466 012742 000305    MOV      #305,-(R2)      ;CONDITIONAL BRANCH INST. AND <= -
4298 012472 005242          INC      -(R2)          ;REPLACE THE MOVE INSTRUCTION <= -
4299 012474 000000          HALT                    ;WHICH FOLLOWS W/ 760 <= -
4300 012476 005737 000000    MDM2D: TST      @#0
4301 012502 001404          BEQ      TS146
4302
4303
4304
4305
4306 012504 012742 000306    MOV      #306,-(R2)      ;MOVE TO MAILBOX # ***** 306 *****
4307 012510 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4308 012512 000000          HALT                    ;DESTINATION DATA INCORRECT
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319 012514 005212          TS146: INC      (R2)      ;UPDATE TEST NUMBER
4320 012516 022712 000146    CMP      #146,(R2)      ;SEQUENCE ERROR?
4321 012522 001046          BNE     TS147-10        ;BR TO ERROR HALT ON SEQ ERROR
4322 012524 005000          CLR      R0            ;R0=0
4323 012526 005010          CLR      (R0)          ;LOC. 0=0
4324 012530 112720 000125    MOVVB   #125,(R0)+     ;TRY DESTINATION MODE 2 W/EVEN BYTE
4325 012534 102402          BVS     MBDM2A         ;BR TO ERROR IF V SET
4326 012536 001401          BEQ     MBDM2A         ;BR TO ERROR IF Z SET
4327 012540 100004          BPL     MBDM2B
4328
4329
4330
4331
4332 012542          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
4333 012542 012742 000307    MBDM2A: MOV      #307,-(R2)      ;CONDITIONAL BRANCH INST. AND <
4334 012546 005242          INC      -(R2)          ;REPLACE THE MOVE INSTRUCTION <
4335 012550 000000          HALT                    ;WHICH FOLLOWS W/ 770 <

```

\*\*\*\*\*  
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  
\*\*\*\*\*

```

4336 012552 022700 000001      MBDM2B: CMP      #1,R0
4337 012556 001404              BEQ      MBDM2C
4338                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--- =
4339                               ; CONDITIONAL BRANCH INST. AND <--- =
4340                               ; REPLACE THE MOVE INSTRUCTION < -
4341                               ; WHICH FOLLOWS W/ 761 < ---
4342 012560 012742 000310      MOV      #310,-(R2) ;MOVE TO MAILBOX # ***** 310 *****
4343 012564 005242              INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4344 012566 000000              HALT                    ;REGISTER NOT INCREMENTED BY ONE
4345 012570 112720 000252      MBDM2C: MOVVB   #252,(R0)+ ;TRY DESTINATION MODE 2 W/ODD BYTE
4346 012574 102402              BVS     MBDM2D
4347 012576 001401              BEQ     MBDM2D
4348 012600 100404              BMI     MBDM2E
4349                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
4350                               ; CONDITIONAL BRANCH INST. AND < - =
4351                               ; REPLACE THE MOVE INSTRUCTION < -
4352                               ; WHICH FOLLOWS W/ 750 < ---
4353 012602                      MBDM2D:
4354 012602 012742 000311      MOV      #311,-(R2) ;MOVE TO MAILBOX # ***** 311 *****
4355 012606 005242              INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4356 012610 000000              HALT                    ;CC'S NOT SET CORRECT
4357 012612 022700 00C002      MBDM2E: CMP      #2,R0
4358 012616 001404              BEQ     MBDM2F
4359                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== =
4360                               ; CONDITIONAL BRANCH INST. AND <====
4361                               ; REPLACE THE MOVE INSTRUCTION <====
4362                               ; WHICH FOLLOWS W/ 741 <--- =
4363 012620 012742 000312      MOV      #312,-(R2) ;MOVE TO MAILBOX # ***** 312 *****
4364 012624 005242              INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4365 012626 000000              HALT                    ;REGISTER NOT INCREMENTED BY ONE
4366 012630 022737 125125 000000 MBDM2F: CMP      #125125,@#0 ;CHECK DATA
4367 012636 001404              BEQ     TS147
4368                               ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- ==
4369                               ; CONDITIONAL BRANCH INST. AND <====
4370                               ; REPLACE THE MOVE INSTRUCTION <--- -
4371                               ; WHICH FOLLOWS W/ 731 <== =
4372 012640 012742 000313      MOV      #313,-(R2) ;MOVE TO MAILBOX # ***** 313 *****
4373 012644 005242              INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4374 012646 000000              HALT                    ;DESTINATION DATA INCORRECT
4375                               ; OR SEQUENCE ERROR

```

```

*****
:
: 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, MOVVB
: 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 012650 005212
4386 012652 022712 000147
4387 012656 001057
4388 012660 012700 000400
4389 012664 005010
4390 012666 005037 000000
4391 012672 012730 125252

```

```
4392 012676 102402      BVS      MDM3A      ;BR TO ERROR IF V SET
4393 012700 001401      BEQ      MDM3A      ;BR TO ERROR IF Z SET
4394 012702 100404      BMI      MDM3B
4395                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
4396                                     ;          CONDITIONAL BRANCH INST. AND < -
4397                                     ;          REPLACE THE MOVE INSTRUCTION < =
4398                                     ;          WHICH FOLLOWS W/ 765 <==
4399 012704      MDM3A:
4400 012704 012742 000314      MOV      #314,-(R2) ;MOVE TO MAILBOX # ***** 314 *****
4401 012710 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4402 012712 000000      HALT
4403 012714 022700 000402      MDM3B:  CMP      #402,R0 ;CC'S INCORRECT
4404 012720 001404      BEQ      MDM3C      ;CHECK DEST. MODE REGISTER
4405                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4406                                     ;          CONDITIONAL BRANCH INST. AND <====
4407                                     ;          REPLACE THE MOVE INSTRUCTION <====
4408                                     ;          WHICH FOLLOWS W/ 756 <====
4409 012722 012742 000315      MOV      #315,-(R2) ;MOVE TO MAILBOX # ***** 315 *****
4410 012726 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4411 012730 000000      HALT
4412 012732 022737 125252 000000  MDM3C:  CMP      #125252,@#0 ;REGISTER NOT INCREMENTED BY 2
4413 012740 001404      BEQ      MDM3D      ;CHECK DESTINATION DATA
4414                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < - -
4415                                     ;          CONDITIONAL BRANCH INST. AND < - -
4416                                     ;          REPLACE THE MOVE INSTRUCTION < - =
4417                                     ;          WHICH FOLLOWS W/ 746 < -
4418 012742 012742 000316      MOV      #316,-(R2) ;MOVE TO MAILBOX # ***** 316 *****
4419 012746 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4420 012750 000000      HALT
4421 012752 112737 000125 000000  MDM3D:  MOVB    #125,@#0 ;DESTINATION DATA INCORRECT
4422 012760 022737 125125 000000  CMP      #125125,@#0 ;TRY MOV B DESTINATION MODE Z EVEN BYTE
4423 012766 001404      BEQ      MDM3E      ;CHECK DATA
4424                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
4425                                     ;          CONDITIONAL BRANCH INST. AND <
4426                                     ;          REPLACE THE MOVE INSTRUCTION <
4427                                     ;          WHICH FOLLOWS W/ 733 <
4428 012770 012742 000317      MOV      #317,-(R2) ;MOVE TO MAILBOX # ***** 317 *****
4429 012774 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4430 012776 000000      HALT
4431 013000 112737 000525 000001  MDM3E:  MOVB    #525,@#1 ;DESTINATION DATA INCORRECT
4432 013006 022737 052525 000000  CMP      #52525,@#0 ;TRY MOV B DESTINATION MODE 2 ODD BYTE
4433 013014 001404      BEQ      TS150      ;CHECK DATA
4434                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
4435                                     ;          CONDITIONAL BRANCH INST. AND <
4436                                     ;          REPLACE THE MOVE INSTRUCTION < -
4437                                     ;          WHICH FOLLOWS W/ 720 < -
4438 013016 012742 000320      MOV      #320,-(R2) ;MOVE TO MAILBOX # ***** 320 *****
4439 013022 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4440 013024 000000      HALT
4441
4442 .....
4443
4444 ; THIS TEST VERIFIES THE MOV DESTINATION MODE 4 INSTRUCTION.
4445 ; SOP INSTRUCTIONS ON R0 ARE USED TO CLEAR TARGET LOCATION 0.
4446 ; R4 IS USED AS THE MODE 4 ADDRESSING REGISTER, AND
4447 ; CONDITIONAL BRANCHES ARE USED TO VERIFY THE DATA.
```

```
4448  
4449  
4450 :TEST 150 TEST MOV DESTINATION MODE 4  
4451 :*****  
4452 013026 005212 TS150: INC (R2) ;UPDATE TEST NUMBER  
4453 013030 022712 000150 CMP #150,(R2) ;SEQUENCE ERROR?  
4454 013034 001026 BNE TS151-10 ;BR TO ERROR HALT ON SEQ ERROR  
4455 013036 005000 CLR R0 ;R0=0  
4456 013040 005010 CLR (R0) ;LOC 0=0  
4457 013042 012704 000002 MOV #2,R4 ;R4=2  
4458 013046 012744 012345 MOV #12345,-(R4) ;TRY MOV DEST. MODE 4  
4459 013052 102402 BVS MDM4A ;BR TO ERROR IF V-BIT SET  
4460 013054 001401 BEQ MDM4A ;BR TO ERROR IF Z-BIT SET  
4461 013056 100004 BPL MDM4B  
4462 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===  
4463 ; CONDITIONAL BRANCH INST. AND <===  
4464 ; REPLACE THE MOVE INSTRUCTION <===  
4465 ; WHICH FOLLOWS W/ 766 <===  
4466 013060 MDM4A: MOV #321,-(R2) ;MOVE TO MAILBOX # ***** 321 *****  
4467 013064 012742 000321 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4468 013066 005242 HALT ;CC'S NOT CORRECT  
4469 013070 000000 MDM4B: TST R4 ;CHECK DECREMENTING OF MODE 4 REG.  
4470 013072 005704 BEQ MDM4C  
4471 013072 001404  
4472 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <= -  
4473 ; CONDITIONAL BRANCH INST. AND <- -  
4474 ; REPLACE THE MOVE INSTRUCTION < -  
4475 ; WHICH FOLLOWS W/ 760 <  
4476 013074 012742 000322 MOV #322,-(R2) ;MOVE TO MAILBOX # ***** 322 *****  
4477 013100 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4478 013102 000000 HALT ;DESTINATION MODE REGISTER NOT DECREMENTED BY 2  
4479 013104 022710 012345 MDM4C: CMP #12345,(R0) ;CHECK DESTINATION DATA  
4480 013110 001404 BEQ TS151  
4481 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- - -  
4482 ; CONDITIONAL BRANCH INST. AND <=  
4483 ; REPLACE THE MOVE INSTRUCTION < -  
4484 ; WHICH FOLLOWS W/ 751 < -  
4485 013112 012742 000323 MOV #323,-(R2) ;MOVE TO MAILBOX # ***** 323 *****  
4486 013116 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
4487 013120 000000 HALT ;DESTINATION DATA INCORRECT  
4488 ; OR SEQUENCE ERROR  
4489  
4490 :*****  
4491 : THIS TEST VERIFIES THE MOV(B) DESTINATION MODE 4 INSTRUCTION  
4492 : ON BOTH ODD AND EVEN BYTES. SOP INSTRUCTIONS ON R4 ARE  
4493 : USED TO CLEAR TARGET LOCATION 0. R0 IS USED AS THE MODE 4  
4494 : ADDRESSING REGISTER, AND CMP AND CONDITIONAL BRANCH  
4495 : INSTRUCTIONS ARE USED TO VERIFY THE DATA.  
4496 :  
4497 :*****  
4498 :  
4499 :TEST 151 TEST MOV(B) DESTINATION MODE 4  
4500 :*****  
4501 013122 005212 TS151: INC (R2) ;UPDATE TEST NUMBER  
4502 013124 022712 000151 CMP #151,(R2) ;SEQUENCE ERROR?  
4503 013130 001046 BNE TS152-10 ;BR TO ERROR HALT ON SEQ ERROR
```

```

4504 013132 005004          CLR      R4          ;R4=0
4505 013134 005014          CLR      (R4)       ;LOC. 0=0
4506 013136 012700 000002  MOV      #2,R0      ;R0 = 2
4507 013142 112740 125125  MOV8     #125125,-(R0);TRY MOV8 DEST. MODE 4-ODD BYTE
4508 013146 020027 000001  CMP      R0,#1      ;CHECK THAT DEST. REG. WAS DECREMENTED
4509 013152 001404          BEQ      MBDM4A
4510          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
4511          ; CONDITIONAL BRANCH INST. AND <
4512          ; REPLACE THE MOVE INSTRUCTION <
4513          ; WHICH FOLLOWS W/ 766 <
4514 013154 012742 000324  MOV      #324,-(R2) ;MOVE TO MAILBOX # ***** 324 *****
4515 013160 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4516 013162 000000          HALT
4517 013164 021427 052400  MBDM4A: CMP      (R4),#52400 ;DESTINATION REG. NOT DECREMENTED BY 1
4518 013170 001404          BEQ      MBDM4B    ;CHECK DEST. DATA
4519          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < =
4520          ; CONDITIONAL BRANCH INST. AND <=
4521          ; REPLACE THE MOVE INSTRUCTION <-- -
4522          ; WHICH FOLLOWS W/ 757 <
4523 013172 012742 000325  MOV      #325,-(R2) ;MOVE TO MAILBOX # ***** 325 *****
4524 013176 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4525 013200 000000          HALT
4526 013202 112740 125125  MBDM4B: MOV8     #125125,-(R0);TRY MOV8 DEST. MODE 4--EVEN BYTE
4527 013206 102402          BVS     MBDM4C      ;BR. TO ERROR IF V-BIT SET
4528 013210 001401          BFQ     MBDM4C      ;BR TO ERROR IF Z-BIT SET
4529 013212 100004          BPL     MBDM4D
4530          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < ---
4531          ; CONDITIONAL BRANCH INST. AND <=
4532          ; REPLACE THE MOVE INSTRUCTION <---=
4533          ; WHICH FOLLOWS W/ 746 <=
4534 013214          MBDM4C:
4535 013214 012742 000326  MOV      #326,-(R2) ;MOVE TO MAILBOX # ***** 326 *****
4536 013220 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4537 013222 000000          HALT
4538 013224 005700          MBDM4D: TST      R0      ;COND. CODES INCORRECT
4539 013226 001404          BEQ      MBDM4E    ;CHECK MODE 4 DEST. REGISTER
4540          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
4541          ; CONDITIONAL BRANCH INST. AND <
4542          ; REPLACE THE MOVE INSTRUCTION <-
4543          ; WHICH FOLLOWS W/ 740 <
4544 013230 012742 000327  MOV      #327,-(R2) ;MOVE TO MAILBOX # ***** 327 *****
4545 013234 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4546 013236 000000          HALT
4547 013240 021427 052525  MBDM4E: CMP      (R4),#52525 ;DESTINATION REG NOT DECREMENTED BY 1
4548 013244 001404          BEQ      TS152     ;CHECK DEST. DATA
4549          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
4550          ; CONDITIONAL BRANCH INST. AND <===
4551          ; REPLACE THE MOVE INSTRUCTION <===
4552          ; WHICH FOLLOWS W/ 731 <--
4553 013246 012742 000330  MOV      #330,-(R2) ;MOVE TO MAILBOX # ***** 330 *****
4554 013252 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4555 013254 000000          HALT
4556          ; DESTINATION DATA INCORRECT
4557          ; OR SEQUENCE ERROR
4558          ; *****
4559          ;

```

```

4560 : THIS TEST VERIFIES THE MOV DESTINATION MODE 5 AND THE MOV8
4561 : DESTINATION MODE 5 - EVEN BYTE INSTRUCTIONS. R4 IS A
4562 : POINTER TO TARGET LOCATION 0 AND R0 IS SETUP TO
4563 : POINT TO LOCATION 376 FOR THE MOV, AND LOCATION 404 FOR
4564 : THE MOV8 INSTRUCTIONS. CMP INSTRUCTIONS ARE USED TO VERIFY
4565 : PROPER ADDRESSING AND DATA.
4566 :
4567 : *****
4568 : TEST 152 TEST MOV DESTINATION MODE 5
4569 : *****
4570 TS152: INC (R2) ;UPDATE TEST NUMBER
4571 013256 005212 000152 CMP #152,(R2) ;SEQUENCE ERROR?
4572 013260 022712 000152 BNE TS153-10 ;BR TO ERROR HALT ON SEQ ERROR
4573 013264 001051 CLR R4 ;R4=0
4574 013270 005014 CLR (R4) ;LOC. 0 = 0
4575 013272 012700 000400 MOV #400,R0 ;R0=400
4576 013276 012750 004321 MOV #4321,@-(R0) ;TRY MOV DEST. MODE 5
4577 013302 102402 BVS MDM5A ;BR TO ERROR IF V-BIT SET
4578 013304 001401 BEQ MDM5A ;BR TO ERROR IF Z-BIT SET
4579 013306 100004 BPL MDM5B
4580 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== =
4581 : CONDITIONAL BRANCH INST. AND < ==-
4582 : REPLACE THE MOVE INSTRUCTION <==
4583 : WHICH FOLLOWS W/ 766 <==--
4584 013310 MDM5A:
4585 013310 012742 000331 MOV #331,-(R2) ;MOVE TO MAILBOX # ***** 331 *****
4586 013314 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4587 013316 000000 HALT ;COND. CODES INCORRECT
4588 013320 022700 000376 MDM5B: CMP #376,R0 ;CHECK MODE 5 REG. WAS DECREMENTED
4589 013324 001404 BEQ MDM5C
4590 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
4591 : CONDITIONAL BRANCH INST. AND <====
4592 : REPLACE THE MOVE INSTRUCTION <==
4593 : WHICH FOLLOWS W/ 757 <==--
4594 013326 012742 000332 MOV #332,-(R2) ;MOVE TO MAILBOX # ***** 332 *****
4595 013332 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4596 013334 000000 HALT ;MODE 5 REGISTER NOT DECREMENTED BY 2
4597 013336 022714 004321 MDM5C: CMP #4321,(R4) ;CHECK DEST. DATA
4598 013342 001404 BEQ MDM5D
4599 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <----=
4600 : CONDITIONAL BRANCH INST. AND < =
4601 : REPLACE THE MOVE INSTRUCTION <= --
4602 : WHICH FOLLOWS W/ 750 <====
4603 013344 012742 000333 MOV #333,-(R2) ;MOVE TO MAILBOX # ***** 333 *****
4604 013350 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4605 013352 000000 HALT ;DEST. DATA INCORRECT
4606 013354 012700 000406 MDM5D: MOV #406,R0 ;R0=406
4607 013360 112750 000377 MOV8 #377,@-(R0) ;TRY MOV DEST. MODE 5 --EVEN BYTE
4608 013364 022700 000404 CMP #404,R0 ;CHECK MODE 5 REG.
4609 013370 001404 BEQ MDM5E
4610 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4611 : CONDITIONAL BRANCH INST. AND <====
4612 : REPLACE THE MOVE INSTRUCTION <====
4613 : WHICH FOLLOWS W/ 735 <====
4614 013372 012742 000334 MOV #334,-(R2) ;MOVE TO MAILBOX # ***** 334 *****
4615 013376 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR

```

```
4616 013400 000000
4617 013402 022714 177721
4618 013406 001404
4619
4620
4621
4622
4623 013410 012742 000335
4624 013414 005242
4625 013416 000000
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639 013420 005212
4640 013422 022712 000153
4641 013426 001054
4642 013430 005000
4643 013432 005010
4644 013434 005200
4645 013436 012760 052525 177777
4646 013444 102402
4647 013446 001401
4648 013450 100004
4649
4650
4651
4652
4653 013452
4654 013452 012742 000336
4655 013456 005242
4656 013460 000000
4657 013462 022700 000001
4658 013466 001404
4659
4660
4661
4662
4663 013470 012742 000337
4664 013474 005242
4665 013476 000000
4666 013500 022737 052525 000000
4667 013506 001404
4668
4669
4670
4671
```

```
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 MOV B - 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 <---=

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 <-
```

```

4672 013510 012742 000340      MOV      #340,-(R2)      ;MOVE TO MAILBOX # ***** 340 *****
4673 013514 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4674 013516 000000              HALT                    ;DEST. DATA INCORRECT
4675 013520 012700 000002      MDM6D: MOV      #2,R0      ;R0=2
4676 013524 112760 000377 177777  MOVB     #377,-1(R0)    ;TRY MOVB DEST. MODE 6
4677 013532 022700 000002      CMP      #2,R0         ;CHECK DEST. REGISTER UNALTERED
4678 013536 001404              BEQ      MDM6E         ;
4679                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4680                                ;          CONDITIONAL BRANCH INST. AND <====
4681                                ;          REPLACE THE MOVE INSTRUCTION <====
4682                                ;          WHICH FOLLOWS W/ 733 <====
4683 013540 012742 C00341      MOV      #341,-(R2)    ;MOVE TO MAILBOX # ***** 341 *****
4684 013544 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4685 013546 000000              HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
4686 013550 022737 177525 000000  MDM6E: CMP      #177525,@#0 ;CHECK DEST. DATA
4687 013556 001404              BEQ      TS154         ;
4688                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4689                                ;          CONDITIONAL BRANCH INST. AND <====
4690                                ;          REPLACE THE MOVE INSTRUCTION <====
4691                                ;          WHICH FOLLOWS W/ 723 <====
4692 013560 012742 000342      MOV      #342,-(R2)    ;MOVE TO MAILBOX # ***** 342 *****
4693 013564 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4694 013566 000000              HALT                    ;DEST. DATA INCORRECT
4695                                ; OR SEQUENCE ERROR

```

```

4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727

```

```

:*****
:
: THIS TEST VERIFIES THE MOV DESTINATION MODE 7 AND MOVB - ODD BYTE
: DESTINATION MODE 7 INSTRUCTIONS. R4 POINTS TO TARGET LOC.0 AND R0
: IS USED AS THE MODE 7 ADDRESSING REGISTER. CMP INSTRUCTIONS ARE
: USED TO VERIFY PROPER ADDRESSING AND DATA.
:
:*****
:TEST 154 TEST MOV DESTINATION MODE 7
:*****

```

```

4707 013570 005212      TS154: INC      (R2)          ;UPDATE TEST NUMBER
4708 013572 022712 000154      CMP      #154,(R2)     ;SEQUENCE ERROR?
4709 013576 001053      BNE     TS155-10      ;BR TO ERROR HALT ON SEQ ERROR
4710 013600 005004      CLR     R4            ;R4=0
4711 013602 005014      CLR     (R4)          ;LOC.0=0
4712 013604 012700 000403      MOV     #403,R0       ;R0=403
4713 013610 012770 070707 177777  MOV     #70707,@-1(R0) ;TRY MOV W/DEST MODE 7
4714 013616 102402      BVS     MDM7A         ;BR. TO ERROR IF V-BIT SET
4715 013620 001401      BEQ     MDM7A         ;BR. TO ERROR IF Z-BIT SET
4716 013622 100004      BPL     MDM7B         ;
4717                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4718                                ;          CONDITIONAL BRANCH INST. AND <====
4719                                ;          REPLACE THE MOVE INSTRUCTION <====
4720                                ;          WHICH FOLLOWS W/ 765 <====
4721 013624      MDM7A:
4722 013624 012742 000343      MOV     #343,-(R2)    ;MOVE TO MAILBOX # ***** 343 *****
4723 013630 005242              INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
4724 013632 000000              HALT                    ;COND. CODES INCORRECT
4725 013634 022700 000403      MDM7B: CMP     #403,R0   ;CHECK DEST. REGISTER
4726 013640 001404              BEQ     MDM7C         ;
4727                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====

```



```

4728 ; ; ; CONDITIONAL BRANCH INST. AND <====
4729 ; ; ; REPLACE THE MOVE INSTRUCTION <====
4730 ; ; ; WHICH FOLLOWS W/ 756 <====
4731 013642 012742 000344 MOV #344,-(R2) ;MOVE TO MAILBOX # ***** 344 *****
4732 013646 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4733 013650 000000 HALT ;DEST. REGISTER INCORRECTLY ALTERED
4734 013652 022737 070707 000000 MDM7C: CMP #70707,@#0 ;CHECK DEST. DATA
4735 013660 001404 BEQ MDM7D
4736 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4737 ; ; ; CONDITIONAL BRANCH INST. AND <====
4738 ; ; ; REPLACE THE MOVE INSTRUCTION <====
4739 ; ; ; WHICH FOLLOWS W/ 746 <====
4740 013662 012742 000345 MOV #345,-(R2) ;MOVE TO MAILBOX # ***** 345 *****
4741 013666 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4742 013670 000000 HALT ;DEST. DATA INCORRECT
4743 013672 112770 107070 000001 MDM7D: MOVB #107070,@1(R0) ;TRY MOVB W/DEST MODE 7--ODD BYTE
4744 013700 022700 000403 CMP #403,R0 ;CHECK MODE 7 DEST. REG.
4745 013704 001404 BEQ MDM7E
4746 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4747 ; ; ; CONDITIONAL BRANCH INST. AND <====
4748 ; ; ; REPLACE THE MOVE INSTRUCTION <====
4749 ; ; ; WHICH FOLLOWS W/ 734 <
4750 013706 012742 000346 MOV #346,-(R2) ;MOVE TO MAILBOX # ***** 346 *****
4751 013712 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4752 013714 000000 HALT ;DEST. DATA INCORRECT
4753 013716 022737 034307 000000 MDM7E: CMP #34307,@#0 ;CHECK DEST. DATA
4754 013724 001404 BEQ TS155
4755 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
4756 ; ; ; CONDITIONAL BRANCH INST. AND <
4757 ; ; ; REPLACE THE MOVE INSTRUCTION <
4758 ; ; ; WHICH FOLLOWS W/ 724 <
4759 013726 012742 000347 MOV #347,-(R2) ;MOVE TO MAILBOX # ***** 347 *****
4760 013732 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4761 013734 000000 HALT ;DESTINATION DATA INCORRECT
4762 ; OR SEQUENCE ERROR

```

```

4763 ;*****
4764 ;
4765 ; THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.
4766 ; THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A
4767 ; TABLE OF OPERANDS. THE TABLE OF OPERANDS AND THE WORK LOCATION IS
4768 ; STORED FOLLOWING THE TEST CODE. A SERIES OF 5 DOP INSTRUCTIONS UTILIZES
4769 ; THE DATA IN THE TABLE TO CYCLE THE WORK LOCATION THRU A SET OF
4770 ; VALUE. THE DATA HAS BEEN CHOSEN TO INSURE THAT NO SINGLE ERROR WILL
4771 ; GO UNDETECTED. WORD AND BYTE INSTRUCTION ACCESSING BOTH EVEN AND
4772 ; ODD ADDRESSES ARE USED IN THE TEST. THE LISTING SHOWS THE
4773 ; EXPECTED INTERMEDIATE RESULT AS EACH INSTRUCTION IS EXECUTED.
4774 ;
4775 ;*****
4776 ;
4777 ; TEST 155 TEST MODE 4 W/ DOP INSTS.
4778 ;*****
4779 013736 005212 TS155: INC (R2) ;UPDATE TEST NUMBER
4780 013740 022712 000155 CMP #155,(R2) ;SEQUENCE ERROR?
4781 013744 001015 BNE DOP4 ;BR TO ERROR HALT ON SEQ ERROR
4782 013746 012700 014020 MOV #TBL1,R0 ;INITIALIZE R0
4783 013752 014037 014020 MOV -(R0),@#TBL1 ;TBL1=125252

```

4784	013756	064037	014020
4785	013762	144037	C14020
4786	013766	154037	014021
4787	013772	024037	014020
4788	013776	001411	

```

ADD -(R0),@#TBL1 ;TBL1=000377
BICB -(R0),@#TBL1 ;TBL1=000252
B,SB -(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 <--=

```

4793	014000		
4794	014000	012742	000350
4795	014004	005242	
4796	014006	000000	
4797			
4798			
4799	014010	125252	
4800	014012	052652	
4801	014014	053125	
4802	014016	125252	
4803	C14020	000000	

```

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

4804			
4805			
4806			
4807			
4808			
4809			
4810			
4811			
4812			
4813			
4814			
4815			

```

:*****
:
: 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.

4816			
4817	014022	005212	
4818	014024	022712	000156
4819	014030	001015	
4820	014032	012700	014106
4821	014036	015037	014020
4822	014042	065037	014020
4823	014046	145037	014020
4824	014052	155037	014021
4825	014056	025037	014020
4826	014062	001411	

```

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 <

```

4827			
4828			
4829			
4830			
4831	014064		
4832	014064	012742	000351
4833	014070	005242	
4834	014072	000000	
4835			
4836	014074	014010	
4837	014076	014012	
4838	014100	014013	
4839	014102	014014	

```

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

```

JKDB-D DCF11-AA CPU DIAG.  
JKDBD.P11 24-NOV-80 11:07

MACY11 30A(1052) 14-JAN-81 11:46 E 8  
1156 TEST MODE 5 W/ DOP INSTS. PAGE 95

484L 014104 014016

TBL2: TBL1-2

```

4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854 014106 005212
4855 014110 022712 000157
4856 014114 001022
4857 014116 012700 014014
4858 014122 016037 000002 014020
4859 014130 066037 000000 014020
4860 014136 146037 177777 014020
4861 014144 156037 177776 014021
4862 014152 026037 177774 014020
4863 014160 001404
4864
4865
4866
4867
4868 014162 012742 000352
4869 014166 005242
4870 014170 000000
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885 014172 005212
4886 014174 022712 000160
4887 014200 001022
4888 014202 012700 014100
4889 014206 017037 000004 014020
4890 014214 067037 000002 014020
4891 014222 147037 000000 014020
4892 014230 157037 177776 014021
4893 014236 027037 177774 014020
4894 014244 001404
4895
4896

```

```

:*****
:
: 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. RO 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,RO ;INITIALIZE RO
MOV 2(RO),@#TBL1 ;TBL1=125252
ADD 0(RO),@#TBL1 ;TBL1=000377
BICB -1(RO),@#TBL1 ;TBL1=000252
BISB -2(RO),@#TBL1+1 ;TBL1=125252
CMP -4(RO),@#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.
: RO 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,RO ;INITIALIZE RO
MOV @4(RO),@#TBL1 ;TBL1=125252
ADD @2(RO),@#TBL1 ;TBL1=000377
BICB @0(RO),@#TBL1 ;TBL1=000252
BISB @-2(RO),@#TBL1+1 ;TBL1=125252
CMP @-4(RO),@#TBL1 ;CHECK RESULT
BEQ TS161

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

```

```

4897                                     :
4898                                     :
4899 014246 012742 000353             MOV #353,-(R2)           :
4900 014252 005242                   INC -(R2)              :
4901 014254 000000                   HALT                   :
4902                                     :
4903                                     :
4904                                     :
4905                                     :
4906                                     :
4907                                     :
4908                                     :
4909                                     :
4910                                     :
4911                                     :
4912                                     :
4913                                     :

```

```

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.
: RO 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.
:*****

```

```

4914 014256 005212
4915 014260 022712 000161
4916 014264 001026
4917 014266 012700 125252
4918 014272 000261
4919 014274 006100
4920 014276 102004
4921 014300 103003
4922 014302 022700 052525
4923 014306 001404
4924
4925
4926
4927

```

```

:TEST 161 TEST ROTATE INSTRUCTIONS OF MODE 0
:*****
TS161: INC (R2) ;UPDATE TEST NUMBER
CMP #161,(R2) ;SEQUENCE ERROR?
BNE TS162-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #125252,R0 ;INITIALIZE DATA
SEC ;SET C-BIT
ROL RO ;TRY ROL W/ MODE 0
BVC R0TOA ;CC=0011
R0TOA
BCC R0TOA
CMP #052525,R0 ;CHECK DATA
BEQ R0TOB
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
; CONDITIONAL BRANCH INST. AND <-
; REPLACE THE MOVE INSTRUCTION <- --
; WHICH FOLLOWS W/ 766 <- -

```

```

4928 014310
4929 014310 012742 000354
4930 014314 005242
4931 014316 000000
4932 014320 012700 125252
4933 014324 000261
4934 014326 106100
4935 014330 102004
4936 014332 103003
4937 014334 022700 125125
4938 014340 001404
4939
4940
4941
4942
4943 014342
4944 014342 012742 000355
4945 014346 005242
4946 014350 000000
4947

```

```

R0TOA: MOV #354,-(R2) ;MOVE TO MAILBOX # ***** 354 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ROL MODE 0 FAILED
R0TOB: MOV #125252,R0 ;INITIALIZE DATA
SEC ;SET C-BIT
ROLB RO ;TRY ROL W/ MODE 0 EVEN BYTE
BVC R0TOC ;CC=0011
R0TOC
BCC R0TOC
CMP #125125,R0 ;CHECK DATA
BEQ TS162
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
; CONDITIONAL BRANCH INST. AND <
; REPLACE THE MOVE INSTRUCTION <
; WHICH FOLLOWS W/ 751 <
R0TOC: MOV #355,-(R2) ;MOVE TO MAILBOX # ***** 355 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ROLB MODE 0 FAILED
; OR SEQUENCE ERROR

```

4948  
4949  
4950  
4951  
4952  
4953  
4954  
4955  
4956  
4957  
4958  
4959  
4960  
4961  
4962  
4963  
4964  
4965  
4966  
4967  
4968  
4969  
4970  
4971  
4972  
4973  
4974  
4975  
4976  
4977  
4978  
4979  
4980  
4981  
4982  
4983  
4984  
4985  
4986  
4987  
4988  
4989  
4990  
4991  
4992  
4993  
4994  
4995  
4996  
4997  
4998  
4999  
5000  
5001  
5002  
5003

014352 005212  
014354 022712 000162  
014360 001051  
014362 005000  
014364 012710 052525  
014370 000241  
014372 006110  
014374 102005  
014376 103404  
014400 023727 000000 125252  
014406 001404  
  
014410  
014410 012742 000356  
014414 005242  
014416 000000  
014420 000261  
014422 012710 125252  
014426 106110  
014430 102005  
014432 103004  
014434 022737 125125 000000  
014442 001404  
  
014444  
014444 012742 000357  
014450 005242  
014452 000000  
014454 012710 125252  
014460 005000  
014462 005200  
014464 000261  
014466 106110  
014470 102005  
014472 103004  
014474 022737 052652 000000  
014502 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 ROLOB W/ MODE 1 EVEN BYTE  
BVC ROT1C ;CC=1011  
BCL 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 ROLOB W/ MODE 1 ODD BYTE  
BVC ROT1E ;CC=0011  
BCC ROT1E  
CMP #052652,@#0 ;CHECK DATA  
BEQ TS163
```



```
5060 014622 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5061 014624 000000          HALT                    ;ROLB W/ MODE 2 EVEN BYTE FAILED
5062 014626 005000          ROT2D: CLR      RO      ;POINT TO LOC 0
5063 014630 012710 004040    MOV      #4040,(RO)    ;INITIALIZE DATA
5064 014634 005200          INC      RO            ;POINT TO ODD BYTE OF DATA
5065 014636 000261          SEC                    ;SET C-BIT
5066 014640 106120          ROLB     (RO)+         ;TRY ROL W/ MODE 2 ODD BYTE
5067 014642 103407          BCS     ROT2E         ;CHECK C-BIT
5068 014644 022737 010440 000000' CMP      #10440,a#0    ;CHECK DATA
5069 014652 001003          BNE     ROT2E         ;BRANCH IF DATA INCORRECT
5070 014654 005300          DEC      RO            ;CHECK RO
5071 014656 005300          DEC      RO
5072 014660 001404          BEQ     TS164
5073
5074
5075
5076
5077 014662
5078 014662 012742 000363    ROT2E: MOV      #363,-(R2) ;MOVE TO MAILBOX # ***** 363 *****
5079 014666 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
5080 014670 000000          HALT                    ;ROLB W/ MODE 2 ODD BYTE FAILED
5081
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--
; CONDITIONAL BRANCH INST. AND <--
; REPLACE THE MOVE INSTRUCTION <----
; WHICH FOLLOWS W/ 720 <---
```



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  
5131  
5132  
5133  
5134  
5135  
5136  
5137

014672 005212  
014674 022712 000164  
014700 001051  
014702 012737 052525 000000  
014710 000261  
014712 006137 000000  
014716 103404  
014720 022737 125253 000000  
014726 001404  
  
014730  
014730 012742 000364  
014734 005242  
014736 000000  
014740 012737 125252 000000  
014746 000241  
014750 106137 000000  
014754 103004  
014756 023727 000000 125124 4\$:  
014764 001404  
  
014766  
014766 012742 000365  
014772 005242  
014774 000000  
014776 012737 125252 000000  
015004 000261  
015006 106137 000001  
015012 103004  
015014 022737 052652 000000  
015022 001404  
  
015024  
015024 012742 000366  
015030 005242  
015032 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 3. 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
      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
      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
```

: OR SEQUENCE ERROR

5138  
5139  
5140  
5141  
5142  
5143  
5144  
5145  
5146  
5147  
5148  
5149  
5150  
5151  
5152  
5153  
5154  
5155  
5156  
5157  
5158  
5159  
5160  
5161  
5162  
5163  
5164  
5165  
5166  
5167  
5168  
5169  
5170  
5171  
5172  
5173  
5174  
5175  
5176  
5177  
5178  
5179  
5180  
5181  
5182  
5183  
5184  
5185  
5186  
5187  
5188  
5189  
5190  
5191  
5192  
5193

015034 005212  
015036 022712 000165  
015042 001016  
015044 012737 070707 000000  
015052 012700 000002  
015056 000261  
015060 006140  
015062 103406  
015064 022737 161617 000000  
015072 001002  
015074 005700  
015076 001404  
  
015100  
015100 012742 000367  
015104 005242  
015106 000000  
  
015110  
015112 022712 000166  
015116 001021  
015120 012737 015172 000000  
015126 012700 000002  
015132 012767 107070 000032  
015140 000241  
015142 006150

```
*****  
: 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
```

```
5194 015144 103006          BCC      ROT5      :CHECK C-BIT
5195 015146 022737 016160 015172  CMP      #016160,@#ROTX :CHECK DATA
5196 015154 001002          BNE      ROT5      :BRANCH IF DATA INCORRECT
5197 015156 005700          TST      R0        :CHECK MODE 5 REGISTER
5198 015160 001405          BEQ      TS167
5199
5200
5201
5202
5203 015162
5204 015162 012742 000370  ROT5:  MOV      #370,-(R2)  :MOVE TO MAILBOX # ***** 370 *****
5205 015166 005242          INC      -(R2)      :SET MSGTYP TO FATAL ERROR
5206 015170 000000          HALT
5207
5208 015172 000000  ROTX:  0
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220 015174 005212          TS167:  INC      (R2)      :UPDATE TEST NUMBER
5221 015176 022712 000167  CMP      #167,(R2)  :SEQUENCE ERROR?
5222 015202 001013          BNE      TS170-10   :BR TO ERROR HALT ON SEQ ERROR
5223 015204 012737 125252 015172  MOV      #125252,@#ROTX :INITIALIZE DATA
5224 015212 000261          SEC
5225 015214 006167 177752  ROL      ROTX      :SET C-BIT
5226 015220 103004          ROL      ROT6      :TRY ROL W/ MODE 6
5227 015222 022737 052525 015172  BCC      ROT6      :CHECK C-BIT
5228 015230 001404          CMP      #52525,@#ROTX :CHECK DATA
5229
5230
5231
5232
5233 015232
5234 015232 012742 000371  ROT6:  MOV      #371,-(R2)  :MOVE TO MAILBOX # ***** 371 *****
5235 015236 005242          INC      -(R2)      :SET MSGTYP TO FATAL ERROR
5236 015240 000000          HALT
5237
```

\*\*\*\*\*  
: 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).  
\*\*\*\*\*  
: TEST 167 TEST MODE 6 W/ ROTATE INSTRUCTIONS  
\*\*\*\*\*

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

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



CKDB-D DCF11-AA CPU DIAG.  
CKDBD.P11 24-NOV-80 11:07

MACY11 30A(1052) 14-JAN-81 11:46 PAGE 105  
T171 TEST MODE 0 W/ SWAB INST.

SEQ 0105

5294 015350 022700 000377  
5295 015354 001404

SBO: CMP #377,R0 ;CHECK RESULT  
BEQ TS172

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

5300 015356 012742 000374  
5301 015362 005242  
5302 015364 000000

MOV #374,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 374 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF SWAB MODE 0 FAILED  
; OR SEQUENCE ERROR

5303  
5304  
5305  
5306

\*\*\*\*\*  
: THIS TEST VERIFIES MODE 1 SWAB INSTRUCTION. THE TEST  
: PATTERN IS MOVED TO LOC 0. R0 IS CLEARED AND USED AS THE ADDRESSING  
: REGISTER IN THE MODE 1 SWAB. THE DATA RESULTS ARE CHECKED WITH  
: A COMPARE.  
: \*\*\*\*\*

5307  
5308  
5309  
5310  
5311

: TEST 172 TEST MODE 1 W/ SWAB INST  
: \*\*\*\*\*

5312  
5313  
5314

5315 015366 005212  
5316 015370 022712 000172  
5317 015374 001011  
5318 015376 012737 125652 000000

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 R0 ;R0=0  
SWAB (R0) ;TRY SWAB MODE 1  
CMP #125253,@#0 ;CHECK RESULT  
BEQ TS173

5319 015404 005000  
5320 015406 000310  
5321 015410 022737 125253 000000  
5322 015416 001404

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

5323  
5324  
5325  
5326

5327 015420 012742 000375  
5328 015424 005242  
5329 015426 000000

MOV #375,-(R2) ;MOVE TO MAILBOX # \*\*\*\*\* 375 \*\*\*\*\*  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;RESULT OF SWAB MODE 1 FAILED  
; OR SEQUENCE ERROR

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  
5380  
5381  
5382  
5383  
5384  
5385  
5386

015430 005212  
015432 022712 000173  
015436 001020  
015440 012737 125152 000000  
015446 005000  
015450 000320  
015452 022737 065252 000000  
015460 001404  
  
015462 012742 000376  
015466 005242  
015470 000000  
015472 162700 000002  
015476 001404  
  
015500 012742 000377  
015504 005242  
015506 000000  
  
015510 005212  
015512 022712 000174  
015516 001011  
015520 012737 000377 000000  
015526 000337 000000  
015532 022737 177400 000000  
015540 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 <--- -
      MOV #376,-(R2) ;MOVE TO MAILBOX # ***** 376 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;RESULT OF SWAB MODE 0 FAILED
SB2: SUB #2,R0 ;CHECK EFFECT OF REG.
      BEQ TS174
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--==
; CONDITIONAL BRANCH INST. AND <- -
; REPLACE THE MOVE INSTRUCTION <- --=
; WHICH FOLLOWS W/ 757 <- -
      MOV #377,-(R2) ;MOVE TO MAILBOX # ***** 377 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;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
```

CJKDB-D DCF11-AA CPU DIAG.  
CJKDBD.P11 24-NOV-80 11:07

MACY11 3CA(1052) 14-JAN-81 11:46 PAGE 107  
T174 TEST MODE 3 W/SWAB INST.

D 9

SEQ 0107

5387  
5388  
5389  
5390  
5391 015542 012742 000400  
5392 015546 005242  
5393 015550 000000  
5394

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





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  
5468  
5469  
5470  
5471  
5472  
5473  
5474

015632 005212  
015634 022712 000176  
015640 001021  
015642 012700 015720  
015646 012767 125125 000040  
015654 000350  
015656 022767 052652 000030  
015664 001404  
  
015666 012742 000403  
015672 005242  
015674 000000  
015676 020027 015716  
015702 001406  
  
015704  
015704 012742 000404  
015710 005242  
015712 000000  
  
015714 000000  
015716 015714

```
*****
: 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 RO IS SET TO TWO PLUS THE ADDRESS OF SB5XAD. FOLLOWING
: THE MODE 5 SWAB SB5X IS CHECKED FOR THE PROPER DATA. RO IS
: CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.
*****
: TEST 176 TEST MODE 5 W/ SWAB INST.
*****
S176: INC (R2) ;UPDATE TEST NUMBER
      CMP #176,(R2) ;SEQUENCE ERROR?
      BNE SB5 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #SB5XAD+2,RO ;SET UP POINTER TO WORK LOCATION
      MOV #125125,SB5X ;MOVE PATTERN TO WORK LOCATION
      SWAB @-(RO) ;TRY SWAB MODE 5
      CMP #52652,SB5X ;CHECK RESULT
      BEQ SB5A
; TO SCOPE: CLEAR THE RIGHT BYTE OF TH; <
; 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 RO,#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
; WORK LOCATION
SB5X: 0
SB5XAD: SB5X
```

5475  
5476  
5477  
5478  
5479  
5480  
5481  
5482  
5483  
5484  
5485  
5486  
5487  
5488 015720 005212  
5489 015722 022712 000177  
5490 015726 001013  
5491 015730 012767 125125 000030  
5492 015736 012700 015760  
5493 015742 000360 000006  
5494 015746 022760 052652 000006  
5495 015754 001405  
5496  
5497  
5498  
5499  
5500 015756  
5501 015756 012742 000405  
5502 015762 005242  
5503 015764 000000  
5504  
5505 015766 000000  
5506

```
*****
:
: 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
```

5507  
5508  
5509  
5510  
5511  
5512  
5513  
5514  
5515  
5516  
5517  
5518  
5519  
5520  
5521 015770 005212  
5522 015772 022712 000200  
5523 015776 001013  
5524 016000 012767 177400 000030  
5525 016006 012700 015746  
5526 016012 000370 000072  
5527 016016 027027 000072 000377  
5528 016024 001406  
5529  
5530  
5531  
5532  
5533 016026  
5534 016026 012742 000406  
5535 016032 005242  
5536 016034 000000  
5537  
5538 016036 000000  
5539 016040 016036  
5540

```
*****
:
:   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. RO 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,RO ;MOVE OFFSET POINTER TO RO
        SWAB   @72(RO)       ;TRY SWAB MODE 7
        CMP     @72(RO),#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)
        HALT
:
SB7x:   0
SB7xAD: SB7X ;POINTER TO WORK LOCATION
```



```

5597 016112 012742 000410      MOV      #410,-(R2)      ;MOVE TO MAILBOX # ***** 410 *****
5598 016116 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5599 016120 000000              HALT                    ;SHOULD BE HERE FROM JMP MODE 2 ONLY
5600 016122 012700 016134      JMP3B:  MOV      #I JMP4,R0      ;POINT R0 TO INDIRECT JMP ADDR.
5601 016126 005267 000252      INC      JMPSEQ          ;UPDATE SEQUENCE CHECKER
5602 016132 000130              JMP      @ (R0)+        ;TRY JMP MODE 3
5603 016134 016166      I JMP4:  JMP4          ;ADDRESS INDIRECT JUMP
5604
5605 016136 005767 000242      JMP2:   TST      JMPSEQ          ;CHECK THAT JMPS ARE IN SEQUENCE: JMPSEQ=0?
5606 016142 001404              BEQ      JMP2A          ;
5607
5608
5609
5610
5611 016144 012742 000411      MOV      #411,-(R2)      ;MOVE TO MAILBOX # ***** 411 *****
5612 016150 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5613 016152 000000              HALT                    ;SHOULD BE HERE FROM JMP MODE 1 ONLY
5614 016154 005267 000224      JMP2A:  INC      JMPSEQ          ;UPDATE SEQUENCE CHECKER
5615 016160 012700 016064      MOV      #JMP3,R0        ;SET R0=JUMP TARGET
5616 016164 000120              JMP      (R0)+          ;TRY A JUMP MODE 2 TO "JMP3"
5617 016166 022700 016136      JMP4:   CMP      #I JMP4+2,R0     ;CHECK RESULT OF REGISTER IN MODE 3 JUMP
5618 016172 001404              BEQ      JMP4A          ;
5619
5620
5621
5622
5623 016174 012742 000412      MOV      #412,-(R2)      ;MOVE TO MAILBOX # ***** 412 *****
5624 016200 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5625 016202 000000              HALT                    ;REGISTER VALUE AFTER MODE 3 JUMP INCORRECT
5626 016204 022767 000002 000172  JMP4A:  CMP      #2,JMPSEQ          ;CHECK JUMP SEQUENCE: JMPSEQ=2?
5627 016212 001404              BEQ      JMP4B          ;
5628
5629
5630
5631
5632 016214 012742 000413      MOV      #413,-(R2)      ;MOVE TO MAILBOX # ***** 413 *****
5633 016220 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5634 016222 000000              HALT                    ;SHOULD BE ONLY FROM MODE 3 JUMP
5635 016224 012700 016274      JMP4B:  MOV      #JMP5+2,R0     ;SET UP POINTER TO JUMP TARGET
5636 016230 005267 000150      INC      JMPSEQ          ;UPDATE SEQUENCE CHECKER
5637 016234 000140              JMP      -(R0)          ;TRY JUMP MODE 4 TO "JMP4"
5638
5639 016236 022767 000004 000140  JMP6:   CMP      #4,JMPSEQ          ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?
5640 016244 001404              BEQ      JMP6A          ;
5641
5642
5643
5644
5645 016246 012742 000414      MOV      #414,-(R2)      ;MOVE TO MAILBOX # ***** 414 *****
5646 016252 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5647 016254 000000              HALT                    ;SHOULD BE HERE ONLY FROM MODE 5 JUMP
5648 016256 012700 016724      JMP6A:  MOV      #JMP7+376,R0    ;SET UP OFFSET POINTER TO JUMP TARGET
5649 016262 005267 000116      INC      JMPSEQ          ;UPDATE JUMP SEQUENCE
5650 016266 000160 177402      JMP      -376(R0)       ;TRY MODE 6 JUMP
5651
5652 016272 022767 000003 000104  JMP5:   CMP      #3,JMPSEQ          ;CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?

```

```

5653 016300 001404          BEQ      JMP5A          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5654                                     ;          CONDITIONAL BRANCH INST. AND <====
5655                                     ;          REPLACE THE MOVE INSTRUCTION <====
5656                                     ;          WHICH FOLLOWS W/ 663          <====
5657                                     ;
5658 016302 012742 000415    MOV      #415,-(R2)      ;MOVE TO MAILBOX # ***** 415 *****
5659 016306 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5660 016310 000000          HALT                     ;SHOULD ONLY BE HERE FROM MODE 4 JUMP
5661 016312 012700 016326    JMP5A:  MOV      #IJMP5+2,R0 ;SET UP POINTER TO INDIRECT JUMP ADDR.
5662 016316 005267 000062          INC      JMPSEQ         ;UPDATE JUMP SEQUENCE
5663 016322 000150          JMP      @-(R0)         ;TRY JUMP MODE 5 TO 'JMP6'
5664 016324 016236          IJMP5: JMP6           ;INDIRECT ADDRESS POINTER
5665
5666 016326 022767 000005 000050 JMP7:   CMP      #5,JMPSEQ   ;CHECK JUMPS IN SEQUENCE: JMPSEQ=5?
5667 016334 001404          BEQ      JMP7A          ;
5668                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5669                                     ;          CONDITIONAL BRANCH INST. AND <====
5670                                     ;          REPLACE THE MOVE INSTRUCTION <====
5671                                     ;          WHICH FOLLOWS W/ 645          <====
5672 016336 012742 000416    MOV      #416,-(R2)      ;MOVE TO MAILBOX # ***** 416 *****
5673 016342 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5674 016344 000000          HALT                     ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5675 016346 012700 016372    JMP7A:  MOV      #IJMP+10,R0 ;SET UP OFFSET POINTER TO INDIRECT ADDR.
5676 016352 005267 000026          INC      JMPSEQ         ;UPDATE JUMP SEQUENCE
5677 016356 000170          JMP      @-10(R0)       ;TRY MODE 7 JUMP
5678 016362 016364          IJMP:   JMPCK          ;INDIRECT ADDRESS
5679
5680 016364 026727 000014 000006 JMPCK:  CMP      JMPSEQ,#6   ;CHECK JUMPS IN SEQUENCE: JMPSEQ
5681 016372 001405          BEQ      TS202          ;
5682                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
5683                                     ;          CONDITIONAL BRANCH INST. AND <
5684                                     ;          REPLACE THE MOVE INSTRUCTION <--
5685                                     ;          WHICH FOLLOWS W/ 626          <-
5686 016374 012742 000417    MOV      #417,-(R2)      ;MOVE TO MAILBOX # ***** 417 *****
5687 016400 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5688 016402 000000          HALT                     ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5689                                     ; OR SEQUENCE ERROR
5690 016404 000000          JMPSEQ: 0

```

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  
5740  
5741  
5742  
5743  
5744  
5745  
5746

016406 005212  
016410 022712 000202  
016414 001001  
016416 000402  
016420 000137 017054  
016424 012706 001000  
016430 012700 016536  
016434 005037 017034  
016440 005001  
016442 005101  
016444 004110  
016446  
016446 012742 000420  
016452 005242  
016454 000000  
016456 022737 000001 017034  
016464 001014  
016466 020127 016620  
016472 001011  
016474 022706 000776  
016500 001006  
016502 022716 125252  
016506 001003  
016510 022700 016460  
016514 001404  
016516  
016516 012742 000421  
016522 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

```

5747	016524	000000			HALT				:JSR MODE 3 MALFUNCTIONED	
5748	016526	005237	017034		ISR3B: INC	@#JSRSEQ			:UPDATE SEQUENCE CHECKER	
5749	016532	004137	016620		JSR	R1,@#JSR4			:TRY JSR MODE 4	
5750										
5751	016536	005737	017034		JSR2: TST	@#JSRSEQ			:CHECK SEQUENCE: JSRSEQ=0?	
5752	016542	001011			BNE	JSR2A			:BRANCH IF OUT OF SEQUENCE	
5753	016544	020127	016446		CMP	R1,#JSR1A			:PROPER PC SAVED?	
5754	016550	001006			BNE	JSR2A			:BRANCH IF PC WRONG	
5755	016552	022706	000776		CMP	#STBOT-2,R6			:R6 DECREMENT?	
5756	016556	001003			BNE	JSR2A			:BRANCH IF R6 IS INCORRECT	
5757	016560	021627	177777		CMP	(R6),#-1			:REGISTER SAVED?	
5758	016564	001404			BEQ	JSR2B				
5759									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<-- -
5760									CONDITIONAL BRANCH INST. AND	<== -
5761									REPLACE THE MOVE INSTRUCTION	<=== =
5762									WHICH FOLLOWS W/ 713	<====-
5763	016566				JSR2A: MOV	#422,-(R2)			:MOVE TO MAILBOX # ***** 422 *****	
5764	016566	012742	000422		INC	-(R2)			:SET MSGTYP TO FATAL ERROR	
5765	016572	005242			HALT				:JSR MODE 1 MALFUNCTIONED	
5766	016574	000000			JSR2B: MOV	#STBOT,R6			:INITIALIZE R6	
5767	016576	012706	001000		MOV	#125252,R1			:INITIALIZE R1	
5768	016602	012701	125252		INC	@#JSRSEQ			:UPDATE SEQUENCE CHECKER	
5769	016606	005237	017034		MOV	#JSR3,R0			:SET TARGET ADDRESS	
5770	016612	012700	016456		JSR	R1,(R0)+			:TRY JSR MODE 2	
5771	016616	004120								
5772										
5773	016620	022737	000002	017034	JSR4: CMP	#2,@#JSRSEQ			:CHECK SEQUENCE: JSRSEQ=2?	
5774	016626	001003			BNE	JSR4A			:BRANCH IF OUT OF SEQUENCE	
5775	016630	022701	016536		CMP	#JSR2,R1			:PROPER PC SAVED?	
5776	016634	001404			BEQ	JSR4B				
5777									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
5778									CONDITIONAL BRANCH INST. AND	<===
5779									REPLACE THE MOVE INSTRUCTION	<== -
5780									WHICH FOLLOWS W/ 667	<====-
5781	016636				JSR4A: MOV	#423,-(R2)			:MOVE TO MAILBOX # ***** 423 *****	
5782	016636	012742	000423		INC	-(R2)			:SET MSGTYP TO FATAL ERROR	
5783	016642	005242			HALT				:JSR MODE 3 MALFUNCTIONED	
5784	016644	000000			JSR4B: INC	@#JSRSEQ			:UPDATE SEQUENCE CHECKER	
5785	016646	005237	017034		MOV	#JSR5+2,R0			:SET TARGET ADDRESS	
5786	016652	012700	016726		JSR	R1,-(R0)			:TRY JSR MODE 4	
5787	016656	004140								
5788										
5789	016660	022767	000004	000146	JSR6: CMP	#4,JSRSEQ			:CHECK SEQUENCE: JSRSEQ=4?	
5790	016666	001006			BNE	JSR6A			:BRANCH IF OUT OF SEQUENCE	
5791	016670	022701	016772		CMP	#JSR7,R1			:PROPER PC SAVED?	
5792	016674	001003			BNE	JSR6A			:BRANCH IF PC WRONG	
5793	016676	022700	017030		CMP	#JSR6AD,R0			:MODE 5 REGISTER CORRECT?	
5794	016702	001404			BEQ	JSR6B				
5795									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
5796									CONDITIONAL BRANCH INST. AND	<====
5797									REPLACE THE MOVE INSTRUCTION	<====
5798									WHICH FOLLOWS W/ 644	<====
5799	016704				JSR6A: MOV	#424,-(R2)			:MOVE TO MAILBOX # ***** 424 *****	
5800	016704	012742	000424		INC	-(R2)			:SET MSGTYP TO FATAL ERROR	
5801	016710	005242			HALT				:JSR MODE 5 FAILED	
5802	016712	000000								





5857  
 5858  
 5859  
 5860  
 5861  
 5862  
 5863  
 5864  
 5865  
 5866  
 5867 017064 005212  
 5868 017066 022712 000203  
 5869 017072 001016  
 5870 017074 012706 001000  
 5871 017100 012746 052525  
 5872 017104 012700 017122  
 5873 017110 000200  
 5874  
 5875  
 5876 017112 012742 000430  
 5877 017116 005242  
 5878 017120 000000  
 5879 017122 022700 052525  
 5880 017126 001404  
 5881  
 5882  
 5883  
 5884  
 5885 017130 012742 000431  
 5886 017134 005242  
 5887 017136 000000  
 5888

```

:*****
:
:   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     TS204-0     ;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

```



```

5945 017234 012700 100001      MOV      #100001,R0
5946 017240 000277              SCC                      ;CL=0110
5947 017242 000251      +CLN!CLC
5948 017244 032700 100000      BIT      #100000,R0      ;CC=1000
5949 017250 101402      BLOS    BIT1
5950 017252 102401      BVS     BIT1
5951 017254 100404      BMI     BIT2
5952              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5953              ; CONDITIONAL BRANCH INST. AND <====
5954              ; REPLACE THE MOVE INSTRUCTION <====
5955              ; WHICH FOLLOWS W/ 766 <====
5956 017256      BIT1:
5957 017256 012742 000434      MOV      #434,-(R2)      ;MOVE TO MAILBOX # ***** 434 *****
5958 017262 005242      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5959 017264 000000      HALT    ;BIT DID NOT SET CC'S CORRECTLY
5960
5961 017266 000277      BIT2:  SCC              ;CC=1011
5962 017270 000244      CLZ
5963 017272 032700 077776      BIT      #77776,R0      ;CC=0101
5964 017276 101002      BHI    BIT3
5965 017300 102401      BVS    BIT3
5966 017302 100004      BPL    TS206
5967              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
5968              ; CONDITIONAL BRANCH INST. AND <
5969              ; REPLACE THE MOVE INSTRUCTION <
5970              ; WHICH FOLLOWS W/ 753 <
5971 017304      BIT3:
5972 017304 012742 000435      MOV      #435,-(R2)      ;MOVE TO MAILBOX # ***** 435 *****
5973 017310 005242      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5974 017312 000000      HALT    ;BIT DID NOT SET CC'S CORRECTLY
5975              ; OR SEQUENCE ERROR
5976
5977      ;*****
5978      ;TEST 206      TEST BIC INSTRUCTION
5979      ;*****
5979 017314 005212      TS206: INC      (R2)           ;UPDATE TEST NUMBER
5980 017316 022712 000206      CMP     #206,(R2)       ;SEQUENCE ERROR?
5981 017322 001024      BNE     TS207-10       ;BR TO ERROR HALT ON SEQ ERROR
5982 017324 012700 177777      MOV     #177777,R0
5983 017330 000277      SCC                      ;CC=0110
5984 017332 000251      +CLN!CLC
5985 017334 042700 077777      BIC     #77777,R0      ;CC=1000
5986 017340 101402      BLOS    BIC1
5987 017342 102401      BVS     BIC1
5988 017344 100404      BMI     BIC2
5989              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
5990              ; CONDITIONAL BRANCH INST. AND <
5991              ; REPLACE THE MOVE INSTRUCTION <
5992              ; WHICH FOLLOWS W/ 766 <
5993 017346      BIC1:
5994 017346 012742 000436      MOV      #436,-(R2)      ;MOVE TO MAILBOX # ***** 436 *****
5995 017352 005242      INC     -(R2)           ;SET MSGTYP TO FATAL ERROR
5996 017354 000000      HALT    ;BIC DID NOT SET CC'S CORRECTLY
5997 017356 000277      BIC2:  SCC              ;CC=1011
5998 017360 000244      CLZ
5999 017362 042700 100000      BIC     #100000,R0      ;CC=0101
6000 017366 101002      BHI    BIC3

```

6001 017370 102401  
6002 017372 100004  
6003  
6004  
6005  
6006  
6007 017374  
6008 017374 012742 000437  
6009 017400 005242  
6010 017402 000000  
6011  
6012  
6013  
6014  
6015 017404 005212  
6016 017406 022712 000207  
6017 017412 001025  
6018 017414 005000  
6019 017416 000277  
6020 017420 000251  
6021 017422 052700 000000  
6022 017426 103403  
6023 017430 102402  
6024 017432 100401  
6025 017434 001404  
6026  
6027  
6028  
6029  
6030 017436  
6031 017436 012742 000440  
6032 017442 005242  
6033 017444 000000  
6034 017446 000277  
6035 017450 000250  
6036 017452 052700 177777  
6037 017456 103003  
6038 017460 102402  
6039 017462 001401  
6040 017464 100404  
6041  
6042  
6043  
6044  
6045 017466  
6046 017466 012742 000441  
6047 017472 005242  
6048 017474 000000  
6049

BVS BIC3  
BPL TS207  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===  
: CONDITIONAL BRANCH INST. AND < ==  
: REPLACE THE MOVE INSTRUCTION < - =  
: WHICH FOLLOWS W/ 753 < ---  
BIC3: MOV #437, -(R2) :MOVE TO MAILBOX # \*\*\*\*\* 437 \*\*\*\*\*  
INC -(R2) :SET MSGTYP TO FATAL ERROR  
HALT :BIC DID NOT SET CC'S CORRECTLY  
: OR SEQUENCE ERROR  
:\*\*\*\*\*  
:TEST 207 TEST BIS INSTRUCTION  
:\*\*\*\*\*  
TS207: INC (R2) :UPDATE TEST NUMBER  
CMP #207, (R2) :SEQUENCE ERROR?  
BNE TS210-10 :BR TO ERROR HALT ON SEQ ERROR  
CLR R0 :R0=0  
SCC :CC=1010  
+CLN!CLC  
BIS #0, R0 :CC=0100 R0=0  
BCS BIS1  
BVS BIS1  
BMI BIS1  
BEQ BIS2  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <  
: CONDITIONAL BRANCH INST. AND <-  
: REPLACE THE MOVE INSTRUCTION <  
: WHICH FOLLOWS W/ 766 <  
BIS1: MOV #440, -(R2) :MOVE TO MAILBOX # \*\*\*\*\* 440 \*\*\*\*\*  
INC -(R2) :SET MSGTYP TO FATAL ERROR  
HALT :BIS DID NOT SET CC'S CORRECTLY  
BIS2: SCC :CC=0111  
CLN  
BIS #177777, R0 :CC=1001  
BCL BIS3  
BVS BIS3  
BEQ BIS3  
BMI TS210  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <  
: CONDITIONAL BRANCH INST. AND <  
: REPLACE THE MOVE INSTRUCTION <  
: WHICH FOLLOWS W/ 752 <  
BIS3: MOV #441, -(R2) :MOVE TO MAILBOX # \*\*\*\*\* 441 \*\*\*\*\*  
INC -(R2) :SET MSGTYP TO FATAL ERROR  
HALT :BIS DID NOT SET CC'S CORRECTLY  
: OR SEQUENCE ERROR

6050  
6051  
6052  
6053  
6054  
6055  
6056  
6057  
6058  
6059  
6060  
6061  
6062  
6063  
6064  
6065 017476 005212  
6066 017500 022712 000210  
6067 017504 001037  
6068 017506 012700 077777  
6069 017512 000257  
6070 017514 000264  
6071 017516 005200  
6072 017520 101402  
6073 017522 100001  
6074 017524 102404  
6075  
6076  
6077  
6078  
6079 017526  
6080 017526 012742 000442  
6081 017532 005242  
6082 017534 000000  
6083 017536 052700 077777  
6084 017542 000261  
6085 017544 000244  
6086 017546 005200  
6087 017550 100403  
6088 017552 102402  
6089 017554 103001  
6090 017556 001404  
6091  
6092  
6093  
6094  
6095 017560  
6096 017560 012742 000443  
6097 017564 005242  
6098 017566 000000  
6099  
6100 017570 000277  
6101 017572 000241  
6102 017574 005200  
6103 017576 101402  
6104 017600 100401  
6105 017602 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  
INL 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 MOVF 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
```



6162 017714 012742 000447  
6163 017720 005242  
6164 017722 000000  
6165 017724 042700 077777  
6166 017730 000277  
6167 017732 000252  
6168 017734 005300  
6169 017736 100403  
6170 017740 001402  
6171 017742 102001  
6172 017744 103404  
6173  
6174  
6175  
6176  
6177 017746  
6178 017746 012742 000450  
6179 017752 005242  
6180 017754 000000  
6181  
6182

DEC6: MOV #447,-(R2) ;MOVE TO MAILBOX # ..... 447 .....  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEC DID NOT SET CC'S CORRECTLY  
BIC #77777,R0 ;RO=100000  
SCC ;CC=0101  
\*(LN:CLV  
DEC R0 ;CC=1011 RO=77777  
BMI DEC7 ;CC=0011  
BEQ DEC7  
BVC DE:7  
BCS TS212

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

DEC7: MOV #450,-(R2) ;MOVE TO MAILBOX # ..... 450 .....  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DEC DID NOT SET CC'S CORRECTLY  
; OR SEQUENCE ERROR



6183  
6184  
6185  
6186  
6187  
6188  
6189  
6190  
6191  
6192  
6193  
6194  
6195  
6196 017756 005212  
6197 017760 022712 000212  
6198 017764 001007  
6199 017766 000277  
6200 017770 000244  
6201 017772 005000  
6202 017774 100403  
6203 017776 102402  
6204 020000 103401  
6205 020002 001404  
6206  
6207  
6208  
6209  
6210 020004  
6211 020004 012742 000451  
6212 020010 005242  
6213 020012 000000  
6214  
6215  
6216  
6217  
6218  
6219 020014 005212  
6220 020016 022712 000213  
6221 020022 001022  
6222 020024 000277  
6223 020026 000244  
6224 020030 005700  
6225 020032 100403  
6226 020034 102402  
6227 020036 103401  
6228 020040 001404  
6229  
6230  
6231  
6232  
6233 020042  
6234 020042 012742 000452  
6235 020046 005242  
6236 020050 000000  
6237 020052 005300  
6238 020054 000277

```
.....  
: THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE CLR,  
: ST, 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 RO ;CC=0100 RO=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 RO ;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 RO ;MAKE RO NEGATIVE  
SCC ;CC=0111  
  
1
```

```
6239 020056 000250          CLN
6240 020060 005700          TST      R0          ;CC=1000
6241 020062 101402          BLOS    TEST3
6242 020064 102401          BVS    TEST3
6243 020066 100404          BMI    TS214
6244
6245
6246
6247
6248 020070
6249 020070 012742 000453      TEST3:  MOV    #453,-(R2)      ;MOVE TO MAILBOX # ***** 453 *****
6250 020074 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6251 020076 000000          HALT
6252
6253
6254
6255
6256 020100 005212
6257 020102 022712 000214      TS214:  INC    (R2)          ;UPDATE TEST NUMBER
6258 020106 001023          CMP    #214,(R2)     ;SEQUENCE ERROR?
6259 020110 012700 170000      BNE    TS215-10      ;BR TO ERROR HALT ON SEQ ERROR
6260 020114 000277          MOV    #170000,R0   ;R0=170000
6261 020116 000250          SCC
6262 020120 000300          CLN
6263 020122 101402          SWAB   R0          ;CC=1000  R0-360
6264 020124 102401          BLOS   SWB1
6265 020126 100404          BVS   SWB1
6266          BMI   SWB2
6267
6268
6269
6270 020130
6271 020130 012742 000454      SWB1:  MOV    #454,-(R2)      ;MOVE TO MAILBOX # ***** 454 *****
6272 020134 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6273 020136 000000          HALT
6274 020140 000277          SWB2:  SCC
6275 020142 000244          CLZ
6276 020144 000300          SWAB   R0          ;CC=0100  R0-170000
6277 020146 102403          BVS   SWB3
6278 020150 103402          BCS   SWB3
6279 020152 100401          BMI   SWB3
6280 020154 001404          BEQ   TS215
6281
6282
6283
6284
6285 020156
6286 020156 012742 000455      SWB3:  MOV    #455,-(R2)      ;MOVE TO MAILBOX # ***** 455 *****
6287 020162 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6288 020164 000000          HALT

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

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

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
; CONDITIONAL BRANCH INST. AND <
; REPLACE THE MOVE INSTRUCTION <
; WHICH FOLLOWS W/ 754 <
```

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  
6338  
6339  
6340  
6341  
6342  
6343  
6344

020166 005212  
020170 022712 000215  
020174 001062  
020176 012700 040000  
020202 000277  
020204 062700 030000  
020210 101402  
020212 102401  
020214 100004  
  
020216 012742 000456  
020222 005242  
020224 000000  
020226 000264  
  
020230 062700 010000  
020234 101402  
020236 102001  
020240 100404  
  
020242 012742 000457  
020246 005242  
020250 000000  
020252 000257  
020254 000270  
020256 062700 100000  
020262 101002  
020264 102001  
020266 100004  
  
020270

```
.....
: 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~ ;BR TO ERROR HALT ON SEQ ERROR
MOV #40000, ) ;RO=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
;CC=1010 40=100000
ADD #10000,R0
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:
```

```

6345 020270 012742 000460      MOV      #460,-(R2)      ;MOVE TO MAILBOX # ***** 460 *****
6346 020274 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6347 020276 000000              HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6348 020300 062700 177777      ADD6:   ADD      #177777,R0 ;CC=1000 R0=177777
6349 020304 101402              BLOS    ADD7
6350 020306 102401              BVS     ADD7
6351 020310 100404              BMI     ADD8
6352                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
6353                                     ; CONDITIONAL BRANCH INST. AND <===
6354                                     ; REPLACE THE MOVE INSTRUCTION <===
6355                                     ; WHICH FOLLOWS W/ 731 <===
6356 020312
6357 020312 012742 000461      ADD7:   MOV      #461,-(R2) ;MOVE TO MAILBOX # ***** 461 *****
6358 020316 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6359 020320 000000              HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6360 020322 000277      ADD8:   SCC
6361 020324 000245              +CLC!CLZ              ;CC=1010 R=0
6362 020326 062700 000001      ADD     #1,R0
6363 020332 102403              BVS     ADD9
6364 020334 103002              BCC     ADD9
6365 020336 100401              BMI     ADD9
6366 020340 001404              BEQ     TS216
6367                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
6368                                     ; CONDITIONAL BRANCH INST. AND <===
6369                                     ; REPLACE THE MOVE INSTRUCTION <= --
6370                                     ; WHICH FOLLOWS W/ 715 <- -
6371 020342
6372 020342 012742 000462      ADD9:   MOV      #462,-(R2) ;MOVE TO MAILBOX # ***** 462 *****
6373 020346 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6374 020350 000000              HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6375                                     ; OR SEQUENCE ERROR
6376
6377 :*****
6378 :TEST 216 TEST ADC INSTRUCTION
6379 :*****
6380 020352 005212      TS216:  INC      (R2)          ;UPDATE TEST NUMBER
6381 020354 022712 000216      CMP     #216,(R2)      ;SEQUENCE ERROR?
6382 020360 001037              BNE     TS217-10      ;BR TO ERROR HALT ON SEQ ERROR
6383 020362 012700 077777      MOV     #077777,R0
6384 020366 000277              SCC
6385 020370 000252              +CLN!CLV              ;CC=0101
6386 020372 005500              ADC     R0              ;CC=1010
6387 020374 101402              BLOS    ADC1
6388 020376 102001              BVC     ADC1
6389 020400 100404              BMI     ADC2
6390                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=
6391                                     ; CONDITIONAL BRANCH INST. AND <=
6392                                     ; REPLACE THE MOVE INSTRUCTION <=-
6393                                     ; WHICH FOLLOWS W/ 767 <-
6394 020402
6395 020402 012742 000463      ADC1:   MOV      #463,-(R2) ;MOVE TO MAILBOX # ***** 463 *****
6396 020406 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6397 020410 000000              HALT                    ;ADC DID NOT SET CC'S CORRECTLY
6398 020412 052700 077777      ADC2:   BIS     #77777,R0
6399 020416 000277              SCC
6400 020420 000244              CLZ

```



6429  
6430  
6431  
6432  
6433  
6434  
6435  
6436  
6437  
6438  
6439  
6440  
6441  
6442  
6443 020470 005212  
6444 020472 022712 000217  
6445 020476 001042  
6446 020500 012700 000001  
6447 020504 000277  
6448 020506 000251  
6449 020510 005400  
6450 020512 103003  
6451 020514 102402  
6452 020516 001401  
6453 020520 100404  
6454  
6455  
6456  
6457  
6458 020522  
6459 020522 012742 000466  
6460 020526 005242  
6461 020530 000000  
6462 020532 042700 077777  
6463 020536 000257  
6464 020540 000264  
6465 020542 005400  
6466 020544 102003  
6467 020546 103002  
6468 020550 001401  
6469 020552 100404  
6470  
6471  
6472  
6473  
6474 020554  
6475 020554 012742 000467  
6476 020560 005242  
6477 020562 000000  
6478 020564 005000  
6479 020566 000277  
6480 020570 000244  
6481 020572 005400  
6482 020574 102403  
6483 020576 103402  
6484 020600 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 R0 ;CC=1001 R0=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 R0 ;CC=1011 R0=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 R0  
SCC ;CC=1011  
CLZ  
NEG R0 ;CC=0100 R0=0  
BVS NEG5  
BVS NEG5  
BNE NEG5
```

```
6485 020602 100004          BPL      TS220
6486
6487
6488
6489
6490 020604
6491 020604 012742 000470    NEG5:   MOV      #470,-(R2)      ;MOVE TO MAILBOX # ***** 470 *****
6492 020610 005242          INC      -(R2)                ;SET MSGTYP TO FATAL ERROR
6493 020612 000000          HALT
6494
6495
6496
6497
6498
6499 020614 005212          TS220:  INC      (R2)            ;UPDATE TEST NUMBER
6500 020616 022712 000220    CMP      #220,(R2)           ;SEQUENCE ERROR?
6501 020622 001060          BNE     TS221-10            ;BR TO ERROR HALT ON SEQ ERROR
6502 020624 012700 000005    MOV      #5,R0
6503 020630 000257          CCC
6504 020632 000271          +SEN.SEC
6505 020634 022700 000005    CMP      #5,R0              ;CC=0101
6506 020640 101002          BHI     CMP1
6507 020642 102401          BVS     CMP1
6508 020644 100004          BPL     CMP2
6509
6510
6511
6512
6513 020646
6514 020646 012742 000471    CMP1:   MOV      #471,-(R2)      ;MOVE TO MAILBOX # ***** 471 *****
6515 020652 005242          INC      -(R2)                ;SET MSGTYP TO FATAL ERROR
6516 020654 000000          HALT                          ;CMP DID NOT SET CC'S CORRECTLY
6517 020656 012700 100000    CMP2:   MOV      #100000,R0
6518 020662 000277          SCC
6519 020664 000242          CLV
6520 020666 020027 077777    CMP      R0,#77777          ;CC=0010
6521 020672 101402          BLUS    CMP3
6522 020674 102001          BVC     CMP3
6523 020676 100004          BPL     CMP4
6524
6525
6526
6527
6528 020700
6529 020700 012742 000472    CMP3:   MOV      #472,-(R2)      ;MOVE TO MAILBOX # ***** 472 *****
6530 020704 005242          INC      -(R2)                ;SET MSGTYP TO FATAL ERROR
6531 020706 000000          HALT                          ;CMP DID NOT SET CC'S CORRECTLY
6532 020710 052700 040000    CMP4:   BIS      #40000,R0
6533 020714 000257          CCC
6534 020716 000264          SEZ
6535 020720 022700 040000    CMP      #40000,R0          ;CC=1011
6536 020724 102003          BVC     CMP5
6537 020726 103002          BCC     CMP5
6538 020730 001401          BEQ     CMP5
6539 020732 100404          BMI     CMP6
6540
```

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

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

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

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





6587  
6588  
6589  
6590  
6591  
6592  
6593  
6594  
6595  
6596  
6597  
6598  
6599  
6600  
6601 021034 005212  
6602 021036 022712 000222  
6603 021042 001055  
6604 021044 012700 125252  
6605 021050 000257  
6606 021052 000271  
6607 021054 162700 125252  
6608 021060 101002  
6609 021062 102401  
6610 021064 100004  
6611  
6612  
6613  
6614  
6615 021066  
6616 021066 012742 000476  
6617 021072 005242  
6618 021074 000000  
6619 021076 052700 100000  
6620 021102 000277  
6621 021104 000242  
6622 021106 162700 077777  
6623 021112 101402  
6624 021114 102001  
6625 021116 100004  
6626  
6627  
6628  
6629  
6630 021120  
6631 021120 012742 000477  
6632 021124 005242  
6633 021126 000000  
6634 021130 005100  
6635 021132 000277  
6636  
6637 021134 162700 100000  
6638 021140 101402  
6639 021142 102401  
6640 021144 100004  
6641  
6642

```
*****
:
: 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) ;SFT MSGTYP TO FATAL ERROR
      HALT
SUB4: COM R0
      SCC ;R0=177777
      ;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 <-
```

```
6643 : REPLACE THE MOVE INSTRUCTION <===
6644 : WHICH FOLLOWS W/ 736 <==
6645 021146 SUB5: MOV #500,-(R2) ;MOVE TO MAILBOX # ***** 500 *****
6646 021146 012742 000500 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6647 021152 005242 HALT ;SUB DID NOT SET CC'S CORRECTLY
6648 021154 000000 SUB6: CCC ;CC=0100
6649 021156 000257 SEZ
6650 021160 000264 SUB #140000,R0 ;CC=1011
6651 021162 162700 140000 BVC SUB37
6652 021166 102003 BCC SUB7
6653 021170 103002 BEQ SUB7
6654 021172 001401 BMI TS223
6655 021174 100404
6656 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6657 : CONDITIONAL BRANCH INST. AND <====
6658 : REPLACE THE MOVE INSTRUCTION <====
6659 : WHICH FOLLOWS W/ 722 <====
6660 021176 SUB7: MOV #501,-(R2) ;MOVE TO MAILBOX # ***** 501 *****
6661 021176 012742 000501 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6662 021202 005242 HALT
6663 021204 000000
6664 :*****
6665 :TEST 223 TEST SBC INSTRUCTION
6666 :*****
6667 TS223: INC (R2) ;UPDATE TEST NUMBER
6668 021206 005212 CMP #223,(R2) ;SEQUENCE ERROR?
6669 021210 022712 000223 BNE TS224-10 ;BR TO ERROR HALT ON SEQ ERROR
6670 021214 001053 MOV #1,R0
6671 021216 012700 000001 SCC ;CC=1011
6672 021222 000277 CLZ ;CC=0100 R=0
6673 021224 000244 SBC R0
6674 021226 005600 BCS SBC1
6675 021230 103403 BVS SBC1
6676 021232 102402 BMI SBC1
6677 021234 100401 BEQ SBC2
6678 021236 001404
6679 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
6680 : CONDITIONAL BRANCH INST. AND <
6681 : REPLACE THE MOVE INSTRUCTION <-
6682 : WHICH FOLLOWS W/ 766 <
6683 021240 SBC1: MOV #502,-(R2) ;MOVE TO MAILBOX # ***** 502 *****
6684 021240 012742 000502 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6685 021244 005242 HALT ;SBC DID NOT SET CC'S CORRECTLY
6686 021246 000000 SBC2: SCC ;CC=1010
6687 021250 000277 +CLZ!CLC
6688 021252 000245 SBC R0 ;CC=0100 R=C
6689 021254 005600 BCS SBC3
6690 021256 103403 BVS SBC3
6691 021260 102402 BMI SBC3
6692 021262 100401 BEQ SBC4
6693 021264 001404
6694 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
6695 : CONDITIONAL BRANCH INST. AND <-
6696 : REPLACE THE MOVE INSTRUCTION <
6697 : WHICH FOLLOWS W/ 753 <
6698 021266 SBC3:
```



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  
6783  
6784  
6785  
6786  
6787  
6788  
6789

021354 005212  
021356 022712 000224  
021362 001053  
021364 012700 144000  
021370 000257  
021372 000266  
021374 006100  
021376 103003  
021400 102402  
021402 001401  
021404 100404  
  
021406  
021406 012742 000506  
021412 005242  
021414 000000  
021416 000277  
021420 000243  
021422 006100  
021424 103003  
021426 102002  
021430 001401  
021432 100004  
  
021434  
021434 012742 000507  
021440 005242  
021442 000000  
021444 000277  
021446 000250  
021450 006100  
021452 101402  
021454 102401  
021456 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 COMMULATIVE 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 ;RO=144000  
CCC ;CC=0110  
+SEZ.SEV  
ROL R0 ;CC=1001 RO=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 RO=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 ;ROL DID NOT SET CC'S CORRECTLY  
ROL4: SCC ;CC=0111  
CLN  
ROL R0 ;CC=0000 RO=040001  
BLOS ROL5  
BVS ROL5  
BPL ROL6  
  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -  
; CONDITIONAL BRANCH INST. AND <- -  
; REPLACE THE MOVE INSTRUCTION <- -

```

6790
6791 021460
6792 021460 012742 000510
6793 021464 005242
6794 021466 000000
6795 021470 000257
6796 021472 000265
6797 021474 006100
6798 021476 101405
6799 021500 102004
6800 021502 100003
6801 021504 022700 100003
6802 021510 001404
6803
6804
6805
6806
6807 021512
6808 021512 012742 000511
6809 021516 005242
6810 021520 000000
6811
6812
6813
6814
6815 021522 005212
6816 021524 022712 000225
6817 021530 001051
6818 021532 012700 000023
6819 021536 000277
6820 021540 000250
6821 021542 006000
6822 021544 102403
6823 021546 103002
6824 021550 001401
6825 021552 100404
6826
6827
6828
6829
6830 021554
6831 021554 012742 000512
6832 021560 005242
6833 021562 000000
6834 021564 000257
6835 021566 000274
6836 021570 006000
6837 021572 102003
6838 021574 103002
6839 021576 001401
6840 021600 100004
6841
6842
6843
6844
6845

```

```

: 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
      ROL R0 ;CC=1010 R0=100003
      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
      ROR R0 ;CC=0011 R0=00004
      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:

```

```
6846 021602 012742 000513      MOV      #513,-(R2)      ;MOVE TO MAILBOX # ***** 513 *****
6847 021606 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6848 021610 000000              HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6849 021612 000277      ROR4:   SCC            ;CC=1110
6850 021614 000241              CLC
6851 021616 006000              ROR      RO            ;CC=0000  RO=020002
6852 021620 101403              BLOS     ROR5
6853 021622 102402              BVS     ROR5
6854 021624 001401              BEQ     ROR5
6855 021626 100004              BPL     ROR6
6856                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
6857                                ; CONDITIONAL BRANCH INST. AND <-
6858                                ; REPLACE THE MOVE INSTRUCTION <-
6859                                ; WHICH FOLLOWS W/ 740 <-
6860 021630
6861 021630 012742 000514      ROR5:   MOV      #514,-(R2)      ;MOVE TO MAILBOX # ***** 514 *****
6862 021634 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6863 021636 000000              HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6864 021640 000257      ROR6:   CCC            ;CC=0101
6865 021642 000265              +SEC .SEZ
6866 021644 006000              ROR      RO            ;CC=1010  RO=110001
6867 021646 101402              BLOS     ROR7
6868 021650 102001              BVC     ROR7
6869 021652 100404              BMI     TS226
6870                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
6871                                ; CONDITIONAL BRANCH INST. AND <-
6872                                ; REPLACE THE MOVE INSTRUCTION <-
6873                                ; WHICH FOLLOWS W/ 726 <-
6874 021654
6875 021654 012742 000515      ROR7:   MOV      #515,-(R2)      ;MOVE TO MAILBOX # ***** 515 *****
6876 021660 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6877 021662 000000              HALT                    ;ROR DID NOT PRODUCE CORRECT RESULTS
6878                                ; OR SEQUENCE ERROR
6879
6880 ;*****
6881 ;TEST 226 TEST ASL INSTRUCTION
6882 ;*****
6882 021664 005212      TS226:  INC      (R2)          ;UPDATE TEST NUMBER
6883 021666 022712 000226      CMP      #226,(R2)      ;SEQUENCE ERROR?
6884 021672 001054              BNE     TS227-10        ;BR TO ERROR HALT ON SEQ ERROR
6885 021674 012700 144000      MOV      #144000,RO     ;RO=14000
6886 021700 000257              CCC            ;CC=0110
6887 021702 000271              +SEN!SEC
6888 021704 006300              ASL      RO            ;CC=1001  RO=110000
6889 021706 103003              BCC     ASL1
6890 021710 102402              BVS     ASL1
6891 021712 001401              BEQ     ASL1
6892 021714 100404              BMI     ASL2
6893                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
6894                                ; CONDITIONAL BRANCH INST. AND <-
6895                                ; REPLACE THE MOVE INSTRUCTION <-
6896                                ; WHICH FOLLOWS W/ 766 <-
6897 021716
6898 021716 012742 000516      ASL1:   MOV      #516,-(R2)      ;MOVE TO MAILBOX # ***** 516 *****
6899 021722 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6900 021724 000000              HALT
6901 021726 000277      ASL2:   SFC            ;CC 1100
```

```
6902 021730 000243 +CIV.CLC
6903 021732 006300 ASL RO ;CC=0011 R0=020000
6904 021734 103003 BCC ASL3
6905 021736 102002 BVC ASL3
6906 021740 001401 BEQ ASL3
6907 021742 100004 BPL ASL4
6908
6909 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6910 ; CONDITIONAL BRANCH INST. AND <====
6911 ; REPLACE THE MOVE INSTRUCTION <====
6912 ; WHICH FOLLOWS W/ 753 <====
6912 021744 ASL3:
6913 021744 012742 000517 MOV #517,-(R2) ;MOVE TO MAILBOX # ***** 517 *****
6914 021750 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6915 021752 000000 HALT ;ASL DID NOT SET CC'S CORRECTLY
6916 021754 000277 ASL4: SLC ;CC=0111
6917 021756 000250 CLN
6918 021760 006300 ASL RO ;CC=0000 R0=040000
6919 021762 101402 BLOS ASL5
6920 021764 102401 BVS ASL5
6921 021766 100004 BPL ASL6
6922
6923 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6924 ; CONDITIONAL BRANCH INST. AND <====
6925 ; REPLACE THE MOVE INSTRUCTION <====
6926 ; WHICH FOLLOWS W/ 741 <====
6926 021770 ASL5:
6927 021770 012742 000520 MOV #520,-(R2) ;MOVE TO MAILBOX # ***** 520 *****
6928 021774 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6929 021776 000000 HALT ;ASL DID NOT SET CC'S CORRECTLY
6930 022000 000257 ASL6: CCC ;CC=0101
6931 022002 000265 +SEZ!SEC
6932 022004 006300 ASL RO ;CC=1010 R0=100000
6933 022006 103406 BCS ASL7
6934 022010 001405 BEQ ASL7
6935 022012 102004 BVC ASL7
6936 022014 100003 BPL ASL7
6937 022016 022700 100000 CMP #100000,R0
6938 022022 001404 BEQ TS227
6939
6940 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=
6941 ; CONDITIONAL BRANCH INST. AND <=
6942 ; REPLACE THE MOVE INSTRUCTION <=
6943 ; WHICH FOLLOWS W/ 723 <=
6943 022024 ASL7:
6944 022024 012742 000521 MOV #521,-(R2) ;MOVE TO MAILBOX # ***** 521 *****
6945 022030 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6946 022032 000000 HALT ;ASL MALFUNCTIONED
6947 ; OR SEQUENCE ERROR
```

```
6948
6949
6950
6951 022034 005212
6952 022036 022712 000227
6953 022042 001060
6954 022044 012700 100023
6955 022050 000277
6956 022052 000250
6957 022054 006200
6958 022056 102403
6959 022060 103002
6960 022062 001401
6961 022064 100404
6962
6963
6964
6965
6966 022066
6967 022066 012742 0C0522
6968 022072 005242
6969 022074 000000
6970 022076 042700 100000
6971 022102 000277
6972 022104 000243
6973 022106 006200
6974 022110 102003
6975 022112 103002
6976 022114 001401
6977 022116 100004
6978
6979
6980
6981
6982 022120
6983 022120 012742 000523
6984 022124 005242
6985 022126 000000
6986 022130 000277
6987
6988 022132 006200
6989 022134 101403
6990 022136 102402
6991 022140 001401
6992 022142 100004
6993
6994
6995
6996
6997 022144
6998 022144 012742 000524
6999 022150 005242
7000 022152 000000
7001 022154 052700 100000
7002 022160 000257
7003 022162 000265

:*****
:TEST 227 TEST ASL 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

ASR: ASR R0 ;CC=0000 R0=010002
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
```



```
7004 022164 006200 ASR R0 ;C=1010 R0=144001
7005 022166 101406 BLOS ASR7
7006 022170 102005 BVC ASR7
7007 022172 100004 BPL ASR7
7008 022174 001403 BEQ ASR7
7009 022176 022700 144001 CMP #144001,R0 ;CHECK RESULT OF ASR'S
7010 022202 001404 BEQ TS230
7011 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7012 ; CONDITIONAL BRANCH INST. AND <====
7013 ; REPLACE THE MOVE INSTRUCTION <====
7014 ; WHICH FOLLOWS W/ 717 <====
7015 022204 ASR7:
7016 022204 012742 000525 MOV #525,-(R2) ;MOVE TO MAILBOX # ***** 525 *****
7017 022210 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7018 022212 000000 HALT ;ASR DID NOT FUNCTION CORRECTLY
7019 ; OR SEQUENCE ERROR
7020
7021
7022
7023
7024
7025
7026
7027
7028
7029
7030
7031
7032
7033
7034 022214 005212 TS230: INC (R2) ;UPDATE TEST NUMBER
7035 022216 022712 000230 CMP #230,(R2) ;SEQUENCE ERROR?
7036 022222 001033 BNE TS231-10 ;BR TO ERROR HALT ON SEQ ERROR
7037 022224 005000 CLR R0
7038 022226 000277 SCC ;SET CC=1011
7039 022230 000244 CLZ
7040 022232 006700 SXT R0 ;TRY SXT
7041 022234 100006 BPL SXT0 ;TEST CC=1001
7042 022236 001405 BEQ SXT0
7043 022240 102404 BVS SXT0
7044 022242 103003 BCC SXT0
7045 022244 022700 177777 CMP #-1,R0 ;CHECK DATA RESULT
7046 022250 001404 BEQ SXT1
7047 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-- -
7048 ; CONDITIONAL BRANCH INST. AND <- -
7049 ; REPLACE THE MOVE INSTRUCTION <= =
7050 ; WHICH FOLLOWS W/ 764 <--- -
7051 022252 SXT0:
7052 022252 012742 000526 MOV #526,-(R2) ;MOVE TO MAILBOX # ***** 526 *****
7053 022256 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7054 022260 000000 HALT ;RESULTS OF SXT INCORRECT
7055 022262 005000 SXT1: CLR R0 ;R0=0
7056 022264 005010 CLR (R0) ;LOC. 0=0
7057 022266 005110 COM (R0) ;LOC. 0=177777
7058 022270 000257 CCC ;SET CC=0110
7059 022272 000266 +SEZ.SEV
```

7060	022274	006710		SXT	(R0)				
7061	022276	001005		BNE	SXT2			:TEST CC=0100	
7062	022300	103404		BCS	SXT2				
7063	022302	102403		BVS	SXT2				
7064	022304	100402		BMI	SXT2				
7065	022306	005710		TST	(R0)				
7066	022310	001404		BEQ	TS231				
7067								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
7068								CONDITIONAL BRANCH INST. AND	<====
7069								REPLACE THE MOVE INSTRUCTION	<====
7070								WHICH FOLLOWS W/ 744	<====
7071	022312			SXT2:					
7072	022312	012742	000527	MOV	#527, -(R2)			:MOVE TO MAILBOX # ***** 527 *****	
7073	022316	005242		INC	-(R2)			:SET MSGTYP TO FATAL ERROR	
7074	022320	000000		HALT				:RESULTS OF SXT INCORRECT	
7075								: OR SEQUENCE ERROR	

7076  
7077  
7078  
7079  
7080  
7081  
7082  
7083  
7084  
7085  
7086  
7087 022322 005212  
7088 022324 022712 000231  
7089 022330 001035  
7090 022332 012700 007463  
7091 022336 012701 031525  
7092 022342 000277  
7093 022344 000241  
7094 022346 074100  
7095 022350 101406  
7096 022352 102405  
7097 022354 001404  
7098 022356 100403  
7099 022360 022700 036146  
7100 022364 001404  
7101  
7102  
7103  
7104  
7105 022366  
7106 022366 012742 000530  
7107 022372 005242  
7108 022374 000000  
7109 022376 010104  
7110 022400 000261  
7111 022402 000241  
7112 022404 074400  
7113 022406 101406  
7114 022410 102405  
7115 022412 001404  
7116 022414 100403  
7117 022416 022700 007463  
7118 022422 001404  
7119  
7120  
7121  
7122  
7123 022424  
7124 022424 012742 000531  
7125 022430 005242  
7126 022432 000000  
7127

```
*****
:
: 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
      HALT
:
XOR2: MOV R1,R4
      SEC ;CC=1110
      CLC
      XOR R4,R0 ;TRY XOR MGDE 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
      HALT ;RESULT OF XOR INCORRECT
: OR SEQUENCE ERROR
```

```

7128
7129
7130
7131
7132
7133
7134
7135
7136
7137
7138 022434 005212
7139 022436 022712 000232
7140 022442 001023
7141 022444 012700 000525
7142 022450 010004
7143 022452 000277
7144 022454 101002
7145 022456 100001
7146 022460 102404
7147
7148
7149
7150
7151 022462
7152 022462 012742 000532
7153 022466 005242
7154 022470 000000
7155 022472 005304
7156 022474 000277
7157 022476 077012
7158 022500 101004
7159 022502 100003
7160 022504 102002
7161 022506 005704
7162 022510 001404
7163
7164
7165
7166
7167 022512
7168 022512 012742 000533
7169 022516 005242
7170 022520 000000
7171

```

```

:*****
: THIS TEST VERIFIES THE SOB INSTRUCTION. R4 IS USED AS A
: COUNTER WHILE R0 IS THE ADDRESS REGISTER. CONDITIONAL
: BRANCHES ARE USED TO VERIFY PROPER TRANSFER OF CONTROL
: WHILE R4 IS CHECKED TO INSURE PROPER DECREMENTING OF R0.
:*****
: TEST 232      TEST SOB INSTRUCTION
:*****
TS232:  INC      (R2)           :UPDATE TEST NUMBER
        CMP      #232,(R2)     :SEQUENCE ERROR?
        BNE     TS233-10      :BR TO ERROR HALT ON SEQ ERROR
        MOV     #525,R0
        MOV     R0,R4
SOB1:   SCC      :SET CC=1111
        BHI     SOB2         :CC=1111?
        BPL     SOB2
        BVS     SOB3
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
:          CONDITIONAL BRANCH INST. AND <--
:          REPLACE THE MOVE INSTRUCTION <-
:          WHICH FOLLOWS W/ 770 <
SOB2:   MOV     #532,-(R2)     :MOVE TO MAILBOX # ***** 532 *****
        INC     -(R2)         :SET MSGTYP TO FATAL ERROR
        HALT
SOB3:   DEC     R4             :COUNT ITERATIONS
        SCC      :CC=1111
        SOB     R0,SOB1       :DO SOB W/ R0
        BHI     SOB4         :CHECK CC=1111
        BPL     SOB4
        BVC     SOB4
        TST     R4            :ITERATION COUNT OK?
        BEQ     TS233
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -
:          CONDITIONAL BRANCH INST. AND <-- =
:          REPLACE THE MOVE INSTRUCTION <==
:          WHICH FOLLOWS W/ 754 <==
SOB4:   MOV     #533,-(R2)     :MOVE TO MAILBOX # ***** 533 *****
        INC     -(R2)         :SET MSGTYP TO FATAL ERROR
        HALT                 :INCORRECT # OF BRANCHES OR CC'S CHANGED
:                               : OR SEQUENCE ERROR

```

```

7172
7173
7174
7175
7176
7177
7178
7179
7180
7181
7182 022522 005212
7183 022524 022712 000233
7184 022530 001061
7185 022532 012706 001000
7186 022536 012746 125252
7187 022542 162706 000074
7188 022546 012705 022572
7189 022552 012746 006436
7190 022556 000277
7191 022560 000116
7192 022562 012742 000534
7193 022566 005242
7194 022570 000000
7195 022572 101010
7196 022574 100007
7197 022576 102006
7198 022600 020527 125252
7199 022604 001003
7200 022606 022706 001000
7201 022612 001404
7202
7203
7204
7205
7206 022614
7207 022614 012742 000535
7208 022620 005242
7209 022622 000000
7210 022624 012746 052525
7211 022630 012746 006400
7212 022634 010605
7213 022636 004737 022646
7214 022642 000137 022660
7215 022646 000205
7216 022650 012742 000536
7217 022654 005242
7218 022656 000000
7219 022660 022706 001000
7220 022664 001003
7221 022666 022705 052525
7222 022672 001404
7223
7224
7225
7226
7227 022674

```

```

:*****
:
:   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 INSTRJCTION
:*****
TS233:  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 CNTL 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     PC,@MRK4
        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:

```

CJKDB-D DCF11-AA CPU DIAG.  
CJKDBD.P11 24-NOV-80 11:07

MACY11 30A(1052) 14-JAN-81 11:46 D 12 PAGE 146  
T233 TEST MARK INSTRUCTION

SEQ 0146

7228 022674 012742 000537  
7229 022700 005242  
7230 022702 000000  
7231

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

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

7232 177776

PS=177776

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

7244 022704 005212  
7246 022706 022712 000234  
7247 022712 0C1024  
7248 022714 012700 000377  
7249 022720 000257  
7250 022722 106400  
7251 022724 022767 000357 155044  
7252 022732 001404

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 <- -

7257 022734 012742 000540  
7258 022740 005242  
7259 022742 000000  
7260 022744 005000  
7261 022746 005010  
7262 022750 000277  
7263 022752 106410  
7264 022754 100403  
7265 022756 102402  
7266 022760 103401  
7267 022762 001004

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 <- =

7272 022764  
7273 022764 012742 000541  
7274 022770 005242  
7275 022772 000000

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

7281 022774 005212  
7282 022776 022712 000235  
7283 023002 001021  
7284 023004 005000  
7285 023006 012710 177777  
7286 023012 005037 177776  
7287 023016 106420

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

```
7288 023020 022737 000357 177776      CMP      #357,@#PS      ;CHECK DATA
7289 023026 001404                      BEQ      MTPS2
7290                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
7291                      ; CONDITIONAL BRANCH INST. AND <-
7292                      ; REPLACE THE MOVE INSTRUCTION <-
7293                      ; WHICH FOLLOWS W/ 765 <-
7294 023030 012742 000542      MOV      #542,-(R2)    ;MOVE TO MAILBOX # ***** 542 *****
7295 023034 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7296 023036 000000      HALT                      ;DEST. DATA INCORRECT
7297 023040 022700 000001      MTPS2: CMP      #1,R0    ;CHECK DEST. REGISTER.
7298 023044 001404                      BEQ      TS236
7299                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7300                      ; CONDITIONAL BRANCH INST. AND <====
7301                      ; REPLACE THE MOVE INSTRUCTION <====
7302                      ; WHICH FOLLOWS W/ 756 <====
7303 023046 012742 000543      MOV      #543,-(R2)    ;MOVE TO MAILBOX # ***** 543 *****
7304 023052 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7305 023054 000000      HALT                      ;DEST REGISTER NOT INCREMENTED BY 1
7306                      ; OR SEQUENCE ERROR
7307
7308 :*****
7309 :TEST 236      TEST MTPS MODE 3
7310 :*****
7311 023056 005212      TS236: INC      (R2)        ;UPDATE TEST NUMBER
7312 023060 022712 000236      CMP      #236,(R2)    ;SEQUENCE ERROR?
7313 023064 001024      BNE     TS237-10      ;BR TO ERROR HALT ON SEQ ERROR
7314 023066 012700 000402      MOV      #402,R0      ;R0=402
7315 023072 005010      CLR     (R0)          ;LOC. 402=0
7316 023074 012737 052652 000000      MOV      #52652,@#0   ;LOC. 0=52652
7317 023102 005037 177776      CLR     @#PS          ;PS=0
7318 023106 106430      MTPS   @(R0)+         ;TRY MTPS W/MODE 3
7319 023110 022737 000252 177776      CMP      #252,@#PS    ;CHECK DEST. DATA
7320 023116 001404      BEQ     MTPS3
7321                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
7322                      ; CONDITIONAL BRANCH INST. AND <- =
7323                      ; REPLACE THE MOVE INSTRUCTION <- -
7324                      ; WHICH FOLLOWS W/ 762 <-
7325 023120 012742 000544      MOV      #544,-(R2)    ;MOVE TO MAILBOX # ***** 544 *****
7326 023124 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7327 023126 000000      HALT                      ;DEST. DATA INCORRECT
7328 023130 022700 000404      MTPS3: CMP      #404,R0 ;CHECK MODE 3 REGISTER.
7329 023134 001404      BEQ     TS237
7330                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=
7331                      ; CONDITIONAL BRANCH INST. AND < = -
7332                      ; REPLACE THE MOVE INSTRUCTION <- -
7333                      ; WHICH FOLLOWS W/ 753 <=-
7334 023136 012742 000545      MOV      #545,-(R2)    ;MOVE TO MAILBOX # ***** 545 *****
7335 023142 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7336 023144 000000      HALT                      ;MODE 3 REGISTER INCORRECT
7337                      ; OR SEQUENCE ERROR
7338
7339 :*****
7340 :TEST 237      TEST MTPS MODE 4
7341 :*****
7342 023146 005212      S237: INC      (R2)        ;UPDATE TEST NUMBER
7343 023150 022712 000237      CMP      #237,(R2)    ;SEQUENCE ERROR?
```



```
7344 023154 001022 BNE TS240-10 ;BR TO ERROR HALT ON SEQ ERROR
7345 023156 012700 000001 MOV #1,R0 ;RO=1
7346 023162 012737 125125 000000 MOV #125125,@#0 ;LOC. 0 = 125125
7347 023170 005037 177776 CLR @#PS ;PS=0
7348 023174 106440 MTPS -(R0) ;TRY MTPS W/MODE 4
7349 023176 022737 000105 177776 CMP #105,@#PS ;CHECK DEST. DATA
7350 023204 001404 BEQ MTPS4
7351 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
7352 ; CONDITIONAL BRANCH INST. AND <===
7353 ; REPLACE THE MOVE INSTRUCTION <===
7354 ; WHICH FOLLOWS W/ 763 <===
7355 023206 012742 000546 MOV #546,-(R2) ;MOVE TO MAILBOX # ***** 546 *****
7356 023212 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7357 023214 000000 HALT ;DEST. DATA INCORRECT
7358 023216 005700 MTPS4: TST R0 ;CHECK MODE 4 REGISTER
7359 023220 001404 BEQ TS240
7360 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
7361 ; CONDITIONAL BRANCH INST. AND <===
7362 ; REPLACE THE MOVE INSTRUCTION <===
7363 ; WHICH FOLLOWS W/ 755 <=
7364 023222 012742 000547 MOV #547,-(R2) ;MOVE TO MAILBOX # ***** 547 *****
7365 023226 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7366 023230 000000 HALT ;MODE 4 REGISTER NOT DECREMENTED BY 1
7367 ; OR SEQUENCE ERROR
7368
7369 :*****
7370 :TEST 240 TEST MTPS MODE 5
7371 :*****
7372 023232 005212 TS240: INC (R2) ;UPDATE TEST NUMBER
7373 023234 022712 000240 CMP #240,(R2) ;SEQUENCE ERROR?
7374 023240 001021 BNE TS241-10 ;BR TO ERROR HALT ON SEQ ERROR
7375 023242 012700 000404 MOV #404,R0 ;RO=404
7376 023246 012737 177400 000000 MOV #177400,@#0 ;LOC. 0=177400
7377 023254 000277 SCC ;SET ALL COND. CODES
7378 023256 106450 MTPS @-(R0) ;TRY MTPS W/MODE 5
7379 023260 005737 177776 TST @#PS ;CHECK DEST. DATA.
7380 023264 001404 BEQ MTPS5
7381 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
7382 ; CONDITIONAL BRANCH INST. AND <
7383 ; REPLACE THE MOVE INSTRUCTION <
7384 ; WHICH FOLLOWS W/ 765 .
7385 023266 012742 000550 MOV #550,-(R2) ;MOVE TO MAILBOX # ***** 550 *****
7386 023272 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7387 023274 000000 HALT ;DESTINATION DATA INCORRECT
7388 023276 022700 000402 MTPS5: CMP #402,R0 ;CHECK MODE 5 REGISTER
7389 023302 001404 BEQ TS241
7390 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
7391 ; CONDITIONAL BRANCH INST. AND <===
7392 ; REPLACE THE MOVE INSTRUCTION <===
7393 ; WHICH FOLLOWS W/ 756 <
7394 023304 012742 000551 MOV #551,-(R2) ;MOVE TO MAILBOX # ***** 551 *****
7395 023310 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7396 023312 000000 HALT ;MODE 5 REGISTER NOT DECREMENTED BY 2
7397 ; OR SEQUENCE ERROR
7398
7399 :*****
```



JKDB-D DCF11-AA CPU DIAG.  
JKDBD.P11 24-NOV-80 11:07

MACY11 3CA(1052) 14-JAN-81 11:46 I 12  
T242 TEST MTF'S MODE 7 PAGE 151

SEQ 0151

7456 023472 000000  
7457  
7458

HALT

:MODE 7 REGISTER MODIFIED  
: OR SEQUENCE ERROR

7459  
7460  
7461  
7462  
7463  
7464  
7465  
7466  
7467  
7468  
7469  
7470  
7471  
7472  
7473  
7474  
7475  
7476  
7477  
7478  
7479  
7480  
7481  
7482  
7483  
7484  
7485  
7486  
7487  
7488  
7489  
7490  
7491  
7492  
7493  
7494  
7495  
7496  
7497  
7498  
7499  
7500  
7501  
7502  
7503  
7504  
7505  
7506  
7507  
7508  
7509  
7510  
7511  
7512  
7513  
7514

023474 005212  
023476 022712 000243  
023502 001025  
023504 012737 000377 177776  
023512 106700  
023514 022700 177757  
023520 001404  
  
023522 012742 000556  
023526 005242  
023530 000000  
  
023532 005000  
023534 012737 177777 000000  
023542 005037 177776  
023546 106710  
023550 105737 000000  
023554 001404  
  
023556 012742 000557  
023562 005242  
023564 000000

.....  
: 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  
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 \*\*\*\*\*  
INL -(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

S244: 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





```
7627
7628
7629 024104 005212
7630 024106 022112 000247
7631 024112 001033
7632 024114 012700 000410
7633 024120 012737 177777 000000
7634 024126 005037 177776
7635 024132 106750
7636 024134 103403
7637 024136 102402
7638 024140 100401
7639 024142 001404
7640
7641
7642
7643
7644 024144
7645 024144 012742 000571
7646 024150 005242
7647 024152 000000
7648 024154 022737 000377 000000
7649 024162 001404
7650
7651
7652
7653
7654 024164 012742 000572
7655 024170 005242
7656 024172 000000
7657 024174 020027 000406
7658 024200 001404
7659
7660
7661
7662
7663 024202 012742 000573
7664 024206 005242
7665 024210 000000
7666
7667
7668
7669
7670
7671 024212 005212
7672 024214 022712 000250
7673 024220 001034
7674 024222 012700 000401
7675 024226 005037 000000
7676 024232 012737 000252 177776
7677 024240 106760 177377
7678 024244 102403
7679 024246 103402
7680 024250 001401
7681 024252 100404
7682

;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 ;RO=410
      CLR #-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 ;RO=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 <
```







T253 TEST USER MODE R6 CAN HOLD A ONE IN EVERY POSITION

```

7795 024524 006106      USP1:  ROL      R6          ;ROTATE 1 POSITION
7796 024526 103376      BCC     USP1      ;BR IF NOT ALL DONE
7797 024530 001407      BEQ     USP1A     ;BR IF NO BITS PICKED
7798 024532 042767 140C00 153236  BIC     #USRM,PS  ;CLEAR USER MODE
7799 024540 012742 000603      MOV     #603,-(R2) ;MOVE TO MAILBOX # ***** 603 *****
7800 024544 005242      INC     -(R2)     ;SET MSGTYP TO FATAL ERROR
7801 024546 000000      HALT                    ;USER MODE R6 PICKED A BIT
7802 024550 042767 140000 153220  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

```

7815 024556 005212      TS254: INC     (R2)          ;UPDATE TEST NUMBER
7816 024560 022712 000254      CMP     #254,(R2)  ;SEQUENCE ERROR?
7817 024564 001046      BNE     USP4-10   ;BR TO ERROR HALT ON SEQ ERROR
7818 024566 012767 000340 153202  MOV     #340,PS    ;LOCK OUT INTERRUPTS WHILE PLAYING WITH R6
7819 024574 052767 140000 153174  BIS     #USRM,PS   ;SET USER MODE
7820 024602 012706 177777      MOV     #-1,R6    ;SET USER R6 TO ALL ONES
7821 024606 022706 177777      CMP     #-1,R6    ;READ AND CHECK USER R6
7822 024612 001407      BEQ     USP2      ;BR IF NO ERROR
7823 024614 042767 140000 153154  BIC     #USRM,PS  ;CLEAR USER MODE
7824 024622 012742 000604      MOV     #604,-(R2) ;MOVE TO MAILBOX # ***** 604 *****
7825 024626 005242      INC     -(R2)     ;SET MSGTYP TO FATAL ERROR
7826 024630 000000      HALT                    ;USER R6 WILL NOT HOLD ALL ONES
7827 024632 042767 140000 153136  USP2:  BIC     #USRM,PS  ;SET KERNEL MODE
7828 024640 022706 177777      CMP     #-1,R6    ;KERNEL MODE R6 ADDR. FROM USER MODE?>>
7829 024644 001004      BNE     USP3

```

```

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

```

```

7834 024646 012742 000605      MOV     #605,-(R2) ;MOVE TO MAILBOX # ***** 605 *****
7835 024652 005242      INC     -(R2)     ;SET MSGTYP TO FATAL ERROR
7836 024654 000000      HALT                    ;DUAL ADDRESSING ERROR USER/KERNEL R6
7837 024656 005006      USP3:  CLR     R6          ;CLEAR KERNEL MODE SP
7838 024660 052767 140000 153110  BIS     #USRM,PS   ;SET USER MODE
7839 024666 022706 177777      CMP     #-1,R6    ;CHECK USER R6 NOT ADDR. FROM KERNEL MODE
7840 024672 042767 140000 153076  BIC     #USRM,PS  ;CLEAR USER MODE
7841 024700 001404      BEQ     USP4      ;BR IF NO ERROR
7842 024702 012742 000606      MOV     #606,-(R2) ;MOVE TO MAILBOX # ***** 606 *****
7843 024706 005242      INC     -(R2)     ;SET MSGTYP TO FATAL ERROR
7844 024710 000000      HALT                    ;DUAL ADDRESSING ERROR OR SEQUENCE ERROR
7845 024712 012706 001000 153052  USP4:  MOV     #STBOT,R6 ;RESTORE SP USER
7846 024716 042767 140000      BIC     #USRM,PS  ;SET KERNEL MODE
7847 024724 012706 001000      MOV     #STBOT,R6 ;RESTORE SP KERNEL

```

```

:*****
:

```

: THESE NEXT TWO TESTS VERIFY MFPI AND MTPi INSTRUCTIONS  
: WITH R6 IN MODE 0.

```

7851 :
7852 :
7853 :
7854 :
7855 : TEST 255 TEST MFPI WITH R6 IN MODE 0
7856 :
7857 TS255: INC (R2) ;UPDATE TEST NUMBER
7858 CMP #255,(R2) ;SEQUENCE ERROR?
7859 BNE TS256-10 ;BR TO ERROR HALT ON SFQ ERROR
7860 MOV #STBOT,R6 ;INITIALIZE KERNEL STACK POINTER
7861 MOV #USRM,PS ;SET USER MODE,PREVIOUS KERNEL
7862 MOV #USTBOT,R6 ;INITIALIZE USER STACK POINTER
7863 MFPI R6 ;TRY MFPI WITH MODE 0
7864 CMP #140000,PS ;CHECK PSW
7865 BEQ MFPI0 ;BR IF NO ERROR
7866 BIC #USRM,PS ;CLEAR USER MODE
7867 MOV #607,-(R2) ;MOVE TO MAILBOX # ***** 607 *****
7868 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7869 HALT ;INCORRECT PSW FROM MFPI
7870 MFPI0: BIC #USRM,PS ;CLEAR USER MODE
7871 CMP #STBOT,USTBOT-2 ;CHECK DATA ON STACK
7872 BEQ MFPI0A ;BR IF NO ERROR
7873 MOV #610,-(R2) ;MOVE TO MAILBOX # ***** 610 *****
7874 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7875 HALT ;INCORRECT DATA FROM MFPI
7876 MFPI0A:
7877 :
7878 :
7879 : TEST 256 TEST MTPi WITH R6 IN MODE 0
7880 :
7881 TS256: INC (R2) ;UPDATE TEST NUMBER
7882 CMP #256,(R2) ;SEQUENCE ERROR?
7883 BNE TS257-10 ;BR TO ERROR HALT ON SEQ ERROR
7884 MOV #340,PS ;SET KERNEL MODE AND LOCK OUT INTERRUPTS
7885 CLR R6 ;INITIALIZE KERNEL R6
7886 BIS #USRM,PS ;SET USER MODE/PREVIOUS KERNEL
7887 MOV #USTBOT,R6 ;INITIALIZE USER STACK POINTER
7888 MOV #STBOT,-(R6) ;SET UP TARGET DATA
7889 MTPi R6 ;TRY MODE 0 MTPi
7890 CMP #140340,PS ;CHECK PSW
7891 BEQ MTPi0 ;BR IF NO ERROR
7892 BIC #USRM,PS ;CLEAR USER MODE
7893 MOV #611,-(R2) ;MOVE TO MAILBOX # ***** 611 *****
7894 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7895 HALT ;PS INCORRECT FOLLOWING MTPi
7896 MTPi0: BIC #USRM,PS ;SET KERNEL MODE
7897 CMP R6,#STBOT ;CHECK TARGET DATA
7898 BEQ TS257
7899 :
7900 : ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-- -
7901 : ; CONDITIONAL BRANCH INST. AND <- -
7902 : ; REPLACE THE MOVE INSTRUCTION <-- -
7903 : ; WHICH FOLLOWS W/ 742 <- -
7904 :
7905 :
7906 :
7903 MOV #612,-(R2) ;MOVE TO MAILBOX # ***** 612 *****
7904 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7905 HALT ;DATA INCORRECT FOLLOWING MTPi
; OR SEQUENCE ERROR

```

CJKDB-D DEF11-AA CPU DIAG.  
CJKDBD.P11 24-NOV-80 11:07

MACY11 3CA(1052) 14-JAN-81 11:46 PAGE 160  
T256 TEST MPI WITH R6 IN MODE 0

SEQ 0160

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  
7960  
7961  
7962  
7963

025146 005212  
025150 022712 000257  
025154 001062  
025156 012700 027214  
025162 012704 027252  
025166 012767 000017 000142  
025174 012067 000110  
025200 012401  
025202 012767 177777 000074  
025210 012703 000020  
025214 005267 000064  
025220 032701 100000  
025224 013705 177776  
025230 042705 177773  
025234 000165 025240  
025240 000167 000020  
025244 012767 025340 000042  
025252 012767 025322 000040  
025260 000167 000014  
025264 012767 025322 000022  
025272 012767 025340 000020  
025300 006101  
025302 012737  
025304 000000  
025306 177776  
025310 000000  
025312 000137  
025314 000000  
025316 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  ;
        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 OCCRS
        JMP     @(PC)+      ;THIS JUMP IF BRANCH OCCRS

```

7964 025320 000000  
7965 025322 012702 000304  
7966 025326 012742 000613  
7967 025332 005242  
7968 025334 000000  
7969 025336 000000  
7970 025340 005303  
7971 025342 013705 177776  
7972 025346 042705 177773  
7973 025352 000165 025356  
7974 025356 000167 177632  
7975 025362 005367 177750  
7976 025366 013705 177776  
7977 025372 042705 177773  
7978 025376 000165 025402  
7979 025402 000167 177566

YBR: 0 ;WHERE TO GO IF BRANCH OCCURS  
FR: MOV #STESTN,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  
SETCC  
DEC BRCT ;BR'S DONE?  
MOV @#177776,R5 ;SIMULATE A JNE  
BIC #177773,R5 ; (JUMP NOT EQUAL)  
JMP .+4(R5) ; TO SETBR  
SETBR

```

7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7990
7991 025406 005212
7992 025410 022712 000260
7993 025414 001051
7994 025416 005000
7995 025420 005001
7996 025422 005002
7997 025424 005003
7998 025426 005004
7999 025430 005005
8000 025432 052700 000001
8001 025436 052701 000002
8002 025442 052702 000004
8003 025446 052703 000010
8004 025452 052704 000020
8005 025456 052705 000040
8006 025462 052706 001000
8007 025466 022706 001000
8008 025472 001022
8009 025474 022705 000040
8010 025500 001017
8011 025502 022704 000020
8012 025506 001014
8013 025510 022703 000010
8014 025514 001011
8015 025516 022702 000004
8016 025522 001006
8017 025524 022701 000002
8018 025530 001003
8019 025532 022700 000001
8020 025536 001404
8021
8022
8023
8024
8025 025540
8026 025540 012742 000614
8027 025544 005242
8028 025546 000000
8029 025550 012702 000304
8030

```

```

:.....
: 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
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
BITCHK: BIS #1000,R6 ;R6=1000
CMP #1000,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/ 726 <
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

```

```

8031
8032
8033
8034
8035
8036
8037
8038
8039
8040 025554 005212
8041 025556 022712 000261
8042 025562 001012
8043 025564 052737 170357 177776
8044 025572 105037 177776
8045 025576 013700 177776
8046 025602 022700 170004
8047 025606 001006
8048 025610 005037 177776
8049 025614 012742 000615
8050 025620 005242
8051 025622 000000
8052 025624 005037 177776
8053
8054
8055
8056
8057
8058
8059
8060
8061
8062
8063 025630 005212
8064 025632 022712 000262
8065 025636 001010
8066 025640 000277
8067 025642 000252
8068 025644 000167 000000
8069 025650 100403
8070 025652 001002
8071 025654 102401
8072 025656 03404
8073
8074
8075
8076
8077 025660
8078 025660 012742 000616
8079 025664 005242
8080 025666 000000
8081

```

```

:*****
:      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
        CLR    @#PS         ;CLR PR LEVEL AND CC'S
        MOV     @#PS,R0      ;COPY CONTENTS OF PSW
        CMP     #170004,R0   ;TEST THAT CLRB AFFECTED ONLY LOW BYTE
        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                    ;CC-0101
        +CLN.CLV              ;
        JMP     JMPT         ;JUMP TO TEST PSW
JMPT:   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 IF SET
        BCS     TS263
:      TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
:      CONDITIONAL BRANCH INST. AND <---
:      REPLACE THE MOVE INSTRUCTION <---
:      WHICH FOLLOWS W/ 767 <
JMPPERR: MOV     #616,-(R2)   ;MOVE TO MAILBOX # ***** 616 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT                    ;JMP INSTRUCTION AFFECTED CC'S
:      OR SEQUENCE ERROR

```





ADB-D DCF11-AA PU DIAG.  
DBD.P11 24-NOV-80 11:07

MA.V11 30A(1052) 14-JAN-81 11:46 PAGE 166  
1263 TEST SET CC AND CLEAR CC INSTRUCTIONS

SEQ 0164

8138	026044		
8140	026044	012742	000620
8141	026050	005242	
8142	026052	000000	
8143	026054	005267	177760
8144	026060	005267	177740
8145	026064	026727	177734 000277
8146	026072	003753	
8147	026074	000167	000006

```

; WHICH FOLLOWS W/ 715
CCERR: MOV #620,-(R2) ;MOVE TO MAILBOX # ***** 620 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;SET CC FAILED OR SEQUENCE ERROR
CON2: INC SC4 ;SET NEXT OCTAL MAP
        INC SC3 ;PREPARE NEXT SET CC INSTRUCTION
        CMP SC3,#277 ;FINISHED?
        BLE SETCD ;BR IF NO
        JMP MO:0 ;JUMP TO NEXT TESTS

```

```
8148 :*****
8149 :SBITL TEST INSTRUCTIONS USING SAME REGISTER FOR SOURCE & DESTINATION
8150 :
8151 :IN AUTO INCREMENT (DECREMENT) MODES AND
8152 :AUTO INCREMENT (DECREMENT) DEFERRED MODES,
8153 :CONTENTS OF THE REGISTER IN USED ARE
8154 :INCREMENTED (DECREMENTED) BY 2
8155 :BEFORE USED AS THE SOURCE OPERAND.
8156 :
8157 026100 000000 000000 000000 A: .WORD 0,0,0
8158 026106 MORO:
8159 :*****
8160 :TEST 264 TEST AUTO-INCREMENT MODE, USING R0
8161 :*****
8162 026106 005212 TS264: INC (R2) ;UPDATE TEST NUMBER
8163 026110 022712 000264 CMP #264,(R2) ;SEQUENCE ERROR?
8164 026114 001020 BNE TS265-10 ;BR TO ERROR HALT ON SEQ ERROR
8165 026116 005037 026100 CLR @#A ;CLEAR LOC A
8166 026122 012700 026100 MOV #A,R0 ;R0 STORES ADDR OF A
8167 026126 060020 ADD R0,(R0)+ ;CHECK THAT R0 IS INCR BY 2 BEFORE
8168 : ;BEING USED AS THE SOURCE OPERAND
8169 026130 022700 026102 CMP #A+2,R0 ;R0 INCR BY 2?
8170 026134 001404 BEQ MOR1
8171 : ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
8172 : ; CONDITIONAL BRANCH INST. AND <- --
8173 : ; REPLACE THE MOVE INSTRUCTION <- -
8174 : ; WHICH FOLLOWS W/ 767 < -
8175 026136 012742 000621 MOV #621,-(R2) ;MOVE TO MAILBOX # ***** 621 *****
8176 026142 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8177 026144 000000 HALT ;R0 WAS NOT INCREMENTED BY 2
8178 :
8179 026146 022737 026102 026100 MOR1: CMP #A+2,@#A ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
8180 : ;BEING USED IN THE "ADD" INSTR
8181 : ;LOC A CONTAINS (A+2)?
8182 026154 001404 BEQ TS265
8183 : ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8184 : ; CONDITIONAL BRANCH INST. AND <====
8185 : ; REPLACE THE MOVE INSTRUCTION <====
8186 : ; WHICH FOLLOWS W/ 757 <====
8187 026156 012742 000622 MOV #622,-(R2) ;MOVE TO MAILBOX # ***** 622 *****
8188 026162 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8189 026164 000000 HALT ;WRONG SUM IN LOC A
8190 : ; OR SEQUENCE ERROR
8191 :
8192 :*****
8193 :TEST 265 AUTO-DECREMENT MODE, USING R0
8194 :*****
8195 026166 005212 TS265: INC (R2) ;UPDATE TEST NUMBER
8196 026170 022712 000265 CMP #265,(R2) ;SEQUENCE ERROR?
8197 026174 001020 BNE TS266-10 ;BR TO ERROR HALT ON SEQ ERROR
8198 026176 005037 026100 CLR @#A ;CLEAR LOC A
8199 026202 012700 026102 MOV #A+2,R0 ;R0 STORES ADDR OF A+2
8200 026206 060040 ADD R0,-(R0) ;CHECK THAT R0 IS DECR BY 2 BEFORE
8201 : ;BEING USED AS THE SOURCE OPERAND
8202 026210 022700 026100 CMP #A,R0 ;R0 DECR BY 2?
8203 026214 001404 BEQ MOR2
```

```
8204 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8205 : CONDITIONAL BRANCH INST. AND <====
8206 : REPLACE THE MOVE INSTRUCTION <====
8207 : WHICH FOLLOWS W/ 767 <====
8208 026216 012742 000623 MOV #623,-(R2) ;MOVE TO MAILBOX # ***** 623 *****
8209 026222 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8210 026224 000000 HALT ;RO WAS NOT DECREMENTED BY 2
8211 :
8212 026226 022737 026100 026100 MOR2: CMP #A,@#A ;CONTENT OF RO WAS DECR BY 2 BEFORE
8213 : BEING USED IN THE 'ADD' INSTR
8214 : LOC A CONTAINS (RO)
8215 026234 001404 BEQ TS266
8216 :
8217 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8218 : CONDITIONAL BRANCH INST. AND <====
8219 : REPLACE THE MOVE INSTRUCTION <====
8220 : WHICH FOLLOWS W/ 757 <====
8220 026236 012742 000624 MOV #624,-(R2) ;MOVE TO MAILBOX # ***** 624 *****
8221 026242 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8222 026244 000000 HALT ;WRONG SUM IN LOC A
8223 : OR SEQUENCE ERROR
8224 :
8225 :
8226 :*****
8227 :TEST 266 TEST AUTO-INCREMENT DEFERRED MODE, USING RO
8228 :*****
8228 026246 005212 TS266: INC (R2) ;UPDATE TEST NUMBER
8229 026250 022712 000266 CMP #266,(R2) ;SEQUENCE ERROR?
8230 026254 001044 BNE TS267-10 ;BR TO ERROR HALT ON SEQ ERROR
8231 026256 005037 026100 CLR @#A ;CLEAR LOC A
8232 026262 005037 026104 CLR @#A+4 ;CLEAR LOC A+4
8233 026266 012737 026100 026102 MOV #A,@#A+2 ;STORE ADDR A IN LOC A+2
8234 026274 012700 026102 MOV #A+2,RO ;RO STORES ADDR A+2
8235 026300 060030 ADD RO,@(RO)+ ;CHECK THAT RO IS INCR BY 2 BEFORE
8236 : BEING USED AS THE SOURCE OPERAND
8237 026302 022700 026104 CMP #A+4,RO ;RO INCR BY 2?
8238 026306 001404 BEQ MOR3
8239 :
8240 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
8241 : CONDITIONAL BRANCH INST. AND <--
8242 : REPLACE THE MOVE INSTRUCTION <=
8243 : WHICH FOLLOWS W/ 762 <-
8243 026310 012742 000625 MOV #625,-(R2) ;MOVE TO MAILBOX # ***** 625 *****
8244 026314 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8245 026316 000000 HALT ;RO WAS NOT INCREMENTED BY 2
8246 :
8247 026320 022737 026100 026102 MOR3: CMP #A,@#A+2 ;LOC A+2 STILL STORES ADDR A?
8248 026326 001404 BEQ MOR4
8249 :
8250 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
8251 : CONDITIONAL BRANCH INST. AND <--
8252 : REPLACE THE MOVE INSTRUCTION <==
8253 : WHICH FOLLOWS W/ 752 <--
8253 026330 012742 000626 MOV #626,-(R2) ;MOVE TO MAILBOX # ***** 626 *****
8254 026334 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8255 026336 000000 HALT ;LOC A+2 STORES WRONG DATA
8256 :
8257 026340 022737 026104 026100 MOR4: CMP #A+4,@#A ;CHECK CONTENT OF RO WAS INCR BY 2 BEFORE
8258 : BEING USED IN THE 'ADD' INSTR
8259 026346 001404 BEQ MOR5
```

```

8260 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8261 ; CONDITIONAL BRANCH INST. AND <====
8262 ; REPLACE THE MOVE INSTRUCTION <====
8263 ; WHICH FOLLOWS W/ 742 <====
8264 026350 012742 000627 MOV #627,-(R2) ;MOVE TO MAILBOX # ***** 627 *****
8265 026354 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8266 026356 000000 HALT ;LOC A STORES WRONG DATA
8267 ;
8268 026360 005737 026104 MOR5: TST @#A+4 ;LOC A+4 STILL STORES 0?
8269 026364 001404 BEQ TS267
8270 ;
8271 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8272 ; CONDITIONAL BRANCH INST. AND <====
8273 ; REPLACE THE MOVE INSTRUCTION <====
8274 ; WHICH FOLLOWS W/ 733 <====
8274 026366 012742 000630 MOV #630,-(R2) ;MOVE TO MAILBOX # ***** 630 *****
8275 026372 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8276 026374 000000 HALT ;LOC A+4 DID NOT STAY CLEAR
8277 ; OR SEQUENCE ERROR
8278 ;
8279 ;*****
8280 ;TEST 267 TEST AUTO-DECREMENT DEFERRED, USING R0
8281 ;*****
8282 026376 005212 TS267: INC (R2) ;UPDATE TEST NUMBER
8283 026400 022712 000267 CMP #267,(R2) ;SEQUENCE ERROR?
8284 026404 001044 BNE TS270-10 ;BR TO ERROR HALT ON SEQ ERROR
8285 026406 005037 026100 CLR @#A ;CLEAR LOC A
8286 026412 005037 026104 CLR @#A+4 ;CLEAR LOC A+4
8287 026416 012700 026104 MOV #A+4,R0 ;R0 STORES ADDR A+4
8288 026422 012737 026100 026102 MOV #A,@#A+2 ;STORE ADDR A IN LOC A+2
8289 026430 060050 ADD R0,@-(R0) ;CHECK THAT R0 IS DECR BY 2 BEFORE
8290 ; BEING USED AS THE SOURCE OPERAND
8291 026432 022700 026102 CMP #A+2,R0 ;R0 DECREMENTED BY 2?
8292 026436 001404 BEQ MOR6
8293 ;
8294 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
8295 ; CONDITIONAL BRANCH INST. AND <
8296 ; REPLACE THE MOVE INSTRUCTION <
8297 ; WHICH FOLLOWS W/ 762 <
8297 026440 012742 000631 MOV #631,-(R2) ;MOVE TO MAILBOX # ***** 631 *****
8298 026444 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8299 026446 000000 HALT ;R0 WAS NOT DECREMENTED BY 2
8300 ;
8301 026450 022737 026102 026100 MOR6: CMP #A+2,@#A ;CHECK CONTENT OF R0 WAS DECR BY 2 BEFORE
8302 ; BEING USED IN THE 'ADD' INSTR
8303 026456 001404 BEQ MOR7
8304 ;
8305 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
8306 ; CONDITIONAL BRANCH INST. AND < -
8307 ; REPLACE THE MOVE INSTRUCTION <
8308 ; WHICH FOLLOWS W/ 752 < -
8308 026460 012742 000632 MOV #632,-(R2) ;MOVE TO MAILBOX # ***** 632 *****
8309 026464 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8310 026466 000000 HALT ;LOC A STORES WRONG DATA
8311 ;
8312 ;
8313 026470 022737 026100 026102 MOR7: CMP #A,@#A+2 ;LOC A+2 STILL STORES A?
8314 026476 001404 BEQ MOR8
8315 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <

```



```

8334 :*****
8335 :SBTTL INSTRUCTION USING PC AS SOURCE REGISTER
8336 :
8337 :IN INDEX, INDEX DEFERRED, RELATIVE, AND
8338 :RELATIVE DEFERRED MODES, DESTINATION WILL CONTAIN
8339 :THE PC COUNT OF THE CURRENT INSTRUCTION +4.
8340 :
8341 :*****
8342 :TEST 270 TEST PC AS SOURCE IN MODE 0, USING R0
8343 :*****
8344 026526 005212 TS270: INC (R2) ;UPDATE TEST NUMBER
8345 026530 022712 000270 CMP #270,(R2) ;SEQUENCE ERROR?
8346 026534 001006 BNE TS271-10 ;BR TO ERROR HALT ON SEQ ERROR
8347 026536 012700 177777 MOV #-1,R0 ;SET ALL 1 IN R0
8348 026542 010700 PCN01: MOV PC,R0 ;STORES PC IN R0
8349 026544 022700 026544 CMP #PCN01+2,R0 ;R0 STORES PC+2?
8350 026550 001404 BEQ TS271
8351 :
8352 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--- -
8353 : CONDITIONAL BRANCH INST. AND <---=-
8354 : REPLACE THE MOVE INSTRUCTION <====
8355 : WHICH FOLLOWS W/ 771 <= -=
8355 026552 012742 000635 MOV #635,-(R2) ;MOVE TO MAILBOX # ***** 635 *****
8356 026556 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8357 026560 000000 HALT ;R0 STORED WRONG VALUE
8358 : OR SEQUENCE ERROR
8359 :
8360 :*****
8361 :TEST 271 TEST PC AS SOURCE IN MODE 6, USING R0
8362 :*****
8363 026562 005212 TS271: INC (R2) ;UPDATE TEST NUMBER
8364 026564 022712 000271 CMP #271,(R2) ;SEQUENCE ERROR?
8365 026570 001010 BNE TS272-10 ;BR TO ERROR HALT ON SEQ ERROR
8366 026572 012700 026100 MOV #A,R0 ;R0 STORES ADDR A
8367 026576 010760 000004 PCN2: MOV PC,4(R0) ;EFFECTIVE ADDR IS A+4
8368 026602 022737 026602 026104 CMP #PCN2+4,@#A+4 ;LOC A+4 STORES PC+4?
8369 026610 001404 BEQ TS272
8370 :
8371 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8372 : CONDITIONAL BRANCH INST. AND <===-
8373 : REPLACE THE MOVE INSTRUCTION <=-=-
8374 : WHICH FOLLOWS W/ 767 <= -=
8374 026612 012742 000636 MOV #636,-(R2) ;MOVE TO MAILBOX # ***** 636 *****
8375 026616 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8376 026620 000000 HALT ;LOC A+4 STORED WRONG VALUE
8377 : OR SEQUENCE ERROR
8378 :
8379 :*****
8380 :TEST 272 TEST PC AS SOURCE IN MODE 7, USING R0
8381 :*****
8382 026622 005212 TS272: INC (R2) ;UPDATE TEST NUMBER
8383 026624 022712 000272 CMP #272,(R2) ;SEQUENCE ERROR?
8384 026630 001013 BNE TS273-10 ;BR TO ERROR HALT ON SEQ ERROR
8385 026632 012737 026100 026104 MOV #A,@#A+4 ;LOC A+4 STORES ADDR A
8386 026640 012700 026100 MOV #A,R0 ;R0 STORES ADDR A
8387 026644 010770 000004 PCN3: MOV PC,@4(R0) ;EFFECTIVE ADDR IS A
8388 026650 022737 026650 026100 CMP #PCN3+4,@#A ;LOC A STORES PC+4?
8389 026656 001404 BEQ TS273

```

```

8390 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
8391 ; CONDITIONAL BRANCH INST. AND <===
8392 ; REPLACE MOVE INSTRUCTION <===
8393 ; WHICH FOLLOWS W/ 764 <===
8394 026660 012742 000637 MOV #637,-(R2) ;MOVE TO MAILBOX # *** 637 *****
8395 026664 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8396 026666 000000 HALT ;LOC A STORED WRONG VALUE
8397 ; OR SEQUENCE ERROR
8398

```

```

8399 :*****
8400 :TEST 273 TEST PC AS SOURCE IN RELATIVE DEFERRED MODE ,USING R0
8401 :*****

```

```

8402 026670 005212 TS273: INC (R2) ;UPDATE TEST NUMBER
8403 026672 022712 000273 CMP #273,(R2) ;SEQUENCE ERROR?
8404 026676 001011 BNE TS274-10 ;BR TO ERROR HALT ON SEQ ERROR
8405 026700 012737 026102 026100 MOV #A+2,@#A ;LOC A STORES ADDR A+2
8406 026706 010777 177166 PCN4: MOV PC,@A ;EFFECTIVE ADDR IS A+2
8407 026712 022737 026712 026102 CMP #PCN4+4,@#A+2 ;LOC A+2 STORES PC+4?
8408 026720 001404 BEQ TS274

```

```

8409 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
8410 ; CONDITIONAL BRANCH INST. AND <===
8411 ; REPLACE THE MOVE INSTRUCTION <===
8412 ; WHICH FOLLOWS W/ 766 <===
8413 026722 012742 000640 MOV #640,-(R2) ;MOVE TO MAILBOX # ***** 640 *****
8414 026726 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8415 026730 000000 HALT ;LOC A+2 STORED WRONG VALUE
8416 ; OR SEQUENCE ERROR
8417

```

```

8418 :*****
8419 :TEST 274 TEST PC AS SOURCE IN RELATIVE MODE ,USING R0
8420 :*****

```

```

8421 026732 005212 TS274: INC (R2) ;UPDATE TEST NUMBER
8422 026734 022712 000274 CMP #274,(R2) ;SEQUENCE ERROR?
8423 026740 001010 BNE TS275-10 ;BR TO ERROR HALT ON SEQ ERROR
8424 026742 005037 026100 CLR @#A ;CLEAR A
8425 026746 010767 177126 PCN5: MOV PC,A ;EFFECTIVE ADDR IS A
8426 026752 022737 026752 026100 CMP #PCN5+4,@#A ;LOC A STORES PC+4?
8427 026760 001404 BEQ TS275

```

```

8428 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
8429 ; CONDITIONAL BRANCH INST. AND <===
8430 ; REPLACE THE MOVE INSTRUCTION <===
8431 ; WHICH FOLLOWS W/ 767 <===
8432 026762 012742 000641 MOV #641,-(R2) ;MOVE TO MAILBOX # ***** 641 *****
8433 026766 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8434 026770 000000 HALT ;LOCATION A STORED WRONG VALUE
8435 ; OR SEQUENCE ERROR
8436

```

```

8437 :*****
8438 :SBTTL THE NEXT THREE TESTS EXERCISE MASKING ACTION OF MICROCODES.
8439 :*****

```

```

8440 :TEST 275 TEST SUB INSTRUCTION, SM=0, DM=2
8441 :*****

```

```

8442 026772 005212 TS275: INC (R2) ;UPDATE TEST NUMBER
8443 026774 022712 000275 CMP #275,(R2) ;SEQUENCE ERROR?
8444 027000 001013 BNE TS276-10 ;BR TO ERROR HALT ON SEQ ERROR
8445 027002 012737 052525 000000 MOV #052525,@#0 ;SET UP LOC 0

```



```

8446 027010 012701 050505      MOV      #050505,R1      ;SET UP R1
8447 027014 005000              CLR      R0              ;CLEAR R0
8448 027016 160120              SUB      R1,(R0)+        ;SUBTRACTION, SM=0,DM=2
8449 027020 022737 002020  C00000  CMP      #2020,@#0      ;CHECK DIFFERENCE AT LOC 0
8450 027026 001404              BEQ
8451                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
8452                                ;          CONDITIONAL BRANCH INST. AND <==
8453                                ;          REPLACE THE MOVE INSTRUCTION <--
8454                                ;          WHICH FOLLOWS W/ 764 <-
8455 027030 012742 000642      MOV      #6+2,-(R2)     ;MOVE TO MAILBOX # ***** 642 *****
8456 027034 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
8457 027036 000000              HALT                   ;WRONG RESULT FROM SUBTRACTION
8458                                ; OR SEQUENCE ERROR
8459

```

```

*****
:TEST 276      TEST MFPD WITH R0, IN MODE 2
*****

```

```

8460
8461 TS276:      INC      (R2)          ;UPDATE TEST NUMBER
8462              CMP      #276,(R2)      ;SEQUENCE ERROR?
8463              BNE      TS277-10      ;BR TO ERROR HALT ON SEQ ERROR
8464 027040 005212 000276      MOV      #052525,@#0    ;SET UP LOC 0
8465 027042 022712              CLR      R0              ;CLEAR R0
8466 027046 001020              MOV      #170000,PS     ;SET USER MODE ON, CURRENT & PREVIOUS
8467 027050 012737 052525 000000  MOV      #USTBOT,R6     ;SET USER STACK POINTER
8468 027056 005000              MFPD     (R0)+          ;MODE 2, MFPD
8469 027060 012767 170000 150710  CLR      PS              ;SET KERNEL MODE
8470 027066 012706 027324      CMP      #052525,USTBOT-2 ;CHECK DATA ON STACK
8471 027072 106520              BEQ      BRMFPD         ;BR IF NO ERROR
8472 027100 022767 052525 000214  MOV      #643,-(R2)     ;MOVE TO MAILBOX # ***** 643 *****
8473 027106 001404              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
8474 027110 012742 000643      HALT                   ;INCORRECT DATA FROM MFPD
8475 027114 005242
8476 027116 000000
8477 027120      BRMFPD:
8478
8479

```

```

*****
:TEST 277      TEST MTPD WITH R0, IN MODE 2
*****

```

```

8480
8481 TS277:      INC      (R2)          ;UPDATE TEST NUMBER
8482 027120 005212 000277      CMP      #277,(R2)      ;SEQUENCE ERROR?
8483 027122 022712              BNE      END1           ;BR TO ERROR HALT ON SEQ ERROR
8484 027126 001026              MOV      #170000,PS     ;SET USER MODE ON, CURRENT & PREVIOUS
8485 027130 012767 170000 150640  MOV      #USTBOT,R6     ;SET USER STACK POINTER
8486 027136 012706 027324      MOV      #125252,-(R6)  ;PUSH DATA IN USER STACK
8487 027142 012746 125252              MOV      #0,@#0        ;CLEAR LOC 0
8488 027146 012737 000000 000000  CLR      R0              ;CLEAR R0
8489 027154 005000              MTPD     (R0)+          ;MODE 2, MTPD
8490 027156 106620              CLR      PS              ;SET KERNEL MODE
8491 027160 005067 150612      CMP      #125252,@#0    ;CHECK DATA ON LOC 0
8492 027164 022737 125252 000000  BEQ      SECPRT         ;BR TO TRAP TEST IF NO ERROR
8493 027172 001514              MOV      #644,-(R2)     ;MOVE TO MAILBOX # ***** 644 *****
8494 027174 012742 000644      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
8495 027200 005242              HALT                   ;INCORRECT DATA FROM MTPD
8496 027202 000000
8497 027204      END1:
8498 027204 012742 000645      MOV      #645,-(R2)     ;MOVE TO MAILBOX # ***** 645 *****
8499 027210 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
8500 027212 000000              HALT                   ;SEQUENCE ERROR
8501

```

```

8502
8503 027214 000402      BRTAB: BR      .+6
8504 027216 001002      BNE      .+6
8505 027220 001402      BEQ      .+6
8506 027222 002002      BGE      .+6
8507 027224 002402      BLT      .+6
8508 027226 003002      BGT      .+6
8509 027230 003402      BLE      .+6
8510 027232 100002      BPL      .+6
8511 027234 100402      BMI      .+3
8512 027236 101002      BHI      .+6
8513 027240 101402      BLOS     .+6
8514 027242 102002      BVC      .+6
8515 027244 102402      BVS      .+6
8516 027246 103002      BCC      .+6      ;SAME AS BHIS
8517 027250 103402      BCS      .+6      ;SAME AS BLO
8518
8519      000002      .RADIX 2
8520 027252 177777      YNTAB: 1111111111111111      ;BR
8521 027254 170360      1111000011110000      ;BNE: Z=0
8522 027256 007417      0000111100001111      ;BEQ: Z=1
8523 027260 146063      1100110000110011      ;BGE: N XOR V -0
8524 027262 031714      0011001111001100      ;BLT: N XOR V -1
8525 027264 140060      1100000000110000      ;BGT: Z+(N XOR V) =0
8526 027266 037717      0011111111001111      ;BLE: Z+(N XOR V) -1
8527
8528 027270 177400      1111111100000000      ;BPL: N=0
8529 027272 000377      0000000011111111      ;BMI: N=1
8530 027274 120240      1010000010100000      ;BHI: C+Z=0
8531 027276 057237      0101111101011111      ;BLOS: C+Z=1
8532 027300 146314      1100110011001100      ;BVC: V=0
8533 027302 031463      0011001100110011      ;BVS: V=1
8534 027304 125252      1010101010101010      ;BCC: C=0
8535 027306 052525      0101010101010101      ;BCS: C=1
8536      000010      .RADIX 8
8537
8538      .EVEN
8539 027310 000006      .BLKW 6
8540 027324      USTBOT:
8541      ;*****
8542      ; THE FOLLOWING ARE SPECIAL CPU TRAP
8543      ; HANDLERS TO TRAP AND REPORT SPECIAL TRAPS.
8544      ;
8545      ;*****
8546
8547 027324      T04:
8548 027324 012742 000646      MOV      #646,-(R2)      ;MOVE TO MAILBOX # ***** 64 *****
8549 027330 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
8550 027332 000000      HALT                    ;TRAPPED THRU LOC. 4
8551 027334      T010:
8552 027334 012742 000647      MOV      #647,-(R2)      ;MOVE TO MAILBOX # ***** 647 *****
8553 027340 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
8554 027342 000000      HALT                    ;TRAPPED THRU LOC. 10
8555 027344      T014:
8556 027344 012742 000650      MOV      #650,-(R2)      ;MOVE TO MAILBOX # ***** 650 *****
8557 027350 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR

```



8581  
8582 J00000  
8583  
8584  
8585  
8586  
8587  
8588  
8589  
8590  
8591  
8592  
8593  
8594  
8595  
8596  
8597  
8598  
8599  
8600 000006  
8601 000006  
8602 000003  
8603 000001  
8604 000005  
8605 000002  
8606 000000  
8607 000003  
8608 000004  
8609 000004  
8610 000014  
8611 000030  
8612 000020  
8613 000034  
8614 177564  
8615 177560  
8616 177564  
8617 177566  
8618 000240  
8619 000240  
8620 177776  
8621 000077  
8622 000010  
8623 004700  
8624 000100  
8625 177776  
8626 001000  
8627  
8628  
8629

.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

ITCSR=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

```

8630
8631 ;SPECIAL CASE OF ODD;.EVEN .BYTE AND REGISTER 6
8632 000000 HERE=0
8633
8634 027424 000167 000026 JMP TESTN1
8635 027430 000000 K1: 0
8636 027432 000000 K2: 0
8637 027434 000000 K3: 0
8638 027436 000000 K4: 0
8639 027440 000000 K5: 0
8640 027442 000000 K6: 0
8641 027444 052525 K7: 052525
8642 027446 052400 K10: 052400
8643 027450 000000 K11: 0
8644 027452 000000 K12: 0
8645 027454 000176 SWR: 176
8646
8647 027456 012767 000176 177770 TESTN1: MOV #176,SWR ;POINT TO LOC 176 AS SOFTWARE SWITCH REGISTER
8648 027464 032737 000001 000320 BIT #1,@$ENV ;ARE WE RUNNING IN APT MODE
8649 027472 001403 BEQ 1$ ;IF NO USE SOFTWARE SWITCH REGISTER
8650 027474 012767 000322 177752 MOV $$SWREG,SWR ;IF YES USE APT SWITCH REGISTER
8651 027502 032777 000001 177744 1$: BIT #1,@SWR ;IF BIT IS NON-ZERO SHIP ALL TRAPS TESTS
8652 027510 001402 BEQ 2$
8653 027512 000167 013006 JMP THRPRT
8654 027516 2$:
8655 ;*****
8656 ;TEST 300 TEST AUTO INCREMENT AND DECREMENT OF R6 FOR WORD AND BYTES
8657 ;*****
8658 027516 005212 TS300: INC (R2) ;UPDATE TEST NUMBER
8659 027520 022712 000300 CMP #300,(R2) ;SEQUENCE ERROR?
8660 027524 001127 BNE TS301-10 ;BR TO ERROR HALT ON SEQ ERROR
8661 027526 005006 CLR %6
8662 027530 112667 150244 MOV#B (6)+,HERE ;SIX SHOULD INCREMENT BY TWO
8663 027534 020627 000002 CMP %6,#2
8664 027540 001404 BEQ BR1
8665 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
8666 ; CONDITIONAL BRANCH INST. AND <-
8667 ; REPLACE THE MOVE INSTRUCTION <== -
8668 ; WHICH FOLLOWS W/ 771 <= -
8669 027542 012742 000656 MOV #656,-(R2) ;MOVE TO MAILBOX # ***** 656 *****
8670 027546 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8671 027550 000000 HALT ;R6 DID NOT AUTO INCREMENT BY TWO
8672
8673 027552 012706 001000 BR1: MOV #1000,%6
8674 027556 114627 000000 MOV#B -(6),#HERE ;SHOULD DECREMENT BY TWO
8675 027562 020627 000776 CMP %6,#776
8676 027566 001404 BEQ BR2
8677 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
8678 ; CONDITIONAL BRANCH INST. AND <-- -
8679 ; REPLACE THE MOVE INSTRUCTION <- =
8680 ; WHICH FOLLOWS W/ 756 <- -
8681 027570 012742 000657 MOV #657,-(R2) ;MOVE TO MAILBOX # ***** 657 *****
8682 027574 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8683 027576 000000 HALT ;R6 DID NOT AUTO DECREMENT BY 2
8684
8685 027600 005006 BR2: CLR %6

```

```

8686 027602 112626          MOVB  (6)+,(6)+      ;DOUBLES AUTO INCREMENT OF R6
8687 027604 020627 000004  CMP   %6,#4
8688 027610 001404          BEQ   BR3
8689                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- ==
8690                                     ;         CONDITIONAL BRANCH INST. AND <-
8691                                     ;         REPLACE THE MOVE INSTRUCTION <-
8692                                     ;         WHICH FOLLOWS W/ 745 <==
8693 027612 012742 000660  MOV   #660,-(R2)    ;MOVE TO MAILBOX # ***** 660 *****
8694 027616 005242          INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8695 027620 000000          HALT                ;WRONG AUTO INCREMENT OF R6
8696
8697 027622 005006          BR3:  CLR   %6
8698 027624 005004          CLR   %4
8699 027626 122624          CMPB  (6)+,(4)+    ;TEST INCREMENT OF R6
8700 027630 020627 000002  CMP   %6,#2
8701 027634 001404          BEQ   BR4
8702                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
8703                                     ;         CONDITIONAL BRANCH INST. AND <-
8704                                     ;         REPLACE THE MOVE INSTRUCTION <-
8705                                     ;         WHICH FOLLOWS W/ 733 <-
8706 027636 012742 000661  MOV   #661,-(R2)    ;MOVE TO MAILBOX # ***** 661 *****
8707 027642 005242          INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8708 027644 000000          HALT                ;WRONG INCREMENT OF R6
8709
8710 027646 005006          BR4:  CLR   %6
8711 027650 005004          CLR   %4
8712 027652 122426          CMPB  (4)+,(6)+    ;TEST INCREMENT OF R6
8713 027654 020627 000002  CMP   %6,#2
8714 027660 001404          BEQ   BR5
8715                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
8716                                     ;         CONDITIONAL BRANCH INST. AND <-
8717                                     ;         REPLACE THE MOVE INSTRUCTION <-
8718                                     ;         WHICH FOLLOWS W/ 721 <-
8719 027662 012742 000662  MOV   #662,-(R2)    ;MOVE TO MAILBOX # ***** 662 *****
8720 027666 005242          INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8721 027670 000000          HALT                ;WRONG INCREMENT OF R6
8722
8723 027672 005006          BR5:  CLR   %6
8724 027674 005004          CLR   %4
8725 027676 122624          CMPB  (6)+,(4)+    ;TEST INCREMENT OF R4
8726 027700 020427 000001  CMP   %4,#1
8727 027704 001404          BEQ   BR6
8728                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
8729                                     ;         CONDITIONAL BRANCH INST. AND <---
8730                                     ;         REPLACE THE MOVE INSTRUCTION <-
8731                                     ;         WHICH FOLLOWS W/ 707 <---
8732 027706 012742 000663  MOV   #663,-(R2)    ;MOVE TO MAILBOX # ***** 663 *****
8733 027712 005242          INC   -(R2)         ;SET MSGTYP TO FATAL ERROR
8734 027714 000000          HALT                ;WRONG INCREMENT OF R4
8735 027716 005006          BR6:  CLR   %6
8736 027720 005004          CLR   %4
8737 027722 122426          CMPB  (4)+,(6)+    ;TEST INCREMENT OF R6
8738 027724 020627 000002  CMP   %6,#2
8739 027730 001404          BEQ   BR7
8740                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
8741                                     ;         CONDITIONAL BRANCH INST. AND <-

```

```

8742                                     :           REPLACE THE MOVE INSTRUCTION <====
8743                                     :           WHICH FOLLOWS W/ 675           <====
8744 027732 012742 000664                MOV    #664,-(R2)                :MOVE TO MAILBOX # ***** 664 *****
8745 027736 005242                        INC    -(R2)                    :SET MSGTYP TO FATAL ERROR
8746 027740 000000                        HALT                               :WRONG INCREMENT OF R6
8747
8748 027742 005006                BR7:  CLR    %6
8749 027744 005004                CLR    %4
8750 027746 122426                CMPB   (4)+,(6)+                :TEST INCREMENT OF R4
8751 027750 020427 000001                CMP    %4,#1
8752 027754 001404                BEQ    BR10
8753                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8754                                     :           CONDITIONAL BRANCH INST. AND <====
8755                                     :           REPLACE THE MOVE INSTRUCTION <====
8756                                     :           WHICH FOLLOWS W/ 663           <====
8757 027756 012742 000665                MOV    #665,-(R2)                :MOVE TO MAILBOX # ***** 665 *****
8758 027762 005242                        INC    -(R2)                    :SET MSGTYP TO FATAL ERROR
8759 027764 000000                        HALT                               :WRONG INCREMENT OF R4
8760
8761 027766 012706 001000                BR10: MOV    #1000,%6
8762 027772 124627 000000                CMPB   -(6),#HERE                :TEST DECREMENT OF R6
8763 027776 022706 000776                CMP    #776,%6
8764 030002 001404                BEQ    TS301
8765                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -
8766                                     :           CONDITIONAL BRANCH INST. AND <= -
8767                                     :           REPLACE THE MOVE INSTRUCTION <=
8768                                     :           WHICH FOLLOWS W/ 650           <-
8769 030004 012742 000666                MOV    #666,-(R2)                :MOVE TO MAILBOX # ***** 666 *****
8770 030010 005242                        INC    -(R2)                    :SET MSGTYP TO FATAL ERROR
8771 030012 000000                        HALT                               :WRONG DECREMENT OF R6,OR WRONG $TSTNM
8772                                     : OR SEQUENCE ERROR
8773
8774 :*****
8775 :TEST 301 TEST TRANSFER OF .BYTE USING R6
8776 :*****
8776 030014 005212                TS301: INC    (R2)                :UPDATE TEST NUMBER
8777 030016 022712 000301                CMP    #301,(R2)                :SEQUENCE ERROR?
8778 030022 001133                BNE    TS302-10                :BR TO ERROR HALT ON SEQ ERROR
8779 030024 012767 123456 177406                MOV    #123456,K5
8780 030032 012767 050505 177370                MOV    #050505,K1
8781 030040 012705 027430                MOV    #K1,%5                    :%5=(050505)K1
8782 030044 012706 027440                MOV    #K5,%6                    :%6=(123456)K5
8783 030050 112625                MOVB   (6)+,(5)+                :LOW .BYTE OF R6 TO R5
8784 030052 022767 050456 177350                CMP    #050456,K1
8785 030060 001404                BEQ    BR11
8786                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--
8787                                     :           CONDITIONAL BRANCH INST. AND <=-
8788                                     :           REPLACE THE MOVE INSTRUCTION <-
8789                                     :           WHICH FOLLOWS W/ 760           <--
8790 030062 012742 000667                MOV    #667,-(R2)                :MOVE TO MAILBOX # ***** 667 *****
8791 030066 005242                        INC    -(R2)                    :SET MSGTYP TO FATAL ERROR
8792 030070 000000                        HALT                               :FALSE TRANSFER OF .BYTE
8793
8794 030072 012767 123456 177340                BR11: MOV    #123456,K5
8795 030100 012767 050505 177322                MOV    #050505,K1
8796 030106 012705 027430                MOV    #K1,%5                    :%5(050505)K1
8797 030112 012706 027442                MOV    #K6,%6                    :%6(123456)K5

```





```
*****  
:TEST 302 TEST BYTE OPERATION WITH SEQUENTIAL ODD-EVEN ADDRESS  
*****  
8854  
8855  
8856  
8857 030322 005212  
8858 030324 022712 000302  
8859 030330 001074  
8860 030332 126767 177106 177105  
8861 030340 001404  
8862  
8863  
8864  
8865  
8866 030342 012742 000674  
8867 030346 005242  
8868 030350 000000  
8869  
8870 030352 126767 177067 177064 BR15:  
8871 030360 001404  
8872  
8873  
8874  
8875  
8876 030362 012742 000675  
8877 030366 005242  
8878 030370 000000  
8879  
8880 030372 126767 177051 177044 BR16:  
8881 030400 001404  
8882  
8883  
8884  
8885  
8886 030402 012742 000676  
8887 030406 005242  
8888 030410 000000  
8889  
8890 030412 126767 177030 177022 BR17:  
8891 030420 001404  
8892  
8893  
8894  
8895  
8896 030422 012742 000677  
8897 030426 005242  
8898 030430 000000  
8899 030432 126767 177007 177007 BR20:  
8900 030440 001404  
8901  
8902  
8903  
8904  
8905 030442 012742 000700  
8906 030446 005242  
8907 030450 000000  
8908  
8909 030452 126767 176770 176767 BR21:  
*****  
:TS302: INC (R2) :UPDATE TEST NUMBER  
:CMP #302,(R2) :SEQUENCE ERROR?  
:BNE TS303-10 :BR TO ERROR HALT ON SEQ ERROR  
:CMPB K7,K7+1 :SAME .WORD LOW TO HIGH  
:BEQ BR15  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===  
: CONDITIONAL BRANCH INST. AND <===  
: REPLACE THE MOVF INSTRUCTION <===  
: WHICH FOLLOWS W/ 773 <===  
:MOVE TO MAILBOX # ***** 674 *****  
:SET MSGTYP TO FATAL ERROR  
:SHOULD COMPARE LOW TO HIGH  
:CMPB K7+1,K7 :COMPARE ODD TO .EVEN SAME .WORD  
:BEQ BR16  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===  
: CONDITIONAL BRANCH INST. AND <===  
: REPLACE THE MOVE INSTRUCTION <===  
: WHICH FOLLOWS W/ 763 <===  
:MOVE TO MAILBOX # ***** 675 *****  
:SET MSGTYP TO FATAL ERROR  
:ODD TO .EVEN .BYTE FAILURE  
:CMPB K10+1,K7 :SEQUENTIAL .BYTES  
:BEQ BR17  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -  
: CONDITIONAL BRANCH INST. AND <== -  
: REPLACE THE MOVE INSTRUCTION <== -  
: WHICH FOLLOWS W/ 753 <== -  
:MOVE TO MAILBOX # ***** 676 *****  
:SET MSGTYP TO FATAL ERROR  
:ODD TO .EVEN FAILED  
:CMPB K10,K6  
:BEQ BR20  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==  
: CONDITIONAL BRANCH INST. AND <==  
: REPLACE THE MOVE INSTRUCTION <==  
: WHICH FOLLOWS W/ 743 <==  
:MOVE TO MAILBOX # ***** 677 *****  
:SET MSGTYP TO FATAL ERROR  
:.EVEN TO EVEN FAILED  
:CMPB K7+1,K10+1  
:BEQ BR21  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -  
: CONDITIONAL BRANCH INST. AND <== -  
: REPLACE THE MOVE INSTRUCTION <== -  
: WHICH FOLLOWS W/ 733 <== -  
:MOVE TO MAILBOX # ***** 700 *****  
:SET MSGTYP TO FATAL ERROR  
:ODD TO ODD FAILED  
:CMPB K10,K10+1
```

```

8910 030460 001004      BNE      BR22
8911                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8912                                     ;          CONDITIONAL BRANCH INST. AND <====
8913                                     ;          REPLACE THE MOVE INSTRUCTION <====
8914                                     ;          WHICH FOLLOWS W/ 723 <====
8915 030462 012742 000701  MOV      #701,-(R2) ;MOVE TO MAILBOX # ***** 701 *****
8916 030466 005242      INC      -(R2)   ;SET MSGTYP TO FATAL ERROR
8917 030470 000000      HALT     ;LOW TO HIGH IN SAME .WORD FAILED
8918
8919 030472 126767 176751 176747 BR22:  CMPB    K10+1,K10+1
8920 030500 001404      BEQ      BR23
8921                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8922                                     ;          CONDITIONAL BRANCH INST. AND <====
8923                                     ;          REPLACE THE MOVE INSTRUCTION <====
8924                                     ;          WHICH FOLLOWS W/ 713 <====
8925 030502 012742 000702  MOV      #702,-(R2) ;MOVE TO MAILBOX # ***** 702 *****
8926 030506 005242      INC      -(R2)   ;SET MSGTYP TO FATAL ERROR
8927 030510 000000      HALT     ;HIGH TO LOW IN SAME .WORD FAILED
8928
8929 C30512 126767 176730 176725 BR23:  CMPB    K10,K7+1
8930 030520 001004      BNE      TS303
8931                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8932                                     ;          CONDITIONAL BRANCH INST. AND <====
8933                                     ;          REPLACE THE MOVE INSTRUCTION <====
8934                                     ;          WHICH FOLLOWS W/ 703 <====
8935 030522 012742 000703  MOV      #703,-(R2) ;MOVE TO MAILBOX # ***** 703 *****
8936 030526 005242      INC      -(R2)   ;SET MSGTYP TO FATAL ERROR
8937 030530 000000      HALT     ;.EVEN TO ODD FAILED,OR WRONG $STNM
8938                                     ; OR SEQUENCE ERROR
8939
8940
8941
8942
8943
8944 030532 005212      INC      (R2)   ;UPDATE TEST NUMBER
8945 030534 022712 000303  CMP      #303,(R2) ;SEQUENCE ERROR?
8946 030540 001053      BNE      TS304-10 ;BR TO ERROR HALT ON SEQ ERROR
8947 030542 000277      SCC      ;SET STATUS
8948 030544 005067 147226  CLR      STATUS ;CLEAR STATUS
8949 030550 103004      BCC     BR33
8950                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
8951                                     ;          CONDITIONAL BRANCH INST. AND <= -
8952                                     ;          REPLACE THE MOVE INSTRUCTION <=- -
8953                                     ;          WHICH FOLLOWS W/ 773 <==
8954 030552 012742 000704  MOV      #704,-(R2) ;MOVE TO MAILBOX # ***** 704 *****
8955 030556 005242      INC      -(R2)   ;SET MSGTYP TO FATAL ERROR
8956 030560 000000      HALT     ;C NOT CLEAR
8957 030562
8958 030562 102004      BVC     BR34
8959                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=- -
8960                                     ;          CONDITIONAL BRANCH INST. AND <=- -
8961                                     ;          REPLACE THE MOVE INSTRUCTION <--
8962                                     ;          WHICH FOLLOWS W/ 766 <-
8963 030564 012742 000705  MOV      #705,-(R2) ;MOVE TO MAILBOX # ***** 705 *****
8964 030570 005242      INC      -(R2)   ;SET MSGTYP TO FATAL ERROR
8965 030572 000000      HALT     ;V NOT CLEAR

```

\*\*\*\*\*  
:TEST 303 TEST THE CC BITS  
\*\*\*\*\*

```

TS303: INC      (R2)   ;UPDATE TEST NUMBER
      CMP      #303,(R2) ;SEQUENCE ERROR?
      BNE     TS304-10 ;BR TO ERROR HALT ON SEQ ERROR
      SCC      ;SET STATUS
      CLR     STATUS ;CLEAR STATUS
      BCC     BR33

```

BR33:

```

      BVC     BR34
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=- -
      ;          CONDITIONAL BRANCH INST. AND <=- -
      ;          REPLACE THE MOVE INSTRUCTION <--
      ;          WHICH FOLLOWS W/ 766 <-
      MOV     #705,-(R2) ;MOVE TO MAILBOX # ***** 705 *****
      INC     -(R2)   ;SET MSGTYP TO FATAL ERROR
      HALT    ;V NOT CLEAR

```



```
9022 ; OP SEQUENCE ERROR
9023 ;*****
9024 ;TEST 304 TEST THAT A TRAP OCCURS ON A RESERVED INSTRUCTION
9025 ;*****
9026 030700 005212 TS304: INC (R2) ;UPDATE TEST NUMBER
9027 030702 022712 000304 CMP #304,(R2) ;SEQUENCE ERROR?
9028 030706 001006 BNE RETA ;BR TO ERROR HALT ON SEQ ERROR
9029 030710 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9030 030714 012767 030734 147066 MOV #RETAH,RTRAP ;RETURN LOCATION
9031 030722 000077 TRAPA ;RESERVED INSTRUCTION, SHOULD TRAP
9032 030724 RETA:
9033 030724 012742 000714 MOV #714,-(R2) ;MOVE TO MAILBOX # ***** 714 *****
9034 030730 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9035 030732 000000 HALT ;RESERVE INSTRUCTION DIDN'T TRAP,OR WRONG $STNM
9036 030734 RETAH:
9037 ;*****
9038 ;TEST 305 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9039 ;*****
9040 030734 005212 TS305: INC (R2) ;UPDATE TEST NUMBER
9041 030736 022712 000305 CMP #305,(R2) ;SEQUENCE ERROR?
9042 030742 001011 BNE TS306-10 ;BR TO ERROR HALT ON SEQ ERROR
9043 030744 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9044 030750 012767 030760 147032 MOV #RETB,RTRAP ;RETURN POINTER
9045 030756 000077 TRAPA ;RESERVED INSTRUCTION
9046 030760 020627 000774 RETB: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
9047 030764 001404 BEQ TS306
9048 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9049 ; CONDITIONAL BRANCH INST. AND <- -
9050 ; REPLACE THE MOVE INSTRUCTION <- -
9051 ; WHICH FOLLOWS W/ 766 <- -
9052 030766 012742 000715 MOV #715,-(R2) ;MOVE TO MAILBOX # ***** 715 *****
9053 030772 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9054 030774 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $STNM
9055 ; OR SEQUENCE ERROR
9056 ;*****
9057 ;TEST 306 TEST THAT PROPER P.C. IS SAVED
9058 ;*****
9059 030776 005212 TS306: INC (R2) ;UPDATE TEST NUMBER
9060 031000 022712 000306 CMP #306,(R2) ;SEQUENCE ERROR?
9061 031004 001012 BNE TS307-10 ;BR TO ERROR HALT ON SEQ ERROR
9062 031006 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9063 031012 012767 031022 146770 MOV #RETC,RTRAP ;RETURN FROM TRAP POINTER
9064 031020 000077 INSTC: TRAPA ;TRAP ON THIS INSTRUCTION
9065 031022 022767 031022 147744 RETC: CMP #,BUFF-4 ;CHECK FOR INCREMENTED P.C.
9066 031030 001404 BEQ TS307
9067 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
9068 ; CONDITIONAL BRANCH INST. AND <- - =
9069 ; REPLACE THE MOVE INSTRUCTION <-
9070 ; WHICH FOLLOWS W/ 765 <-
9071 031032 012742 000716 MOV #716,-(R2) ;MOVE TO MAILBOX # ***** 716 *****
9072 031036 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9073 031040 000000 HALT ;INCORRECT P.C. OR WRONG $STNM
9074 ; OR SEQUENCE ERROR
9075 ;*****
9076 ;TEST 307 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
9077 ;*****
```

```
9078 031042 005212      TS307: INC      (R2)          ;UPDATE TEST NUMBER
9079 031044 022712 000307  CMP      #307,(R2)       ;SEQUENCE ERROR?
9080 031050 001037      BNE      TS310-10       ;BR TO ERROR HALT ON SEQ ERROR
9081 031052 012706 001000  MOV      #BUFF,SP      ;SET UP
9082 031056 012767 031074 146724  MOV      #RETD,RTRAP   ;SET UP
9083 031064 005067 146706  CLR      CC            ;CLEAR CC AND PRIORITY
9084 031070 000257      CCC
9085 031072 000077      TRAPA
9086 031074 026727 147676 000000 RETD:  CMP      BUFF-2,#0   ;TRAP
9087 031102 001404      BEQ      1$           ;TEST THAT OLD STATUS WENT TO STACK
9088
9089
9090
9091
9092 031104 012742 000717  MOV      #717,-(R2)    ;MOVE TO MAILBOX # ***** 717 *****
9093 031110 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9094 031112 000000      HALT
9095 031114 012706 001000 1$:  MOV      #BUFF,SP      ;INCORRECT STATUS
9096 031120 012767 031140 146662  MOV      #RETE,RTRAP   ;SET UP
9097 031126 012767 000357 146642  MOV      #357,CC       ;SET PRIORITY
9098 031134 000277      SCC
9099 031136 000077      TRAPA
9100 031140 026727 147632 000357 RETE:  CMP      BUFF-2,#357   ;SET CC
9101 031146 001404      BEQ      TS310        ;TRAP
9102
9103
9104
9105
9106 031150 012742 000720  MOV      #720,-(R2)    ;COMPARES STATUS ON STACK
9107 031154 005242      INC      -(R2)        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9108 031156 000000      HALT                 ; CONDITIONAL BRANCH INST. AND <
9109
9110
9111
9112
9113 031160 005212      *****
9114 031162 022712 000310  TS310: INC      (R2)          ;TEST 310 TEST THAT 'NEW' STATUS IS CORRECT
9115 031166 001110      *****
9116 031170 012706 001000  CMP      #310,(R2)     ;UPDATE TEST NUMBER
9117 031174 012767 031210 146606  BNE      STPP          ;SEQUENCE ERROR?
9118 031202 005067 146604  MOV      #BUFF,SP      ;BR TO ERROR HALT ON SEQ ERROR
9119 031206 000077      TRAPA
9120 031210 100004      RETF:  CLR      RTRAP+2  ;CLEAR FUTURE PRIORITY AND CC
9121 031210 100004      BPL      1$           ;TEST FOR 'C' CLEARED
9122
9123
9124
9125
9126 031212 012742 000721  MOV      #721,-(R2)    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9127 031216 005242      INC      -(R2)        ; CONDITIONAL BRANCH INST. AND <==
9128 031220 000000      HALT                 ; REPLACE THE MOVE INSTRUCTION < --
9129 031222 1$:  BNE      2$           ; WHICH FOLLOWS W/ 766 <-
9130 031222 001004      BNE      2$           ;MOVE TO MAILBOX # ***** 721 *****
9131
9132
9133
9134
```

```
9134  
9135 031224 012742 000722      MOV      #722,-(R2)      ; WHICH FOLLOWS W/ 761  
9136 031230 005242      INC      -(R2)          ; MOVE TO MAILBOX # ***** 722 *****  
9137 031232 000000      HALT                    ; SET MSGTYP TO FATAL ERROR  
9138 031234                      2$:                   ; Z NOT CLEARED  
9139 031234 102004      BVC      3$  
9140                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-  
9141                      ; CONDITIONAL BRANCH INST. AND <-  
9142                      ; REPLACE THE MOVE INSTRUCTION <-  
9143                      ; WHICH FOLLOWS W/ 754 <-  
9144 031236 012742 000723      MOV      #723,-(R2)      ; MOVE TO MAILBOX # ***** 723 *****  
9145 031242 005242      INC      -(R2)          ; SET MSGTYP TO FATAL ERROR  
9146 031244 000000      HALT                    ; V NOT CLEARED  
9147 031246                      3$:                   ;  
9148 031246 103004      BCC      4$  
9149                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <  
9150                      ; CONDITIONAL BRANCH INST. AND <  
9151                      ; REPLACE THE MOVE INSTRUCTION <  
9152                      ; WHICH FOLLOWS W/ 747 <-  
9153 031250 012742 000724      MOV      #724,-(R2)      ; MOVE TO MAILBOX # ***** 724 *****  
9154 031254 005242      INC      -(R2)          ; SET MSGTYP TO FATAL ERROR  
9155 031256 000000      HALT                    ; C NOT CLEARED  
9156 031260 032767 000340 146510 4$:  BIT      #340,CC  
9157 031266 001404      BEQ      5$  
9158                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <  
9159                      ; CONDITIONAL BRANCH INST. AND <  
9160                      ; REPLACE THE MOVE INSTRUCTION <  
9161                      ; WHICH FOLLOWS W/ 737 <  
9162 031270 012742 000725      MOV      #725,-(R2)      ; MOVE TO MAILBOX # ***** 725 *****  
9163 031274 005242      INC      -(R2)          ; SET MSGTYP TO FATAL ERROR  
9164 031276 000000      HALT                    ; PRIORITY NOT ZERO  
9165 031300 012706 001000 5$:  MOV      #BUFF,SP  
9166 031304 012767 031322 146476  MOV      #RETG,RTRAP  
9167 031312 012767 000357 146472  MOV      #357,RTRAP+2  
9168 031320 000077      TRAPA  
9169 031322                      RETG:                   ; SET NEW 'CC' AND PRIORITY  
9170 031322 100404      BMI      1$  
9171                      ; TRAP HERE  
9172                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <  
9173                      ; CONDITIONAL BRANCH INST. AND <  
9174                      ; REPLACE THE MOVE INSTRUCTION <  
9175                      ; WHICH FOLLOWS W/ 721 <  
9176 031324 012742 000726      MOV      #726,-(R2)      ; MOVE TO MAILBOX # ***** 726 *****  
9177 031330 005242      INC      -(R2)          ; SET MSGTYP TO FATAL ERROR  
9178 031332 000000      HALT                    ; N NOT SET  
9179 031334 001404      BEQ      2$  
9180                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-  
9181                      ; CONDITIONAL BRANCH INST. AND <-  
9182                      ; REPLACE THE MOVE INSTRUCTION <-  
9183                      ; WHICH FOLLOWS W/ 714 <-  
9184 031336 012742 000727      MOV      #727,-(R2)      ; MOVE TO MAILBOX # ***** 727 *****  
9185 031342 005242      INC      -(R2)          ; SET MSGTYP TO FATAL ERROR  
9186 031344 000000      HALT                    ; Z NOT SET  
9187 031346                      2$:                   ;  
9188 031346 102404      BVS      3$  
9189                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
```









```
9358 032032 012742 000744      MOV      #744,-(R2)      ;MOVE TO MAILBOX # ***** 744 *****
9359 032036 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9360 032040 000000              HALT                    ;PRIORITY NOT ZERO
9361 032042 012706 001000      5$:  MOV      #BUFF,SP
9362 032046 012767 032064 145760  MOV      #RETG1,RTRAP1
9363 032054 012767 000357 145754  MOV      #357,RTRAP1+2 ;SET NEW 'CC' AND PRIORITY
9364 032062 104400              TRAP                    ;TRAP HERE
9365 032064                      RETG1:
9366 032064 100404              BMI      1$
9367                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
9368                                ;          CONDITIONAL BRANCH INST. AND <===
9369                                ;          REPLACE THE MOVE INSTRUCTION <==
9370                                ;          WHICH FOLLOWS W/ 721 <===
9371 032066 012742 000745      MOV      #745,-(R2)      ;MOVE TO MAILBOX # ***** 745 *****
9372 032072 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9373 032074 000000              HALT                    ;N NOT SET
9374 032076                      1$:
9375 032076 001404              BEQ      2$
9376                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
9377                                ;          CONDITIONAL BRANCH INST. AND <===
9378                                ;          REPLACE THE MOVE INSTRUCTION <==
9379                                ;          WHICH FOLLOWS W/ 714 <=
9380 032100 012742 000746      MOV      #746,-(R2)      ;MOVE TO MAILBOX # ***** 746 *****
9381 032104 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9382 032106 000000              HALT                    ;Z NOT SET
9383 032110                      2$:
9384 032110 102404              BVS      3$
9385                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
9386                                ;          CONDITIONAL BRANCH INST. AND <==
9387                                ;          REPLACE THE MOVE INSTRUCTION <=
9388                                ;          WHICH FOLLOWS W/ 707 <=
9389 032112 012742 000747      MOV      #747,-(R2)      ;MOVE TO MAILBOX # ***** 747 *****
9390 032116 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9391 032120 000000              HALT                    ;V NOT SET
9392 032122                      3$:
9393 032122 103404              BCS      4$
9394                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < =
9395                                ;          CONDITIONAL BRANCH INST. AND <
9396                                ;          REPLACE THE MOVE INSTRUCTION <===
9397                                ;          WHICH FOLLOWS W/ 702 < ==
9398 032124 012742 000750      MOV      #750,-(R2)      ;MOVE TO MAILBOX # ***** 750 *****
9399 032130 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9400 032132 000000              HALT                    ;C NOT SET
9401 032134 016706 145636      4$:  MOV      CC,SP
9402 032140 042706 000017      BIC      #17,SP
9403 032144 022706 000340      CMP      #340,SP
9404 032150 001404              BEQ      TS316
9405                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
9406                                ;          CONDITIONAL BRANCH INST. AND <=
9407                                ;          REPLACE THE MOVE INSTRUCTION <==
9408                                ;          WHICH FOLLOWS W/ 667 < ==
9409 032152 012742 000751      MOV      #751,-(R2)      ;MOVE TO MAILBOX # ***** 751 *****
9410 032156 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9411 032160 000000              HALT                    ;PRIORITY WAS CHANGED,OR WRONG $1STNM
9412                                ; OR SEQUENCE ERROR
9413
```

\*\*\*\*\*

```
9414 ;TEST 316 TEST THAT ALL COMBINATION OF 'TRAP' WILL CAUSE A TRAP
9415 ;*****
9416 032162 005212 TS316: INC (R2) ;UPDATE TEST NUMBER
9417 032164 022712 000316 CMP #316,(R2) ;SEQUENCE ERROR?
9418 032170 001011 BNE BR45 ;BR TO ERROR HALT ON SEQ ERROR
9419 ;***** F11 **** ADD +376 TO SHORTEN TEST
9420 032172 012767 104776 000012 MOV #TRAP+376,RB1 ;INITIALIZE BASE TRAP INSTRUCTION
9421 032200 012767 032224 145626 MOV #RA1,34 ;RETURN FROM TRAP TO RA1
9422 032206 012706 001000 RC1: MOV #BUFF,SP ;SET UP STACK POINTER
9423 032212 104400 RB1: TRAP ;TRAP INST WILL BE MODIFIED TO TRAP+377
9424 032214 BR45:
9425 032214 012742 000752 MOV #752,-(R2) ;MOVE TO MAILBOX # ***** 752 *****
9426 032220 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9427 032222 000000 HALT ;PREVIOUS INST FAILED TO TRAP,OR WRONG $STNM
9428 032224 005267 177762 RA1: INC RB1 ;INCREMENT TRAP INSTRUCTION
9429 032230 022767 104777 177754 CMP #104777,RB1 ;TRAP+377 TO UPPER LIMIT
9430 032236 103363 BHIS RC1 ;HAVE WE TESTED ALL
9431 032240 012767 000036 145566 MOV #36,34
9432 032246 005067 145564 CLR 36
9433 ;*****
9434 ;TEST 317 TEST THAT A TRAP OCCURES ON AN 'IOT' INSTRUCTION
9435 ;*****
9436 032252 005212 TS317: INC (R2) ;UPDATE TEST NUMBER
9437 032254 022712 000317 CMP #317,(R2) ;SEQUENCE ERROR?
9438 032260 001006 BNE TS320-10 ;BR TO ERROR HALT ON SEQ ERROR
9439 032262 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9440 032266 012767 032306 145524 MOV #RETA2,RTRAP2 ;RETURN LOCATION
9441 032274 000004 IOT ;RESERVE INSTRUCTION, SHOULD TRAP
9442 032276 012742 000753 MOV #753,-(R2) ;MOVE TO MAILBOX # ***** 753 *****
9443 032302 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9444 032304 000000 HALT ;IOT DIDN'T TRAP,OR WRONG $STNM
9445 032306 RETA2:
9446 ;*****
9447 ;TEST 320 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9448 ;*****
9449 032306 005212 TS320: INC (R2) ;UPDATE TEST NUMBER
9450 032310 022712 000320 CMP #320,(R2) ;SEQUENCE ERROR?
9451 032314 001011 BNE TS321-10 ;BR TO ERROR HALT ON SEQ ERROR
9452 032316 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9453 032322 012767 032332 145470 MOV #RETB2,RTRAP2 ;RETURN POINTER
9454 032330 000094 IOT ;RESERVED INSTRUCTION
9455 032332 020627 000774 RETB2: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
9456 032336 001404 BEQ TS321
9457 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
9458 ; CONDITIONAL BRANCH INST. AND <
9459 ; REPLACE THE MOVE INSTRUCTION <==
9460 ; WHICH FOLLOWS W/ 766 <==
9461 032340 012742 000754 MOV #754,-(R2) ;MOVE TO MAILBOX # ***** 754 *****
9462 032344 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9463 032346 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $STNM
9464 ; OR SEQUENCE ERROR
9465 ;*****
9466 ;TEST 321 TEST THAT PROPER P.C. IS SAVED
9467 ;*****
9468 032350 005212 TS321: INC (R2) ;UPDATE TEST NUMBER
9469 032352 022712 000321 CMP #321,(R2) ;SEQUENCE ERROR?
```

```
9470 032356 001012      BNE    TS322-10      ;BR TO ERROR HALT ON SEQ ERROR
9471 032360 012706 001000  MOV    #BUFF,SP      ;STACK POINTER SETUP
9472 032364 012767 032374 145426  MOV    #RETC2,RTRAP2 ;RETURN FROM TRAP POINTER
9473 032372 000004      IOT                    ;TRAP ON THIS INSTRUCTION
9474 032374 022767 032374 146372  RETC2: CMP    #.BUFF-4    ;CHECK FOR INCREMENTED P.C.
9475 032402 001404      BEQ    TS322
9476                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <= -
9477                                     ;          CONDITIONAL BRANCH INST. AND    < --
9478                                     ;          REPLACE THE MOVE INSTRUCTION   <--
9479                                     ;          WHICH FOLLOWS W/ 765          <-
9480 032404 012742 000755      MOV    #755,-(R2)    ;MOVE TO MAILBOX # ***** 755 *****
9481 032410 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
9482 032412 000000      HALT                ;INCORRECT P.C.,OR WRONG $STNM
9483                                     ; OR SEQUENCE ERROR
9484 :*****
9485 :TEST 322      TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
9486 :*****
9487 032414 005212      TS322: INC    (R2)         ;UPDATE TEST NUMBER
9488 032416 022712 000322      CMP    #322,(R2)    ;SEQUENCE ERROR?
9489 032422 001037      BNE    TS323-10     ;BR TO ERROR HALT ON SEQ ERROR
9490 032424 012706 001000      MOV    #BUFF,SP      ;SET UP
9491 032430 012767 032446 145362  MOV    #RETD2,RTRAP2 ;SET UP
9492 032436 005067 145334      LR     CC            ;CLEAR CC AND PRIORITY
9493 032442 000257      CCC
9494 032444 000004      IOT                    ;TRAP
9495 032446 026727 146324 000000  RETD2: CMP    BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK
9496 032454 001404      BEQ    1$
9497                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <=-
9498                                     ;          CONDITIONAL BRANCH INST. AND    <
9499                                     ;          REPLACE THE MOVE INSTRUCTION   <=
9500                                     ;          WHICH FOLLOWS W/ 762          <
9501 032456 012742 000756      MOV    #756,-(R2)    ;MOVE TO MAILBOX # ***** 756 *****
9502 032462 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
9503 032464 000000      HALT                ;INCORRECT STATUS
9504 032466 012706 001000  1$:  MOV    #BUFF,SP      ;SET UP
9505 032472 012767 032512 145320  MOV    #RETE2,RTRAP2 ;SET UP
9506 032500 012767 000357 145270  MOV    #357,CC       ;SET PRIORITY
9507 032506 000277      SCC
9508 032510 000004      IOT                    ;TRAP
9509 032512 026727 146260 000357  RETE2: CMP    BUFF-2,#357 ;COMPARES STATUS ON STACK
9510 032520 001404      BEQ    TS323
9511                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  < --
9512                                     ;          CONDITIONAL BRANCH INST. AND    <
9513                                     ;          REPLACE THE MOVE INSTRUCTION   <-
9514                                     ;          WHICH FOLLOWS W/ 740          <- -
9515 032522 012742 000757      MOV    #757,-(R2)    ;MOVE TO MAILBOX # ***** 757 *****
9516 032526 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
9517 032530 000000      HALT                ;INCORRECT STATUS ON STACK,OR WRONG $STNM
9518                                     ; OR SEQUENCE ERROR
9519 :*****
9520 :TEST 323      TEST THAT 'NEW' STATUS IS CORRECT
9521 :*****
9522 032532 005212      TS323: INC    (R2)         ;UPDATE TEST NUMBER
9523 032534 022712 000323      CMP    #323,(R2)    ;SEQUENCE ERROR?
9524 032540 001110      BNE    BR46         ;BR TO ERROR HALT ON SEQ ERROR
9525 032542 012706 001000      MOV    #BUFF,SP
```

```

9526 032546 012767 032562 145244      MOV      #RETF2,RTRAP2
9527 032554 005067 145242      CLR      RTRAP2+2      ;CLEAR FUTURE PRIORITY AND CC
9528 032560 000004                      IOT
9529 032562 100004      RETF2:      ;TEST FOR 'C' CLEARED
9530 032562 100004      BPL      1$
9531                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
9532                      ;          CONDITIONAL BRANCH INST. AND
9533                      ;          REPLACE THE MOVE INSTRUCTION
9534                      ;          WHICH FOLLOWS W/ 766
9535 032564 012742 000760      MOV      #760,-(R2)    ;MOVE TO MAILBOX # ***** 760 *****
9536 032570 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9537 032572 000000      HALT
9538 032574          1$:      BNE      2$
9539 032574 001004      BNE      2$
9540                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
9541                      ;          CONDITIONAL BRANCH INST. AND
9542                      ;          REPLACE THE MOVE INSTRUCTION
9543                      ;          WHICH FOLLOWS W/ 761
9544 032576 012742 000761      MOV      #761,-(R2)    ;MOVE TO MAILBOX # ***** 761 *****
9545 032602 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9546 032604 000000      HALT
9547 032606          2$:      BVC      3$
9548 032606 102004      BVC      3$
9549                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
9550                      ;          CONDITIONAL BRANCH INST. AND
9551                      ;          REPLACE THE MOVE INSTRUCTION
9552                      ;          WHICH FOLLOWS W/ 754
9553 032610 012742 000762      MOV      #762,-(R2)    ;MOVE TO MAILBOX # ***** 762 *****
9554 032614 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9555 032616 000000      HALT
9556 032620          3$:      BCC      4$
9557 032620 103004      BCC      4$
9558                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
9559                      ;          CONDITIONAL BRANCH INST. AND
9560                      ;          REPLACE THE MOVE INSTRUCTION
9561                      ;          WHICH FOLLOWS W/ 747
9562 032622 012742 000763      MOV      #763,-(R2)    ;MOVE TO MAILBOX # ***** 763 *****
9563 032626 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9564 032630 000000      HALT
9565 032632 032767 000340 145136 4$:      BIT      #340,CC
9566 032640 001404      BEQ      5$
9567                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
9568                      ;          CONDITIONAL BRANCH INST. AND
9569                      ;          REPLACE THE MOVE INSTRUCTION
9570                      ;          WHICH FOLLOWS W/ 737
9571 032642 012742 000764      MOV      #764,-(R2)    ;MOVE TO MAILBOX # ***** 764 *****
9572 032646 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9573 032650 000000      HALT
9574 032652 012706 001000 5$:      MOV      #BUFF,SP
9575 032656 012767 032674 145134      MOV      #RETF2,RTRAP2
9576 032664 012767 000357 145130      MOV      #357,RTRAP2+2 ;SET NEW 'CC' AND PRIORITY
9577 032672 000004      IOT      ;TRAP HERE
9578 032674          RETG2:   BMI      1$
9579 032674 100404      BMI      1$
9580                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
9581                      ;          CONDITIONAL BRANCH INST. AND

```

```

9582                                     ; REPLACE THE MOVE INSTRUCTION <====
9583                                     ; WHICH FOLLOWS W/ 721 <====
9584 032676 012742 000765             MOV #765,-(R2) ;MOVE TO MAILBOX # ***** 765 *****
9585 032702 005242                   INC -(R2) ;SET MSGTYP TO FATAL ERROR
9586 032704 000000                   HALT ;N NOT SET
9587                                     ;
9588 032706 001404                   BEQ 2$ ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9589                                     ; CONDITIONAL BRANCH INST. AND <====
9590                                     ; REPLACE THE MOVE INSTRUCTION <====
9591                                     ; WHICH FOLLOWS W/ 714 <====
9592                                     ;
9593 032710 012742 000766             MOV #766,-(R2) ;MOVE TO MAILBOX # ***** 766 *****
9594 032714 005242                   INC -(R2) ;SET MSGTYP TO FATAL ERROR
9595 032716 000000                   HALT ;Z NOT SET
9596                                     ;
9597 032720 102404                   BVS 3$ ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9598                                     ; CONDITIONAL BRANCH INST. AND <====
9599                                     ; REPLACE THE MOVE INSTRUCTION <====
9600                                     ; WHICH FOLLOWS W/ 707 <====
9601                                     ;
9602 032722 012742 000767             MOV #767,-(R2) ;MOVE TO MAILBOX # ***** 767 *****
9603 032726 005242                   INC -(R2) ;SET MSGTYP TO FATAL ERROR
9604 032730 000000                   HALT ;V NOT SET
9605                                     ;
9606 032732 103404                   BCS 4$ ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9607                                     ; CONDITIONAL BRANCH INST. AND <====
9608                                     ; REPLACE THE MOVE INSTRUCTION <====
9609                                     ; WHICH FOLLOWS W/ 702 <====
9610                                     ;
9611 032734 012742 000770             MOV #770,-(R2) ;MOVE TO MAILBOX # ***** 770 *****
9612 032740 005242                   INC -(R2) ;SET MSGTYP TO FATAL ERROR
9613 032742 000000                   HALT ;C NOT SET
9614 032744 016706 145026             MOV CC,SP
9615 032750 042706 000017             BIC #17,SP
9616 032754 022706 000340             CMP #340,SP
9617 032760 001404                   BEQ BR46A
9618                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9619                                     ; CONDITIONAL BRANCH INST. AND <====
9620                                     ; REPLACE THE MOVE INSTRUCTION <====
9621                                     ; WHICH FOLLOWS W/ 667 <====
9622 032762                               BR46:
9623 032762 012742 000771             MOV #771,-(R2) ;MOVE TO MAILBOX # ***** 771 *****
9624 032766 005242                   INC -(R2) ;SET MSGTYP TO FATAL ERROR
9625 032770 000000                   HALT ;PRIORITY WAS CHANGED,OR WRONG $STNM
9626 032772 012767 000022 145020     BR46A: MOV #22,20 ;.+2
9627 033000 005067 145016             CLR 22 ;HALT
9628 ;*****
9629 ;TEST 324 TEST THAT A TRAP OCCURS ON AN EMT INSTRUCTION
9630 ;*****
9631 033004 005212                               TS324: INC (R2) ;UPDATE TEST NUMBER
9632 033006 022712 000324             CMP #324,(R2) ;SEQUENCE ERROR?
9633 033012 001006                               BNE TS325-10 ;BR TO ERROR HALT ON SEQ ERROR
9634 033014 012706 001000             MOV #BUFF,SP ;STACK POINTER SETUP
9635 033020 012767 033040 145002     MOV #RETA3,RTRAP3 ;RETURN LOCATION
9636 033026 104000                               EMT ;RESERVE INSTRUCTION, SHOULD TRAP
9637 033030 012742 000772             MOV #772,-(R2) ;MOVE TO MAILBOX # ***** 772 *****

```

```
9638 033034 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9639 033036 000000          HALT                   ;EMT DIDN'T TRAP,OR WRONG $STNM
9640 033040
9641
9642
9643
9644 033040 005212          RETA3:
;*****
;TEST 325      TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
;*****
9645 033042 022712 000325      TS325:  INC      (R2)          ;UPDATE TEST NUMBER
9646 033046 001011          CMP      #325,(R2)      ;SEQUENCE ERROR?
9647 033050 012706 001000          BNE     TS326-10        ;BR TO ERROR HALT ON SEQ ERROR
9648 033054 012767 033064 144746      MOV     #BUFF,SP        ;STACK POINTER SETUP
9649 033062 104000          MOV     #RETB3,RTRAP3  ;RETURN POINTER
9650 033064 020627 000774      EMT
9651 033070 001404          RETB3:  CMP      SP,#BUFF-4 ;RESERVED INSTRUCTION
;TEST DECREMENT OF SP
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
;          CONDITIONAL BRANCH INST. AND <===
;          REPLACE THE MOVE INSTRUCTION <===
;          WHICH FOLLOWS W/ 766 <===
9652
9653
9654
9655
9656 033072 012742 000773      MOV     #773,-(R2)     ;MOVE TO MAILBOX # ***** 773 *****
9657 033076 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9658 033100 000000          HALT                   ;NOT DECREMENTED TWO WORDS,OR WRONG $STNM
9659
9660
9661
9662
9663 033102 005212          ;*****
;TEST 326      TEST THAT PROPER P.C. IS SAVED
;*****
9664 033104 022712 000326      TS326:  INC      (R2)          ;UPDATE TEST NUMBER
9665 033110 001012          CMP     #326,(R2)      ;SEQUENCE ERROR?
9666 033112 012706 001000          BNE     TS327-10        ;BR TO ERROR HALT ON SEQ ERROR
9667 033116 012767 033126 144704      MOV     #BUFF,SP        ;STACK POINTER SETUP
9668 033124 104000          MOV     #RETC3,RTRAP3  ;RETURN FROM TRAP POINTER
9669 033126 022767 033126 145640      EMT
9670 033134 001404          RETC3:  CMP     #,BUFF-4  ;TRAP ON THIS INSTRUCTION
;CHECK FOR INCREMENTED P.C.
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=
;          CONDITIONAL BRANCH INST. AND <
;          REPLACE THE MOVE INSTRUCTION <=
;          WHICH FOLLOWS W/ 765 <=
9671
9672
9673
9674
9675 033136 012742 000774      MOV     #774,-(R2)     ;MOVE TO MAILBOX # ***** 774 *****
9676 033142 005242          INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
9677 033144 000000          HALT                   ;INCORRECT P.C.,OR WRONG $STNM
9678
9679
9680
9681
9682 033146 005212          ;*****
;TEST 327      TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
;*****
9683 033150 022712 000327      TS327:  INC      (R2)          ;UPDATE TEST NUMBER
9684 033154 001037          CMP     #327,(R2)      ;SEQUENCE ERROR?
9685 033156 012706 001000          BNE     TS330-10        ;BR TO ERROR HALT ON SEQ ERROR
9686 033162 012767 033200 144640      MOV     #BUFF,SP        ;SET UP
9687 033170 005067 144602          MOV     #RETD3,RTRAP3  ;SET UP
9688 033174 000257          CLR     CC              ;CLEAR CC AND PRIORITY
9689 033176 104000          CCC
9690 033200 026727 145572 000000      RETD3:  EMT
;TRAP
;TEST THAT OLD STATUS WENT TO STACK
9691 033206 001404          CMP     BUFF-2,#0
9692
9693
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
;          CONDITIONAL BRANCH INST. AND
```

```

9694                                     :           REPLACE THE MOVE INSTRUCTION
9695                                     :           WHICH FOLLOWS W/ 762
9696 033210 012742 000775                MOV    #775,-(R2)  ;MOVE TO MAILBOX # ***** 775 *****
9697 033214 005242                        INC    -(R2)       ;SET MSGTYP TO FATAL ERROR
9698 033216 000000                        HALT                                     ;INCORRECT STATUS
9699 033220 012706 001000                1$:    MOV    #BUFF,SP  ;SET UP
9700 033224 012767 033244 144576        MOV    #RETE3,RTRAP3 ;SET UP
9701 033232 012767 000357 144536        MOV    #357,CC     ;SET PRIORITY
9702 033240 000277                        SCC                                     ;SET CC
9703 033242 104000                        EMT                                     ;TRAP
9704 033244 026727 145526 000357 RETE3:  CMP    BUFF-2,#357 ;COMPARES STATUS ON STACK
9705 033252 001404                        BEQ    TS330
9706                                     :   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=
9707                                     :   CONDITIONAL BRANCH INST. AND <====
9708                                     :   REPLACE THE MOVE INSTRUCTION <====
9709                                     :   WHICH FOLLOWS W/ 740 <====
9710 033254 012742 000776                MOV    #776,-(R2)  ;MOVE TO MAILBOX # ***** 776 *****
9711 033260 005242                        INC    -(R2)       ;SET MSGTYP TO FATAL ERROR
9712 033262 000000                        HALT                                     ;INCORRECT STATUS ON STACK,OR WRONG $STNM
9713                                     :   OR SEQUENCE ERROR
9714 :*****
9715 :TEST 330 TEST THAT 'NEW' STATUS IS CORRECT
9716 :*****
9717 033264 005212                        TS330: INC    (R2)       ;UPDATE TEST NUMBER
9718 033266 022712 000330                CMP    #330,(R2)  ;SEQUENCE ERROR?
9719 033272 001106                        BNE    TS331-10   ;BR TO ERROR HALT ON SEQ ERROR
9720 033274 012706 001000                MOV    #BUFF,SP
9721 033300 012767 033314 144522        MOV    #RETF3,RTRAP3
9722 033306 005067 144520                CLR    RTRAP3+2  ;CLEAR FUTURE PRIORITY AND CC
9723 033312 104000                        EMT
9724 033314                        RETF3:
9725 033314 100004                        BPL    1$         ;TEST FOR 'C' CLEARED
9726                                     :   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== =
9727                                     :   CONDITIONAL BRANCH INST. AND <====
9728                                     :   REPLACE THE MOVE INSTRUCTION <====
9729                                     :   WHICH FOLLOWS W/ 766 <====
9730 033316 012742 000777                MOV    #777,-(R2)  ;MOVE TO MAILBOX # ***** 777 *****
9731 033322 005242                        INC    -(R2)       ;SET MSGTYP TO FATAL ERROR
9732 033324 000000                        HALT                                     ;C NOT CLEARED
9733 033326                1$:
9734 033326 001004                        BNE    2$
9735                                     :   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== =
9736                                     :   CONDITIONAL BRANCH INST. AND <====
9737                                     :   REPLACE THE MOVE INSTRUCTION <====
9738                                     :   WHICH FOLLOWS W/ 761 <====
9739 033330 012742 001000                MOV    #1000,-(R2) ;MOVE TO MAILBOX # ***** 1000 *****
9740 033334 005242                        INC    -(R2)       ;SET MSGTYP TO FATAL ERROR
9741 033336 000000                        HALT                                     ;Z NOT CLEARED
9742 033340                2$:
9743 033340 102004                        BVC    3$
9744                                     :   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== =
9745                                     :   CONDITIONAL BRANCH INST. AND <====
9746                                     :   REPLACE THE MOVE INSTRUCTION <====
9747                                     :   WHICH FOLLOWS W/ 754 <====
9748 033342 012742 001001                MOV    #1001,-(R2) ;MOVE TO MAILBOX # ***** 1001 *****
9749 033346 005242                        INC    -(R2)       ;SET MSGTYP TO FATAL ERROR

```



```

9750 033350 000000          HALT          ;V NOT CLEARED
9751 033352          3$:          BCC          4$
9752 033352 103004          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9753          ;                   CONDITIONAL BRANCH INST. AND <====
9754          ;                   REPLACE THE MOVE INSTRUCTION <====
9755          ;                   WHICH FOLLOWS W/ 747 <====
9756          ;
9757 033354 012742 001002    MOV          #1002,-(R2) ;MOVE TO MAILBOX # ***** 1002 *****
9758 033360 005242          INC          -(R2) ;SET MSGTYP TO FATAL ERROR
9759 033362 000000          HALT          ;C NOT CLEARED
9760 033364 032767 000340 144404 4$:    BIT          #340,CC ;TEST PRIORITY
9761 033372 001404          BEQ          5$
9762          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9763          ;                   CONDITIONAL BRANCH INST. AND <====
9764          ;                   REPLACE THE MOVE INSTRUCTION <====
9765          ;                   WHICH FOLLOWS W/ 737 <====
9766 033374 012742 001003    MOV          #1003,-(R2) ;MOVE TO MAILBOX # ***** 1003 *****
9767 033400 005242          INC          -(R2) ;SET MSGTYP TO FATAL ERROR
9768 033402 000000          HALT          ;PRIORITY NOT ZERO
9769 033404 012706 001000 5$:    MOV          #BUFF,SP
9770 033410 012767 033426 144412    MOV          #RETG3,RTRAP3
9771 033416 012767 000357 144406    MOV          #357,RTRAP3+2 ;SET NEW 'CC' AND PRIORITY
9772 033424 104000          EMT          ;TRAP HERE
9773 033426          RETG3:
9774 033426 100404          BMI          1$
9775          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9776          ;                   CONDITIONAL BRANCH INST. AND <====
9777          ;                   REPLACE THE MOVE INSTRUCTION <====
9778          ;                   WHICH FOLLOWS W/ 721 <====
9779 033430 012742 001004    MOV          #1004,-(R2) ;MOVE TO MAILBOX # ***** 1004 *****
9780 033434 005242          INC          -(R2) ;SET MSGTYP TO FATAL ERROR
9781 033436 000000          HALT          ;N NOT SET
9782 033440          1$:
9783 033440 001404          BEQ          2$
9784          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9785          ;                   CONDITIONAL BRANCH INST. AND <====
9786          ;                   REPLACE THE MOVE INSTRUCTION <====
9787          ;                   WHICH FOLLOWS W/ 714 <====
9788 033442 012742 001005    MOV          #1005,-(R2) ;MOVE TO MAILBOX # ***** 1005 *****
9789 033446 005242          INC          -(R2) ;SET MSGTYP TO FATAL ERROR
9790 033450 000000          HALT          ;Z NOT SET
9791 033452          2$:
9792 033452 102404          BVS          3$
9793          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9794          ;                   CONDITIONAL BRANCH INST. AND <====
9795          ;                   REPLACE THE MOVE INSTRUCTION <====
9796          ;                   WHICH FOLLOWS W/ 707 <====
9797 033454 012742 001006    MOV          #1006,-(R2) ;MOVE TO MAILBOX # ***** 1006 *****
9798 033460 005242          INC          -(R2) ;SET MSGTYP TO FATAL ERROR
9799 033462 000000          HALT          ;V NOT SET
9800 033464          3$:
9801 033464 103404          BCS          4$
9802          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9803          ;                   CONDITIONAL BRANCH INST. AND <====
9804          ;                   REPLACE THE MOVE INSTRUCTION <====
9805          ;                   WHICH FOLLOWS W/ 702 <====

```

```

9806 033466 012742 001007      MOV      #1007,-(R2)      ;MOVE TO MAILBOX # ***** 1007 *****
9807 033472 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9808 033474 000000              HALT                    ;C NOT SET
9809 033476 000257              CCC
9810 033500 022767 000340 144270 4$:  CMP      #340,CC
9811 033506 001404              BEQ      TS331
9812
9813
9814
9815
9816 033510 012742 001010      MOV      #1010,-(R2)    ;MOVE TO MAILBOX # ***** 1010 *****
9817 033514 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9818 033516 000000              HALT                    ;PRIORITY WAS CHANGED,OR WRONG $STNM
9819
9820
9821
9822
9823 033520 005212              TS331:  INC      (R2)          ;UPDATE TEST NUMBER
9824 033522 022712 000331      CMP      #331,(R2)      ;SEQUENCE ERROR?
9825 033526 001011              BNE      BR47           ;BR TO ERROR HALT ON SEQ ERROR
9826
9827 033530 012767 104376 000012      MOV      #EMT+376,RB    ;***** F11 **** ADD +376 TO SHORTEN TEST
9828 033536 012767 033562 144264      MOV      #RA,30         ;INITIALIZE BASE EMT INSTRUCTION
9829 033544 012706 001000      MOV      #BUFF,SP      ;RETURN FROM TRAP TO RA
9830 033550 104000              EMT                    ;SET UP STACK POINTER
9831 033552
9832 033552 012742 001011      BR47:  MOV      #1011,-(R2) ;TRAP INST. WILL BE MODIFIED TO EMT+377
9833 033556 005242              INC      -(R2)          ;MOVE TO MAILBOX # ***** 1011 *****
9834 033560 000000              HALT                    ;SET MSGTYP TO FATAL ERROR
9835 033562 005267 177762      RA:    INC      RB        ;PREVIOUS INST FAILED TO TRAP,OR WRONG $STNM
9836 033566 022767 104377 177754      CMP      #104377,RB    ;INCREMENT TRAP INSTRUCTION
9837 033574 103363              BHIS    RC             ;EMT+377 TO EMT?
9838
9839 033576 012767 000032 144224      MOV      #32,30        ;HAVE WE TESTED ALL
9840 033604 005067 144222      CLR      32            ;YES
9841
9842
9843
9844 033610 005212              ;TEST 332 TEST THAT A TRAP OCCURES ON AN 'TRACE-TRT' INSTRUCTION
9845 033612 022712 000332      TS332:  INC      (R2)          ;UPDATE TEST NUMBER
9846 033616 001006              CMP      #332,(R2)      ;SEQUENCE ERROR?
9847 033620 012706 001000      BNE      TS333-10      ;BR TO ERROR HALT ON SEQ ERROR
9848 033624 012767 033644 144162      MOV      #BUFF,SP      ;STACK POINTER SETUP
9849 033632 000003              MOV      #RETA4,RTRAP4 ;RETURN LOCATION
9850 033634 012742 001012      TRT                    ;RESERVED INSTRUCTION, SHOULD TRAP
9851 033640 005242              MOV      #1012,-(R2)   ;MOVE TO MAILBOX # ***** 1012 *****
9852 033642 000000              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9853 033644              HALT                    ;TRT DIDN'T TRAP,OR WRONG $STNM
9854
9855
9856
9857 033644 005212              RETA4:
9858 033646 022712 000333      ;TEST 333 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9859 033652 001011              TS333:  INC      (R2)          ;UPDATE TEST NUMBER
9860 033654 012706 001000      CMP      #333,(R2)      ;SEQUENCE ERROR?
9861 033660 012767 033670 144126      BNE      TS334-10      ;BR TO ERROR HALT ON SEQ ERROR
          MOV      #BUFF,SP ;STACK POINTER SETUP
          MOV      #RETB4,RTRAP4 ;RETURN POINTER

```

```

9862 033666 000003
9863 033670 020627 C00774 RETB4: CMP SP,#BUFF-4 ;RESERVED INSTRUCTION
9864 033674 001404 BEQ TS334 ;TEST DECREMENT OF SP
9865 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9866 ; CONDITIONAL BRANCH INST. AND <
9867 ; REPLACE THE MOVE INSTRUCTION <
9868 ; WHICH FOLLOWS W/ 766 <
9869 033676 012742 001013 MOV #1013,-(R2) ;MOVE TO MAILBOX # ***** 1013 *****
9870 033702 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9871 033704 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $STNM
9872 ; OR SEQUENCE ERROR
9873 ;*****
9874 ;TEST 334 TEST THAT PROPER P.C. IS SAVED
9875 ;*****
9876 033706 005212 TS334: INC (R2) ;UPDATE TEST NUMBER
9877 033710 022712 000334 CMP #334,(R2) ;SEQUENCE ERROR?
9878 033714 001012 BNE TS335-10 ;BR TO ERROR HALT ON SEQ ERROR
9879 033716 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9880 033722 012767 033732 144064 MOV #RETC4,RTRAP4 ;RETURN FROM TRAP POINTER
9881 C33730 000003 TRT ;TRAP ON THIS INSTRUCTION
9882 033732 022767 033732 145034 RETC4: CMP #.BUFF-4 ;CHECK FOR INCREMENTED P.C.
9883 033740 001404 BEQ TS335
9884 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9885 ; CONDITIONAL BRANCH INST. AND <
9886 ; REPLACE THE MOVE INSTRUCTION <
9887 ; WHICH FOLLOWS W/ 765 <
9888 033742 012742 001014 MOV #1014,-(R2) ;MOVE TO MAILBOX # ***** 1014 *****
9889 033746 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9890 033750 000000 HALT ;INCORRECT P.C.,OR WRONG $STNM
9891 ; OR SEQUENCE ERROR
9892 ;*****
9893 ;TEST 335 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
9894 ;*****
9895 033752 005212 TS335: INC (R2) ;UPDATE TEST NUMBER
9896 033754 022712 000335 CMP #335,(R2) ;SEQUENCE ERROR?
9897 033760 001037 BNE TS336-10 ;BR TO ERROR HALT ON SEQ ERROR
9898 033762 012706 001000 MOV #BUFF,SP ;SET UP
9899 033766 012767 034004 144020 MOV #RETD4,RTRAP4 ;SET UP
9900 033774 005067 143776 CLR CC ;CLEAR CC AND PRIORITY
9901 034000 000257 CCC
9902 034002 000003 TRT ;TRAP
9903 034004 026727 144766 000000 RETD4: CMP BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK
9904 ;TEST FOR ALL ZEROS
9905 034012 001404 BEQ 1$
9906 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9907 ; CONDITIONAL BRANCH INST. AND <
9908 ; REPLACE THE MOVE INSTRUCTION <
9909 ; WHICH FOLLOWS W/ 762 <
9910 034014 012742 001015 MOV #1015,-(R2) ;MOVE TO MAILBOX # ***** 1015 *****
9911 034020 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9912 034022 000000 HALT ;INCORRECT STATUS
9913 034024 012706 001000 1$: MOV #BUFF,SP ;SET UP
9914 034030 012767 034050 143756 MOV #RETE4,RTRAP4 ;SET UP
9915 034036 012767 000357 143732 MOV #357,CC ;SET PRIORITY
9916 034044 000277 SCC ;SET-SET CC
9917 034046 000003 TRT ;TRAP

```

CJKDB-D DCF11-AA CPU DIAG.  
CJKDBD.P11 24-NOV-80 11:07

MACY11 30A(1052) 14-JAN-81 11:46 PAGE 200  
T335 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK

```

9918 034050 026727 144722 000357 RETE4: CMP      BUFF-2,#357      ;COMPARES STATUS ON STACK
9919 034056 001404          BEQ      TS336          ;
9920          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
9921          ;          CONDITIONAL BRANCH INST. AND <=-
9922          ;          REPLACE THE MOVE INSTRUCTION <-
9923          ;          WHICH FOLLOWS W/ 740 <-
9924 034060 012742 001016          MOV      #1016,-(R2)    ;MOVE TO MAILBOX # ***** 1016 *****
9925 034064 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9926 034066 000000          HALT          ;INCORRECT STATUS ON STACK,OR WRONG $ISTNM
9927          ; OR SEQUENCE ERROR
9928          ;*****
9929          ;TEST 336      TEST THAT 'NEW' STATUS IS CORRECT
9930          ;*****
9931 034070 005212          TS336: INC      (R2)      ;UPDATE TEST NUMBER
9932 034072 022712 000336          CMP      #336,(R2)    ;SEQUENCE ERROR?
9933 034076 001110          BNE     BR51          ;BR TO ERROR HALT ON SEQ ERROR
9934 034100 012706 001000          MOV      #BUFF,SP
9935 034104 012767 034120 143702          MOV      #RETF4,RTRAP4
9936 034112 005067 143700          CLR     RTRAP4+2    ;CLEAR FUTURE PRIORITY AND CC
9937 034116 000003          TRT
9938          RETF4:          ;TEST FOR 'C' CLEARED
9939 034120 100004          BPL     1$
9940          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
9941          ;          CONDITIONAL BRANCH INST. AND <-
9942          ;          REPLACE THE MOVE INSTRUCTION <-
9943          ;          WHICH FOLLOWS W/ 766 <-
9944 034122 012742 001017          MOV      #1017,-(R2)  ;MOVE TO MAILBOX # ***** 1017 *****
9945 034126 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9946 034130 000000          HALT          ;C NOT CLEARED
9947          1$:
9948 034132 001004          BNE     2$
9949          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
9950          ;          CONDITIONAL BRANCH INST. AND <=-
9951          ;          REPLACE THE MOVE INSTRUCTION <-
9952          ;          WHICH FOLLOWS W/ 761 <-
9953 034134 012742 001020          MOV      #1020,-(R2)  ;MOVE TO MAILBOX # ***** 1020 *****
9954 034140 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9955 034142 000000          HALT          ;Z NOT CLEARED
9956          2$:
9957 034144 102004          BVC     3$
9958          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
9959          ;          CONDITIONAL BRANCH INST. AND <=-
9960          ;          REPLACE THE MOVE INSTRUCTION <=-
9961          ;          WHICH FOLLOWS W/ 754 <-
9962 034146 012742 001021          MOV      #1021,-(R2)  ;MOVE TO MAILBOX # ***** 1021 *****
9963 034152 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9964 034154 000000          HALT          ;V NOT CLEARED
9965          3$:
9966 034156 103004          BCC     4$
9967          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
9968          ;          CONDITIONAL BRANCH INST. AND <---
9969          ;          REPLACE THE MOVE INSTRUCTION <-
9970          ;          WHICH FOLLOWS W/ 747 <-
9971 034160 012742 001022          MOV      #1022,-(R2)  ;MOVE TO MAILBOX # ***** 1022 *****
9972 034164 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9973 034166 000000          HALT          ;C NOT CLEARED

```

```

9974 034170 032767 000340 143600 4$: BIT #340,CC ;TEST PRIORITY
9975 034176 001404 BEQ 5$
9976 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
9977 ; CONDITIONAL BRANCH INST. AND <===
9978 ; REPLACE THE MOVE INSTRUCTION <===
9979 ; WHICH FOLLOWS W/ 737 <===
9980 034200 012742 001023 MOV #1023,-(R2) ;MOVE TO MAILBOX # ***** 1023 *****
9981 034204 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9982 034206 000000 HALT ;PRIORITY NOT ZERO
9983 034210 012706 001000 5$: MOV #BJFF,SP
9984 034214 012767 034232 143572 MOV #RETG4,RTRAP4
9985 034222 012767 000357 143566 MOV #357,RTRAP4+2 ;SET NEW 'CC' AND PRIORITY
9986 034230 000003 TRT ;TRAP HERE
9987 034232 RETG4:
9988 034232 100404 BMI 1$
9989 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
9990 ; CONDITIONAL BRANCH INST. AND <===
9991 ; REPLACE THE MOVE INSTRUCTION <===
9992 ; WHICH FOLLOWS W/ 721 <===
9993 034234 012742 001024 MOV #1024,-(R2) ;MOVE TO MAILBOX # ***** 1024 *****
9994 034240 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9995 034242 000000 HALT ;N NOT SET
9996 034244 1$:
9997 034244 001404 BEQ 2$
9998 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
9999 ; CONDITIONAL BRANCH INST. AND <===
10000 ; REPLACE THE MOVE INSTRUCTION <=
10001 ; WHICH FOLLOWS W/ 714 <=
10002 034246 012742 001025 MOV #1025,-(R2) ;MOVE TO MAILBOX # ***** 1025 *****
10003 034252 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10004 034254 000000 HALT ;Z NOT SET
10005 034256 2$:
10006 034256 102404 BVS 3$
10007 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
10008 ; CONDITIONAL BRANCH INST. AND <===
10009 ; REPLACE THE MOVE INSTRUCTION <=
10010 ; WHICH FOLLOWS W/ 707 <=
10011 034260 012742 001026 MOV #1026,-(R2) ;MOVE TO MAILBOX # ***** 1026 *****
10012 034264 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10013 034266 000000 HALT ;V NOT SET
10014 034270 3$:
10015 034270 103404 BCS 4$
10016 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
10017 ; CONDITIONAL BRANCH INST. AND <=
10018 ; REPLACE THE MOVE INSTRUCTION <=
10019 ; WHICH FOLLOWS W/ 702 <=
10020 034272 012742 001027 MOV #1027,-(R2) ;MOVE TO MAILBOX # ***** 1027 *****
10021 034276 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10022 034300 000000 HALT ;C NOT SET
10023 034302 016706 143470 4$: MOV CC,SP
10024 034306 042706 000017 BIC #17,SP
10025 034312 022706 000340 CMP #340,SP
10026 034316 001404 BEQ BR51A
10027 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=-
10028 ; CONDITIONAL BRANCH INST. AND <=
10029 ; REPLACE THE MOVE INSTRUCTION <=

```

```
10030 ; WHICH FOLLOWS W/ 667
10031 034320 BR51:
10032 034320 012742 001030 MOV #1030,-(R2) ;MOVE TO MAILBOX # ***** 1030 *****
10033 034324 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10034 034326 000000 HALT ;PRIORITY WAS CHANGED,OR WRONG $STNM
10035 034330 012767 000016 143456 BR51A: MOV #16,14
10036 034336 005067 143454 CLR 16
10037
10038 ;PDP-11 ILLEGAL AND ADDRESS INSTRUCTION TEST
10039 ;ALL INSTRUCTIONS THAT ARE RESERVED
10040 ;SHOULD TRAP TO LOCATION 4, AND THE
10041 ;PC THAT POINTS TO THE TRAPPING INSTRUCTION
10042 ;SHOULD BE PLACED ON THE STACK
10043
10044 ;*****
10045 ;TEST 337 TEST THAT A TRAP OCCURS ON AN ILLEGAL INSTRUCTION
10046 ;*****
10047 034342 005212 TS337: INC (R2) ;UPDATE TEST NUMBER
10048 034344 022712 000337 CMP #337,(R2) ;SEQUENCE ERROR?
10049 034350 001006 BNE TS340-10 ;BR TO ERROR HALT ON SEQ ERROR
10050 034352 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
10051 034356 012767 034376 143420 MOV #RETA5,RTRAP5 ;RETURN LOCATION
10052 034364 000100 JMP %0 ;ILLEGAL INSTRUCTION, SHOULD TRAP
10053 034366 012742 001031 MOV #1031,-(R2) ;MOVE TO MAILBOX # ***** 1031 *****
10054 034372 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10055 034374 000000 HALT ;ILLEGAL INSTRUCTION DIDN'T TRAP,OR WRONG $STNM
10056 034376
10057 RETA5:
10058 ;*****
10059 ;TEST 340 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
10060 ;*****
10061 034376 005212 TS340: INC (R2) ;UPDATE TEST NUMBER
10062 034400 022712 000340 CMP #340,(R2) ;SEQUENCE ERROR?
10063 034406 012706 001000 BNE TS341-10 ;BR TO ERROR HALT ON SEQ ERROR
10064 034412 012767 034422 143364 MOV #BUFF,SP ;STACK POINTER SETUP
10065 034420 000100 MOV #RETB5,RTRAP5 ;RETURN POINTER
10066 034422 020627 000774 RETB5: CMP SP,#BUFF-4 ;RESERVED INSTRUCTION
10067 034426 001404 BEQ TS341 ;TEST DECREMENT OF SP
10068
10069 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
10070 ; CONDITIONAL BRANCH INST. AND <
10071 ; REPLACE THE MOVE INSTRUCTION <
10072 ; WHICH FOLLOWS W/ 766 <
10073 034430 012742 001032 MOV #1032,-(R2) ;MOVE TO MAILBOX # ***** 1032 *****
10074 034434 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10075 034436 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $STNM
10076 ; OR SEQUENCE ERROR
10077 ;*****
10078 ;TEST 341 TEST THAT PROPER P.C. IS SAVED
10079 ;*****
10080 034440 005212 TS341: INC (R2) ;UPDATE TEST NUMBER
10081 034442 022712 000341 CMP #341,(R2) ;SEQUENCE ERROR?
10082 034446 001012 BNE TS342-10 ;BR TO ERROR HALT ON SEQ ERROR
10083 034450 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
10084 034454 012767 034464 143322 MOV #RETC5,RTRAP5 ;RETURN FROM TRAP POINTER
10085 034462 000100 JMP %0 ;TRAP ON THIS INSTRUCTION
10085 034464 022767 034464 144302 RETC5: CMP #.,BUFF-4 ;CHECK FOR INCREMENTED P.C.
```

```
10086 034472 001404          BEQ      TS342          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10087                                     ;          CONDITIONAL BRANCH INST. AND <====
10088                                     ;          REPLACE THE MOVE INSTRUCTION <====
10089                                     ;          WHICH FOLLOWS W/ 765 <====
10090                                     ;
10091 034474 012742 001033    MOV      #1033,-(R2)    ;MOVE TO MAILBOX # ***** 1033 *****
10092 034500 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
10093 034502 000000          HALT                    ;INCORRECT P.C. OR WRONG $STNM
10094                                     ; OR SEQUENCE ERROR
10095 ;*****
10096 ;TEST 342 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
10097 ;*****
10098 034504 005212          TS342: INC      (R2)          ;UPDATE TEST NUMBER
10099 034506 022712 000342    CMP      #342,(R2)     ;SEQUENCE ERROR?
10100 034512 001037          BNE     TS343-10      ;BR TO ERROR HALT ON SEQ ERROR
10101 034514 012706 001000    MOV      #BUFF,SP     ;SET UP
10102 034520 012767 034536 143256  MOV      #RETD5,RTRAP5 ;SET UP
10103 034526 005067 143244    CLR      CC           ;CLEAR CC AND PRIORITY
10104 034532 000257          CCC
10105 034534 000100          JMP      %0           ;TRAP
10106 034536 026727 144234 000000 RETD5:  CMP      BUFF-2,#0     ;TEST THAT OLD STATUS WENT TO STACK
10107 034544 001404          BEQ      1$
10108                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
10109                                     ;          CONDITIONAL BRANCH INST. AND <-
10110                                     ;          REPLACE THE MOVE INSTRUCTION <=-
10111                                     ;          WHICH FOLLOWS W/ 762 <=-
10112 034546 012742 001034    MOV      #1034,-(R2)  ;MOVE TO MAILBOX # ***** 1034 *****
10113 034552 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
10114 034554 000000          HALT                    ;INCORRECT STATUS
10115 034556 012706 001000    1$:  MOV      #BUFF,SP     ;SET UP
10116 034562 012767 034602 143214  MOV      #RETE5,RTRAP5 ;SET UP
10117 034570 012767 000357 143200  MOV      #357,CC      ;SET PRIORITY
10118 034576 000277          SCC
10119 034600 000100          JMP      %0           ;TRAP
10120 034602 026727 144170 000357 RETE5:  CMP      BUFF-2,#357   ;COMPARES STATUS ON STACK
10121 034610 001404          BEQ      TS343
10122                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
10123                                     ;          CONDITIONAL BRANCH INST. AND <
10124                                     ;          REPLACE THE MOVE INSTRUCTION <=-
10125                                     ;          WHICH FOLLOWS W/ 740 <-
10126 034612 012742 001035    MOV      #1035,-(R2)  ;MOVE TO MAILBOX # ***** 1035 *****
10127 034616 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
10128 034620 000000          HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $STNM
10129                                     ; OR SEQUENCE ERROR
10130 ;*****
10131 ;TEST 343 TEST THAT 'NEW' STATUS IS CORRECT
10132 ;*****
10133 034622 005212          TS343: INC      (R2)          ;UPDATE TEST NUMBER
10134 034624 022712 000343    CMP      #343,(R2)     ;SEQUENCE ERROR?
10135 034630 001106          BNE     TS344-10      ;BR TO ERROR HALT ON SEQ ERROR
10136 034632 012706 001000    MOV      #BUFF,SP     ;
10137 034636 012767 034652 143140  MOV      #RETF5,RTRAP5 ;
10138 034644 005067 143136    CLR      RTRAP5+2     ;CLEAR FUTURE PRIORITY AND CC
10139 034650 000100          JMP      %0           ;
10140 034652          RETF5: ;TEST FOR 'C' CLEARED
10141 034652 100004          BPL      1$
```





```
10198 034776  
10199 034776 001404  
10200  
10201  
10202  
10203  
10204 035000 012742 001044  
10205 035004 005242  
10206 035006 000000  
10207 035010  
10208 035010 102404  
10209  
10210  
10211  
10212  
10213 035012 012742 001045  
10214 035016 005242  
10215 035020 000000  
10216 035022  
10217 035022 103404  
10218  
10219  
10220  
10221  
10222 035024 012742 001046  
10223 035030 005242  
10224 035032 000000  
10225 035034 016706 142736  
10226 035040 022706 000357  
10227 035044 001404  
10228  
10229  
10230  
10231  
10232 035046 012742 001047  
10233 035052 005242  
10234 035054 000000  
10235  
10236  
10237  
10238  
10239 035056 005212  
10240 035060 022712 000344  
10241 035064 001006  
10242 035066 012706 001000  
10243 035072 012767 035112 142704  
10244 035100 004000  
10245 035102 012742 001050  
10246 035106 005242  
10247 035110 000000  
10248 035112  
10249  
10250  
10251  
10252 035112 005212  
10253 035114 022712 000345
```

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 # \*\*\*\*\* 1044 \*\*\*\*\*  
: 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 <====

3\$: BCS 4\$  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <----  
: WHICH FOLLOWS W/ 702 <----

4\$: MOV #1046,-(R2) : MOVE TO MAILBOX # \*\*\*\*\* 1046 \*\*\*\*\*  
: INC -(R2) : SET MSGTYP TO FATAL ERROR  
: HALT : C NOT SET  
: 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 <--  
: 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 FROR  
: 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?

```

10254 035120 001011          BNE      TS346-10      ;BR TO ERROR HALT ON SEQ ERROR
10255 035122 012706 001000    MOV      #BUFF,SP     ;STACK POINTER SETUP
10256 035126 012767 035136 142650  MOV      #RETJ,RTRAP5 ;RETURN POINTER
10257 035134 004000          JSR      %0,%0        ;RESERVED INSTRUCTION
10258 035136 020627 000774    RETJ:   CMP      SP,#BUFF-4 ;TEST DECREMENT OF SP
10259 035142 001404          BEQ      TS346
10260                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
10261                                     ;          CONDITIONAL BRANCH INST. AND <===
10262                                     ;          REPLACE THE MOVE INSTRUCTION <===
10263                                     ;          WHICH FOLLOWS W/ 766 <===
10264 035144 012742 001051          MOV      #1051,-(R2) ;MOVE TO MAILBOX # ***** 1051 *****
10265 035150 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10266 035152 000000          HALT     ;NOT DECREMENTED TWO WORDS,OR WRONG $STMM
10267                                     ; OR SEQUENCE ERROR
10268 :*****
10269 :TEST 346      TEST THAT PROPER P.C. IS SAVED
10270 :*****
10271 035154 005212          TS346:  INC      (R2)      ;UPDATE TEST NUMBER
10272 035156 022712 000346    CMP      #346,(R2)   ;SEQUENCE ERROR?
10273 035162 001012          BNE      TS347-10   ;BR TO ERROR HALT ON SEQ ERROR
10274 035164 012706 001000    MOV      #BUFF,SP     ;STACK POINTER SETUP
10275 035170 012767 035200 142606  /        #RETK,RTRAP5 ;RETURN FROM TRAP POINTER
10276 035176 004000          INSTK: JSR      %0,%0 ;TRAP ON THIS INSTRUCTION
10277 035200 022767 035200 143566  RETK:   CMP      #INSTK+2,BUFF-4 ;CHECK FOR INCREMENTED P.C.
10278 035206 001404          BEQ      TS347
10279                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
10280                                     ;          CONDITIONAL BRANCH INST. AND <---
10281                                     ;          REPLACE THE MOVE INSTRUCTION <---
10282                                     ;          WHICH FOLLOWS W/ 765 <-
10283 035210 012742 001052          MOV      #1052,-(R2) ;MOVE TO MAILBOX # ***** 1052 *****
10284 035214 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10285 035216 000000          HALT     ;INCORRECT P.C.,OR WRONG $STMM
10286                                     ; OR SEQUENCE ERROR
10287 :*****
10288 :TEST 347      TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
10289 :*****
10290 :*****
10291 035220 005212          TS347:  INC      (R2)      ;UPDATE TEST NUMBER
10292 035222 022712 000347    CMP      #347,(R2)   ;SEQUENCE ERROR?
10293 035226 001037          BNE      TS350-10   ;BR TO ERROR HALT ON SEQ ERROR
10294 035230 012706 001000    MOV      #BUFF,SP     ;SET UP
10295 035234 012767 035252 142542  MOV      #RETL,RTRAP5 ;SET UP
10296 035242 005067 142530    CLR      CC          ;CLEAR CC AND PRIORITY
10297 035246 000257          CCC
10298 035250 004000          JSR      %0,%0        ;TRAP
10299 035252 026727 143520 000000  RETL:  CMP      BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK
10300 035260 001404          BEQ      1$
10301                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
10302                                     ;          CONDITIONAL BRANCH INST. AND <---
10303                                     ;          REPLACE THE MOVE INSTRUCTION <=
10304                                     ;          WHICH FOLLOWS W/ 762 <---
10305 035262 012742 001053          MOV      #1053,-(R2) ;MOVE TO MAILBOX # ***** 1053 *****
10306 035266 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10307 035270 000000          HALT     ;INCORRECT STATUS
10308 035272 012706 001000 1$:   MOV      #BUFF,SP     ;SET UP
10309 035276 012767 035316 142500  MOV      #RETM,RTRAP5 ;SET UP

```

```

10310 035304 012767 000357 142464      MOV      #357,CC      ;SET PRIORITY
10311 035312 000277          SCC          ;SET CC
10312 035314 004000          JSR      %0,%0      ;TRAP
10313 035316 026727 143454 000357 RETM:  CMP      BUFF-2,#357  ;COMPARES STATUS ON STACK
10314 035324 001404          BEQ      TS350
10315          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10316          ;          CONDITIONAL BRANCH INST. AND <====
10317          ;          REPLACE THE MOVE INSTRUCTION <====
10318          ;          WHICH FOLLOWS W/ 740 <====
10319 035326 012742 001054      MOV      #1054,-(R2) ;MOVE TO MAILBOX # ***** 1054 *****
10320 035332 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10321 035334 000000          HALT          ;INCORRECT STATUS ON STACK,OR WRONG $TSTM
10322          ; OR SEQUENCE ERROR
10323          ;*****
10324          ;TEST 350      TEST THAT 'NEW' STATUS IS CORRECT
10325          ;*****
10326 035336 005212          TS350: INC      (R2)      ;UPDATE TEST NUMBER
10327 035340 022712 000350      CMP      #350,(R2)  ;SEQUENCE ERROR?
10328 035344 001105          BNE      TS351-10   ;BR TO ERROR HALT ON SEQ ERROR
10329 035346 012706 001000      MOV      #BUFF,SP
10330 035352 012767 035366 142424      MOV      #RETN,RTRAP5
10331 035360 005067 142422          CLR      RTRAP5+2  ;CLEAR FUTURE PRIORITY AND CC
10332 035364 004000          JSR      %0,%0
10333 035366          RETN:          ;TEST FOR 'C' CLEARED
10334 035366 100004          BPL      1$
10335          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10336          ;          CONDITIONAL BRANCH INST. AND <====
10337          ;          REPLACE THE MOVE INSTRUCTION <====
10338          ;          WHICH FOLLOWS W/ 766 <====
10339 035370 012742 001055      MOV      #1055,-(R2) ;MOVE TO MAILBOX # ***** 1055 *****
10340 035374 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10341 035376 000000          HALT          ;C NOT CLEARED
10342 035400          1$:          BNE      2$
10343 035400 001004          BNE      2$
10344          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10345          ;          CONDITIONAL BRANCH INST. AND <====
10346          ;          REPLACE THE MOVE INSTRUCTION <====
10347          ;          WHICH FOLLOWS W/ 761 <====
10348 035402 012742 001056      MOV      #1056,-(R2) ;MOVE TO MAILBOX # ***** 1056 *****
10349 035406 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10350 035410 000000          HALT          ;Z NOT CLEARED
10351 035412          2$:          BVC      3$
10352 035412 102004          BVC      3$
10353          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10354          ;          CONDITIONAL BRANCH INST. AND <====
10355          ;          REPLACE THE MOVE INSTRUCTION <====
10356          ;          WHICH FOLLOWS W/ 754 <====
10357 035414 012742 001057      MOV      #1057,-(R2) ;MOVE TO MAILBOX # ***** 1057 *****
10358 035420 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10359 035422 000000          HALT          ;V NOT CLEARED
10360 035424          3$:          BCC      4$
10361 035424 103004          BCC      4$
10362          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10363          ;          CONDITIONAL BRANCH INST. AND <====
10364          ;          REPLACE THE MOVE INSTRUCTION <====
10365          ;          WHICH FOLLOWS W/ 747 <====

```

```
10366 035426 012742 001060      MOV    #1060,-(R2)      ;MOVE TO MAILBOX # ***** 1060 *****
10367 035432 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10368 035434 000000              HALT                    ;C NOT CLEARED
10369 035436 016700 142334      4$:  MOV    CC,%0       ;TEMP STORAGE
10370 035442 001404              BEQ    5$
10371              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10372              ;          CONDITIONAL BRANCH INST. AND <====
10373              ;          REPLACE THE MOVE INSTRUCTION <====
10374              ;          WHICH FOLLOWS W/ 740 <====
10375 035444 012742 001061      MOV    #1061,-(R2)      ;MOVE TO MAILBOX # ***** 1061 *****
10376 035450 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10377 035452 000000              HALT                    ;PRIORITY NOT ZERO
10378 035454 012706 001000      5$:  MOV    #BUFF,SP
10379 035460 012767 035476 142316  MOV    #RETO,RTRAP5
10380 035466 012767 000357 142312  MOV    #357,RTRAP5+2
10381 035474 004000              JSR    %0,%0           ;SET NEW 'CC' AND PRIORITY
10382 035476              RETO:  JSR    %0,%0           ;TRAP HERE
10383 035476 100404              BMI    1$
10384              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10385              ;          CONDITIONAL BRANCH INST. AND <====
10386              ;          REPLACE THE MOVE INSTRUCTION <====
10387              ;          WHICH FOLLOWS W/ 722 <====
10388 035500 012742 001062      MOV    #1062,-(R2)      ;MOVE TO MAILBOX # ***** 1062 *****
10389 035504 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10390 035506 000000              HALT                    ;N NOT SET
10391 035510              1$:
10392 035510 001404              BEQ    2$
10393              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10394              ;          CONDITIONAL BRANCH INST. AND <====
10395              ;          REPLACE THE MOVE INSTRUCTION <====
10396              ;          WHICH FOLLOWS W/ 715 <====
10397 035512 012742 001063      MOV    #1063,-(R2)      ;MOVE TO MAILBOX # ***** 1063 *****
10398 035516 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10399 035520 000000              HALT                    ;Z NOT SET
10400 035522              2$:
10401 035522 102404              BVS    3$
10402              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10403              ;          CONDITIONAL BRANCH INST. AND <====
10404              ;          REPLACE THE MOVE INSTRUCTION <====
10405              ;          WHICH FOLLOWS W/ 710 <====
10406 035524 012742 001064      MOV    #1064,-(R2)      ;MOVE TO MAILBOX # ***** 1064 *****
10407 035530 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10408 035532 000000              HALT                    ;V NOT SET
10409 035534              3$:
10410 035534 103404              BCS    4$
10411              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10412              ;          CONDITIONAL BRANCH INST. AND <====
10413              ;          REPLACE THE MOVE INSTRUCTION <====
10414              ;          WHICH FOLLOWS W/ 703 <====
10415 035536 012742 001065      MOV    #1065,-(R2)      ;MOVE TO MAILBOX # ***** 1065 *****
10416 035542 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10417 035544 000000              HALT                    ;C NOT SET
10418 035546 016700 142224      4$:  MOV    CC,%0
10419 035552 022700 000357      CMP    #357,%0
10420 035556 001404              BEQ    TS351
10421              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
```

```
0424 035580 012742 001066      MOV    #1066,-(R2)      ; MOVE TO MAILBOX # ***** 1066 *****
0425 035584 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
0426 035586 000000              HALT                    ; PRIORITY WAS CHANGED,OR WRONG $STNM
                                ; OR SEQUENCE ERROR
                                ;
                                ;          CONDITIONAL BRANCH INST. AND <====
                                ;          REPLACE THE MOVE INSTRUCTION <====
                                ;          WHICH FOLLOWS W/ 672 <====
0427 035570 005212              TS351: INC    (R2)       ; UPDATE TEST NUMBER
0428 035572 022712 000351        CMP    #351,(R2)      ; SEQUENCE ERROR?
0429 035576 001006              BNE   TS352-10        ; BR TO ERROR HALT ON SEQ ERROR
0430 035600 012706 000150        MOV    #150,%6        ; R6 = 150
0431 035604 012767 035624 142172  MOV    #TDEC1,4       ; STACK OVERFLOW TRAP POINTER
0432 035612 005746              TST   -(6)            ; WITH R6 = 150 SHOULD TRAP
0433 035614 012742 001067        MOV    #1067,-(R2)    ; MOVE TO MAILBOX # ***** 1067 *****
0434 035620 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
0435 035622 000000              HALT                    ; SHOULD HAVE TRAPPED,OR WRONG $STNM
                                ;
                                ; *****
                                ; TEST 351      TEST THAT DECREMENT R6 TO A VALUE LESS THAN 400 TRAPS
                                ; *****
                                ;
                                ; TDEC1:
0436 035624 005212              TS352: INC    (R2)       ; UPDATE TEST NUMBER
0437 035626 022712 000352        CMP    #352,(R2)      ; SEQUENCE ERROR?
0438 035632 001011              BNE   TS353-10        ; BR TO ERROR HALT ON SEQ ERROR
0439 035634 012706 000150        MOV    #150,%6        ; R6 = 150
0440 035640 012767 035650 142136  MOV    #TDEC2,4       ; TRAP POINTER
0441 035646 005746              TST   -(6)            ; WITH R6 = 150 SHOULD TRAP
0442 035650 020627 000142        TDEC2: CMP    %6,#142  ; DID R6 DECREMENT
0443 035654 001404              BEQ   TS353           ;
                                ;
                                ; *****
                                ; TEST 352      TEST FOR DECREMENT OF R6 ON OVERFLOW TRAP
                                ; *****
                                ;
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
                                ;          CONDITIONAL BRANCH INST. AND <
                                ;          REPLACE THE MOVE INSTRUCTION <-
                                ;          WHICH FOLLOWS W/ 766 <-
0444 035656 012742 001070        MOV    #1070,-(R2)    ; MOVE TO MAILBOX # ***** 1070 *****
0445 035662 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
0446 035664 000000              HALT                    ; R6 NOT = 142,OR WRONG $STNM
                                ; OR SEQUENCE ERROR
                                ;
                                ; *****
                                ; TEST 353      TEST DIFFERENT TYPES OF OVERFLOW
                                ; *****
0447 035666 005212              TS353: INC    (R2)       ; UPDATE TEST NUMBER
0448 035670 022712 000353        CMP    #353,(R2)      ; SEQUENCE ERROR?
0449 035674 001041              BNE   TS354-10        ; BR TO ERROR HALT ON SEQ ERROR
0450 035676 012706 000150        MOV    #150,%6        ;
0451 035702 005067 142240        CLR   146             ; STATUS WORD OF LOC 10
0452 035706 012767 035716 142070  MOV    #TDEC3,4       ; RETURN TO LOC 4
0453 035714 005246              INC    -(6)           ;
0454 035716 005767 142224        TDEC3: TST   146       ;
0455 035722 001004              BNE   1$              ;
                                ;
                                ; *****
                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=
                                ;          CONDITIONAL BRANCH INST. AND <=
0456 035722 001004              BNE   1$              ;
```



```
0534 036136 001011          BNE      VDEC6          ;BR TO ERROR HALT ON SEQ ERROR
0535 036140 012706 000400    MOV      #400,%6        ;SET UP STACK TO OVERFLOW
0536 036144 012767 036162 141656  MOV      #VDEC6,30      ;SET UP EMT VECTOR
0537 036152 012767 036172 141624  MOV      #VDEC5,4        ;SET UP OVERFLOW VECTOR
0538 036160 104000          EMT                    ;THIS TRAP SHOULD CAUSE OVERFLOW
0539 036162          VDEC6:
0540 036162 012742 001076    MOV      #1076,-(R2)    ;MOVE TO MAILBOX # ***** 1076 *****
0541 036166 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
0542 036170 000000          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $STNM
0543 036172 012767 000032 141630  VDEC5: MOV      #30+2,30
0544          ;*****
0545          ;TEST 357      TEST THAT AN TRAP CAUSES AN OVERFLOW TRAP
0546          ;*****
0547 036200 005212          TS357: INC      (R2)          ;UPDATE TEST NUMBER
0548 036202 022712 000357    CMP      #357,(R2)      ;SEQUENCE ERROR?
0549 036206 001011          BNE      VDEC8          ;BR TO ERROR HALT ON SEQ ERROR
0550 036210 012706 000400    MOV      #400,%6        ;SET UP STACK TO OVERFLOW
0551 036214 012767 036232 141612  MOV      #VDEC8,34      ;SET UP TRAP VECTOR
0552 036222 012767 036242 141554  MOV      #VDEC7,4        ;SET UP OVERFLOW VECTOR
0553 036230 104400          TRAP                    ;THIS TRAP SHOULD CAUSE OVERFLOW
0554 036232          VDEC8:
0555 036232 012742 001077    MOV      #1077,-(R2)    ;MOVE TO MAILBOX # ***** 1077 *****
0556 036236 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
0557 036240 000000          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $STNM
0558 036242 012767 000036 141564  VDEC7: MOV      #34+2,34
0559          ;*****
0560          ;TEST 360      TEST THAT AN TRT CAUSES AN OVERFLOW TRAP
0561          ;*****
0562 036250 005212          TS360: INC      (R2)          ;UPDATE TEST NUMBER
0563 036252 022712 000360    CMP      #360,(R2)      ;SEQUENCE ERROR?
0564 036256 001011          BNE      VDEC10         ;BR TO ERROR HALT ON SEQ ERROR
0565 036260 012706 000400    MOV      #400,%6        ;SET UP STACK TO OVERFLOW
0566 036264 012767 036302 141522  MOV      #VDEC10,14     ;SET UP TRT VECTOR
0567 036272 012767 036312 141504  MOV      #VDEC9,4        ;SET UP OVERFLOW VECTOR
0568 036300 000003          TRT                      ;THIS TRAP SHOULD CAUSE OVERFLOW
0569 036302          VDEC10:
0570 036302 012742 001100    MOV      #1100,-(R2)    ;MOVE TO MAILBOX # ***** 1100 *****
0571 036306 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
0572 036310 000000          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $STNM
0573 036312 012767 000016 141474  VDEC9: MOV      #14+2,14
0574          ;*****
0575          ;TEST 361      TEST THAT AN ILLA CAUSES AN OVERFLOW TRAP
0576          ;*****
0577 036320 005212          TS361: INC      (R2)          ;UPDATE TEST NUMBER
0578 036322 022712 000361    CMP      #361,(R2)      ;SEQUENCE ERROR?
0579 036326 001011          BNE      VDEC11         ;BR TO ERROR HALT ON SEQ ERROR
0580 036330 012706 000400    MOV      #400,%6        ;SET UP STACK TO OVERFLOW
0581 036334 012767 036352 141446  MOV      #VDEC11,10     ;SET UP ILLA VECTOR
0582 036342 012767 036362 141434  MOV      #VDEC12,4        ;SET UP OVERFLOW VECTOR
0583 036350 004700          ILLA                      ;THIS TRAP SHOULD CAUSE OVERFLOW
0584 036352          VDEC11:
0585 036352 012742 001101    MOV      #1101,-(R2)    ;MOVE TO MAILBOX # ***** 1101 *****
0586 036356 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
0587 036360 000000          HALT                    ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $STNM
0588 036362 012767 000012 141420  VDEC12: MOV      #10+2,10
0589 036370 020627 000370    CMP      %6,#370        ;STACK PUSHED FOUR WORDS?
```

```

10591 036374 001404          BEQ      TS362
10592                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
10593                                     ;          CONDITIONAL BRANCH INST. AND <==
10594                                     ;          REPLACE THE MOVE INSTRUCTION <===
10595                                     ;          WHICH FOLLOWS W/ 754 <===
10595 036376 012742 001102    MOV      #1102,-(R2) ;MOVE TO MAILBOX # ***** 1102 *****
10596 036402 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10597 036404 000000          HALT     ;TRAP OVERFLOW DID NOT OCCUR
10598                                     ; OR SEQUENCE ERROR
10599
10600 ;*****
10600 ;TEST 362      TEST THAT AN ILLB CAUSES AN OVERFLOW TRAP
10601 ;*****
10602 036406 005212          TS362:  INC      (R2)      ;UPDATE TEST NUMBER
10603 036410 022712 000362    CMP      #362,(R2)   ;SEQUENCE ERROR?
10604 036414 001011          BNE     VDEC13      ;BR TO ERROR HALT ON SEQ ERROR
10605 036416 012706 000400    MOV      #400,%6    ;SET UP STACK TO OVERFLOW
10606 036422 012767 036440 141360  MOV      #VDEC13,10 ;SET UP ILLB VECTOR
10607 036430 012767 036450 141346  MOV      #VDEC14,4  ;SET UP OVERFLOW VECTOR
10608 036436 000100          ILLB     ;THIS TRAP SHOULD CAUSE OVERFLOW
10609 036440
10610 036440 012742 001103    VDEC13: MOV      #1103,-(R2) ;MOVE TO MAILBOX # ***** 1103 *****
10611 036444 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10612 036446 000000          HALT     ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $STNM
10613 036450 012767 000012 141332  VDEC14: MOV      #10+2,10
10614
10615 ;*****
10616 ;TEST 363      TEST FOR FALSE OVERFLOW TRAP
10617 ;*****
10618 036456 005212          TS363:  INC      (R2)      ;UPDATE TEST NUMBER
10619 036460 022712 000363    CMP      #363,(R2)   ;SEQUENCE ERROR?
10620 036464 001023          BNE     FOVER       ;BR TO ERROR HALT ON SEQ ERROR
10621
10622 036466 012767 036534 141310  MOV      #FOVER,4    ;SET UP OVERFLOW POINTER
10623 036474 012706 001002    MOV      #1002,%6
10624 036500 005746          TST     -(6)        ;SHOULD NOT OVERFLOW
10625 036502 012706 002002    MOV      #2002,%6
10626 036506 005746          TST     -(6)        ;SHOULD NOT OVERFLOW
10627 036510 012706 004002    MOV      #4002,%6
10628 036514 005746          TST     -(6)        ;SHOULD NOT OVERFLOW
10629 036516 012706 010002    MOV      #10002,%6
10630 036522 005746          TST     -(6)
10631 036524 012706 020000    MOV      #20000,%6  ;SHOULD NOT OVERFLOW
10632 036530 005746          TST     -(6)
10633 036532 000404          BR      STP
10634
10635                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
10636                                     ;          CONDITIONAL BRANCH INST. AND <--
10637                                     ;          REPLACE THE MOVE INSTRUCTION <= -
10638                                     ;          WHICH FOLLOWS W/ 754 <===
10638 036534
10639 036534 012742 001104    FOVER:  MOV      #1104,-(R2) ;MOVE TO MAILBOX # ***** 1104 *****
10640 036540 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10641 036542 000000          HALT     ;IT OVERFLOWED,OR WRONG $STNM

```



```
10642 036544 012767 000006 141232 STP:   MOV    #6,4
10643 036552 005067 141230          CLR    6
;*****
;TEST 364      TEST THAT BIT 4 PSW WILL CAUSE A TRAP TO 14
;*****
10644
10645
10646
10647 036556 005212          TS364:  INC    (R2)          ;UPDATE TEST NUMBER
10648 036560 022712 000364          CMP    #364,(R2)        ;SEQUENCE ERROR?
10649 036564 001013          BNE   TS365-10         ;BR TO ERROR HALT ON SEQ ERROR
10650 036566 012706 001000          MOV    #BUFF,SP
10651 036572 012767 036624 141214          MOV    #RETAT,RTRAP4   ;SET UP TO TRAP TO 14
10652 036600 012746 000020          MOV    #20,-(SP)       ;PUSH T BIT
10653 036604 012746 036612          MOV    #.+6,-(SP)      ;PUSH PC
10654 036610 000002          RTI                    ;SET T BIT
10655 036612 000240          NOP                    ;TRAP HERE
10656 036614 012742 001105          MOV    #1105,-(R2)     ;MOVE TO MAILBOX # ***** 1105 *****
10657 036620 005242          INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10658 036622 000000          HALT                   ;TRACE BIT DID NOT TRAP!,OR WRONG $TESTN
10659 036624
RETAT:
;*****
;TEST 365      TEST STACK POINTER DECREMENTS
;*****
10660
10661
10662
10663 036624 005212          TS365:  INC    (R2)          ;UPDATE TEST NUMBER
10664 036626 022712 000365          CMP    #365,(R2)        ;SEQUENCE ERROR?
10665 036632 001022          BNE   TS366-10         ;BR TO ERROR HALT ON SEQ ERROR
10666 036634 012706 001000          MOV    #BUFF,SP
10667 036640 012767 036672 141146          MOV    #RETBT,RTRAP4   ;PUSH T BIT
10668 036646 012746 000020          MOV    #20,-(SP)       ;PUSH PC
10669 036652 012746 036660          MOV    #.+6,-(SP)      ;PUSH PC
10670 036656 000002          RTI                    ;SET T BIT
10671 036660 000240          NOP                    ;TRAP HERE
10672 036662 012742 001106          MOV    #1106,-(R2)     ;MOVE TO MAILBOX # ***** 1106 *****
10673 036666 005242          INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10674 036670 000000          HALT                   ;TRACE BIT DID NOT TRAP!
10675 036672 020627 000774          RETBT:  CMP    SP,#BUFF-4
10676 036676 001404          BEQ   TS366
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
;          CONDITIONAL BRANCH INST. AND <--
;          REPLACE THE MOVE INSTRUCTION <-
;          WHICH FOLLOWS W/ 755 <-
10677
10678
10679
10680
10681 036700 012742 001107          MOV    #1107,-(R2)     ;MOVE TO MAILBOX # ***** 1107 *****
10682 036704 005242          INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10683 036706 000000          HALT                   ;STACK POINTER WAS NOT PUSHED BY TRAP,OR WRONG $TESTN
10684
; OR SEQUENCE ERROR
;*****
;TEST 366      TEST FOR PROPER PC ON STACK
;*****
10685
10686
10687
10688 036710 005212          TS366:  INC    (R2)          ;UPDATE TEST NUMBER
10689 036712 022712 000366          CMP    #366,(R2)        ;SEQUENCE ERROR?
10690 036716 001016          BNE   TS367-10         ;BR TO ERROR HALT ON SEQ ERROR
10691 036720 012706 001000          MOV    #BUFF,SP
10692 036724 012767 036744 141062          MOV    #RETCT,RTRAP4   ;PUSH T BIT
10693 036732 012746 000020          MOV    #20,-(SP)       ;PUSH PC
10694 036736 012746 036744          MOV    #.+6,-(SP)      ;PUSH PC
10695 036742 000002          RTI                    ;SET T BIT
10696
10697 036744 022767 036744 142022          RETCT:  CMP    #.,BUFF-4
;TRAP HERE
```

0698 036754 012742 001110  
0700 036760 005242  
0701 036762 000000  
0702  
0703  
0704  
0705  
0706  
0707  
0708  
0709  
0710  
0711  
0712 036764 005212  
0713 036766 022712 000367  
0714 036772 001015  
0715  
0716 036774 012706 001000  
0717 037000 005001  
0718 037002 012746 000020  
0719 037006 012746 037022  
0720 037012 012767 037036 140774  
0721 037020 000006  
0722 037022 000240  
0723 037024 001404  
0724  
0725  
0726  
0727  
0728 037026 012742 001111  
0729 037032 005242  
0730 037034 000000  
0731  
0732  
0733 037036  
0734  
0735  
0736  
0737 037036 005212  
0738 037040 022712 000370  
0739 037044 001030  
0740 037046 012705 177777  
0741 037052 012706 001000  
0742 037056 012746 000020  
0743 037062 012746 037100  
0744 037066 012767 037116 140720  
0745 037074 005001  
0746 037076 000006  
0747 037100 005201  
0748 037102 005205  
0749 037104 001762  
0750 037106 012742 001112  
0751 037112 005242  
0752 037114 000000  
0753 037116 005301

BEQ TS367  
MOV #1110,-(R2)  
INC -(R2)  
HALT  
TS367: INC (R2)  
CMP #367,(R2)  
BNE TS370-10  
MOV #BUFF,SP  
CLR R1  
MOV #20,-(SP)  
MOV #RTT1,-(SP)  
MOV #RTT2,14  
RTT1: NOP  
BEQ TS370  
MOV #1111,-(R2)  
INC -(R2)  
HALT  
RTT2:  
TS370: INC (R2)  
CMP #370,(R2)  
BNE TS371-10  
MOV #177777,%5  
RTT5: MOV #BUFF,SP  
MOV #20,-(SP)  
MOV #RTT3,-(SP)  
MOV #RTT4,14  
CLR R1  
RTT3: INC R1  
INC %5  
BEQ RTT5  
MOV #1112,-(R2)  
INC -(R2)  
HALT  
RTT4: DEC R1

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <=====  
: REPLACE THE MOVE INSTRUCTION <=====  
: WHICH FOLLOWS W/ 761 <=====  
: MOVE TO MAILBOX # \*\*\*\*\* 1110 \*\*\*\*\*  
: SET MSGTYP TO FATAL ERROR  
: CORRECT PC WAS NOT SAVED ON STACK,OR WRONG \$TESTN  
: OR SEQUENCE ERROR

\*\*\*\*\*  
:TEST 367 TEST THAT RTT POPS T- RTT  
\*\*\*\*\*

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

:CLEAR R1

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <  
: CONDITIONAL BRANCH INST. AND <  
: REPLACE THE MOVE INSTRUCTION <  
: WHICH FOLLOWS W/ 762 <  
: MOVE TO MAILBOX # \*\*\*\*\* 1111 \*\*\*\*\*  
: SET MSGTYP TO FATAL ERROR  
: T-BIT DID NOT TRAP,OR WRONG \$TESTN  
: OR SEQUENCE ERROR

\*\*\*\*\*  
:TEST 370 TEST THAT RTT ALLOWS ONE INST. BEFORE TRAP  
\*\*\*\*\*

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

:CLEAR R0  
:SET T-BIT

:DO THIS TEST NO MORE THAN 2 TIMES  
:MOVE TO MAILBOX # \*\*\*\*\* 1112 \*\*\*\*\*  
:SET MSGTYP TO FATAL ERROR  
:DID NOT TRAP  
:SEE IF RTT ALLOWS 1 INST.

```

10754 037120 001406      BEQ      RTT6
10755 037122 005205      INC      %5          ;DO THIS TEST NO MORE THAN TWO TIMES
10756 037124 001752      BEQ      RTT5
10757                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10758                      ;          CONDITIONAL BRANCH INST. AND <====
10759                      ;          REPLACE THE MOVE INSTRUCTION <====
10760                      ;          WHICH FOLLOWS W/ 747 <====
10761 037126 012742 001113  MOV      #1113,-(R2) ;MOVE TO MAILBOX # ***** 1113 *****
10762 037132 005247      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10763 037134 000000      HALT          ;RTT DID NOT ALLOW 1 INST.,OR WRONG $TESTN
10764 037136
10765
10766
10767
10768 037136 005212      TS371: INC      (R2)          ;UPDATE TEST NUMBER
10769 037140 022712 000371  CMP      #371,(R2)      ;SEQUENCE ERROR?
10770 037144 001022      BNE      TS372-10      ;BR TO ERROR HALT ON SEQ ERROR
10771 037146 012706 001000  MOV      #BUFF,SP
10772 037152 012746 000020  MOV      #20,-(SP)
10773 037156 012746 037174  MOV      #RTI1,-(SP)
10774 037162 012767 037206 140624  MOV      #RTI2,14
10775 037170 005001      CLR      R1
10776 037172 000002      RTI
10777 037174 005201      RTI1: INC      R1          ;SET T-BIT
10778 037176 012742 001114  MOV      #1114,-(R2)   ;RTI SHOULD NOT ALLOW THIS
10779 037202 005242      INC      -(R2)      ;MOVE TO MAILBOX # ***** 1114 *****
10780 037204 000000      HALT          ;SET MSGTYP TO FATAL ERROR
10781 037206 005701      RTI2: TST      R1      ;T- BIT DID NOT CAUSE TRAP
10782
10783 037210 001404      BEQ      TS372          ;RTI SHOULD NOT ALLOW 1 INST. BEFORE TRAP
10784
10785                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10786                      ;          CONDITIONAL BRANCH INST. AND <====
10787                      ;          REPLACE THE MOVE INSTRUCTION <====
10788                      ;          WHICH FOLLOWS W/ 755 <====
10788 037212 012742 001115  MOV      #1115,-(R2)   ;MOVE TO MAILBOX # ***** 1115 *****
10789 037216 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10790 037220 000000      HALT          ;RTI DID ALLOW 1 INST. BEFORE TRAP,OR WRONG $TESTN
10791
10792                      ; OR SEQUENCE ERROR
10793
10794
10795
10796 037222 005212      ;*****
10797 037224 022712 000372  ;TEST 372 TEST TRAP ON TRAP THAT TRACE BIT TRAPS ARE INHIBITED ON TRAP INST
10798 037230 001026      ;*****
10799
10800 037232 012706 001000  TS372: INC      (R2)          ;UPDATE TEST NUMBER
10801 037236 012767 037276 140550  CMP      #372,(R2)      ;SEQUENCE ERROR?
10802 037244 005027 000016  BNE      BR70          ;BR TO ERROR HALT ON SEQ ERROR
10803 037250 005027 000022  MOV      #BUFF,%6
10804 037254 012767 037316 140536  MOV      #TRACE,14      ;TRACE TRAP
10805 037262 012746 000020  CLR      #16
10806 037266 012746 037274  CLR      #22
10807 037272 000006  MOV      #TONT1,20      ;IOT TRAP
10808 037274 000004  MOV      #20,-(SP)      ;PUSH T BIT
10809 037276 000000  MOV      #.+6,-(SP)     ;PUSH PC
10809
TRACE:

```

```

081 037276 012742 001116      MOV    #1116,-(R2)      ;MOVE TO MAILBOX # ***** 1116 *****
082 037302 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
083 037304 000000              HALT                    ;TRACE TRAP WAS NOT INHIBITED
084 037306 012742 001117      BR70:  MOV    #1117,-(R2)      ;MOVE TO MAILBOX # ***** 1117 *****
085 037312 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
086 037314 000000              HALT                    ;WRONG TSTNM,OR WRONG $TSTNM
087 037316 012767 000016 140470 TONT1: MOV    #16,14
088 037324 012767 000022 140466      MOV    #22,20
089
090
091
092
093
094
095
096
097
098
099
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
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
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
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
296
297
298
299
300
301
302
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
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865

```

```

10865 037470 012706 001000      MOV      #BUFF,SP      ;SET STACK POINTER
10867 037474 105720      NOR:     TSTB          (0)+      ;IF OUTSIDE OF CORE, TRAP TO 4
10868 037476 020027      CMP      R0,(PC)+      ;IS POINTER INSIDE 28K (30K) CORE
10869 037500 160000      HICORE: .WORD        160000      ;MAY BE CHANGED TO 170000 IF 30K
10870 037502 103774      BLO      NOR           ;TEST THE REST OF CORE
10871 037504 012737 037526 000004      MOV      #ROTRAP,@#4    ;SET UP NEW VECTOR POINTER
10872 037512 105737 177700      TSTB     @#1/7700      ;SHOULD CAUSE A TRAP
10873 037516
10874 037516 012742 001121      MOV      #1121,-(R2)    ;MOVE TO MAILBOX # ***** 1121 *****
10875 037522 005242      INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
10876 037524 000000      HALT
10877      ;TRAP TO HERE IF FORCING TRAP BY TESTING 177700
10878 037526 106767 140244      ROTRAP: MFFS        STATUS
10879 037532 005767 140240      TST     STATUS        ;TEST PSW
10880 037536 001404      BEQ     1$
10881      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
10882      ;          CONDITIONAL BRANCH INST. AND <==
10883      ;          REPLACE THE MOVE INSTRUCTION <==
10884      ;          WHICH FOLLOWS W/ 733 <
10885 037540 012742 001122      MOV      #1122,-(R2)    ;MOVE TO MAILBOX # ***** 1122 *****
10886 037544 005242      INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
10887 037546 000000      HALT
10888 037550 026727 141220 037516 1$:  CMP      BUFF-4,#TRPADR ;NEW PSW SHOULD HAVE BEEN ZERO
10889 037556 001453      BEQ     TRAPB         ;TEST OLD PC AT STACK
10890      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
10891      ;          CONDITIONAL BRANCH INST. AND <
10892      ;          REPLACE THE MOVE INSTRUCTION <
10893      ;          WHICH FOLLOWS W/ 723 <
10894 037560 012742 001123      MOV      #1123,-(R2)    ;MOVE TO MAILBOX # ***** 1123 *****
10895 037564 005242      INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
10896 037566 000000      HALT
10897      ;RETURN HERE ON AN ADDRESS TRAP FROM MEMORY BELOW 28K (OR 30K)
10898 037570 005300      ATRAP:  DEC      R0
10899 037572 010067 000032      MOV      R0,CORH      ;MOVE THE FIRST NXM LOCATION IN CORH
10900      ;THIS ROUTINE DOES NXM TRAPS UNTIL IT FINDS AN EXISTENT MEMORY LOCATION
10901 037576 013700 037500      MOV      @#HICORE,R0   ;SET UP THE HIGHEST MEM LOCATION
10902 037602 005300      DEC      R0           ;MAKE 1 LESS THAN THE HIGHEST CORE BOUNDARY
10903 037604 000402      BR      NOSUB         ;DON'T SUBTRACT 1K FIRST TIME
10904 037606 162700 001000      CTRAP:  SUB      #1000,R0 ;SUBTRACT 1K OCTAL BYTE FROM ADDRESS
10905      ;TO SPEED UP TESTING
10906 037612 012767 037644 140164      NOSUB:  MOV      #BTRAP,4 ;SET UP THE VECTOR
10907 037620 012706 001000      MOV      #BUFF,SP
10908 037624 005710      TST     (R0)         ;DOES THIS MEMORY EXIST?
10909      ;IF NXM, TRAP TO BTRAP
10910 037626 020027      DTRAP1: CMP      R0,(PC)+ ;IF EXISTS, IS THIS THE SAME TRAP THAT CAUSED
10911      ;TRAP TO ATRAP
10912 037630 000000      CORH:   .WORD        C
10913 037632 101425      BLOS    TRAPB
10914      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10915      ;          CONDITIONAL BRANCH INST. AND <==
10916      ;          REPLACE THE MOVE INSTRUCTION <==
10917      ;          WHICH FOLLOWS W/ 675 <==
10918 037634 012742 001124      MOV      #1124,-(R2)    ;MOVE TO MAILBOX # ***** 1124 *****
10919 037640 005242      INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
10920 037642 000000      HALT
10921      ;CONTENTS OF R0 SHOULD BE LESS THAN OR EQUAL TO CORH
10921      ;IF THIS COMPARISON FAILS IT MEANS

```

```

1092.                                     :THAT SOME LEGAL ADDRESS TRAPPED, OR
10923                                     :THAT AN ILLEGAL ADDRESS DID NOT TRAP
10924 037644 106767 140126      BTRAP: MFPS      STATUS
10925 037650 005767 140122      TST        STATUS
10926 037654 001404      BEQ        1$
10927                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
10928                                     ;          CONDITIONAL BRANCH INST. AND <
10929                                     ;          REPLACE THE MOVE INSTRUCTION <
10930                                     ;          WHICH FOLLOWS W/ 664 <
10931 037656 012742 001125      MOV        #1125,-(R2) ;MOVE TO MAILBOX # ***** 1125 *****
10932 037662 005242      INC        -(R2) ;SET MSGTYP TO FATAL ERROR
10933 037664 000000      HALT      ;NEW PSW SHOULD HAVE BEEN ZERO
10934 037666 026727 141102 037626 1$: CMP        BUFF-4,#DTRAP1 ;CHECK IF TRAP PC IS OK
10935 037674 001744      BEQ        CTRAP
10936                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
10937                                     ;          CONDITIONAL BRANCH INST. AND <- --
10938                                     ;          REPLACE THE MOVE INSTRUCTION <=
10939                                     ;          WHICH FOLLOWS W/ 654 <==
10940 037676      AUTO1: MOV        #1126,-(R2) ;MOVE TO MAILBOX # ***** 1126 *****
10941 037676 012742 0C1126      INC        -(R2) ;SET MSGTYP TO FATAL ERROR
10942 037702 005242      HALT      ;OLD PC WAS NOT SAVED OR WRONG $TESTN
10943 037704 000000      TRAPB: MOV        #6,4 ;RESET TRAP CATCHER
10944 037706 012767 000006 140070 CLR        6 ;RESET TRAP CATCHER
10945 037714 005067 140066
10946
10947                                     ;THIS ROUTINE WILL FIGURE OUT IF YOU HAVE A DL11W
10948                                     CLR        PROFTE
10949 037720 095067 000020      MOV        #BUFF,SP ;SET UP THE STACK POINTER
10950 037724 012706 001000      MOV        #DL11W,4 ;SET UP THE TRAP VECTOR
10951 037730 012767 037746 140046 TST        TPS ;TEST THE PUNCH STATUS REGISTER
10952 037736 005767 137622      BR        DL11W1
10953 037742 000403      PROFTE: 000000
10954 037744 000000      DL11W: INC        PROFTE ;INCR IF NO DL11W
10955 037746 005267 177772      DL11W1: MOV        #6,4
10956 037752 012767 000006 140024
10957
10958 037760      SKP104:
10959                                     ;*****
10960                                     ;TEST 375 TEST THAT A TTY INTERRUPT CAUSES AN OVERFLOW TRAP
10961                                     ;*****
10962 037760 005212      TS375: INC        (R2) ;UPDATE TEST NUMBER
10963 037762 022712 000375      CMP        #375,(R2) ;SEQUENCE ERROR?
10964 037766 001037      BNE        TDEC8 ;BR TO ERROR HALT ON SEQ ERROR
10965 037770 005767 177750      TST        PROFTE
10966 037774 001047      BNE        R7TRX
10967 037776 122767 000001 140314 CMPB       #APTENV,$ENV ;RUNING IN APT MODE?
10968 040004 001003      BNE        2$ ;IF NOT, DO THIS TEST
10969 040006 005767 140274      TST        $PASS ;IS THIS THE FIRST PASS?
10970 040012 001040      BNE        R7TRX ;IF NOT FIRST PASS, SKIP TEST
10971 040014
10972 040014 000005      2$: RESET
10973 040016 012767 000340 137752 MOV        #340,STATUS ;LOCK OUT INTERRUPT
10974 040024 012706 000400 MOV        #400,26 ;SET UP STACK TO OVERFLOW
10975 040030 012767 040076 137746 MOV        #TDEC77,4 ;SET UP OVERFLOW TRAP
10976 040036 012767 040066 140020 MOV        #TDEC8,64 ;SET UP INTERRUPT VECTOR
10977 040044 012767 000100 137512 MOV        #100,TTCSR ;SET INTERRUPT ENABLE

```

T375 TEST THAT A TTY INTERRUPT CAUSES AN OVERFLOW TRAP

040052 005067 137720
040056 012742 C01127
040062 005242
040064 000000
040066 001130
040072 005242
040074 000000
040076 005067 137462
040102 012767 000006 137674
040110 005067 137672
040114

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 ENABLF
MOV #6,4
CLR 6

R7TRX:
:\*\*\*\*\*
:TEST 376 TEST THAT A PENDING INTERRUPT OCCURS BEFORE TRAP
:\*\*\*\*\*

040114 005212
040116 022712 000376
040122 001045
040124 005767 177614
040130 001053
040132 122767 000001 140160
040140 001003
040142 005767 140140
040146 001044
040150
040154 012706 001000
040154 012767 000340 137614
040162 012767 040226 137674
040170 012767 000100 137366
040176 012767 040236 137630
040204 012767 040246 137652
040212 012767 000340 137616
040220 005067 137552
040224 104400
040226
040226 012742 001131
040232 005242
040234 000000
040236
040236 012742 001132
040242 005242
040244 000000
040246 005067 137564
040252 042767 000100 137304
040260

TS376: 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 #TR0,64
MOV #100,TTCSR ;INTERRUPT FOR TTY PUNCH/PRINTER
MOV #BR71,34 ;TRAP VECTOR
MOV #TR2,64 ;TTY VECTOR
MOV #340,36 ;IF TRAP TRAPS, MOVE 340 TO PRIORITY
CLR STATUS ;SHOULD INTERRUPT AT END OF CLR INST
TRAP ;TTY INTERRUPT SHOULD OVERRIDE TRAP
TRO: 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
:\*\*\*\*\*
TS377: 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?

```
11034 040312 001054          BNE      NODL1          ;IF NOT FIRST PASS, SKIP TEST
11035 040314          2$:      MOV      #BUFF,%6
11036 040314 012706 001000          MOV      #340,STATUS
11037 040320 012767 000340 137450          MOV      #100,TRCSR
11038 040326 012767 000100 137230          MOV      #TR3,34          ;TRAP
11039 040334 012767 040374 137472          MOV      #TR4,64          ;TTY OUTPUT
11040 040342 012767 040406 137514          MOV      #340,66          ;TTY OUTPUT PRIORITY
11041 040350 012767 000340 137510          MOV      #TR5,20          ;IOT
11042 040356 012767 040376 137434          MOV      #340,22          ;IOT PRIORITY
11043 040364 012767 000340 137430          TRAP          ;THE ACT OF TRAPPING LOWER PRIORITY
11044 040372 104400          TR3:      IOT          ;INTERRUPT SHOULD OCCUR IN PLACE OF IOT TRAP
11045 040374 000004          TR5:
11046 040376          MOV      #1133,-(R2)      ;MOVE TO MAILBOX # ***** 1133 *****
11047 040376 012742 001133          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
11048 040402 005242          HALT          ;NO INTERRUPT BETWEEN TRAPS,OR WRONG $STNM
11049 040404 010000          TR4:      CLR      22          ;CLR IOT PRIORITY
11050 040406 005067 137410          CLR      66
11051 040412 005067 137450          MOV      #36,34
11052 040416 012767 000036 137410          MOV      #66,64
11053 040424 012767 000066 137432          MOV      #22,20
11054 040432 012767 000022 137360          CLR      TRCSR
11055 040440 005067 137120          NODL1:
11056 040444
11057
11058 ;*****
11059 ;TEST 400 TEST THAT 'RESET' GOES TO OUTSIDE WORLD
11060 ;*****
11061 040444 005212          TS400:  INC      (R2)          ;UPDATE TEST NUMBER
11062 040446 022712 000400          CMP      #400,(R2)      ;SEQUENCE ERROR?
11063 040452 001030          BNE      TS401-10       ;BR TO ERROR HALT ON SEQ ERROR
11064 040454 005767 177264          TST     PROFTE
11065 040460 001031          BNE      NODL2
11066 040462 122767 000001 137630          CMPB   #APTENV,$ENV    ;RUNNING IN APT MODE?
11067 040470 001003          BNE      1$           ;IF NO DO TEST.
11068 040472 005767 137610          TST     $PASS          ;IF YES THEN ARE WE ON FIRST PASS?
11069 040476 001022          BNE      NODL2        ;IF IN APT MODE AND FIRST PASS, SKIP TEST.
11070 040500 016700 137056          1$:     MOV      TKB,R0        ;MAKE SURE RECEIVER DONE IS SET
11071 040504 012767 000100 137046          MOV      #100,TRCSR    ;SET INTERRUPT ENABLE
11072 040512 000005          RESET   ;SHOULD CLEAR INTERRUPT ENABLE
11073 040514 032767 000100 137036          BIT     #100,TRCSR     ;TEST FOR CLEAR
11074 040522 001410          BEQ     TS401
11075          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11076          ;          CONDITIONAL BRANCH INST. AND <====
11077          ;          REPLACE THE MOVE INSTRUCTION <====
11078          ;          WHICH FOLLOWS W/ 753 <====
11079 040524 012742 001134          MOV      #1134,-(R2)    ;MOVE TO MAILBOX # ***** 1134 *****
11080 040530 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
11081 040532 000000          HALT     ;RESET FAILED TO CLEAR TRCSR
11082          ; OR SEQUENCE ERROR
11083 040534 012742 001135          MOV      #1135,-(R2)    ;MOVE TO MAILBOX # ***** 1135 *****
11084 040540 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
11085 040542 000000          HALT     ;WRONG $STNM
11086 040544          NODL2:
11087
11088 ;*****
11089 ;TEST 401 TEST THAT RESET HAS NO EFFECT ON THE TRACE TRAP
11090 ;*****
```



```

090 040544 005212          TS401: INC (R2)          ;UPDATE TEST NUMBER
091 040546 022712 006401    CMP #401,(R2)        ;SEQUENCE ERROR?
092 040552 001023          BNE RESET3         ;BR TO ERROR HALT ON SEQ ERROR
093 040554 122767 000001 137536 CMPB #APTENV,$ENV   ;RUNNING IN APT MODE?
094 040562 001003          BNE 2$            ;IF NO DO TEST.
095 040564 005767 137516    TST $PASS         ;IF YES THEN ARE WE ON FIRST PASS?
096 040570 001027          BNE SKTST2        ;IF IN APT MODE AND FIRST PASS, SKIP TEST.
097 040572 012706 001000    2$: MOV #BUFF,%6     ;SET STACK
098 040576 012767 040632 137210 MOV #RESET2,14    ;SET UP TRACE VECTOR
099 040604 012746 000020    MOV #20,-(R6)    ;SET THE T-BIT ON STACK
100 040610 012746 040616    MOV #1$,-(R6)   ;MOVE NEW PC ON STACK
101 040614 000006          RTT
102 040616 000005          1$: RESET          ;SHOULD HAVE NO EFFECT
103 040620 000005          RESET          ;NO EFFECT
104 040622
105 040622 012742 001136    RESET3: MOV #1136,-(R2) ;MOVE TO MAILBOX # ***** 1136 *****
106 040626 005242          INC -(R2)        ;SET MSGTYP TO FATAL ERROR
107 040630 000000          HALT           ;TRACE TRAP FAILED,OR WRONG $STNM
108 040632 005067 137140    RESET2: CLR STATUS ;CLEAR TRACK
109 040636 005067 137154    CLR 16          ;TRACE STATUS
110 040642 012767 000016 137144 MOV #16,14
111 040650
112 :*****
113 :TEST 402 TEST THAT WHEN TTY INTERRUPTS IT POPS NEW STATUS
114 :*****
115 040650 005212          TS402: INC (R2)          ;UPDATE TEST NUMBER
116 040652 022712 000402    CMP #402,(R2)   ;SEQUENCE ERROR?
117 040656 001060          BNE TTY11       ;BR TO ERROR HALT ON SEQ ERROR
118 040660 005767 177060    TST PROFIE
119 040664 001063          BNE NODL3
120 040666 122767 000001 137424 CMPB #APTENV,$ENV ;RUNING IN APT MODE?
121 040674 001003          BNE 2$         ;IF NOT, DO THIS TEST
122 040676 005767 137404    TST $PASS      ;IS THIS THE FIRST PASS?
123 040702 001054          BNE NODL3      ;IF NOT FIRST PASS, SKIP TEST
124 040704
125 040704 000005          2$: RESET
126 040706 012706 001000    MOV #BUFF,%6   ;SET UP STACK
127 040712 012767 040740 137144 MOV #TTY3,64   ;INTERRUPT VECTOR
128 040720 005067 137052    CLR STATUS     ;DROP PROCESSOR PRIORITY
129 040724 012767 000357 137134 MOV #357,66    ;HIGH PRIORITY ON INTERRUPT
130 040732 052767 000300 136624 BIS #300,TTCSR ;SHOULD SET INTERRUPT ENABLE & INTERRUPT
131 040740 026727 137032 000357 TTY3: CMP STATUS,#357
132 040746 001404          BEQ 1$
133 :
134 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
135 : CONDITIONAL BRANCH INST. AND <=
136 : REPLACE THE MOVE INSTRUCTION <-
137 : WHICH FOLLOWS W/ 743 <-
138 040750 012742 001137    MOV #1137,-(R2) ;MOVE TO MAILBOX # ***** 1137 *****
139 040754 005242          INC -(R2)      ;SET MSGTYP TO FATAL ERROR
140 040756 000000          HALT         ;INTERRUPT DID NOT POP CORRECT STATUS
141 040760 000005          1$: RESET     ;CLR INTERRUPT ENABLE
142 040762 012706 001000    MOV #BUFF,%6   ;STACK SET UP
143 040766 012767 041012 137070 MOV #TTY4,64   ;INTERRUPT VECTOR
144 040774 005067 137066    CLR 66        ;CLR NEW STATUS
145 041000 012767 000157 136770 MOV #157,STATUS ;PROCESSOR STATUS
146 041006 105167 136552    COMB TTCSR    ;SET INTERRUPT ENABLE

```

```

11146 041012 00576~ 136760      TTY4:  TST      STATUS
11147 041016 001404              BEQ      TTT37
11148                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
11149                                     ;                                     <===
11150                                     ;                                     <===
11151                                     ;                                     <===
11152 041020              TTY11:
11153 041020 012742 001140      MOV      #1140,-(R2) ;MOVE TO MAILBOX # ***** 1140 *****
11154 041024 005242              INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
11155 041026 000000              HALT                    ;INCORRECT STATUS,OR WRONG $STNM
11156 041030 105167 136530      TTT37:  COMB      TTCSR
11157 041034              NODL3:
11158
11159
11160 ;*****
11161 ;TEST 403      TEST THE 'WAIT' INSTRUCTION
11162 ;*****
11162 041034 005212              TS403:  INC      (R2)      ;UPDATE TEST NUMBER
11163 041036 022712 000403      CMP      #403,(R2)    ;SEQUENCE ERROR?
11164 041042 001062              BNE      STP4         ;BR TO ERROR HALT ON SEQ ERROR
11165 041044 122767 000001 137246  CMPB     #APTENV,$ENV ;RUNING IN APT MODE?
11166 041052 001003              BNE      1$          ;IF NOT, DO THIS TEST
11167 041054 005767 137226      TST      $PASS       ;IS THIS THE FIRST PASS?
11168 041060 001057              BNE      STP4E       ;IF NOT FIRST PASS, SKIP TEST
11169 041062
11170 041062 042767 000100 136474 1$:      BIC      #100,TPS    ;CLEAR INTERRUPT ENABLE
11171 041070 012706 001000      MOV      #BUFF,SP    ;SET UP THE STACK
11172 041074 012767 041162 136762      MOV      #WATE,64    ;SET UP THE INTERRUPT VECTOR
11173 041102 005067 136760      CLR      66
11174 041106 105767 136452      WATE1:  TSTB     TPS    ;WAIT FOR READY
11175 041112 100375              BPL     WATE1        ;TO BE UP
11176 041114 012767 000015 136444      MOV      #15,TPB    ;DO A CARRIAGE RETURN
11177 041122 105767 136436      WATE2:  TSTB     TPS    ;WAIT FOR READY TO COME UP
11178 041126 100375              BPL     WATE2
11179 041130 012767 000015 136430      MOV      #15,TPB    ;DO ANOTHER CARRIAGE RETURN
11180 041136 052767 000100 136420      BIS      #100,TPS   ;SET THE INTERRUPT ENABLE
11181 041144 005067 136626      CLR     STATUS      ;CLEAR THE PSW
11182 041150 000001              WATE3:  WAIT                    ;WAIT FOR THE INTERRUPT
11183 041152 012742 001141      MOV      #1141,-(R2) ;MOVE TO MAILBOX # ***** 1141 *****
11184 041156 005242              INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
11185 041160 000000              HALT                    ;WAIT INSTRUCTION DID NOT LOOP
11186 041162 005767 136610      WATE:  TST      STATUS ;IS THE PSW CORRECT?
11187 041166 001404              BEQ      1$
11188                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
11189                                     ;                                     <-
11190                                     ;                                     <-
11191                                     ;                                     <-
11192 041170 012742 001142              MOV      #1142,-(R2) ;MOVE TO MAILBOX # ***** 1142 *****
11193 041174 005242              INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
11194 041176 000000              HALT                    ;NEW PSW SHOULD HAVE BEEN ZERO
11195 041200 026727 137570 041152 1$:      CMP      BUFF-4,#WATE3+2 ;IS THE OLD PC SAVED
11196 041206 001404              BEQ      STP4E
11197
11198                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <= -
11199                                     ;                                     <= -
1200                                     ;                                     <= -
1201 041210              STP4:

```

```
11202 041210 012742 001143      MOV    #1143,-(R2)      ;MOVE TO MAILBOX # ***** 1143 *****
11203 041214 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
11204 041216 000000              HALT                   ;OLD PC WAS NOT SAVED OR WRONG $TESTN
11205 041220
11206
11207
11208
11209 041220 005212              STP4E:
11210 041222 022712 000404      ;*****
11211 041226 001017      TS404:  INC    (R2)           ;UPDATE TEST NUMBER
11212
11213
11214
11215
11216 041230 012706 000000      MOV    #BUFF,SP        ;SET STACK POINTER
11217 041234 012737 041256 000004  MOV    #RETR1,@#RTRAP5 ;SET TRAP RETURN ADDR
11218 041242 005737 177700      PCN1:  TST    @#177700    ;BAD ADDR REFERENCE, TRAP TO 4
11219 041246 012742 001144      MOV    #1144,-(R2)     ;MOVE TO MAILBOX # ***** 1144 *****
11220 041252 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
11221 041254 000000              HALT                   ;REFERENCING 177700 DID NOT CAUSE TIME OUT
11222 041256 022767 041246 137510 RETR1:  CMP    #PCN1+4,BUFF-4  ;PROPER PC STORED ON STACK?
11223 041264 001404      BEQ    TS405
11224
11225
11226
11227
11228 041266 012742 001145      MOV    #1145,-(R2)     ;MOVE TO MAILBOX # ***** 1145 *****
11229 041272 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
11230 041274 000000              HALT                   ;OLD PC WAS NOT SAVED IN STACK
11231
11232
11233
11234
11235
11236
11237
11238
11239 041276 005212              ;*****
11240 041300 022712 000405      ;ODD ADDRESS USED BY A 'WORD' INSTRUCTION SHOULD NOT
11241 041304 001013      ;CAUSE A TRAP, BUT THE LOW ORDER ADDRESS BIT WOULD BE IGNORED.
11242
11243
11244
11245
11246
11247
11248
11249
11250
11251
11252 041334
11253 041334 012742 001146      ;*****
11254 041340 005242              ;TEST 405
11255 041342 000000      ;TEST ODD ADDRESS TRAP IS NOT IMPLEMENTED.
11256
11257
11258
11259
11260
11261
11262
11263
11264
11265
11266
11267
11268
11269
11270
11271
11272
11273
11274
11275
11276
11277
11278
11279
11280
11281
11282
11283
11284
11285
11286
11287
11288
11289
11290
11291
11292
11293
11294
11295
11296
11297
11298
11299
11300
11301
11302
11303
11304
11305
11306
11307
11308
11309
11310
11311
11312
11313
11314
11315
11316
11317
11318
11319
11320
11321
11322
11323
11324
11325
11326
11327
11328
11329
11330
11331
11332
11333
11334
11335
11336
11337
11338
11339
11340
11341
11342
11343
11344
11345
11346
11347
11348
11349
11350
11351
11352
11353
11354
11355
11356
11357
11358
11359
11360
11361
11362
11363
11364
11365
11366
11367
11368
11369
11370
11371
11372
11373
11374
11375
11376
11377
11378
11379
11380
11381
11382
11383
11384
11385
11386
11387
11388
11389
11390
11391
11392
11393
11394
11395
11396
11397
11398
11399
11400
11401
11402
11403
11404
11405
11406
11407
11408
11409
11410
11411
11412
11413
11414
11415
11416
11417
11418
11419
11420
11421
11422
11423
11424
11425
11426
11427
11428
11429
11430
11431
11432
11433
11434
11435
11436
11437
11438
11439
11440
11441
11442
11443
11444
11445
11446
11447
11448
11449
11450
11451
11452
11453
11454
11455
11456
11457
11458
11459
11460
11461
11462
11463
11464
11465
11466
11467
11468
11469
11470
11471
11472
11473
11474
11475
11476
11477
11478
11479
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
11521
11522
11523
11524
11525
11526
11527
11528
11529
11530
11531
11532
11533
11534
11535
11536
11537
11538
11539
11540
11541
11542
11543
11544
11545
11546
11547
11548
11549
11550
11551
11552
11553
11554
11555
11556
11557
11558
11559
11560
11561
11562
11563
11564
11565
11566
11567
11568
11569
11570
11571
11572
11573
11574
11575
11576
11577
11578
11579
11580
11581
11582
11583
11584
11585
11586
11587
11588
11589
11590
11591
11592
11593
11594
11595
11596
11597
11598
11599
11600
11601
11602
11603
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
11660
11661
11662
11663
11664
11665
11666
11667
11668
11669
11670
11671
11672
11673
11674
11675
11676
11677
11678
11679
11680
11681
11682
11683
11684
11685
11686
11687
11688
11689
11690
11691
11692
11693
11694
11695
11696
11697
11698
11699
11700
11701
11702
11703
11704
11705
11706
11707
11708
11709
11710
11711
11712
11713
11714
11715
11716
11717
11718
11719
11720
11721
11722
11723
11724
11725
11726
11727
11728
11729
11730
11731
11732
11733
11734
11735
11736
11737
11738
11739
11740
11741
11742
11743
11744
11745
11746
11747
11748
11749
11750
11751
11752
11753
11754
11755
11756
11757
11758
11759
11760
11761
11762
11763
11764
11765
11766
11767
11768
11769
11770
11771
11772
11773
11774
11775
11776
11777
11778
11779
11780
11781
11782
11783
11784
11785
11786
11787
11788
11789
11790
11791
11792
11793
11794
11795
11796
11797
11798
11799
11800
11801
11802
11803
11804
11805
11806
11807
11808
11809
11810
11811
11812
11813
11814
11815
11816
11817
11818
11819
11820
11821
11822
11823
11824
11825
11826
11827
11828
11829
11830
11831
11832
11833
11834
11835
11836
11837
11838
11839
11840
11841
11842
11843
11844
11845
11846
11847
11848
11849
11850
11851
11852
11853
11854
11855
11856
11857
11858
11859
11860
11861
11862
11863
11864
11865
11866
11867
11868
11869
11870
11871
11872
11873
11874
11875
11876
11877
11878
11879
11880
11881
11882
11883
11884
11885
11886
11887
11888
11889
11890
11891
11892
11893
11894
11895
11896
11897
11898
11899
11900
11901
11902
11903
11904
11905
11906
11907
11908
11909
11910
11911
11912
11913
11914
11915
11916
11917
11918
11919
11920
11921
11922
11923
11924
11925
11926
11927
11928
11929
11930
11931
11932
11933
11934
11935
11936
11937
11938
11939
11940
11941
11942
11943
11944
11945
11946
11947
11948
11949
11950
11951
11952
11953
11954
11955
11956
11957
11958
11959
11960
11961
11962
11963
11964
11965
11966
11967
11968
11969
11970
11971
11972
11973
11974
11975
11976
11977
11978
11979
11980
11981
11982
11983
11984
11985
11986
11987
11988
11989
11990
11991
11992
11993
11994
11995
11996
11997
11998
11999
12000
```

```

11258
11259
11260
11261
11262
11263
11264
11265
11266 041344 005212
11267 041346 022712 000406
11268 041352 001016
11269 041354 012737 041402 000004
11270 041362 012700 177700
11271 041366 012720 001234
11272 041372 012742 001147
11273 041376 005242
11274 041400 000000
11275 041402 022700 177702
11276 041406 001404
11277
11278
11279
11280
11281 041410 012742 001150
11282 041414 005242
11283 041416 000000
11284
11285
11286
11287
11288
11289
11290
11291
11292
11293
11294
11295
11296
11297 041420 005212
11298 041422 022712 000407
11299 041426 001062
11300 041430 012737 041506 000004
11301 041436 012737 000340 000006
11302 041444 012737 041476 000010
11303 041452 012737 000340 000012
11304 041460 012706 177700
11305 041464 000077
11306 041466 012742 001151
11307 041472 005242
11308 041474 000000
11309 041476
11310 041476 012742 001152
11311 041502 005242
11312 041504 000000
11313 041506 022737 041466 000000

```

```

:*****
:USING ADDRESS 177700 IN MODE 2, CAUSES BUS ERROR, BUT
:THE REGISTER IN USE WILL BE INCREMENTED.
:*****
:TEST 406 TEST THAT IN MODE 2, BAD ADDRESS REFERENCE CAUSES BUS ERROR.
:*****
TS406: INC (R2) ;UPDATE TEST NUMBER
CMP #406,(R2) ;SEQUENCE ERROR?
BNE TS407-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #RETR3,@#RTRAP5 ;SET TRAP RETURN ADDR
MOV #177700,R0 ;STORES BAD MEMORY REFERENCE
MOV #1234,(R0)+ ;BAD ADDR REFERENCE, TRAP TO LOC 4
MOV #1147,-(R2) ;MOVE TO MAILBOX # ***** 1147 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ADDRESSING 177700 DID NOT CAUSE TRAP
RETR3: CMP #177702,R0 ;WAS R0 INCREMENTED?
BEQ TS407
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
; CONDITIONAL BRANCH INST. AND <=
; REPLACE THE MOVE INSTRUCTION <-
; WHICH FOLLOWS W/ 761 <-
MOV #1150,-(R2) ;MOVE TO MAILBOX # ***** 1150 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;R0 WAS NOT INCREMENTED
; OR SEQUENCE ERROR
:*****
:
: AFTER THE FIRST BUS ERROR WAS ENCOUNTERED, AN ATTEMPT WAS MADE
: TO PUSH PC AND PS INTO THE STACK. HOWEVER, IF THE STACK POINTER
: WAS BAD, A DOUBLE BUS ERROR OCCURED. THE STACK POINTER WOULD
: THEN BE SET TO LOCATION 4, OLD PC AND PS WERE PUSHED INTO
: LOCATIONS 0 AND 2. THE PROCESSOR WOULD TRAP TO 4 AND CONTINUE
: EXECUTION.
:*****
:TEST 407 TEST FOR DOUBLE BUS ERROR.
:*****
TS407: INC (R2) ;UPDATE TEST NUMBER
CMP #407,(R2) ;SEQUENCE ERROR?
BNE TS410-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #DBE1,@#RTRAP5 ;SET TRAP RETURN ADDR
MOV #340,@#6 ;SET UP PS
MOV #DBE2,@#RTRAP ;SET TRAP RETURN ADDR
MOV #340,@#12 ;SET UP PS
MOV #177700,SP ;SET ILLEGAL SP
DBE: TRAPA ;ILLEGAL INSTRUCTION
MOV #1151,-(R2) ;MOVE TO MAILBOX # ***** 1151 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DOUBLE BUS ERROR DID NOT CAUSE TRAP
DBE2: MOV #1152,-(R2) ;MOVE TO MAILBOX # ***** 1152 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TRAP TO WRONG LOCATION
DBE1: CMP #DBE+2,@#0 ;OLD PC GOT SAVED?

```

```
11314 041514 001404          BEQ      DBE3          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11315                          ;          CONDITIONAL BRANCH INST. AND <====
11316                          ;          REPLACE THE MOVE INSTRUCTION <====
11317                          ;          WHICH FOLLOWS W/ 744 <====
11318                          ;
11319 041516 012742 001153    MOV      #1153,-(R2)    ;MOVE TO MAILBOX # ***** 1153 *****
11320 041522 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
11321 041524 000000          HALT                    ;OLD PC DID NOT GET SAVEDD
11322 041526 022737 000340 000002 DBE3:    CMP      #340,@#2     ;CORRECT PS SAVED?
11323 041534 001404          BEQ      DBE4          ;
11324                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11325                          ;          CONDITIONAL BRANCH INST. AND <====
11326                          ;          REPLACE THE MOVE INSTRUCTION <====
11327                          ;          WHICH FOLLOWS W/ 734 <====
11328 041536 012742 001154    MOV      #1154,-(R2)    ;MOVE TO MAILBOX # ***** 1154 *****
11329 041542 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
11330 041544 000000          HALT                    ;CORRECT PS DID NOT GET SAVE
11331 041546 022706 000000    DBE4:    CMP      #0,SP        ;SP POINTS TO LOC 0?
11332 041552 001404          BEQ      DBE5          ;
11333                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11334                          ;          CONDITIONAL BRANCH INST. AND <====
11335                          ;          REPLACE THE MOVE INSTRUCTION <====
11336                          ;          WHICH FOLLOWS W/ 725 <====
11337 041554 012742 001155    MOV      #1155,-(R2)    ;MOVE TO MAILBOX # ***** 1155 *****
11338 041560 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
11339 041562 000000          HALT                    ;SP IS NOT POINTING TO LOC 0
11340 041564 012737 027324 000004 DBE5:    MOV      #T04,@#RTRAP5 ;RESET TIMEOUT VECTOR
11341 041572 012737 027334 000010    MOV      #T010,@#RTRAP ;RESET ILLEGAL INST. VECTOR
11342 041600 012706 001000    MOV      #STBOT,SP     ;RESET SP
11343
11344
11345 ;*****
11346 ;TEST 410      TEST MFPT
11347 ;*****
11348 041604 005212          TS410: INC      (R2)         ;UPDATE TEST NUMBER
11349 041606 022712 000410    CMP      #410,(R2)     ;SEQUENCE ERROR?
11350 041612 001023          BNE     TS411-10      ;BR TO ERROR HALT ON SEQ ERROR
11351 ;THIS TESTS THE MFPT INSTRUCTION- MOVE FROM PROCESSOR TYPE
11352 ;UPON EXECUTION, R0 WILL RECEIVE THE PROCESSOR MODEL CODE
11353 ;WHICH IS '000003' FOR F11.
11354 MFPT=000007
11355 041614 012706 001000    MOV      #STBOT,R6     ;INIT. SP
11356 041620 013746 000010    MOV      @#10,-(SP)    ;SAVE TRAP VECTOR
11357 041624 012737 041654 000010    MOV      #1$,@#10     ;SET UP ILLEGAL INSTRUCTION TRAP
11358 041632 010046          MOV      R0,-(SP)     ;SAVE R0
11359 041634 000007          MFPT                    ;GET PROCESSOR MODEL
11360 041636 010037 041664    MOV      R0,@#CPUTYP   ;STORE IT
11361 041642 012600          MOV      (SP)+,R0     ;RESTORE R0
11362 041644 022737 000003 041664    CMP      #3,@#CPUTYP  ;CHECK MODEL TYPE
11363 041652 001405          BEQ      XXT          ;
11364                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
11365                          ;          CONDITIONAL BRANCH INST. AND <
11366                          ;          REPLACE THE MOVE INSTRUCTION <
11367                          ;          WHICH FOLLOWS W/ 757 <
11368 041654          1$:
11369 041654 012742 001156    MOV      #1156,-(R2)    ;MOVE TO MAILBOX # ***** 1156 *****
```

```
11370 041660 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
11371 041662 000000          HALT                    ;ILLEGAL INSTR TRAP OR WRONG MODEL TYPE
11372
11373 041664 000000          CPUTYP: .WORD 0
11374 041666                    XXT:
11375 041666 012637 000010      MOV      (SP)+,@#10    ;RESTORE TRAP VECTOR
11376
11377                          ;*****
11378                          ;THIS TEST WILL CHECK THE SERVICE ROUTINE FOR A CONTROL CHIP ERROR.
11379                          ;THIS IS DONE BY EXECUTING INSTRUCTIONS WHICH JUMP TO NON-EXISTENT
11380                          ;CONTROL-CHIP. THE TEST THEREFORE DOES A 'FIS' INSTRUCTION. THE
11381                          ;CTLERR TRAPS TO LOCATION 10.
11382                          ;THE RESET LINE IS ALSO ASSERTED FOR 1 CYCLE.
11383                          ;*****
11384                          ;TEST 411 TEST CTLERR SERVICE ROUTINE
11385                          ;*****
11385 041672 005212          TS411: INC      (R2)          ;UPDATE TEST NUMBER
11386 041674 022712 000411      CMP      #411,(R2)     ;SEQUENCE ERROR?
11387 041700 001013          BNE     TS412-10      ;BR TO ERROR HALT ON SEQ ERROR
11388 041702 012706 001000      MOV      #STBOT,R6    ;INIT STACK POINTER
11389 041706 012737 041734 000010  MOV      #1$,@#10     ;SET UP RETURN ADDR FROM TRAP
11390 041714 012737 000340 000012  MOV      #340,@#12    ;SET TRAP PRIORITY=7
11391 041722 075006          FADD     R6           ;EXECUTE FIS INSTR..SHOULD CAUSE CTLERR
11392 041724 012742 001157      MOV      #1157,-(R2)  ;MOVE TO MAILBOX # ***** 1157 *****
11393 041730 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
11394 041732 000000          HALT
11395 041734 012706 001000      1$: MOV      #STBOT,R6    ;RE-INIT STACK POINTER
11396
11397                          ;*****
11398                          ;TEST 412 TEST THAT ALL RESERVED INSTRUCTIONS TRAP
11399                          ;*****
11400 041740 005212          TS412: INC      (R2)          ;UPDATE TEST NUMBER
11401 041742 022712 000412      CMP      #412,(R2)     ;SEQUENCE ERROR?
11402 041746 001140          BNE     RET4          ;BR TO ERROR HALT ON SEQ ERROR
11403 041750 042767 000100 135606  BIC      #100,TPS
11404 041756 012737 042004 000244  MOV      #TRAP244,@#244 ; SET UP TO SEE IF
11405 041764 013767 000010 000044  MOV      @#10,TENSAVE  ; THIS PROCESSOR HAS THE
11406 041772 012737 042014 000010  MOV      #TRAP10,@#10 ; FLOATING POINT OPTION
11407 042000 170007          .WORD   170007        ; AN ILLEGAL FPP INSTRUCTION
11408 042002 000416          BR      AROUND        ; THE FOLLOWING
11409 042004          TRAP244:           ; IF FPP IN--
11410 042004 013767 042402 000400  MOV      @#FPP,FINISH ; RESET END OF TABLE POINTER
11411 042012 000002          RTI                    ; AND RETURN
11412 042014          TRAP10:           ; LEAVE THE TABLE ALONE
11413 042014 000002          RTI                    ; AND RETURN
11414
11415 042016 042777 000004 165430  ;* IF NO CIS OPTION TRAP TO HERE
11416 042024 012716 042072  CISTRP: BIC      #4,@SWR ;CLEAR CIS OPTION BIT IN SWR
11417 042030 000002          MOV      #CONCIS,(SP) ;CHANGE RETURN ADDRESS TO CONCIS LOCATION
11418          RTI                    ;RETURN
11419 042032 000000          CISADR: .WORD 0      ;DATA FOR CIS INSTRUCTION
11420 042034 000000          .WORD 0
11421
11422 042036 000000          TENSARE: .WORD 0      ; A PLACE TO STORE CONTENTS OF 10
11423
11424 042040          AROUND:           ; CONTINUATION POINT
11425 042040 012737 000246 000244  MOV      #246,@#244    ; RESTORE THE TRAP VECTOR
```

```

11426 042046 012737 042016 000010      MOV      #CISTRP,@#10      ;SET UP TO SEE IF
11427                                     ; THIS PROCESSOR HAS THE CIS OPTION
11428 042054 076144      .WORD    076144            ; EXECUTE A CMPCI INSTRUCTION
11429 042056 042032      .WORD    CISADR            ; OPERANDS
11430 042060 042032      .WORD    CISADR            ;
11431 042062 000000      .WORD    0                  ; FOR CIS
11432 042064 052777 000004 165362      BIS      #4,@SWR          ; INSTRUCTION
11433 042072                                     ; SET CIS PRESENT FLAG IN SWR
11434 042072 016737 177740 000010      MOV      TENSARE,@#10      ; RESTORE THE ILLEGAL INST. VECTOR
11435 042100 012703 042272      MOV      #TABLE,TAB       ; TABLE POINTER
11436 042104 012305      GIN1:   MOV      (TAB)+,FIRST ; FIRST OR CURRENT INSTRUCTION
11437 042106 012301      MOV      (TAB)+,LAST      ; LAST INSTRUCTION OR GROUP
11438 042110 020537 042346      CMP      FIRST,@#CIS      ; HAVE WE TESTED UP TO THE CIS INSTRUCTION SPACE
11439 042114 001007      BNE     1$                ; IF NO CONTINUE TESTING
11440 042116 032777 000004 165330      BIT      #4,@SWR          ; IF YES,CHECK IF CIS CHIPS PRESENT
11441 042124 001403      BEQ     1$                ; IF THEY ARE NOT CONTINUE TESTING
11442 042126 012703 042402      MOV      #FPP,TAB        ; IF CIS HERE SKIP CIS INSTRUCTION SPACE
11443 042132 000764      BR      GIN1
11444 042134 020567 000252      1$:    CMP      FIRST,FINISH ; TESTED ALL
11445 042140 001415      BEQ     GIN3              ; YES BRANCH
11446 042142 010567 000246      MOV      FIRST,INST      ; SET UP INST
11447 042146 005267 000242      GIN2:  INC      INST
11448 042152 012767 042204 135630      MOV      #RET,10         ; SET UP RETURN FROM TRAP
11449 042160 012706 001000      MOV      #BUFF,SP        ; SET UP STACK POINTER
11450 042164 005067 135606      CLR     CC                ; CLEAR PRIORITY
11451 042170 000167 000220      JMP     INST              ; EXECUTE RESERVED INSTRUCTION
11452 042174 012700 000370      GIN3:  MOV      #370,R0     ; RESET RESERVED AREA 370-402
11453 042200 000167 000320      JMP     THRPRT           ; JUMP TO EIS TEST
11454
11455      ;TRAPPING SHOULD SEND YOU HERE
11456 042204 020527 000774      RET:   CMP      SP,#BUFF-4 ; TEST DECREMENT OF SP
11457 042210 001404      BEQ     RET1
11458 042212 012742 001160      MOV     #1160,-(R2)       ; MOVE TO MAILBOX # ***** 1160 *****
11459 042216 005242      INC     -(R2)            ; SET MSGTYP TO FATAL ERROR
11460 042220 000000      HALT
11461 042222 026727 136546 042416      RET1:  CMP      BUFF-4,#INST+2 ; WRONG DECREMENT
11462 042230 001404      BEQ     RET2              ; LOC OF INST UNINCREMENTED
11463 042232 012742 001161      MOV     #1161,-(R2)       ; MOVE TO MAILBOX # ***** 1161 *****
11464 042236 005242      INC     -(R2)            ; SET MSGTYP TO FATAL ERROR
11465 042240 000000      HALT                      ; INST INC ON TRAP
11466 042242 005767 136530      RET2:  TST      BUFF-2
11467 042246 001404      BEQ     RET3
11468
11469      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
11470      ; CONDITIONAL BRANCH INST. AND < -
11471      ; REPLACE THE MOVE INSTRUCTION <-
11472      ; WHICH FOLLOWS W/ 637 < -
11472 042250      RET4:
11473 042250 012742 001162      MOV     #1162,-(R2)       ; MOVE TO MAILBOX # ***** 1162 *****
11474 042254 005242      INC     -(R2)            ; SET MSGTYP TO FATAL ERROR
11475 042256 000000      HALT                      ; CONDITION CODFS SET ON TRAP OR WRONG $TSTNM
11476 042260 026701 000130      RET3:  CMP      INST,LAST
11477 042264 001707      BEQ     GIN1              ; SET UP NEW GROUP
11478 042266 000167 177654      JMP     GIN2              ; FINISH OLD GROUP
11479      ; END OF INSTRUCT,ON GROUP
11480 042272 000007      TABLE: 7                ; END OF OPERATE
11481 042274 000077      77

```

11482	042276	000207	207		
11483	042300	000227	227		
11484	042302	006777	6777		
11485	042304	007777	7777		
11486	042306	075037	075037		
11487	042310	076017	76017		
11488	042312	076032	76032		
11489	042314	076037	76037		
11490	042316	076045	76045		
11491	042320	076047	76047		
11492	042322	076077	76077		
11493	042324	076127	76127		
11494	042326	076132	76132		
11495	042330	076137	76137		
11496	042332	076145	76145		
11497	042334	076147	76147		
11498	042336	076157	76157		
11499	042340	076167	76167		
11500	042342	076177	76177		
11501	042344	076777	76777		
11502	042346	076017	76017	CIS:	
11503	042350	076032	76032		
11504	042352	076037	76037		
11505	042354	076045	76045		
11506	042356	076047	76047		
11507	042360	076077	76077		
11508	042362	076127	76127		
11509	042364	076132	76132		
11510	042366	076137	76137		
11511	042370	076145	76145		
11512	042372	076147	76147		
11513	042374	076157	76157		
11514	042376	076167	76167		
11515	042400	076177	76177		
11516	042402	167777	167777	FPP:	: START OF THE FPP INSTRUCTIONS
11517	042404	177700	177700		
11518	042406	177716	177716		
11519	042410	177777	177777		
11520	042412	042412		FINISH:	:END FLAG
11521	042414	000000		INST:	:WILL CONTINUE RESERVED INST
11522	042416	000000	HALT		:SHOULD TRAP TO LOC 10
11523	042420	000000	HALT		:LOC 10 SHOULD SEND YOU TO
11524	042422	000000	HALT		:RET
11525	042424	000000	HALT		



11526  
11527  
11528  
11529  
11530  
11531  
11532  
11533  
11534  
11535  
11536  
11537  
11538  
11539  
11540  
11541  
11542  
11543  
11544  
11545  
11546  
11547  
11548  
11549  
11550  
11551  
11552  
11553  
11554  
11555  
11556  
11557  
11558  
11559  
11560  
11561  
11562  
11563

000000

.SBTTL \*\* STARTING OF EIS TEST \*\*  
.REPT 0

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 2000 IN  
LOCATION 320

.ENDR

```
11564
11565
11566      000000      DUMMY= 0
11567      000001      ERRNM= 1
11568      000051      F= 5
11569      000176      N= 176
11570      001000      SW09= 1000
11571      002000      SW10= 2000
11572      004000      SW11= 4000
11573      010000      SW12= 10000
11574
11575 042426      COUNT:
11576      042430      .=COUNT+2
11577 042430      PSWORD:
11578      042432      .=PSWORD+2
11579 042432      TEMP1:
11580      042434      .=TEMP1+2
11581 042434      TEMP2:
11582      042436      .=TEMP2+2
11583 042436      TEMP3:
11584      042440      .=TEMP3+2
11585 042440      TEMP4:
11586      042442      .=TEMP4+2
11587 042442 000000      TEMP5: .WORD
11588 042444 000000      TEMP6: .WORD
11589 042446      000      TYPCNT: .BYTE
11590 042447      000      $TPCNT: .BYTE
11591 042450 000007      S0: 7
11592 042452 177771      S1: -7
11593 042454 042452      S2: S1
11594 042456 177772      S3: -6
11595 042460 177777      S4: -1
11596 042462 040000      S5: 40000
11597 042464 042462      S6: S5
11598 042466 040000      S7: 40000
11599 042470 177776      S8: -2
11600 042472 000002      S9: 2
11601 042474 042472      S10: S9
11602 042476 000002      S11: 2
11603 042500 000964      TTYOUT: 64
11604 042502 177566      $TPB: 177566
11605 042504 177564      $TPS: 177564
11606 042506 005015 020040 000040      $CRLF: .ASCIIZ <15><12>/ /
11607 042514 006412 047520 042527      POWER: .ASCIIZ <12><15>/POWER/
11608 042522 000122
11609
11610
11611
11612
11613
11614
11615
11616
11617
11618
11619 042524 012705 000304      THRPR: MOV # $TESTN,R5 ;MAKE R5 POINT TO THE LOCATION $TESTN
```

11620	042530	005037	042426	CLR	@#COUNT	:CLEAR THE COUNTER
11621	042534	012715	000001	MOV	#1,(R5)	:INITIALIZE TEST NUMBER
11622	042540	012706	001000	MOV	#STBOT,SP	:** STACK AT STBOT **
11623	042544	012737	000001	MOV	#1,@#TEMP1	:TEMP1=1
11624	042552	005037	042434	CLR	@#TEMP2	:TEMP2=0
11625	042556	012737	000001	MOV	#1,@#TEMP3	:TEMP3=1
11626	042564	005037	042440	CLR	@#TEMP4	:TEMP4=0
11627	042570	012737	062470	MOV	#\$TRAP,@#34	:SET UP TRAP INSTRUCTION VECTOR
11628	042576	012737	000340	MOV	#340,@#36	:AND VECTOR PLUS TWO
11629	042604	106427	000000	MTPS	#0	:CLEAR PROCESSOR PRIORITY BITS IN PSW
11630						
11631						



11688	042774	001004		BNE	2\$		:IF NOT THEN GO TO 2\$
11689	042776	013702	042434	MOV	@#TEMP2,R2		:OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11690	043002	072102		ASH	R2,R1		:USING R1
11691	043004	000402		BR	4\$		
11692	043006	072167	177422	ASH	TEMP2,%1		:SHIFT R1 BY THE NUMBER SPECIFIED BY TEMP2
11693	043012	106737	042430	MFPS	@#PSWORD		:SAVE PS
11694	043016	123737	042440	CMPB	@#TEMP4,@#PSWORD		:IS THE PS = TEMP4 ?
11695	043024	001403		BEQ	+.10		
11696	043026	004767	015652	JSR	PC,\$HLT		:SEEN AN ERROR, GO TO THE HALT ROUTINE
11697							:THE PS IS NOT EQUAL TO 0
11698	043032	000003		3			:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11699							:BY (013746 000172 000207)
11700							
11701	043034	005237	042426	INC	@#COUNT		:INCREMENT THE COUNTER
11702	043040	023701	042436	CMP	@#TEMP3,%1		:IS THE RESULT IN R1 EQUAL TO TEMP3?
11703	043044	001403		BEQ	+.10		
11704	043046			6\$:			
11705	043046	004767	015632	JSR	PC,\$HLT		:SEEN AN ERROR, GO TO THE HALT ROUTINE
11706							:EITHER INCORRECT R1 OR INCORRECT SEQUENCE
11707	043052	000004		4			:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11708							:BY (013746 000172 000207)
11709							
11710	043054	021537	042426	CMP	(R5),@#COUNT		:IS THE TEST NUMBER EQUAL TO THE COUNTER?
11711	043060	001372		BNE	6\$		:IF NOT GO TO THE HLT ABOVE
11712	043062	005215		INC	(R5)		
11713	043064	010767	135102	MOV	PC,LPADR		:STORE ERROR LOOP ADDRESS
11714	043070	021527	000037	CMP	(R5),#37		:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
11715							:BY 14. AND RIGHT BY 14.?
11716	043074	002011		BGE	8\$		
11717	043076	005237	042434	INC	@#TEMP2		
11718	043102	006367	177330	ASL	TEMP3		:SHIFT TEMP3 LEFT
11719	043106	021527	000020	CMP	(R5),#20		:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11720	043112	001004		BNE	REGR2		
11721	043114	000167	000642	JMP	NEGAT		:IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11722	043120	004767	000664	JSR	PC,TST37		:IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11723	043124	010767	135042	MOV	PC,LPADR		:STORE ERROR LOOP ADDRESS
11724	043130	013702	042432	MOV	@#TEMP1,%2		:LOAD R2 WITH THE CONTENTS OF TEMP1
11725	043134	032737	000001	BIT	#1,@#SPASS		:IS IT AN EVEN PASS ?
11726	043142	001004		BNE	2\$		:IF NOT THEN GO TO 2\$
11727	043144	013703	042434	MOV	@#TEMP2,R3		:OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11728	043150	072203		ASH	R3,R2		:USING R2
11729	043152	000402		BR	4\$		
11730	043154	072267	177254	ASH	TEMP2,%2		:SHIFT R2 BY THE NUMBER SPECIFIED BY TEMP2
11731	043160	106737	042430	MFPS	@#PSWORD		:SAVE PS
11732	043164	123737	042440	CMPB	@#TEMP4,@#PSWORD		:IS THE PS = TEMP4 ?
11733	043172	001403		BEQ	+.10		
11734	043174	004767	015504	JSR	PC,\$HLT		:SEEN AN ERROR, GO TO THE HALT ROUTINE
11735							:THE PS IS NOT EQUAL TO 0
11736	043200	000005		5			:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11737							:BY (013746 000172 000207)
11738							
11739	043202	005237	042426	INC	@#COUNT		
11740	043206	023702	042436	CMP	@#TEMP3,%2		:IS THE RESULT IN R2 EQUAL TO TEMP3.
11741	043212	001403		BEQ	+.10		
11742	043214			6\$:			
11743	043214	004767	015464	JSR	PC,\$HLT		:SEEN AN ERROR, GO TO THE HALT ROUTINE

```

11744
11745 043220 000006
11746
11747
11748 043222 021537 042426
11749 043226 001372
11750 043230 005215
11751 043232 010767 134734
11752 043236 021527 000037
11753
11754 043242 002011
11755 043244 005237 042434
11756 043250 006367 177162
11757 043254 021527 000020
11758 043260 001004
11759 043262 000167 000474
11760 043266 004767 000516
11761 043272 010767 134674
11762 043276 013703 042732
11763 043302 032737 000001 000306
11764 043310 001004
11765 043312 013704 042434
11766 043316 072304
11767 043320 000402
11768 043322 072367 177106
11769 043326 106737 042430
11770 043332 123737 042440 042430
11771 043340 001403
11772 043342 004767 015336
11773
11774 043346 000007
11775
11776
11777 043350 005237 042426
11778 043354 023703 042436
11779 043360 001403
11780 043362
11781 043362 004767 015316
11782
11783 043366 000010
11784
11785
11786 043370 021537 042426
11787 043374 001372
11788 043376 005215
11789 043400 010767 134566
11790 043404 021527 000037
11791
11792 043410 002010
11793 043412 005237 042434
11794 043416 006367 177014
11795 043422 021527 000020
11796 043426 001003
11797 043430 000554
11798 043432 004767 000352
11799 043436 010767 134530

;EITHER INCORRECT R2 OR INCORRECT SEQUENCE
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

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 ;SHIFTED TEMP3 LEFT
CMP (R5),#20 ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
BNE REGR3
JMP NEGAT ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
JSR PC,TST37 ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV @#TEMP1,%3 ;LOAD R3 WITH THE CONTENTS OF TEMP1
BIT #1,@#SPASS ;IS IT AN EVEN PASS?
BNE 2$ ;IF NOT THEN GO TO 2$
MOV @#TEMP2,R4 ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
ASH R4,R3 ;USING R3
BR 4$
ASH TEMP2,%3 ;SHIFT R3 BY THE NUMBER SPECIFIED BY TEMP2
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)

INC @#COUNT
CMP @#TEMP3,%3 ;IS THE RESULT IN R3 EQUAL TO TEMP3?
BEQ .+10

JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;EITHER INCORRECT R3 OR INCORRECT SEQUENCE
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

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 REGR4
BR NEGAT ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
JSR PC,TST37 ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
MOV PC,LPADR ;STORE ERROR LOOP ADDRESS

```

11800	043442	013704	042432		MOV	@#TEMP1,%4	;LOAD R4 WITH THE CONTENTS OF TEMP1
11801	043446	010501			MOV	R5,R1	;SAVE R5
11802	043450	032737	000001	000306	BIT	#1,@#SPASS	;IS IT AN EVEN PASS ?
11803	043456	001004			BNE	2\$	;IF NOT THEN GO TO 2\$
11804	043460	013705	042434		MOV	@#TEMP2,R5	;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11805	043464	072405			ASH	R5,R4	;USING R4
11806	043466	000402			BR	4\$	
11807	043470	072467	176740		2\$: ASH	TEMP2,%4	;SHIFT R4 BY THE NUMBER SPECIFIED BY TEMP2
11808	043474	106737	042430		4\$: MFPS	@#PSWORD	;SAVE PS
11809	043500	123737	042440	042430	CMPB	@#TEMP4,@#PSWORD	;IS PS = TEMP4 ?
11810	043506	001403			BEQ	.+10	
11811	043510	004767	015170		JSR	PC,\$HLT	;SEEN AN ERROR, GO TO THE HALT ROUTINE
11812							;THE PS IS NOT EQUAL TO 0
11813	043514	000011			11		;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11814							;BY (013746 000172 000207)
11815							
11816	043516	005237	042426		INC	@#COUNT	
11817	043522	023704	042436		CMP	@#TEMP3,%4	;IS THE RESULT IN R4 EQUAL TO TEMP3?
11818	043526	001403			BEQ	.+10	
11819	043530				6\$: JSR	PC,\$HLT	;SEEN AN ERROR, GO TO THE HALT ROUTINE
11820	043530	004767	015150				;EITHER INCORRECT R4 OR INCORRECT SEQUENCE
11821							;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11822	043534	000012			12		;BY (013746 000172 000207)
11823							
11824							
11825	043536	010105			MOV	R1,R5	;RESTORE R5
11826	043540	021537	042426		CMP	(R5),@#COUNT	;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11827	043544	001371			BNE	6\$	;IF NOT GO TO THE HLT ABOVE
11828	043546	005215			INC	(R5)	
11829	043550	010767	134416		MOV	PC,LPADR	;STORE ERROR LOOP ADDRESS
11830	043554	021527	000037		CMP	(R5),#37	;HAS THE CONTENTS OF REGISTERS BEEN
11831							;SHIFTED LEFT BY 14. AND RIGHT BY 14.?
11832	043560	002010			BGE	8\$	
11833	043562	005237	042434		INC	@#TEMP2	
11834	043566	006367	176644		ASL	TEMP3	;SHIFT TEMP3 LEFT
11835	043572	021527	000020		CMP	(R5),#20	;HAS THE CONTENTS OF REGISTER BEEN SHIFTED BY 14.?
11836	043576	001003			BNE	REGR5	
11837	043600	000470			BR	NEGAT	;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11838	043602	004767	000202		8\$: JSR	PC,TST37	;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11839	043606	010767	134360		REGR5: MOV	PC,LPADR	;STORE ERROR LOOP ADDRESS
11840	043612	010501			MOV	R5,R1	;SAVE R5
11841	043614	013705	042432		MOV	@#TEMP1,%5	;LOAD R5 WITH THE CONTENTS OF TEMP1
11842	043620	032737	000001	000306	BIT	#1,@#SPASS	;IS IT AN EVEN PASS ?
11843	043626	001004			BNE	2\$	;IF NOT THEN GO TO 2\$
11844	043630	013700	042434		MOV	@#TEMP2,R0	;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11845	043634	072500			ASH	R0,R5	;USING R5
11846	043636	000402			BR	4\$	
11847	043640	072567	176570		2\$: ASH	TEMP2,%5	;SHIFT R5 BY THE NUMBER SPECIFIED BY TEMP2
11848	043644	106737	042430		4\$: MFPS	@#PSWORD	;SAVE PS
11849	043650	123737	042440	042430	CMPB	@#TEMP4,@#PSWORD	;IS PS = TEMP4 ?

11850	043656	001403		BEQ	+.10	
11851	043660	004767	015020	JSR	PC,\$HLT;	SEEN AN ERROR, GO TO THE HALT ROUTINE
11852						:THE PS IS NOT EQUAL TO 0.
11853	043664	000013		13		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11854						:BY (013746 000172 000207)
11855						
11856	043666	005237	042426	INC	@#COUNT	
11857	043672	023705	042436	CMP	@#TEMP3,%5	:IS THE RESULT IN R5 EQUAL TO TEMP3?
11858	043676	001403		BEQ	+.10	
11859	043700			6\$:		
11860	043700	004767	015000	JSR	PC,\$HLT;	SEEN AN ERROR, GO TO THE HALT ROUTINE
11861						:EITHER INCORRECT R5 OR INCORRECT SEQUENCE
11862	043704	000014		14		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11863						:BY (013746 000172 000207)
11864						
11865	043706	021137	042426	CMP	(R1),@#COUNT	:IS THE TEST NUMBER EQUAL TO THE COUNTER?
11866	043712	001372		BNE	6\$	:IF NOT GO TO THE HLT ABOVE
11867	043714	010105		MOV	R1,R5	:RESTORE R5
11868	043716	005215		INC	(R5)	
11869	043720	010767	134246	MOV	PC,LPADR	:STORE ERROR LOOP ADDRESS
11870	043724	021527	000037	CMP	(R5),#37	:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED



```

11871                                     ;LEFT BY 14. AND RIGHT BY 14.?
11872 043730 002010                                     ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11873 043732 005237 042434      BGE      8$
11874 043736 006367 176474      INC      @#TEMP2
11875 043742 021527 000020      ASL      TEMP3      ;SHIFT TEMP3 LEFT
11876 043746 001405      CMP      (R5),#20    ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11877 043750 000402      BEQ      NEGAT      ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11878 043752 004767 000032      BR       10$
11879 043756 000167 176626      8$:     JSR      PC,TST37
11880 043762 012737 040000 042432 10$:     JMP      ASTART      ;GO BACK TO START
11881 043770 012737 177762 042434 NEGAT:  MOV      #40000,@#TEMP1    ;TEMP1=40000
11882 043776 012737 000001 042436      MOV      #177762,@#TEMP2    ;TEMP2=177762
11883 044004 000167 176600      MOV      #1,@#TEMP3        ;TEMP3=1
11884 044010 021527 000037      JMP      ASTART
11885 044014 001013      TST37:  CMP      (R5),#37    ;IS IT TEST 37?
11886 044016 005037 042432      BNE      TST40          ;IF NOT THEN TRY TEST 40
11887 044022 012737 000020 042434      CLR      @#TEMP1        ;0
11888 044030 005037 042436      MOV      #16,@#TEMP2     ;SHIFTED BY 16
11889 044034 012737 000004 042440      CLR      @#TEMP3        ;IS=0
11890 044042 000207      MOV      #4,@#TEMP4     ;AND PS=4
11891 044044 021527 000040      RTS      PC
11892 044050 001003      TST40:  CMP      (R5),#40    ;IS IT TEST 40?
11893 044052 005037 042434      BNE      TST41          ;IF NOT THEN TRY TEST 41
11894 044056 000207      CLR      @#TEMP2        ;0 SHIFTED BY 0=0 AND PS=4
11895 044060 021527 000041      RTS      PC
11896 044064 001004      TST41:  CMP      (R5),#41    ;IS IT TEST 41?
11897 044066 012737 177760 042434      BNE      TST42          ;IF NOT THEN TRY TEST 42
11898 044074 000207      MOV      #-16,@#TEMP2   ;0 SHIFTED BY -16.=0 AND PS=4
11899 044076 021527 000042      RTS      PC
11900 044102 001013      TST42:  CMP      (R5),#42    ;IS IT TEST 42?
11901 044104 012737 100000 042432      BNE      TST43          ;IF NOT THEN TRY TEST 43
11902 044112 005237 042434      MOV      #100000,@#TEMP1 ;100000
11903 044116 005337 042436      INC      @#TEMP2        ;SHIFTED BY -15
11904 044122 012737 000010 042440      DEC      @#TEMP3        ;IS=-1
11905 044130 000207      MOV      #10,@#TEMP4   ;AND PS=10
11906 044132 021527 000043      RTS      PC
11907 044136 001012      TST43:  CMP      (R5),#43    ;IS IT TEST 43?
11908 044140 012737 125252 042432      BNE      TST44          ;IF NOT THEN IF NOT THEN TRY TEST 44
11909 044146 012737 177777 042434      MOV      #125252,@#TEMP1 ;125252
11910 044154 012737 152525 042436      MOV      #-1,@#TEMP2    ;SHIFTED BY -1
11911 044162 000207      MOV      #152525,@#TEMP3 ;IS=152525 AND PS=10
11912 044164 021527 000044      RTS      PC
11913 044170 001012      TST44:  CMP      (R5),#44    ;IS IT TEST 44?
11914 044172 012737 000001 042434      BNE      TST45          ;IF NOT THEN TRY TEST 45
11915 044200 012737 052524 042436      MOV      #1,@#TEMP2     ;125252 SHIFTED BY 1
11916 044206 012737 000003 042440      MOV      #52524,@#TEMP3 ;IS=52524
11917 044214 000207      MOV      #3,@#TEMP4     ;AND PS=3
11918 044216 021527 000045      RTS      PC
11919 044222 001012      TST45:  CMP      (R5),#45    ;IS IT TEST 45?
11920 044224 012737 177776 042434      BNE      TST46          ;IF NOT THEN TRY TEST 46
11921 044232 012737 165252 042436      MOV      #-2,@#TEMP2    ;125252 SHIFTED BY -2
11922 044240 012737 000011 042440      MOV      #165252,@#TEMP3 ;IS=165252
11923 044246 000207      MOV      #11,@#TEMP4   ;AND PS=11
11924 044250 021527 000046      RTS      PC
11925 044254 001014      TST46:  CMP      (R5),#46    ;IS IT TEST 46?
11926 044256 012737 177777 042432      BNE      TST47          ;IF NOT THEN TRY TEST 47
11926 044256 012737 177777 042432      MOV      #-1,@#TEMP1   ;-1

```

```

11927 044264 012737 000020 042434      MOV      #16,@#TEMP2      ;SHIFTED BY 15.
11928 044272 005037 042436      CLR      @#TEMP3          ;IS=0
11929 044276 012737 000007 042440      MOV      #7,@#TEMP4       ;AND PS=7
11930 044304 000207                RTS      PC
11931 044306 021527 000047      TST47:  CMP      (R5),#47   ;IS IT TEST 47?
11932 044312 001011                BNE     TST50             ;IF NOT THEN TRY TEST 50
11933 044314 005337 042434      DEC      @#TEMP2          ;-1 SHIFTED BY 15
11934 044320 012737 100000 042436      MOV      #100000,@#TEMP3  ;IS=100000
11935 044326 012737 000011 042440      MOV      #11,@#TEMP4      ;AND PS=11
11936 044334 000207                RTS      PC
11937 044336 021527 000050      TST50:  CMP      (R5),#50   ;IS IT TEST 50
11938 044342 001007                BNE     ENT51            ;IF NOT THEN TRY TEST 51
11939 044344 012737 137777 042432      MOV      #137777,@#TEMP1  ;137777 SHIFTED BY 15, IS=100000
11940 044352 012737 000013 042440      MOV      #13,@#TEMP4      ;AND PS=13
11941 044360 000207                RTS      PC
11942 044362 021527 000051      ENT51:  CMP      (R5),#51   ;IS IT ENTERING TEST 51?
11943 044366 001403                BEQ     .+10
11944 044370 004767 044310      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11945                                ;TEST NUMBER GOOFED
11946 044374 000015                15          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11947                                ;BY (013746 000172 000207)
11948
11949
11950 044376 005726                TST      (SP)+           ;RESTORE STACK POINTER
11951 044400 012704 177771      MOV      #-7,%4
11952 044404 012702 042452      MOV      #S1,%2
11953 044410 012703 042454      MOV      #S2,%3
11954                                ;*****
11955                                ;TEST:51      ASH      125252 SHIFTED BY #5 = 52500 PS = 3
11956                                ;*****
11957
11958 044414 010767 133552      TST51:  MOV      PC,LPADR   ;STORE ERROR LOOP ADDRESS
11959 044420 012701 125252      MOV      #125252,%1      ;LOAD R1 WITH 125252
11960 044424 072127 000005      ASH     #5,%1            ;SHIFT R1 BY #5
11961 044430 106737 042430      MFPS   @#PSWORD         ;SAVE PS
11962 044434 122737 000003 042430      CMPB   #3,@#PSWORD      ;IS THE PS 3?
11963 044442 001403                BEQ     .+10
11964 044444 004767 014234      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11965                                ;THE PS IS NOT EQUAL TO 3
11966 044450 000016                16          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11967                                ;BY (013746 000172 000207)
11968
11969 044452 022701 052500                CMP      #52500,%1       ;IS THE RESULT 52500?
11970 044456 001403                BEQ     .+10
11971 044460                1$:
11972 044460 004767 014220      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11973                                ;R1 IS NOT EQUAL TO 52500 OR INCORRECT SEQUENCE
11974 044464 000017                17          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11975                                ;BY (013746 000172 000207)
11976
11977 044466 021527 000051      CMP      (R5),#51        ;IS $TESTN = #51
11978 044472 001372                BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
11979 044474 005215                INC      (R5)
11980
11981
11982

```

```
11983
11984
11985
11986
11987 044476 010767 133470
11988 044502 012700 125252
11989 044506 072077 175742
11990 044512 106737 042430
11991 044516 122737 000010 042430
11992 044524 001403
11993 044526 004767 014152
11994
11995 044532 000020
11996
11997
11998 044534 022700 177525
11999 044540 001403
12000 044542
12001 044542 004767 014136
12002
12003 044546 000021
12004
12005
12006 044550 021527 000052
12007 044554 001372
12008 044556 005215
12009
12010
12011
12012
12013
12014
12015
12016 044560 010767 133406
12017 044564 012700 125252
12018 044570 072037 042452
12019 044574 106737 042430
12020 044600 122737 000010 042430
12021 044606 001403
12022 044610 004767 014070
12023
12024 044614 000022
12025
12026
12027 044616 022700 177525
12028 044622 001403
12029 044624
12030 044624 004767 014054
12031
12032 044630 000023
12033
12034
12035 044632 021527 000053
12036 044636 001372
12037 044640 005215
12038
```

```
*****
:TEST:52      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,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;THE PS IS NOT EQUAL TO 10
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)
        20
        1$:
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)
        21
        CMP     (R5),#52        ;IS $TESTN = #52
        BNE     1$             ;IF NOT THEN GO TO HLT ABOVE
        INC     (R5)
*****
:TEST:53      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,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;THE PS IS NOT EQUAL TO 10
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)
        22
        1$:
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)
        23
        CMP     (R5),#53        ;IS $TESTN = #53
        BNE     1$             ;IF NOT THEN GO TO HLT ABOVE
        INC     (R5)
```

12039  
12040  
12041  
12042  
12043  
12044  
12045  
12046  
12047  
12048  
12049  
12050  
12051  
12052  
12053  
12054  
12055  
12056  
12057  
12058  
12059  
12060  
12061  
12062  
12063  
12064  
12065  
12066  
12067  
12068  
12069  
12070  
12071  
12072  
12073  
12074  
12075  
12076  
12077  
12078  
12079  
12080  
12081  
12082  
12083  
12084  
12085  
12086  
12087  
12088  
12089  
12090  
12091  
12092  
12093  
12094

```
.....  
:TEST:54      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)
```

```
.....  
:TEST:55      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
```

12095 045000 005215  
12096  
12097  
12098  
12099  
12100  
12101  
12102  
12103 045002 010767 133164  
12104 045006 012700 125252  
12105 045012 072042  
12106 045014 106737 042430  
12107 045020 122737 000010 042430  
12108 045026 001403  
12109 045030 004767 013650  
12110  
12111 045034 000030  
12112  
12113  
12114 045036 022700 177525  
12115 045042 001403  
12116 045044  
12117 045044 004767 013634  
12118  
12119 045050 000031  
12120  
12121  
12122 045052 021527 000056  
12123 045056 001372  
12124 045060 005215  
12125  
12126  
12127  
12128  
12129  
12130  
12131  
12132 045062 010767 133104  
12133 045066 012700 125252  
12134 045072 072063 000002  
12135 045076 106737 042430  
12136 045102 122737 000011 042430  
12137 045110 001403  
12138 045112 004767 013566  
12139  
12140 045116 000032  
12141  
12142  
12143 045120 022700 177252  
12144 045124 001403  
12145 045126  
12146 045126 004767 013552  
12147  
12148 045132 000033  
12149  
12150

\*\*\*\*\*  
:TEST:56 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)

\*\*\*\*\*  
:TEST:57 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)

12151 045134 021527 000057  
12152 045140 001372  
12153 045142 005215

CMP (R5),#57 ;IS \$TESTN = #57  
BNE 1\$ ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)

\*\*\*\*\*  
:TEST:60 ASH 125252 SHIFTED BY @ (3) = 177525 PS - 10  
\*\*\*\*\*

12154  
12155  
12156  
12157  
12158  
12159  
12160

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  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

12161 045144 010767 133022  
12162 045150 012700 125252  
12163 045154 072073 000000  
12164 045160 106737 042430  
12165 045164 122737 000010 042430  
12166 045172 001403  
12167 045174 004767 013504

12168  
12169 045200 000034  
12170  
12171

1\$: 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 OR INCORRECT SEQUENCE  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

12172 045202 022700 177525  
12173 045206 001403  
12174 045210  
12175 045210 004767 013470  
12176  
12177 045214 000035  
12178  
12179

12180 045216 021527 000060  
12181 045222 001372  
12182 045224 005215

CMP (R5),#60 ;IS \$TESTN - #60  
BNE 1\$ ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)

\*\*\*\*\*  
:TEST:61 ASH 125252 SHIFTED BY @ (3)+ 177525 PS 10  
\*\*\*\*\*

12183  
12184  
12185  
12186  
12187  
12188  
12189

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  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

12190 045226 010767 132740  
12191 045232 012700 125252  
12192 045236 072033  
12193 045240 106737 042430  
12194 045244 122737 000010 042430  
12195 045252 001403  
12196 045254 004767 013424

12197  
12198 045260 000036  
12199

1\$: 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 OR INCORRECT SEQUENCE  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)

12200  
12201 045262 022700 177525  
12202 045266 001403  
12203 045270  
12204 045270 004767 013410  
12205  
12206 045274 000037

12207  
12208  
12209 045276 021527 000061  
12210 045302 001372  
12211 045304 005215  
12212  
12213  
12214  
12215  
12216  
12217  
12218  
12219 045306 010767 132660  
12220 045312 012700 125252  
12221 045316 072053  
12222 045320 106737 042430  
12223 045324 122737 000010 042430  
12224 045332 001403  
12225 045334 004767 013344  
12226  
12227 045340 000040  
12228  
12229  
12230 045342 022700 177525  
12231 045346 001403  
12232 045350  
12233 045350 004767 013330  
12234  
12235 045354 000041  
12236  
12237  
12238 045356 021527 000062  
12239 045362 001372  
12240 045364 005215  
12241  
12242  
12243

:BY (013746 000172 000207)

CMP (R5),#61 ;IS \$TESTN - #61  
BNE 1\$ ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)

\*\*\*\*\*  
:TEST:62 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)  
  
1\$: 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 OR INCORRECT SEQUENCE  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#62 ;IS \$TESTN #62  
BNE 1\$ ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)

```

12244
12245
12246
12247
12248
12249
12250
12251
12252
12253
12254
12255
12256 045366 010767 132600          MOV    PC,LPADR          ;STORE ERROR LOOP ADDRESS
12257 045372 012737 000062 042426    MOV    #62,@#COUNT
12258 045400 005037 042432          CLR    @#TEMP1          ;TEMP1=0
12259 045404 012737 000001 042434    MOV    #1,@#TEMP2      ;TEMP2=1
12260 045412 005037 042436          CLR    @#TEMP3          ;TEMP3=0
12261 045416 005037 042440          CLR    @#TEMP4          ;TEMP4=0
12262 045422 012737 000001 042442    MOV    #1,@#TEMP5      ;TEMP5=1
12263 045430 005037 042444          CLR    @#TEMP6          ;0 1 SHIFTED BY 0=0 1, PS=0
12264
12265 045434 010502          REG01: MOV    R5,R2          ;SAVE R5
12266 045436 013700 042432          MOV    @#TEMP1,%0      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 0
12267 045442 013701 042434          MOV    @#TEMP2,%0.1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 1
12268 045446 000241          CLC
12269 045450 032737 000001 000306    BIT    #1,@#SPASS      ;IS IT AN EVEN PASS ?
12270 045456 001004          BNE    2$              ;IF NOT THEN GO TO 2$
12271 045460 013705 042436          MOV    @#TEMP3,R5      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12272 045464 073005          ASHC   R5,R0           ;USING R0
12273 045466 000402          BR     4$
12274 045470 073067 174742          2$:  ASHC   TEMP3,%0      ;ASHC REGISTER 0 BY THE CONTENTS OF TEMP3
12275 045474 106737 042430          4$:  MFPS   @#PSWORD      ;SAVE PS
12276 045500 123737 042444 042430    CMPSB @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
12277 045506 001403          BEQ    .+10
12278 045510 004767 013170          JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12279
12280 045514 000042          42    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12281
12282
12283 045516 005237 042426          INC    @#COUNT
12284 045522 023700 042440          CMP    @#TEMP4,%0      ;IS THE RESULT IN R0 SAME AS TEMP4?
12285 045526 001403          BEQ    .+10
12286 045530 004767 013150          JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12287
12288 045534 000043          43    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12289
12290
12291 045536 023701 042442          CMP    @#TEMP5,%1      ;IS THE RESULT IN R1 SAME AS TEMP5?
12292 045542 001403          BEQ    .+10            ;TEMP1 TEMP2 SHIFTED BY TEMP3=TEMP4 TEMP5
12293
12294 045544 004767 013134          JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12295
12296 045550 000044          44    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12297
12298
12299 045552 010205          MOV    R2,R5           ;RESTORE R5

```

\*\*\*\*\*  
: ASHC INSTRUCTION TESTS  
:\*\*\*\*\*

\*\*\*\*\*  
: TESTS 63-157  
:\*\*\*\*\*



12300	045554	021537	042426			CMP	(R5),@#COUNT	:IS TEST NUMBER=COUNTER?
12301	045560	001403				BEQ	+.10	
12302	045562	004767	013116			JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
12303								:NO
12304	045566	000045				45		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12305								:BY (013746 000172 000207)
12306								
12307	045570	005215				INC	(R5)	
12308	045572	021527	000160			CMP	(R5),#160	:HAVE THE FIRST 159 TEST BEEN EXECUTED?
12309	045576	002014				BGE	6\$	:YES
12310	045600	005237	042436			INC	@#TEMP3	
12311	045604	000241				CLC		
12312	045606	006137	042442			ROL	@#TEMP5	:ROTATE TEMP5 LEFT BY 1 PLACE
12313	045612	006137	042440			ROL	@#TEMP4	:INTRODUCE CARRY FROM TEMP4 IN TEMP5
12314	045616	021527	000121			CMP	(R5),#121	:IS IT TEST 121?
12315	045622	001004				BNE	REGR23	
12316	045624	004467	000414			JSR	R4,RITSH	:IF SO THEN GO AND INITIATE RIGHT SHIFT
12317	045630	004767	000444			JSR	%7,TST160	
12318	045634	010767	132332		6\$: REGR23:	MOV	PC,LPADR	:STORE ERROR LOOP ADDRESS
12319	045640	013702	042432			MOV	@#TEMP1,%2	:PLACE THE CONTENTS OF TEMP1 IN REGISTER 2
12320	045644	013703	042434			MOV	@#TEMP2,%2!1	:PLACE THE CONTENTS OF TEMP2 IN REGISTER 3
12321	045650	000241				CLC		
12322	045652	032737	000001	000306		BIT	#1,@#\$PASS	:IS IT AN EVEN PASS ?
12323	045660	001004				BNE	2\$	:IF NOT THEN GO TO 2\$
12324	045662	013704	042436			MOV	@#TEMP3,R4	:OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12325	045666	073204				ASHC	R4,R2	:USING R2
12326	045670	000402				BR	4\$	
12327	045672	073267	174540		2\$:	ASHC	TEMP3,%2	:ASHC REGISTER 2 BY THE CONTENTS OF TEMP3
12328	045676	106737	042430		4\$:	MFPS	@#PSWORD	:SAVE PS
12329	045702	123737	042444	042430		CMPB	@#TEMP6,@#PSWORD	:COMPARE PS WITH THE CONTENTS OF TEMP6
12330	045710	001403				BEQ	+.10	
12331	045712	004767	012766			JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
12332								:WRONG PS
12333	045716	000046				46		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12334								:BY (013746 000172 000207)
12335								
12336	045720	005237	042426			INC	@#COUNT	
12337	045724	023702	042440			CMP	@#TEMP4,%2	:IS THE RESULT IN R2 SAME AS TEMP4?
12338	045730	001403				BEQ	+.10	
12339	045732	004767	012746			JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
12340								:WRONG RESULT IN R2
12341	045736	000047				47		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12342								:BY (013746 000172 000207)
12343								
12344	045740	023703	042442			CMP	@#TEMP5,%3	:IS THE RESULT IN R3 SAME AS TEMP5?
12345	045744	001403				BEQ	+.10	:TEMP1 TEMP2 SHIFTED BY TEMP3 TEMP4 TEMP5
12346								:AND PS=TEMP6
12347	045746	004767	012732			JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
12348								:WRONG RESULT IN R1
12349	045752	000050				50		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12350								:BY (013746 000172 000207)
12351								
12352	045754	021537	042426			CMP	(R5),@#COUNT	:IS TEST NUMBER=COUNTER?
12353	045760	001403				BEQ	+.10	
12354	045762	004767	012716			JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
12355								:NO



12412	046172	010105			MOV	R1,R5		:RESTORE R5
12413	046174	005215			INC	(R5)		
12414	046176	021527	000160		CMP	(R5),#160		:HAVE THE FIRST 159 TEST BEEN EXECUTED?
12415	046202	002014			BGE	6\$		:YES
12416	046204	005237	042436		INC	@#TEMP3		
12417	046210	000241			CLC			
12418	046212	006137	042442		ROL	@#TEMP5		:ROTATE TEMP5 LEFT BY 1 PLACE
12419	046216	006137	042440		ROL	@#TEMP4		:INTRODUCE CARRY FROM TEMP5 IN TEMP4
12420	046222	021527	000121		CMP	(R5),#121		:IS IT TEST 121?
12421	046226	001004			BNE	8\$		
12422	046230	004467	000010		JSR	R4,RITSH		:IF SO THEN GO AND INITIATE RIGHT SHIFT
12423	046234	004767	000040		6\$: JSR	%7,TST160		
12424	046240	000167	177170		8\$: JMP	REG01		
12425	046244	022424			RITSH: CMP	(R4)+,(R4)+		:MAKE R4 POINT TO THE NEXT REG TAG
12426	046246	012737	040000	042432	MOV	#40000,@#TEMP1		:TEMP1=4000
12427	046254	005037	042434		CLR	@#TEMP2		:TEMP2=0
12428	046260	012737	177742	042436	MOV	#-30,@#TEMP3		:TEMP3=-30
12429	046266	005037	042440		CLR	@#TEMP4		:TEMP4=0
12430	046272	005237	042442		INC	@#TEMP5		:TEMP5-1
12431	046276	000204			RTS	R4		
12432	046300	021527	000160		TST160: CMP	(R5),#160		:IS IT TEST 160
12433	046304	001010			BNE	TST161		:IF NOT THEN TRY TEST 161
12434	046306	005037	042432		CLR	@#TEMP1		:0 0 SHIFTED BY 0
12435	046312	005037	042440		CLR	@#TEMP4		:IS EQUAL TO 0 0
12436	046316	012737	000004	042444	MOV	#4,@#TEMP6		:AND PS=4
12437	046324	000207			RTS	%7		
12438	046326	021527	000161		TST161: CMP	(R5),#161		:IS IT TEST 161
12439	046332	001004			BNE	TST162		
12440	046334	012737	177746	042436	MOV	#-32,@#TEMP3		:0 0 SHIFTED BY -32=0 0, PS=4
12441	046342	000207			RTS	%7		
12442	046344	021527	000162		TST162: CMP	(R5),#162		:IS IT TEST 162
12443	046350	001004			BNE	TST163		:IF NOT THEN TRY TEST 163
12444	046352	012737	000032	042436	MOV	#32,@#TEMP3		:0 0 SHIFTED BY 32=0 0, PS-4
12445	046360	000207			RTS	%7		
12446	046362	021527	000163		TST163: CMP	(R5),#163		:IS IT TEST 163?
12447	046366	001016			BNE	TST164		:IF NOT THEN TRY TEST 164
12448	046370	012737	052525	042432	MOV	#52525,@#TEMP1		:52525 0
12449	046376	012737	177760	042436	MOV	#-16,@#TEMP3		:SHIFTED BY -16.
12450	046404	005037	042440		CLR	@#TEMP4		
12451	046410	012737	052525	042442	MOV	#52525,@#TEMP5		:IS EQUAL TO 0 52525
12452	046416	005037	042444		CLR	@#TEMP6		:AND PS = 0
12453	046422	000207			RTS	%7		
12454	046424	021527	000164		TST164: CMP	(R5),#164		:IS IT TEST 164?
12455	046430	001014			BNE	TST165		:IF NOT THEN TRY TEST 165
12456	046432	012737	125252	042432	MOV	#125252,@#TEMP1		:125252 0 SHIFTED BY -16.
12457	046440	005337	042440		DEC	@#TEMP4		
12458	046444	012737	125252	042442	MOV	#125252,@#TEMP5		:IS EQUAL TO -1 125252
12459	046452	012737	000010	042444	MOV	#10,@#TEMP6		:AND PS=10
12460	046460	000207			RTS	%7		
12461	046462	021527	000165		TST165: CMP	(R5),#165		:IS IT TEST 165?
12462	046466	001007			BNE	TST166		:IF NOT THEN TRY TEST 166
12463	046470	012737	177777	042432	MOV	#-1,@#TEMP1		:-1 0 SHIFTED BY -16
12464	046476	012737	177777	042442	MOV	#-1,@#TEMP5		:IS EQUAL TO -1 -1, AND PS-10
12465	046504	000207			RTS	%7		
12466	046506	021527	000166		TST166: CMP	(R5),#166		:IS IT TEST 166?
12467	046512	001011			BNE	TST167		:IF NOT THEN TRY TEST 167

```

12468 046514 012737 100000 042432 MOV #100000,@#TEMP1 :100000 0
12469 046522 012737 177740 042436 MOV #-32,@#TEMP3 :SHIFTED BY -32 IS EQUAL TO -1 -1
12470 046530 005237 042444 INC @#TEMP6 :AND PS=11
12471 046534 000207 RTS %7
12472 046536 021527 000167 TST167: CMP (R5),#167 :IS IT TEST 167?
12473 046542 001014 BNE TST170 :IF NOT THEN TRY TEST 170
12474 046544 005037 042432 CLR @#TEMP1
12475 046550 J05337 042434 DEC @#TEMP2 :0 -1
12476 046554 012737 000020 042436 MOV #16,@#TEMP3 :SHIFTED BY 16.
12477 046562 005037 042442 CLR @#TEMP5 :IS EQUAL TO -1 0
12478 046566 005237 042444 INC @#TEMP6 :AND PS=12
12479 046572 000207 RTS %7
12480 046574 021527 000170 TST170: CMP (R5),#170 :IS IT TEST 170?
12481 046600 001007 BNE TST171 :IF NOT THEN TRY TEST 171
12482 046602 012737 125252 042434 MOV #125252,@#TEMP2 :0 125252 SHIFTED BY 16
12483 046610 012737 125252 042440 MOV #125252,@#TEMP4 :IS EQUAL TO 125252 0, AND PS=12
12484 046616 000207 RTS %7
12485 046620 021527 000171 TST171: CMP (R5),#171 :IS IT TEST 171?
12486 046624 001010 BNE TST172 :IF NOT THEN TRY TEST 172
12487 046626 005337 042436 DEC @#TEMP3 :0 125252 SHIFTED BY 15
12488 046632 012737 052525 042440 MOV #52525,@#TEMP4 :IS EQUAL TO 52525 0
12489 046640 005037 042444 CLR @#TEMP6 :AND PS=0
12490 046644 000207 RTS %7
12491 046646 021527 000172 TST172: CMP (R5),#172 :IS IT TEST 172?
12492 046652 001006 BNE TST173 :IF NOT THEN TRY TEST 173
12493 046654 012737 052525 042434 MOV #52525,@#TEMP2 :0 52525
12494 046662 005237 042436 INC @#TEMP3 :SHIFTED BY 16. IS EQUAL TO 52525 0, AND PS C
12495 046666 000207 RTS %7
12496 046670 021527 000173 TST173: CMP (R5),#173 :IS IT TEST 173?
12497 046674 001014 BNE TST174 :IF NOT THEN TRY TEST 174
12498 046676 012737 177777 042434 MOV #-1,@#TEMP2 :0 -1
12499 046704 005337 042436 DEC @#TEMP3 :SHIFTED BY 15.
12500 046710 012737 077777 042440 MOV #77777,@#TEMP4
12501 046716 012737 100000 042442 MOV #100000,@#TEMP5 :IS EQUAL TO 77777 100000. AND PS=0
12502 046724 000207 RTS %7
12503 046726 021527 000174 TST174: CMP (R5),#174 :IS IT TEST 174?
12504 046732 001013 BNE TST175 :IF NOT THEN TRY TEST 175
12505 046734 012737 100000 042432 MOV #100000,@#TEMP1
12506 046742 005337 042434 DEC @#TEMP2 :100000 -2 SHIFTED BY 15.
12507 046746 005037 042442 CLR @#TEMP5 :IS EQUAL TO 77777 0
12508 046752 012737 000002 042444 MOV #2,@#TEMP6 :AND PS=2
12509 046760 000207 RTS %7
12510 046762 021527 000175 TST175: CMP (R5),#175 :IS IT TEST 175?
12511 046766 001015 BNE ENT176 :IF NOT THEN TRY TEST 176
12512 046770 012737 177777 042432 MOV #-1,@#TEMP1
12513 046776 005037 042434 CLR @#TEMP2 :-1 0
12514 047002 005237 042436 INC @#TEMP3 :SHIFTED BY 16.
12515 047006 005037 042440 CLR @#TEMP4 :IS EQUAL TO 0 0
12516 047012 012737 000007 042444 MOV #7,@#TEMP6 :AND PS=7
12517 047020 000207 RTS %7
12518 047022 021527 000176 ENT176: CMP (R5),#176 :IS THE PROGRAM ENTERING TEST 176?
12519 047026 001403 BEQ .+10
12520 047030 004767 011650 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12521
12522 047034 000056 56 :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12523 :BY (013746 000172 000207)

```

```
12524
12525
12526 047036 005726          TST      (SP)+          ;RESTORE STACK POINTER
12527
12528 ;*****
12529 ;TEST:176          1 SHIFTED BY 8. = 400 PS = 0
12530 ;*****
12531
12532 047040 010767 131126      TST176: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRFS
12533 047044 012701 069000      MOV      #DUMMY,%1          ;LOAD R1 WITH DUMMY
12534 047050 012701 000001      MOV      #1,%1:1           ;LOAD R1:1 WITH 1
12535 047054 000241              CLC
12536 047056 073127 000010      ASHC     #8,%1              ;SHIFT R1,R1:1 BY 8.
12537 047062 106737 042430      MFPS     @#PSWORD           ;SAVE PS
12538 047066 122737 000000      CMPB    #0,@#PSWORD        ;IS THE PS 0?
12539 047074 001403              BEQ     .+10
12540 047076 004767 011602      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12541 ;THE PS IS NOT EQUAL TO 0
12542 047102 000057              57          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12543 ;BY (013746 000172 000207)
12544
12545 047104 022701 000400      CMP      #400,%1           ;IS THE RESULT 400?
12546 047110 001403              BEQ     .+10
12547 047112 004767 011566      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12548 ;R1 IS NOT EQUAL TO 400
12549 047116 000060              60          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12550 ;BY (013746 000172 000207)
12551
12552 047120 021527 000176      CMP      (R5),#176         ;IS $TESTN = #176?
12553 047124 001403              BEQ     .+10                ;IF NOT THEN GO TO HLT
12554 047126 004767 011552      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12555 ;TEST IS IN WRONG SEQUENCE
12556 047132 000061              61          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12557 ;BY (013746 000172 000207)
12558
12559 047134 005215              INC      (R5)
12560
12561
12562 ;*****
12563 ;TEST:177          -1 SHIFTED BY 15. = 100000 PS = 11
12564 ;*****
12565
12566 047136 010767 131030      TST177: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
12567 047142 012703 000000      MOV      #DUMMY,%3          ;LOAD R3 WITH DUMMY
12568 047146 012703 177777      MOV      #-1,%3:1          ;LOAD R3:1 WITH -1
12569 047152 000241              CLC
12570 047154 073327 000017      ASHC     #15,%3             ;SHIFT R3,R3:1 BY 15.
12571 047160 106737 042430      MFPS     @#PSWORD           ;SAVE PS
12572 047164 122737 000011      CMPB    #11,@#PSWORD        ;IS THE PS 11?
12573 047172 001403              BEQ     .+10
12574 047174 004767 011504      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12575 ;THE PS IS NOT EQUAL TO 11
12576 047200 000062              62          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12577 ;BY (013746 000172 000207)
12578
12579 047202 022703 100000      CMP      #100000,%3         ;IS THE RESULT 100000?
```

```

12580 047206 001403          BEQ      .+10
12581 047210 004767  C11470   JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12582                                     ;R3 IS NOT EQUAL TO 100000
12583 047214 000063          63       ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12584                                     ;BY (013746 000172 000207)
12585
12586 047216 021527 000177     CMP      (R5),#177          ;IS $TESTN = #177?
12587 047222 001403          BEQ      .+10             ;IF NOT THEN GO TO HLT
12588 047224 004767 011454     JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12589                                     ;TEST IS IN WRONG SEQUENCE
12590 047230 000064          64       ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12591                                     ;BY (013746 000172 000207)
12592
12593 047232 005215          INC      (R5)
12594
12595
12596                                     :*****
12597                                     ;TEST:200      52525 SHIFTED BY 0 = 52525  PS = 0
12598                                     :*****
12599
12600 047234 010767 130732     TST200: MOV     PC,LPADR      ;STORE ERROR LOOP ADDRESS
12601 047240 010501          MOV     R5,R1             ;SAVE R5
12602 047242 012705 000C00     MOV     #DUMMY,%5        ;LOAD R5 WITH DUMMY
12603 047246 012705 052525     MOV     #52525,%5.1      ;LOAD R5:1 WITH 52525
12604 047252 000241          CLC
12605 047254 073527 000000     ASHC   #0,%5             ;SHIFT R5,R5:1 BY 0
12606 047260 106737 042430     MFPS   @#PSWORD         ;SAVE PS
12607 047264 122737 000000 042430 CMPB   #0,@#PSWORD       ;IS THE PS 0?
12608 047272 001403          BEQ     .+10
12609 047274 004767 011404     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12610                                     ;THE PS IS NOT EQUAL TO 0
12611 047300 000065          65       ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12612                                     ;BY (013746 000172 000207)
12613
12614 047302 022705 052525     CMP     #52525,%5        ;IS THE RESULT 52525?
12615 047306 001403          BEQ     .+10
12616 047310 004767 011370     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12617                                     ;R5 IS NOT EQUAL TO 52525
12618 047314 000066          66       ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12619                                     ;BY (013746 000172 000207)
12620
12621 047316 010105          MOV     R1,R5            ;RESTORE R5
12622 047320 021527 000200     CMP     (R5),#200        ;IS $TESTN = #200?
12623 047324 001403          BEQ     .+10             ;IF NOT THEN GO TO HLT
12624 047326 004767 011352     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12625                                     ;TEST IS IN WRONG SEQUENCE
12626 047332 000067          67       ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12627                                     ;BY (013746 000172 000207)
12628
12629 047334 005215          INC     (R5)
12630
12631
12632                                     :*****
12633                                     ;TEST:201      20010 SHIFTED BY -13. - 101  PS = 0
12634                                     :*****
12635

```

```
12636 047336 010767 130630 TST201: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12637 047342 012701 000000 MOV #DUMMY,%1 ;LOAD R1 WITH DUMMY
12638 047346 012701 020010 MOV #20010,%1.1 ;LOAD R1:1 WITH 20010
12639 047352 000241 CLC
12640 047354 073127 177763 ASHC #-13.,%1 ;SHIFT R1,R1:1 BY -13.
12641 047360 106737 042430 MFPS @#PSWORD ;SAVE PS
12642 047364 122737 000000 042430 CMPB #0,@#PSWORD ;IS THE PS 0?
12643 047372 001403 BEQ .+10
12644 047374 004767 011304 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12645 ;THE PS IS NOT EQUAL TO 0
12646 047400 000070 70 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12647 ;BY (013746 000172 000207)
12648
12649 047402 022701 000101 CMP #101,%1 ;IS THE RESULT 101?
12650 047406 001403 BEQ .+10
12651 047410 004767 011270 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12652 ;R1 IS NOT EQUAL TO 101
12653 047414 000071 71 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12654 ;BY (013746 000172 000207)
12655
12656 047416 021527 000201 CMP (R5),#201 ;IS $TESTN = #201?
12657 047422 001403 BEQ .+10 ;IF NOT THEN GO TO HLT
12658 047424 004767 011254 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12659 ;TEST IS IN WRONG SEQUENCE
12660 047430 000072 72 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12661 ;BY (013746 000172 000207)
12662
12663 047432 005215 INC (R5)
12664
12665
12666 ;*****
12667 ;TEST:202 -1 SHIFTED BY 16. = 0 PS = 11
12668 ;*****
12669
12670 047434 010767 130532 TST202: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12671 047440 012703 000000 MOV #DUMMY,%3 ;LOAD R3 WITH DUMMY
12672 047444 012703 177777 MOV #-1,%3:1 ;LOAD R3:1 WITH -1
12673 047450 000241 CLC
12674 047452 073327 000020 ASHC #16.,%3 ;SHIFT R3,R3:1 BY 16.
12675 047456 106737 042430 MFPS @#PSWORD ;SAVE PS
12676 047462 122737 000011 042430 CMPB #11,@#PSWORD ;IS THE PS 11?
12677 047470 001403 BEQ .+10
12678 047472 004767 011206 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12679 ;THE PS IS NOT EQUAL TO 11
12680 047476 000073 73 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12681 ;BY (013746 000172 000207)
12682
12683 047500 022703 000000 CMP #0,%3 ;IS THE RESULT 0?
12684 047504 001403 BEQ .+10
12685 047506 004767 011172 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12686 ;R3 IS NOT EQUAL TO 0
12687 047512 000074 74 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12688 ;BY (013746 000172 000207)
12689
12690 047514 021527 000202 CMP (R5),#202 ;IS $TESTN = #202?
12691 047520 001403 BEQ .+10 ;IF NOT THEN GO TO HLT
```

```
12692 047522 004767 011156 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12693 ;TEST IS IN WRONG SEQUENCE
12694 047526 000075 75 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12695 ;BY (013746 000172 000207)
12696
12697 047530 005215 INC (R5)
12698
12699
12700
12701 ;*****
12702 ;TEST:203 1 SHIFTED BY -1 100000 PS - 1
12703 ;*****
12704 047532 010767 130434 TST203: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12705 047536 010501 MOV R5,R1 ;SAVE R5
12706 047540 012705 000000 MOV #DUMMY,%5 ;LOAD R5 WITH DUMMY
12707 047544 012705 000001 MOV #1,%5.1 ;LOAD R5.1 WITH 1
12708 047550 000241 CLC
12709 047552 073527 177777 ASHC #-1,%5 ;SHIFT R5,R5.1 BY -1
12710 047556 106737 042430 MFPS @#PSWORD ;SAVE PS
12711 047562 122737 0C0001 042430 CMPB #1,@#PSWORD ;IS THE PS 1?
12712 047570 001403 BEQ .+10
12713 047572 004767 011106 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12714 ;THE PS IS NOT EQUAL TO 1
12715 047576 000076 76 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12716 ;BY (013746 000172 000207)
12717
12718 047600 022705 100000 CMP #100000,%5 ;IS THE RESULT 100000?
12719 047604 001403 BEQ .+10
12720 047606 004767 011072 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12721 ;R5 IS NOT EQUAL TO 100000
12722 047612 000077 77 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12723 ;BY (013746 000172 000207)
12724
12725 047614 010105 MOV R1,R5 ;RESTORE R5
12726 047616 021527 000203 CMP (R5),#203 ;IS $TESTN - #203?
12727 047622 001403 BEQ .+10 ;IF NOT THEN GO TO HLT
12728 047624 004767 011054 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12729 ;TEST IS IN WRONG SEQUENCE
12730 047630 000100 100 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12731 ;BY (013746 000172 000207)
12732
12733 047632 005215 INC (R5)
12734
12735
12736 ;*****
12737 ;TEST:204 125252 SHIFTED BY -16. = 125252 PS - 11
12738 ;*****
12739
12740 047634 010767 130332 TST204: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12741 047640 012701 000000 MOV #DUMMY,%1 ;LOAD R1 WITH DUMMY
12742 047644 012701 125252 MOV #125252,%1:1 ;LOAD R1:1 WITH 125252
12743 047650 000241 CLC
12744 047652 073127 177760 ASHC #-16,%1 ;SHIFT R1,R1:1 BY -16.
12745 047656 106737 042430 MFPS @#PSWORD ;SAVE PS
12746 047662 122737 000011 042430 CMPB #11,@#PSWORD ;IS THE PS 11?
12747 047670 001403 BEQ .+10
```



```

12748 047672 004767 011006 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12749 :THE PS IS NOT EQUAL TO 11
12750 047676 000101 101 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12751 :BY (013746 000172 000207)
12752
12753 047700 022701 125252 (MP #125252,%1 ;IS THE RESULT 125252?
12754 047704 001403 BEQ .+10
12755 047706 004767 010772 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12756 :R1 IS NOT EQUAL TO 125252
12757 047712 000102 102 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12758 :BY (013746 000172 000207)
12759
12760 047714 021527 000204 (MP (R5),#204 ;IS $TESTN = #204?
12761 047720 001403 BEQ .+10 ;IF NOT THEN GO TO HLT
12762 047722 004767 010756 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12763 :TEST IS IN WRONG SEQUENCE
12764 047726 000103 103 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12765 :BY (013746 000172 000207)
12766
12767 047730 005215 INC (R5)
12768
12769
12770 :*****
12771 :TEST:205 125252 125252 SHIFTED BY 21. = 52500 000000 PS 3
12772 :*****
12773
12774 047732 010767 130234 TST205: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12775 047736 012702 125252 MOV #125252,%2 ;LOAD R2 WITH 125252
12776 047742 012703 125252 MOV #125252,%2!1 ;LOAD R2!1 WITH 125252
12777 047746 000241 CLC
12778 047750 073227 000025 ASHC #21,%2 ;SHIFT R2,R2!1 BY 21.
12779 047754 106737 042430 MFPS @#PSWORD ;SAVE PS
12780 047760 122737 000003 042430 CMPB #3,@#PSWORD ;IS THE PS 3?
12781 047766 001403 BEQ .+10
12782 047770 004767 010710 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12783 :THE PS IS NOT EQUAL TO 3
12784 047774 000104 104 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12785 :BY (013746 000172 000207)
12786
12787 047776 022702 052500 (MP #52500,%2 ;IS THE RESULT 52500?
12788 050002 001403 BEQ .+10
12789 050004 004767 010674 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12790 :R2 IS NOT EQUAL TO 52500
12791 050010 000105 105 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12792 :BY (013746 000172 000207)
12793
12794 050012 022703 000000 (MP #000000,%2!1 ;IS THE RESULT 000000?
12795 050016 001403 BEQ .+10
12796 050020 004767 010660 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12797 :R2!1 IS NOT EQUAL TO 000000
12798 050024 000106 106 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12799 :BY (013746 000172 000207)
12800
12801 050026 021527 000205 (MP (R5),#205 ;IS $TESTN = #205?
12802 050032 001403 BEQ .+10 ;IF NOT THEN GO TO HLT
12803 050034 004767 010644 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE

```

```

12804
12805 050040 000107          107          ;TEST IS IN WRONG SEQUENCE
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
12806
12807
12808 050042 005215          INC          (R5)
12809
12810
12811
12812 050044 012702 177771          MOV          #-7,%2
12813 050050 012703 042452          MOV          #5,%3
12814 050054 012704 042454          MOV          #52,%4
12815
12816
12817 ;*****
;TEST:206          125252 125252 SHIFTED BY S1 - 177525 52525 PS - 10
;*****
12818
12819
12820 050060 010767 130106          TST206: MOV    PC,LPADR          ;STORE ERROR LOOP ADDRESS
12821 050064 012700 125252          MOV    #125252,%0          ;LOAD R0 WITH 125252
12822 050070 012701 125252          MOV    #125252,%0:1        ;LOAD R0:1 WITH 125252
12823 050074 000241          CLC
12824 050076 073067 172350          ASHC   S1,%0          ;SHIFT R0,R0:1 BY S1
12825 050102 106737 042430          MFPS   @#PSWORD          ;SAVE PS
12826 050106 122737 000010          CMPB  #10,@#PSWORD        ;IS THE PS 10?
12827 050114 001403          BEQ   .+10
12828 050116 004767 010562          JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
12829
12830 050122 000110          110          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
12831
12832
12833 050124 022700 177525          CMP   #177525,%0          ;IS THE RESULT 177525?
12834 050130 001403          BEQ   .+10
12835 050132 004767 010546          JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;R0 IS NOT EQUAL TO 177525
12836
12837 050136 000111          111          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
12838
12839
12840 050140 022701 052525          CMP   #52525,%0:1        ;IS THE RESULT 52525?
12841 050144 001403          BEQ   .+10
12842 050146          1$:
12843 050146 004767 010532          JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;R0:1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12844
12845 050152 000112          112          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
12846
12847
12848 050154 021527 000206          CMP   (R5),#206          ;IS THE $TESTN - #206?
12849 050160 001372          BNE   1$          ;IF NOT THEN GO TO HLT ABOVE
12850 050162 005215          INC   (R5)
12851
12852
12853 ;*****
;TEST:207          125252 125252 SHIFTED BY @S2 = 177525 52525 PS - 10
;*****
12854
12855
12856
12857 050164 010767 130002          TST207: MOV    PC,LPADR          ;STORE ERROR LOOP ADDRESS
12858 050170 012700 125252          MOV    #125252,%0          ;LOAD R0 WITH 125252
12859 050174 012701 125252          MOV    #125252,%0:1        ;LOAD R0:1 WITH 125252

```

12860	050200	000241		CLC		
12861	050202	073077	172246	ASHC	@S2,%0	:SHIFT R0,R0.1 BY @S2
12862	050206	106737	042430	MFPS	@#PSWORD	:SAVE PS
12863	050212	122737	000010	CMPS	#10,@#PSWORD	:IS THE PS 10?
12864	050220	001403		BEQ	+.10	
12865	050222	004767	010456	JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
12866						:THE PS IS NOT EQUAL TO 10
12867	050226	000113		113		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12868						:BY (013746 000172 000207)
12869						
12870	050230	022700	177525	CMPS	#177525,%0	:IS THE RESULT 177525?
12871	050234	001403		BEQ	+.10	
12872	050236	004767	010442	JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
12873						:R0 IS NOT EQUAL TO 177525
12874	050242	000114		114		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12875						:BY (013746 000172 000207)
12876						
12877	050244	022701	052525	CMPS	#52525,%0.1	:IS THE RESULT 52525?
12878	050250	001403		BEQ	+.10	
12879	050252					
12880	050252	004767	010426	JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
12881						:R0.1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12882	050256	000115		115		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12883						:BY (013746 000172 000207)
12884						
12885	050260	021527	000207	CMPS	(R5),#207	:IS THE \$TESTN = #207?
12886	050264	001372		BNE	1\$	:IF NOT THEN GO TO HLT ABOVE
12887	050266	005215		INC	(R5)	
12888						
12889						
12890						:*****
12891						:TEST:210 125252 125252 SHIFTED BY @#S1 = 177525 52525 PS 10
12892						:*****
12893						
12894	050270	010767	127676	TST210: MOV	PC,LPADR	:STORE ERROR LOOP ADDRESS
12895	050274	012700	125252	MOV	#125252,%0	:LOAD R0 WITH 125252
12896	050300	012701	125252	MOV	#125252,%0.1	:LOAD R0.1 WITH 125252
12897	050304	000241		CLC		
12898	050306	073037	042452	ASHC	@#S1,%0	:SHIFT R0,R0.1 BY @#S1
12899	050312	106737	042430	MFPS	@#PSWORD	:SAVE PS
12900	050316	122737	000010	CMPS	#10,@#PSWORD	:IS THE PS 10?
12901	050324	001403		BEQ	+.10	
12902	050326	004767	010352	JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
12903						:THE PS IS NOT EQUAL TO 10
12904	050332	000116		116		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12905						:BY (013746 000172 000207)
12906						
12907	050334	022700	177525	CMPS	#177525,%0	:IS THE RESULT 177525?
12908	050340	001403		BEQ	+.10	
12909	050342	004767	010336	JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
12910						:R0 IS NOT EQUAL TO 177525
12911	050346	000117		117		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12912						:BY (013746 000172 000207)
12913						
12914	050350	022701	052525	CMPS	#52525,%0.1	:IS THE RESULT 52525?
12915	050354	001403		BEQ	+.10	

12916 050356  
12917 050356 004767 010322  
12918  
12919 050362 000120  
12920  
12921  
12922 050364 021527 000210  
12923 050370 001372  
12924 050372 005215  
12925  
12926  
12927  
12928  
12929  
12930

1\$: JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;RO.1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUEN I  
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  
\*\*\*\*\*

12931 050374 010767 127572  
12932 050400 012700 125252  
12933 050404 012701 125252  
12934 050410 000241  
12935 050412 073013  
12936 050414 106737 042430  
12937 050420 122737 000010 042430  
12938 050426 001403  
12939 050430 004767 010250  
12940

TST211: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #125252,%0 ;LOAD RO WITH 125252  
MOV #125252,%0:1 ;LOAD RO:1 WITH 125252  
CLC  
ASHC (3),%0 ;SHIFT RO,RO: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

12941 050434 000121  
12942  
12943  
12944 050436 022700 177525  
12945 050442 001403  
12946 050444 004767 010234  
12947  
12948 050450 000122  
12949  
12950

121 ;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  
;RO IS NOT EQUAL TO 177525  
122 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

12951 050452 022701 052525  
12952 050456 001403  
12953 050460  
12954 050460 004767 010220  
12955  
12956 050464 000123  
12957  
12958  
12959 050466 021527 000211  
12960 050472 001372  
12961 050474 005215  
12962  
12963

1\$: CMP #52525,%0.1 ;IS THE RESULT 52525?  
BEQ .+10  
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;RO:1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE  
123 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
CMP (R5),#211 ;IS THE \$TESTN #211?  
BNE 1\$ ;IF NOT THEN GO TO HLT ABOVE  
INC (R5)

\*\*\*\*\*  
:TEST:212 125252 125252 SHIFTED BY (3)+ - 177525 52525 PS 10  
\*\*\*\*\*

12968 050476 010767 127470  
12969 050502 012700 125252  
12970 050506 012701 125252  
12971 050512 000241

TST212: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #125252,%0 ;LOAD RO WITH 125252  
MOV #125252,%0.1 ;LOAD RO:1 WITH 125252  
CLC

```

12972 050514 073023          ASHC      (3),%0          ;SHIFT R0,R0 1 BY (3)+
12973 050516 106737 042430  MFPS      @#PSWORD        ;SAVE PS
12974 050522 122737 000010 042430  CMPB      #10,@#PSWORD      ;IS THE PS 10?
12975 050530 001403          BEQ       .+10
12976 050532 004767 010146  JSR       PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12977                                ;THE PS IS NOT EQUAL TO 10
12978 050536 000124          124       ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12979                                ;BY (013746 000172 000207)
12980
12981 050540 022700 177525  CMP       #177525,%0    ;IS THE RESULT 177525?
12982 050544 001403          BFQ       .+10
12983 050546 004767 010132  JSR       PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12984                                ;RO IS NOT EQUAL TO 177525
12985 050552 000125          125       ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12986                                ;BY (013746 000172 000207)
12987
12988 050554 022701 052525  CMP       #52525,%0!1   ;IS THE RESULT 52525?
12989 050560 001403          BEQ       .+10
12990                                1$:
12991 050562 004767 010116  JSR       PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12992                                ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12993 050566 000126          126       ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12994                                ;BY (013746 000172 000207)
12995
12996 050570 021527 000212  CMP       (R5),#212     ;IS THE $TESTN - #212?
12997 050574 001372          BNE      1$           ;IF NOT THEN GO TO HLT ABOVE
12998 050576 005215          INC      (R5)
12999
13000
13001                                ;*****
13002                                ;TEST:213      125252 125252 SHIFTED BY -(3) = 177525 52525 PS 10
13003                                ;*****
13004
13005 050600 010767 127366  TST213:  MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
13006 050604 012700 125252  MOV      #125252,%0     ;LOAD R0 WITH 125252
13007 050610 012701 125252  MOV      #125252,%0!1   ;LOAD R0!1 WITH 125252
13008 050614 000241          CLL
13009 050616 073043          ASHC     -(3),%0       ;SHIFT R0,R0!1 BY -(3)
13010 050620 106737 042430  MFPS     @#PSWORD      ;SAVE PS
13011 050624 122737 000010 042430  CMPB     #10,@#PSWORD   ;IS THE PS 10?
13012 050632 001403          BEQ     .+10
13013 050634 004767 010044  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13014                                ;THE PS IS NOT EQUAL TO 10
13015 050640 000127          127       ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13016                                ;BY (013746 000172 000207)
13017
13018 050642 022700 177525  CMP     #177525,%0     ;IS THE RESULT 177525?
13019 050646 001403          BEQ     .+10
13020 050650 004767 010030  JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13021                                ;RO IS NOT EQUAL TO 177525
13022 050654 000130          130       ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13023                                ;BY (013746 000172 000207)
13024
13025 050656 022701 052525  CMP     #52525,%0.1    ;IS THE RESULT 52525?
13026 050662 001403          BEQ     .+10
13027 050664                                1$:

```

```

13028 050664 004767 010014 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13029 ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13030 050670 000131 131 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13031 ;BY (013746 000172 000207)
13032
13033 050672 021527 000213 CMP (R5),#213 ;IS THE $TESTN #213?
13034 050676 001372 BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
13035 050700 005215 INC (R5)
13036
13037
13038
13039 ;*****
13040 ;TEST:214 125252 125252 SHIFTED BY 2(4) - 177252 125252 PS 11
13041 ;*****
13042 050702 010767 127264 TST214: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13043 050706 012700 125252 MOV #125252,%0 ;LOAD RO WITH 125252
13044 050712 012701 125252 MOV #125252,%0!1 ;LOAD RO!1 WITH 125252
13045 050716 000241 CLC
13046 050720 073064 000002 ASHC 2(4),%0 ;SHIFT RO,RO!1 BY 2(4)
13047 050724 106737 042430 MFPS @#PSWORD ;SAVE PS
13048 050730 122737 000011 042430 CMPB #11,@#PSWORD ;IS THE PS 11?
13049 050736 001403 BEQ .+10
13050 050740 004767 007740 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13051 ;THE PS IS NOT EQUAL TO 11
13052 050744 000132 132 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13053 ;BY (013746 000172 000207)
13054
13055 050746 022700 177252 CMP #177252,%0 ;IS THE RESULT 177252?
13056 050752 001403 BEQ .+10
13057 050754 004767 007724 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13058 ;RO IS NOT EQUAL TO 177252
13059 050760 000133 133 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13060 ;BY (013746 000172 000207)
13061
13062 050762 022701 125252 CMP #125252,%0!1 ;IS THE RESULT 125252?
13063 050766 001403 BEQ .+10
13064 050770 004767 007710 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13065 ;RO!1 IS NOT EQUAL TO 125252 OR INCORRECT SEQUENCE
13066 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13067 050774 000134 134 ;BY (013746 000172 000207)
13068
13069
13070 050776 021527 000214 CMP (R5),#214 ;IS THE $TESTN - #214?
13071 051002 001372 BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
13072 051004 005215 INC (R5)
13073
13074
13075
13076 ;*****
13077 ;TEST:215 125252 125252 SHIFTED BY @ (4) = 177525 52525 PS 10
13078 ;*****
13079 051006 010767 127160 TST215: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13080 051012 012700 125252 MOV #125252,%0 ;LOAD RO WITH 125252
13081 051016 012701 125252 MOV #125252,%0!1 ;LOAD RO!1 WITH 125252
13082 051022 000241 CLC
13083 051024 073074 000000 ASHC @ (4),%0 ;SHIFT RO,RO!1 BY @ (4)

```

```
13084 051030 106737 042430 MFPS @#PSWORD ;SAVE PS
13085 051034 122737 000010 042430 CMPB #10,@#PSWORD ;IS THE PS 10?
13086 051042 001403 BEQ .+10
13087 051044 004767 007634 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13088 ;THE PS IS NOT EQUAL TO 10
13089 051050 000135 135 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13090 ;BY (013746 000172 000207)
13091
13092 051052 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
13093 051056 001403 BEQ .+10
13094 051060 004767 007620 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13095 ;RO IS NOT EQUAL TO 177525
13096 051064 000136 136 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13097 ;BY (013746 000172 000207)
13098
13099 051066 022701 052525 CMP #52525,%0.1 ;IS THE RESULT 52525?
13100 051072 001403 BEQ .+10
13101 051074 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13102 051074 004767 007604 1$: ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13103 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13104 051100 000137 137 ;BY (013746 000172 000207)
13105
13106
13107 051102 021527 000215 CMP (R5),#215 ;IS THE $TESTN - #215?
13108 051106 001372 BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
13109 051110 005215 INC (R5)
13110
13111
13112 ;*****
13113 ;TEST:216 125252 125252 SHIFTED BY @ (4)+ = 177525 52525 PS - 10
13114 ;*****
13115
13116 051112 010767 127054 TST216: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13117 051116 012700 125252 MOV #125252,%0 ;LOAD RO WITH 125252
13118 051122 012701 125252 MOV #125252,%0!1 ;LOAD RO!1 WITH 125252
13119 051126 000241 CLC
13120 051130 073034 ASHC @ (4)+,%0 ;SHIFT RO,RO!1 BY @ (4)+
13121 051132 106737 042430 MFPS @#PSWORD ;SAVE PS
13122 051136 122737 000010 042430 CMPB #10,@#PSWORD ;IS THE PS 10?
13123 051144 001403 BEQ .+10
13124 051146 004767 007532 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13125 ;THE PS IS NOT EQUAL TO 10
13126 051152 000140 140 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13127 ;BY (013746 000172 000207)
13128
13129 051154 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
13130 051160 001403 BEQ .+10
13131 051162 004767 007516 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13132 ;RO IS NOT EQUAL TO 177525
13133 051166 000141 141 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13134 ;BY (013746 000172 000207)
13135
13136 051170 022701 052525 CMP #52525,%0!1 ;IS THE RESULT 52525?
13137 051174 001403 BEQ .+10
13138 051176 1$:
13139 051176 004767 007502 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
```

```

13140
13141 051202 000142          142          ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13142                                     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13143                                     ;BY (013746 000172 000207)
13144 051204 021527 000216    CMP      (R5),#216          ;IS THE $TESTN = #216?
13145 051210 001372          BNE     1$                ;IF NOT THEN GO TO HLT ABOVE
13146 051212 005215          INC     (R5)
13147
13148
13149
13150
13151
13152
13153 051214 010767 126752    TST217: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
13154 051220 012700 125252    MOV     #125252,%0        ;LOAD RO WITH 125252
13155 051224 012701 125252    MOV     #125252,%0.1     ;LOAD RO!1 WITH 125252
13156 051230 000241          CLC
13157 051232 073054          ASHC   @-(4),%0          ;SHIFT RO,RO!1 BY @-(4)
13158 051234 106737 042430    MFPS   @#PSWORD         ;SAVE PS
13159 051240 122737 0C0010    CMPB   #10,@#PSWORD     ;IS THE PS 10?
13160 051246 001403          BEQ    .+10
13161 051250 004767 007430    JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13162                                     ;THE PS IS NOT EQUAL TO 10
13163 051254 000143          143          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13164                                     ;BY (013746 000172 000207)
13165
13166 051256 022700 177525    CMP     #177525,%0       ;IS THE RESULT 177525?
13167 051262 001403          BEQ    .+10
13168 051264 004767 007414    JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13169                                     ;RO IS NOT EQUAL TO 177525
13170 051270 000144          144          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13171                                     ;BY (013746 000172 000207)
13172
13173 051272 022701 052525    CMP     #52525,%0!1     ;IS THE RESULT 52525?
13174 051276 001403          BEQ    .+10
13175 051300
13176 051300 004767 007400    JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13177                                     ;RO!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13178 051304 000145          145          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13179                                     ;BY (013746 000172 000207)
13180
13181 051306 021527 000217    CMP     (R5),#217       ;IS THE $TESTN = #217?
13182 051312 001372          BNE     1$                ;IF NOT THEN GO TO HLT ABOVE
13183 051314 005215          INC     (R5)
13184
13185
13186
13187
13188
13189
13190
13191
13192

```

\*\*\*\*\*  
:TEST:217 125252 125252 SHIFTED BY @-(4) = 177525 52525 PS = 10  
\*\*\*\*\*

042430

1\$:



13193  
13194  
13195  
13196  
13197  
13198  
13199  
13200  
13201  
13202  
13203  
13204  
13205  
13206  
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  
13242  
13243  
13244  
13245  
13246  
13247  
13248

\*\*\*\*\*  
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)

```

\*\*\*\*\*  
: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
        ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
        ;BY (013746 000172 000207)

```

```
13249
13250 051452 022700 177777      CMP    # -1,%0      ;IS HIGH ORDER = -1
13251 051456 001403      BEQ    .+10
13252 051460 004767 007220      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13253                                     ;HIGH ORDER IS WRONG
13254 051464 000152      152      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13255                                     ;BY (013746 000172 000207)
13256
13257 051466 022701 177777      CMP    # -1,%0!1    ;IS LOW ORDER = -1
13258 051472 001403      BEQ    .+10
13259 051474                                     1$:
13260 051474 004767 007204      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13261                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13262 051500 000153      153      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13263                                     ;BY (013746 000172 000207)
13264
13265 051502 021527 000221      CMP    (R5),#221
13266 051506 001372      BNE    1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13267 051510 005215      INC    (R5)
13268
13269
13270                                     :*****
13271 :TEST:222      MUL    2 * #2 = 0 4      PS  0
13272 :*****
13273
13274 051512 010767 126454      TST222: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
13275 051516 012702 000002      MOV    #2,%2        ;LOAD MULTIPLICAND WITH 2
13276 051522 070227 000002      MUL    #2,%2        ;MULTIPLY 2 * #2
13277 051526 106737 042430      MFPS   @#PSWORD     ;SAVE PS
13278 051532 122737 000000 042430  CMPB   #0,@#PSWORD  ;IS PS = 0
13279 051540 001403      BEQ    .+10
13280 051542 004767 007136      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13281                                     ;PS IS WRONG
13282 051546 000154      154      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13283                                     ;BY (013746 000172 000207)
13284
13285 051550 022702 000000      CMP    #0,%2        ;IS HIGH ORDER = 0
13286 051554 001403      BEQ    .+10
13287 051556 004767 007122      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13288                                     ;HIGH ORDER IS WRONG
13289 051562 000155      155      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13290                                     ;BY (013746 000172 000207)
13291
13292 051564 022703 000004      CMP    #4,%2!1     ;IS LOW ORDER = 4
13293 051570 001403      BEQ    .+10
13294 051572                                     1$:
13295 051572 004767 007106      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13296                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13297 051576 000156      156      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13298                                     ;BY (013746 000172 000207)
13299
13300 051600 021527 000222      CMP    (R5),#222
13301 051604 001372      BNE    1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13302 051606 005215      INC    (R5)
13303
13304
```

```
13305 :*****
13306 :TEST:223      MUL      1000 * #200 - 1 0      PS      1
13307 :*****
13308
13309 051610 010767 126356 TST223: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
13310 051614 010501      MOV      R5,R1          ;SAVE R5
13311 051616 012704 001000      MOV      #1000,%4      ;LOAD MULTIPLICAND WITH 1000
13312 051622 070427 000200      MUL      #200,%4      ;MULTIPLY 1000 * #200
13313 051626 106737 042430      MFPS    @#PSWORD      ;SAVE PS
13314 051632 122737 000001 042430      CMPB    #1,@#PSWORD   ;IS PS = 1
13315 051640 001403      BEQ     .+10
13316 051642 004767 007036      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13317 :PS IS WRONG
13318 051646 000157      157     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13319 :BY (013746 000172 000207)
13320
13321 051650 022704 000001      CMP     #1,%4          ;IS HIGH ORDER - 1
13322 051654 001403      BEQ     .+10
13323 051656 004767 007022      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13324 :HIGH ORDER IS WRONG
13325 051662 000160      160     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13326 :BY (013746 000172 000207)
13327
13328 051664 022705 000000      CMP     #0,%4!1       ;IS LOW ORDER = 0
13329 051670 001403      BEQ     .+10
13330 051672
13331 051672 004767 007006      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13332 :LOW ORDER IS WRONG OR WRONG SEQUENCE
13333 051676 000161      161     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13334 :BY (013746 000172 000207)
13335
13336 051700 021127 000223      CMP     (R1),#223      ;CHECK THE TEST NUMBER
13337 051704 001372      BNE     1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13338 051706 010105      MOV     R1,R5          ;RESTORE R5
13339 051710 005215      INC     (R5)
13340
13341
13342 :*****
13343 :TEST:224      MUL      2 * #77777 - 0 177776      PS = 1
13344 :*****
13345
13346 051712 010767 126254 TST224: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
13347 051716 012700 000002      MOV      #2,%0        ;LOAD MULTIPLICAND WITH 2
13348 051722 070027 077777      MUL      #77777,%0    ;MULTIPLY 2 * #77777
13349 051726 106737 042430      MFPS    @#PSWORD      ;SAVE PS
13350 051732 122737 000001 042430      CMPB    #1,@#PSWORD   ;IS PS = 1
13351 051740 001403      BEQ     .+10
13352 051742 004767 006736      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13353 :PS IS WRONG
13354 051746 000162      162     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13355 :BY (013746 000172 000207)
13356
13357 051750 022700 000000      CMP     #0,%0          ;IS HIGH ORDER = 0
13358 051754 001403      BEQ     .+10
13359 051756 004767 006722      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13360 :HIGH ORDER IS WRONG
```

```
13361 051762 000163          163          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13362                               ;BY (013746 000172 000207)
13363
13364 051764 022701 177776          CMP      #177776,%0!1      ;IS LOW ORDER = 177776
13365 051770 001403          BEQ      .+10
13366 051772
13367 051772 004767 006706          JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13368                               ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13369 051776 000164          164          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13370                               ;BY (013746 000172 000207)
13371
13372 052000 021527 000224          CMP      (R5),#224
13373 052004 001372          BNE      1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13374 052006 005215          INC      (R5)
13375
13376
13377
13378          :*****
13379          :TEST:225      MUL      7777 * #10 = 0 77770      PS  0
13380          :*****
13381 052010 010767 126156          TST225: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
13382 052014 012702 007777          MOV      #7777,%2          ;LOAD MULTIPLICAND WITH 7777
13383 052020 070227 000010          MUL      #10,%2            ;MULTIPLY 7777 * #10
13384 052024 106737 042430          MFPS    @#PSWORD          ;SAVE PS
13385 052030 122737 000000 042430          CMPB    #0,@#PSWORD        ;IS PS = 0
13386 052036 001403          BEQ      .+10
13387 052040 004767 006640          JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13388                               ;PS IS WRONG
13389 052044 000165          165          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13390                               ;BY (013746 000172 000207)
13391
13392 052046 022702 000000          CMP      #0,%2            ;IS HIGH ORDER = 0
13393 052052 001403          BEQ      .+10
13394 052054 004767 006624          JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13395                               ;HIGH ORDER IS WRONG
13396 052060 000166          166          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13397                               ;BY (013746 000172 000207)
13398
13399 052062 022703 077770          CMP      #77770,%2!1      ;IS LOW ORDER = 77770
13400 052066 001403          BEQ      .+10
13401 052070
13402 052070 004767 006610          JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13403                               ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13404 052074 000167          167          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13405                               ;BY (013746 000172 000207)
13406
13407 052076 021527 000225          CMP      (R5),#225
13408 052102 001372          BNE      1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13409 052104 005215          INC      (R5)
13410
13411
13412          :*****
13413          :TEST:226      MUL      7777 * #7777 = 37777 1      PS = 1
13414          :*****
13415
13416 052106 010767 126060          TST226: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
```

```

13417 052112 010501          MOV    R5,R1          ;SAVE R5
13418 052114 012704 077777    MOV    #77777,%4     ;LOAD MULTIPLICAND WITH 77777
13419 052120 070427 077777    MUL    #77777,%4     ;MULTIPLY 77777 * #77777
13420 052124 106737 042430    MFPS   @#PSWORD      ;SAVE PS
13421 052130 122737 000001 042430    CMPB   #1,@#PSWORD   ;IS PS = 1
13422 052136 001403          BEQ    .+10
13423 052140 004767 006540    JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13424                                ;PS IS WRONG
13425 052144 000170          170                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13426                                ;BY (013746 000172 000207)
13427
13428 052146 022704 037777    CMP    #37777,%4     ;IS HIGH ORDER = 37777
13429 052152 001403          BEQ    .+10
13430 052154 004767 006524    JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13431                                ;HIGH ORDER IS WRONG
13432 052160 000171          171                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13433                                ;BY (013746 000172 000207)
13434
13435 052162 022705 000001    CMP    #1,%4.1       ;IS LOW ORDER = 1
13436 052166 001403          BEQ    .+10
13437 052170                                1$:
13438 052170 004767 006510    JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13439                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13440 052174 000172          172                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13441                                ;BY (013746 000172 000207)
13442
13443 052176 021127 000226    CMP    (R1),#226     ;CHECK THE TEST NUMBER
13444 052202 001372          BNE   1$             ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13445 052204 010105          MOV    R1,R5         ;RESTORE R5
13446 052206 005215          INC    (R5)
13447
13448
13449
13450 :*****
13451 :TEST:227      MUL    -1 * #77777 - -1 100001      PS 10
13452 :*****
13453 052210 010767 125756    TST227: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
13454 052214 012702 177777    MOV    #-1,%2        ;LOAD MULTIPLICAND WITH -1
13455 052220 070227 077777    MUL    #77777,%2     ;MULTIPLY -1 * #77777
13456 052224 106737 042430    MFPS   @#PSWORD      ;SAVE PS
13457 052230 122737 000010 042430    CMPB   #10,@#PSWORD  ;IS PS = 10
13458 052236 001403          BEQ    .+10
13459 052240 004767 006440    JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13460                                ;PS IS WRONG
13461 052244 000173          173                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13462                                ;BY (013746 000172 000207)
13463
13464 052246 022702 177777    CMP    #-1,%2        ;IS HIGH ORDER = -1
13465 052252 001403          BEQ    .+10
13466 052254 004767 006424    JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13467                                ;HIGH ORDER IS WRONG
13468 052260 000174          174                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13469                                ;BY (013746 000172 000207)
13470
13471 052262 022703 100001    CMP    #100001,%2.1  ;IS LOW ORDER = 100001
13472 052266 001403          BEQ    .+10

```

13473 052270  
13474 052270 004767 006410  
13475  
13476 052274 000175  
13477  
13478  
13479 052276 021527 000227  
13480 052302 001372  
13481 052304 005215  
13482  
13483  
13484  
13485  
13486  
13487

1\$: JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;LOW ORDER IS WRONG OR WRONG SEQUENCE  
175 ;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)

\*\*\*\*\*  
:TEST:230 MUL -2 \* #77777 = -1 2 PS - 11  
\*\*\*\*\*

13488 052306 010767 125660  
13489 052312 012700 177776  
13490 052316 070027 077777  
13491 052322 106737 042430  
13492 052326 122737 000011 042430  
13493 052334 001403  
13494 052336 004767 006342  
13495  
13496 052342 000176  
13497  
13498  
13499 052344 022700 177777  
13500 052350 001403  
13501 052352 004767 006326  
13502  
13503 052356 000177  
13504  
13505  
13506 052360 022701 000002  
13507 052364 001403  
13508 052366  
13509 052366 004767 006312  
13510  
13511 052372 000200  
13512  
13513  
13514 052374 021527 000230  
13515 052400 001372  
13516 052402 005215  
13517  
13518  
13519

TEST230: 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  
176 ;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  
177 ;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  
200 ;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)

\*\*\*\*\*  
:TEST:231 MUL 125252 \* #2 = -1 52524 PS = 11  
\*\*\*\*\*

13522  
13523 052404 010767 125562  
13524 052410 012702 125252  
13525 052414 070227 000002  
13526 052420 106737 042430  
13527 052424 122737 000011 042430  
13528 052432 001403

TEST231: 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

```
13529 052434 004767 006244 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13530 ;PS IS WRONG
13531 052440 000201 201 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13532 ;BY (013746 000172 000207)
13533
13534 052442 022702 177777 CMP #-1,%2 ;IS HIGH ORDER = -1
13535 052446 001403 BEQ .+10
13536 052450 004767 006230 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13537 ;HIGH ORDER IS WRONG
13538 052454 000202 202 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13539 ;BY (013746 000172 000207)
13540
13541 052456 022703 052524 CMP #52524,%2.1 ;IS LOW ORDER = 52524
13542 052462 001403 BEQ .+10
13543 052464
13544 052464 004767 006214 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13545 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13546 052470 000203 203 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13547 ;BY (013746 000172 000207)
13548
13549 052472 021527 000231 CMP (R5),#231
13550 052476 001372 BNE '$ ;IF IN WRONG SEQUENCE GO TO THE HALT ABOVE
13551 052500 005215 INC (R5)
13552
13553
13554 :*****
13555 :TEST:232 MUL 125252 * #40000 165252 100000 PS 1'
13556 :*****
13557
13558 052502 010767 125464 TST:232: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13559 052506 010501 MOV R5,R1 ;SAVE R5
13560 052510 012704 125252 MOV #125252,%4 ;LOAD MULTIPLICAND WITH 125252
13561 052514 070427 040000 MUL #40000,%4 ;MULTIPLY 125252 * #40000
13562 052520 106737 042430 MFPS @#PSWORD ;SAVE PS
13563 052524 122737 000011 042430 CMPS #11,@#PSWORD ;IS PS = 11
13564 052532 001403 BEQ .+10
13565 052534 004767 006144 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13566 ;PS IS WRONG
13567 052540 000204 204 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13568 ;BY (013746 000172 000207)
13569
13570 052542 022704 165252 CMP #165252,%4 ;IS HIGH ORDER = 165252
13571 052546 001403 BEQ .+10
13572 052550 004767 006130 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13573 ;HIGH ORDER IS WRONG
13574 052554 000205 205 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13575 ;BY (013746 000172 000207)
13576
13577 052556 022705 100000 CMP #100000,%4.1 ;IS LOW ORDER = 100000
13578 052562 001403 BEQ .+10
13579 052564
13580 052564 004767 006114 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13581 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13582 052570 000206 206 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13583 ;BY (013746 000172 000207)
13584
```

13585 052572 021127 000233  
13586 052576 001372  
13587 052600 010105  
13588 052602 005215  
13589  
13590  
13591

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)

13592  
13593  
13594

\*\*\*\*\*  
:TEST:233 MUL 107070 \* #107070 = 31222 26100 PS 1  
\*\*\*\*\*

13595 052604 010767 125362  
13596 052610 012700 107070  
13597 052614 070027 107070  
13598 052620 106737 042430  
13599 052624 122737 000001 042430  
13600 052632 001403  
13601 052634 004767 006044  
13602

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

13603 052640 000207  
13604  
13605

207 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

13606 052642 022700 031222  
13607 052646 001403  
13608 052650 004767 006030  
13609

CMP #31222,%0 ;IS HIGH ORDER = 31222  
BEQ .+10  
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE

13610 052654 000210  
13611  
13612

210 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

13613 052656 022701 026100  
13614 052662 001403  
13615 052664  
13616 052664 004767 006014  
13617

CMP #26100,%0.1 ;IS LOW ORDER = 26100  
BEQ .+10  
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE

13618 052670 000211  
13619  
13620

211 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)

13621 052672 021527 000233  
13622 052676 001372  
13623 052700 005215  
13624  
13625

CMP (R5),#233 ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE  
BNE 1\$  
INC (R5)

13626  
13627  
13628  
13629

\*\*\*\*\*  
:TEST:234 MUL -1 \* #1 = -1 -1 PS = 10  
\*\*\*\*\*

13630 052702 010767 125264  
13631 052706 012701 177777  
13632 052712 070127 000001  
13633 052716 106737 042430  
13634 052722 122737 000010 042430  
13635 052730 001403  
13636 052732 004767 005746  
13637

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

13638 052736 000212  
13639  
13640

212 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)



```

13641 052740 022701 177777      CMP      #-1,%1      ;IS HIGH ORDER = -1
13642 052744 001403      BEQ      .+10
13643 052746 004767 005732      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13644                                ;HIGH ORDER IS WRONG
13645 052752 000213      213      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13646                                ;BY (013746 000172 000207)
13647
13648 052754 022701 177777      CMP      #-1,%1:1    ;IS LOW ORDER - -1
13649 052760 001403      BEQ      .+10
13650                                1$:
13651 052762 004767 005716      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13652                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13653 052766 000214      214      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13654                                ;BY (013746 000172 000207)
13655
13656 052770 021527 000234      CMP      (R5),#234
13657 052774 001372      BNE      1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13658 052776 005215      INC      (R5)
13659
13660
13661                                ;*****
13662                                ;TEST:235      MUL      -1 * #0 = 0 0      PS - 4
13663                                ;*****
13664
13665 053000 010767 125166      TST235: MOV     PC,LPADR      ;STORE ERROR LOOP ADDRESS
13666 053004 012703 177777      MOV     #-1,%3        ;LOAD MULTIPLICAND WITH -1
13667 053010 070327 000000      MUL     #0,%3         ;MULTIPLY -1 * #0
13668 053014 106737 042430      MFPS   @#PSWORD      ;SAVE PS
13669 053020 122737 000004 042430  CMPB   #4,@#PSWORD    ;IS PS = 4
13670 053026 001403      BEQ     .+10
13671 053030 004767 005650      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13672                                ;PS IS WRONG
13673 053034 000215      215     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13674                                ;BY (013746 000172 000207)
13675
13676 053036 022703 000000      CMP     #0,%3         ;IS HIGH ORDER = 0
13677 053042 001403      BEQ     .+10
13678 053044 004767 005634      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13679                                ;HIGH ORDER IS WRONG
13680 053050 000216      216     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13681                                ;BY (013746 000172 000207)
13682
13683 053052 022703 000000      CMP     #0,%3:1      ;IS LOW ORDER - 0
13684 053056 001403      BEQ     .+10
13685 053060                                1$:
13686 053060 004767 005620      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13687                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13688 053064 000217      217     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13689                                ;BY (013746 000172 000207)
13690
13691 053066 021527 000235      CMP     (R5),#235
13692 053072 001372      BNE     1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13693 053074 005215      INC     (R5)
13694
13695
13696                                ;*****

```

```

13697 ;TEST:236 MUL 7777 * #100000 - 100000 100000 PS = 11
13698 ;*****
13699
13700 053076 010767 125070 TST236: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13701 053102 010501 MOV R5,R1 ;SAVE R5
13702 053104 012705 077777 MOV #77777,%5 ;LOAD MULTIPLICAND WITH 77777
13703 053110 070527 100000 MUL #100000,%5 ;MULTIPLY 77777 * #100000
13704 053114 106737 042430 MFPS @#PSWORD ;SAVE PS
13705 053120 122737 000011 042430 CMPSB #11,@#PSWORD ;IS PS = 11
13706 053126 001403 BEQ .+0
13707 053130 004767 005550 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13708 ;PS IS WRONG
13709 053134 000220 220 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13710 ;BY (013746 000172 000207)
13711
13712 053136 022705 100000 CMP #100000,%5 ;IS HIGH ORDER = 100000
13713 053142 001403 BEQ .+10
13714 053144 004767 005534 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13715 ;HIGH ORDER IS WRONG
13716 053150 000221 221 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13717 ;BY (013746 000172 000207)
13718
13719 053152 022705 100000 CMP #100000,%5.1 ;IS LOW ORDER - 100000
13720 053156 001403 BEQ .+10
13721 053160 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13722 053160 004767 005520 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13723 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13724 053164 000222 222 ;BY (013746 000172 000207)
13725
13726
13727 053166 021127 000236 CMP (R1),#236 ;CHECK THE TEST NUMBER
13728 053172 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13729 053174 010105 MOV R1,R5 ;RESTORE R5
13730 053176 005215 INC (R5)
13731
13732
13733 ;*****
13734 ;TEST:237 MUL -1 * #77777 = 100001 100001 PS 10
13735 ;*****
13736
13737 053200 010767 124766 TST237: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13738 053204 012701 177777 MOV #-1,%1 ;LOAD MULTIPLICAND WITH -1
13739 053210 070127 077777 MUL #77777,%1 ;MULTIPLY -1 * #77777
13740 053214 106737 042430 MFPS @#PSWORD ;SAVE PS
13741 053220 122737 000010 042430 CMPSB #10,@#PSWORD ;IS PS = 10
13742 053226 001403 BEQ .+10
13743 053230 004767 005450 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13744 ;PS IS WRONG
13745 053234 000223 223 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13746 ;BY (013746 000172 000207)
13747
13748 053236 022701 100001 CMP #100001,%1 ;IS HIGH ORDER = 100001
13749 053242 001403 BEQ .+10
13750 053244 004767 005434 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13751 ;HIGH ORDER IS WRONG
13752 053250 000224 224 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)

```

```

13753                                     ;BY (013746 000172 000207)
13754
13755 053252 022701 100001             CMP    #100001,%1.1      ;IS LOW ORDER - 100001
13756 053256 001403                   BEQ    .+10
13757 053260                               1$:
13758 053260 004767 005420             JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13759                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13760 053264 000225                   225    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13761                                     ;BY (013746 000172 000207)
13762
13763 053266 021527 000237             CMP    (R5),#237
13764 053272 001372                   BNE    1$                ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13765 053274 005215                   INC    (R5)
13766
13767
13768
13769
13770
13771
13772 053276 010767 124670             TST240: MOV   PC,LPADR      ;STORE ERROR LOOP ADDRESS
13773 053302 012703 077777             MOV   #77777,%3         ;LOAD MULTIPLICAND WITH 77777
13774 053306 070327 077777             MUL   #77777,%3         ;MULTIPLY 77777 * #77777
13775 053312 106737 042430             MFPS  @#PSWORD          ;SAVE PS
13776 053316 122737 000001 042430     CMPB  #1,@#PSWORD       ;IS PS = 1
13777 053324 001403                   BEQ    .+10
13778 053326 004767 005352             JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13779                                     ;PS IS WRONG
13780 053332 000226                   226    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13781                                     ;BY (013746 000172 000207)
13782
13783 053334 022703 000001             CMP    #1,%3            ;IS HIGH ORDER = 1
13784 053340 001403                   BEQ    .+10
13785 053342 004767 005336             JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13786                                     ;HIGH ORDER IS WRONG
13787 053346 000227                   227    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13788                                     ;BY (013746 000172 000207)
13789
13790 053350 022703 000001             CMP    #1,%3:1         ;IS LOW ORDER = 1
13791 053354 001403                   BEQ    .+10
13792 053356                               1$:
13793 053356 004767 005322             JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13794                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13795 053362 000230                   230    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13796                                     ;BY (013746 000172 000207)
13797
13798 053364 021527 000240             CMP    (R5),#240
13799 053370 001372                   BNE    1$                ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13800 053372 005215                   INC    (R5)
13801
13802

```

```

*****
:TEST:240      MUL      77777 * #77777 = 1 1      PS = 1
*****

```

```

*****
:TEST:241      MUL      2 * #2 = 4 4      PS = 0
*****

```

```

13803
13804
13805
13806
13807 053374 010767 124572             TST241: MOV   PC,LPADR      ;STORE ERROR LOOP ADDRESS
13808 053400 010501                   MOV   R5,R1              ;SAVE R5

```



```
13865
13866 053564 022701 100000      CMP      #100000,%0:1      ;IS LOW ORDER = 100000
13867 053570 001403      BEQ      .+10
13868 053572
13869 053572 004767 005106      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13870                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13871 053576 000236      236      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13872                                     ;BY (013746 000172 000207)
13873
13874 053600 021527 000242      CMP      (R5),#242
13875 053604 001372      BNE      1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13876 053606 005215      INC      (R5)
13877
13878
13879
13880
13881
13882
```

```
:*****
:TEST:243      MUL      125252 * @s6 - 165252 100000      PS = 11
:*****
```

```
13883 053610 010767 124356      TST243: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
13884 053614 012700 125252      MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13885 053620 070077 166640      MUL      @s6,%0      ;MULTIPLY 125252 * @s6
13886 053624 106737 042430      MFPS      @#PSWORD      ;SAVE PS
13887 053630 122737 000011 042430      CMPB     #11,@#PSWORD      ;IS PS = 11
13888 053636 001403      BEQ      .+10
13889 053640 004767 005040      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13890                                     ;PS IS WRONG
13891 053644 000237      237      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13892                                     ;BY (013746 000172 000207)
13893
```

```
13894 053646 022700 165252      CMP      #165252,%0      ;IS HIGH ORDER = 165252
13895 053652 001403      BEQ      .+10
13896 053654 004767 005024      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13897                                     ;HIGH ORDER IS WRONG
13898 053660 000240      240      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13899                                     ;BY (013746 000172 000207)
13900
```

```
13901 053662 022701 100000      CMP      #100000,%0:1      ;IS LOW ORDER - 100000
13902 053666 001403      BEQ      .+10
13903 053670
13904 053670 004767 005010      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13905                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13906 053674 000241      241      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13907                                     ;BY (013746 000172 000207)
13908
13909 053676 021527 000243      CMP      (R5),#243
13910 053702 001372      BNE      1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13911 053704 005215      INC      (R5)
13912
13913
```

```
:*****
:TEST:244      MUL      125252 * @#s5 = 165252 100000      PS 1'
:*****
```

```
13914
13915
13916
13917
13918 053706 010767 124260      TST244: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
13919 053712 012700 125252      MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
13920 053716 070037 042462      MUL      @#s5,%0      ;MULTIPLY 125252 * @#s5
```

```

13921 053722 106737 042430 MFPS @#PSWORD ;SAVE PS
13922 053726 122737 000011 042430 CMPB #11,@#PSWORD ;IS PS = 11
13923 053734 001403 BEQ .+10
13924 053736 004767 004742 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13925 ;PS IS WRONG
13926 053742 000242 242 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13927 ;BY (013746 000172 000207)
13928
13929 053744 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
13930 053750 001403 BEQ .+10
13931 053752 004767 004726 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13932 ;HIGH ORDER IS WRONG
13933 053756 000243 243 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13934 ;BY (013746 000172 000207)
13935
13936 053760 022701 100000 CMP #100000,%0!1 ;IS LOW ORDER = 100000
13937 053764 001403 BEQ .+10
13938 053766 JSR
13939 053766 004767 004712 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13940 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13941 053772 000244 244 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13942 ;BY (013746 000172 000207)
13943
13944 053774 021527 000244 CMP (R5),#244
13945 054000 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13946 054002 005215 INC (R5)
13947
13948
13949
13950 :*****
13951 :TEST:245 MUL 125252 * %2 = 165252 100000 PS 11
13952 :*****
13953 054004 010767 124162 TST245: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13954 054010 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
13955 054014 070002 MUL %2,%0 ;MULTIPLY 125252 * %2
13956 054016 106737 042430 MFPS @#PSWORD ;SAVE PS
13957 054022 122737 000011 042430 CMPB #11,@#PSWORD ;IS PS = 11
13958 054030 001403 BEQ .+10
13959 054032 004767 004646 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13960 ;PS IS WRONG
13961 054036 000245 245 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13962 ;BY (013746 000172 000207)
13963
13964 054040 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
13965 054044 001403 BEQ .+10
13966 054046 004767 004632 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13967 ;HIGH ORDER IS WRONG
13968 054052 000246 246 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13969 ;BY (013746 000172 000207)
13970
13971 054054 022701 100000 CMP #100000,%0.1 ;IS LOW ORDER = 100000
13972 054060 001403 BEQ .+10
13973 054062 JSR
13974 054062 004767 004616 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13975 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13976 054066 000247 247 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)

```

```

13977 ;BY (013746 000172 000207)
13978
13979 054070 021527 000245 (MP (R5),#245
13980 054074 001372 (BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13981 054076 005215 (INC (R5)

```

```

13982
13983
13984 :*****
13985 :TEST:246 MUL 125252 * (3)+ = 165252 100000 PS - 11
13986 :*****

```

```

13987
13988 054100 010767 124066 TST246: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13989 054104 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
13990 054110 070023 MUL (3)+,%0 ;MULTIPLY 125252 * (3)+
13991 054112 106737 042430 MFPS @#PSWORD ;SAVE PS
13992 054116 122737 000011 042430 CMPB #11,@#PSWORD ;IS PS = 11
13993 054124 001403 BEQ .+10
13994 054126 004767 004552 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13995 ;PS IS WRONG
13996 054132 000250 250 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13997 ;BY (013746 000172 000207)

```

```

13998
13999 054134 022700 165252 (MP #165252,%0 ;IS HIGH ORDER = 165252
14000 054140 001403 BEQ .+10
14001 054142 004767 004536 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14002 ;HIGH ORDER IS WRONG
14003 054146 000251 251 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14004 ;BY (013746 000172 000207)

```

```

14005
14006 054150 022701 100000 (MP #100000,%0!1 ;IS LOW ORDER - 100000
14007 054154 001403 BEQ .+10
14008 054156 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14009 054156 004767 004522 1$: ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14010 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14011 054162 000252 252 ;BY (013746 000172 000207)
14012

```

```

14013
14014 054164 021527 000246 (MP (R5),#246
14015 054170 001372 (BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14016 054172 005215 (INC (R5)

```

```

14017
14018
14019 :*****
14020 :TEST:247 MUL 125252 * -(3) = 165252 100000 PS 11
14021 :*****

```

```

14022
14023 054174 010767 123772 TST247: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
14024 054200 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
14025 054204 070043 MUL -(3),%0 ;MULTIPLY 125252 * -(3)
14026 054206 106737 042430 MFPS @#PSWORD ;SAVE PS
14027 054212 122737 000011 042430 CMPB #11,@#PSWORD ;IS PS = 11
14028 054220 001403 BEQ .+10
14029 054222 004767 004456 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14030 ;PS IS WRONG
14031 054226 000253 253 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14032 ;BY (013746 000172 000207)

```

```
14033
14034 054230 022700 165252      CMP      #165252,%0      ;IS HIGH ORDER = 165252
14035 054234 001403      BEQ      .+10
14036 054236 004767 004442      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14037                                     ;HIGH ORDER IS WRONG
14038 054242 000254      254      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14039                                     ;BY (013746 000172 000207)
14040
14041 054244 022701 100000      CMP      #100000,%0.1  ;IS LOW ORDER = 100000
14042 054250 001403      BEQ      .+10
14043 054252 004767 004426      1$:      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14044                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14045                                     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14046 054256 000255      255      ;BY (013746 000172 000207)
14047
14048
14049 054260 021527 000247      CMP      (R5),#247
14050 054264 001372      BNE      1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14051 054266 005215      INC      (R5)
14052
14053
14054 ;*****
14055 ;TEST:250      MUL      125252 * 2(4) = 165252 100000      PS 11
14056 ;*****
14057
14058 054270 010767 123676      TST250: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
14059 054274 012700 125252      MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14060 054300 070064 000002      MUL      2(4),%0      ;MULTIPLY 125252 * 2(4)
14061 054304 106737 042430      MFPS     @#PSWORD      ;SAVE PS
14062 054310 122737 000011 042430      CMPB    #11,@#PSWORD    ;IS PS = 11
14063 054316 001403      BEQ      .+10
14064 054320 004767 004360      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14065                                     ;PS IS WRONG
14066 054324 000256      256      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14067                                     ;BY (013746 000172 000207)
14068
14069 054326 022700 165252      CMP      #165252,%0      ;IS HIGH ORDER = 165252
14070 054332 001403      BEQ      .+10
14071 054334 004767 004344      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14072                                     ;HIGH ORDER IS WRONG
14073 054340 000257      257      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14074                                     ;BY (013746 000172 000207)
14075
14076 054342 022701 100000      CMP      #100000,%0!1  ;IS LOW ORDER 100000
14077 054346 001403      BEQ      .+10
14078 054350 004767 004330      1$:      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14079                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14080                                     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14081 054354 000260      260      ;BY (013746 000172 000207)
14082
14083
14084 054356 021527 000250      CMP      (R5),# 50
14085 054362 001372      BNE      1$      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14086 054364 005215      INC      (R5)
14087
14088
```



```
14089 ;*****
14090 ;TEST:251      MUL      125252 * @ (4) = 165252 100000      PS  11
14091 ;*****
14092
14093 054366 010767 123600 TST251: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
14094 054372 012700 125252      MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14095 054376 070074 000000      MUL      @ (4),%0      ;MULTIPLY 125252 * @ (4)
14096 054402 106737 042430      MFPS     @#PSWORD      ;SAVE PS
14097 054406 122737 000011 042430      CMPB    #11,@#PSWORD    ;IS PS = 11
14098 054414 001403      BEQ     .+10
14099 054416 004767 004262      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14100      ;PS IS WRONG
14101 054422 000261      261      ;TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14102      ;BY (013746 000172 000207)
14103
14104 054424 022700 165252      CMP     #165252,%0      ;IS HIGH ORDER = 165252
14105 054430 001403      BEQ     .+10
14106 054432 004767 004246      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14107      ;HIGH ORDER IS WRONG
14108 054436 000262      262      ;TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14109      ;BY (013746 000172 000207)
14110
14111 054440 022701 100000      CMP     #100000,%0!1    ;IS LOW ORDER  100000
14112 054444 001403      BEQ     .+10
14113 054446      1$:
14114 054446 004767 004232      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14115      ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14116 054452 000263      263      ;TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14117      ;BY (013746 000172 000207)
14118
14119 054454 021527 000251      CMP     (R5),#251
14120 054460 001372      BNE     1$
14121 054462 005215      INC     (R5)
14122      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14123
14124 ;*****
14125 ;TEST:252      MUL      125252 * @ (4)+ = 165252 100000      PS  11
14126 ;*****
14127
14128 054464 010767 123502 TST252: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
14129 054470 012700 125252      MOV      #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14130 054474 070034 000000      MUL      @ (4)+,%0      ;MULTIPLY 125252 * @ (4)+
14131 054476 106737 042430      MFPS     @#PSWORD      ;SAVE PS
14132 054502 122737 000011 042430      CMPB    #11,@#PSWORD    ;IS PS = 11
14133 054510 001403      BEQ     .+10
14134 054512 004767 004166      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14135      ;PS IS WRONG
14136 054516 000264      264      ;TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14137      ;BY (013746 000172 000207)
14138
14139 054520 022700 165252      CMP     #165252,%0      ;IS HIGH ORDER = 165252
14140 054524 001403      BEQ     .+10
14141 054526 004767 004152      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14142      ;HIGH ORDER IS WRONG
14143 054532 000265      265      ;TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14144      ;BY (013746 000172 000207)
```

```
14145
14146 054534 022701 100000          CMP      #100000,%0.1      ;IS LOW ORDER = 100000
14147 054540 001403          BEQ      .+10
14148 054542          1$:
14149 054542 004767 004136          JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14150          ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14151 054546 000266          266          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14152          ;BY (013746 000172 000207)
14153
14154 054550 021527 000252          CMP      (R5),#252
14155 054554 001372          BNE     1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14156 054556 005215          INC     (R5)
14157
14158
14159          :*****
14160          :TEST:253      MUL      125252 * @-(4) - 165252 100000      PS      11
14161          :*****
14162
14163 054560 010767 123406          TST253: MOV     PC,LPADR          ;STORE ERROR LOOP ADDRESS
14164 054564 012700 125252          MOV     #125252,%0          ;LOAD MULTIPLICAND WITH 125252
14165 054570 070054          MUL     @-(4),%0          ;MULTIPLY 125252 * @-(4)
14166 054572 106737 042430          MFPS   @#PSWORD          ;SAVE PS
14167 054576 122737 000011 042430          CMPB   #11,@#PSWORD      ;IS PS = 11
14168 054604 001403          BEQ     .+10
14169 054606 004767 004072          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14170          ;PS IS WRONG
14171 054612 000267          267          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14172          ;BY (013746 000172 000207)
14173
14174 054614 022700 165252          CMP     #165252,%0          ;IS HIGH ORDER = 165252
14175 054620 001403          BEQ     .+10
14176 054622 004767 004056          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14177          ;HIGH ORDER IS WRONG
14178 054626 000270          270          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14179          ;BY (013746 000172 000207)
14180
14181 054630 022701 100000          CMP     #100000,%0!1      ;IS LOW ORDER = 100000
14182 054634 001403          BEQ     .+10
14183 054636          1$:
14184 054636 004767 004042          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14185          ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14186 054642 000271          271          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14187          ;BY (013746 000172 000207)
14188
14189 054644 021527 000253          CMP     (R5),#253
14190 054650 001372          BNE     1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14191 054652 005215          INC     (R5)
14192
14193
```

14196  
14195  
14196  
14197  
14198  
14199  
14200  
14201  
14202  
14203  
14204  
14205  
14206  
14207  
14208  
14209  
14210  
14211  
14212  
14213  
14214  
14215  
14216  
14217  
14218  
14219  
14220  
14221  
14222  
14223  
14224  
14225  
14226  
14227  
14228  
14229  
14230  
14231  
14232  
14233  
14234  
14235  
14236  
14237  
14238  
14239  
14240  
14241  
14242  
14243  
14244  
14245  
14246  
14247  
14248  
14249

\*\*\*\*\*  
: DIV INSTRUCTION TESTS  
\*\*\*\*\*

\*\*\*\*\*  
: TEST:254 DIV 0 4 / #2 = 2 REM = 0 PS = 0  
\*\*\*\*\*

```
TST254: 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

14210 054700 122737 000000 042430 CMPB #0,@#PSWORD ;IS PS = 0
14211 054706 001403 BEQ .+10
14212 054710 004767 003770 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;PS IS WRONG
14214 054714 000272 272 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

14218 054716 022700 000002 CMP #2,%0 ;IS QUOTIENT - 2
14219 054722 001403 BEQ .+10
14220 054724 004767 003754 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;QUOTIENT IS WRONG
14222 054730 000273 273 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

14226 054732 022701 000000 CMP #0,%0+1 ;IS REMAINDER - 0
14227 054736 001403 BEQ .+10
14228 054740 004767 003740 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;WRONG REMAINDER
14230 054744 000274 274 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

14233 054746 021527 000254 CMP (R5),#254
14234 054752 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
14235 054754 004767 003724 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
14237 054760 000275 275 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

14240 054762 005215 INC (R5)
```

\*\*\*\*\*  
: 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
```

14250	055004	106737	042430		MFPS	@#PSWORD	:SAVE PS
14251							
14252	055010	122737	000010	042430	CMPB	#10,@#PSWORD	:IS PS = 10
14253	055016	001403			BEQ	+.10	
14254	055020	004767	003660		JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
14255							:PS IS WRONG
14256	055024	000276			276		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14257							:BY (013746 000172 000207)
14258							
14259							
14260	055026	022702	77775		CMP	#-3,%2	:IS QUOTIENT - -3
14261	055032	001403			BEQ	+.10	
14262	055034	004767	003644		JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
14263							:QUOTIENT IS WRONG
14264	055040	000277			277		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14265							:BY (013746 000172 000207)
14266							
14267							
14268	055042	022703	000000		CMP	#0,%2+1	:IS REMAINDER = 0
14269	055046	001403			BEQ	+.10	
14270	055050	004767	003630		JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
14271							:WRONG REMAINDER
14272	055054	000300			300		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14273							:BY (013746 000172 000207)
14274							
14275	055056	021527	000255		CMP	(R5),#255	
14276	055062	001403			BEQ	+.10	:IF IN WRONG SEQUENCE GO TO THE HLT
14277	055064	004767	003614		JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
14278							:TEST IS IN WRONG SEQUENCE
14279	055070	000301			301		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14280							:BY (013746 000172 000207)
14281							
14282	055072	005215			INC	(R5)	
14283							
14284							
14285							
14286							
14287							
14288	055074	010767	123072		TST256: MOV	PC,LPADR	:STORE ERROR LOOP ADDRESS
14289	055100	010501			MOV	R5,R1	:SAVE R5
14290	055102	012704	000000		MOV	#0,%4	:LOAD HIGH ORDER WITH 0
14291	055106	012705	000011		MOV	#9,%4+1	:LOAD LOW ORDER WITH 9.
14292	055112	071427	000002		MOV	#2,%4	:DIVIDE BY #2
14293	055116	106737	042430		MFPS	@#PSWORD	:SAVE PS
14294							
14295	055122	122737	000000	042430	CMPB	#0,@#PSWORD	:IS PS = 0
14296	055130	001403			BEQ	+.10	
14297	055132	004767	003546		JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	
14298							:PS IS WRONG
14299	055136	000302			302		:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14300							:BY (013746 000172 000207)
14301							
14302							
14303	055140	022704	000004		CMP	#4,%4	:IS QUOTIENT = 4
14304	055144	001403			BEQ	+.10	
14305	055146	004767	003532		JSR	PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE	

.....  
:TEST:256 DIV 0 9. / #2 = 4 REM 1 PS - 0  
:.....

```
14306  
14307 055152 000303 303 ;QUOTIENT IS WRONG  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
14308 ;BY (013746 000172 000207)  
14309  
14310  
14311 055154 022705 000001 CMP #1,%+1 ;IS REMAINDER = 1  
14312 055160 001403 BEQ .+10  
14313 055162 004767 003516 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
14314 ;WRONG REMAINDER  
14315 055166 000304 304 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
14316 ;BY (013746 000172 000207)  
14317  
14318 055170 010105 MOV R1,R5 ;RESTORE R5  
14319 055172 021527 000256 CMP (R5),#256  
14320 055176 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT  
14321 055200 004767 003500 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
14322 ;TEST IS IN WRONG SEQUENCE  
14323 055204 000305 305 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
14324 ;BY (013746 000172 000207)  
14325  
14326 055206 005215 INC (R5)  
14327  
14328  
14329 ;*****  
14330 ;TEST:257 DIV -1 -9. / #2 = -4 REM = -1 PS = 10  
14331 ;*****  
14332 055210 010767 122756 TST257: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
14333 055214 01700 177777 MOV #-1,%0 ;LOAD HIGH ORDER WITH -1  
14334 055220 012701 177767 MOV #-9,%0+1 ;LOAD LOW ORDER WITH -9.  
14335 055224 071027 000002 DIV #2,%0 ;DIVIDE BY #2  
14336 055230 106737 042430 MFPS @#PSWORD ;SAVE PS  
14337  
14338 055234 122737 000010 042430 CMPB #10,@#PSWORD ;IS PS = 10  
14339 055242 001403 BEQ .+10  
14340 055244 004767 003434 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
14341 ;PS IS WRONG  
14342 055250 000306 306 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
14343 ;BY (013746 000172 000207)  
14344  
14345  
14346 055252 022700 177774 CMP #-4,%0 ;IS QUOTIENT = -4  
14347 055256 001403 BEQ .+10  
14348 055260 004767 003420 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
14349 ;QUOTIENT IS WRONG  
14350 055264 000307 307 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
14351 ;BY (013746 000172 000207)  
14352  
14353  
14354 055266 022701 177777 CMP #-1,%0+1 ;IS REMAINDER = -1  
14355 055272 001403 BEQ .+10  
14356 055274 004767 003404 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
14357 ;WRONG REMAINDER  
14358 055300 000310 310 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
14359 ;BY (013746 000172 000207)  
14360  
14361 055302 021527 000257 CMP (R5),#257
```

```

14362 055306 001403          BEQ    .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
14363 055310 004767 003370  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14364                                ;TEST IS IN WRONG SEQUENCE
14365 05-314 000311          311    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14366                                ;BY (013746 000172 000207)
14367
14368 055316 005215          INC    (R5)
14369

```

```

:*****
:TEST:260      DIV      0 2 / #-3 - 0      REM = 2      PS  4
:*****

```

```

14374 055320 010767 122646  TST260: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
14375 055324 012702 000000      MOV    #0,%2          ;LOAD HIGH ORDER WITH 0
14376 055330 012703 000002      MOV    #2,%2+1       ;LOAD LOW ORDER WITH 2
14377 055334 071227 177775      DIV    #-3,%2        ;DIVIDE BY #-3
14378 055340 106737 042430      MFPS   @#PSWORD      ;SAVE PS
14379
14380 055344 122737 000004 042430 (MPB  #4,@#PSWORD    ;IS PS = 4
14381 055352 001403          BEQ    .+10
14382 055354 004767 003324  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14383                                ;PS IS WRONG
14384 055360 000312          312    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14385                                ;BY (013746 000172 000207)
14386

```

```

14387
14388 055362 022702 000000      CMP    #0,%2          ;IS QUOTIENT = 0
14389 055366 001403          BEQ    .+10
14390 055370 004767 003310  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14391                                ;QUOTIENT IS WRONG
14392 055374 000313          313    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14393                                ;BY (013746 000172 000207)
14394

```

```

14395
14396 055376 022703 000002      CMP    #2,%2+1       ;IS REMAINDER = 2
14397 055402 001403          BEQ    .+10
14398 055404 004767 003274  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14399                                ;WRONG REMAINDER
14400 055410 000314          314    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14401                                ;BY (013746 000172 000207)
14402

```

```

14403 055412 021527 000260      (MP   (R5),#260
14404 055416 001403          BEQ    .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
14405 055420 004767 003260  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14406                                ;TEST IS IN WRONG SEQUENCE
14407 055424 000315          315    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14408                                ;BY (013746 000172 000207)
14409

```

```

14410 055426 005215          INC    (R5)
14411

```

```

:*****
:TEST:261      DIV      -1 -2 / #3 = 0      REM = -2      PS - 4
:*****

```

```

14416 055430 010767 122536  TST261: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
14417 055434 010501          MOV    R5,R1         ;SAVE R5

```







```
14530
14531 055746 021527 C00263      CMP      (R5),#263
14532 055752 001403      BEQ      .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT
14533 055754 004767 002724      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14534                                ;TEST IS IN WRONG SEQUENCE
14535 055760 000331      331      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14536                                ;BY (013746 000172 000207)
14537
14538 055762 005215      INC      (R5)
14539
14540 :*****
14541 :TEST:264      DIV      -1 125252 / #2 = 152525      REM - 0      PS      10
14542 :*****
14543
14544 055764 010767 122202      TST264: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
14545 055770 012702 177777      MOV      #-1,%2      ;LOAD HIGH ORDER WITH -1
14546 055774 012703 125252      MOV      #125252,%2+1      ;LOAD LOW ORDER WITH 125252
14547 056000 071227 000002      DIV      #2,%2      ;DIVIDE BY #2
14548 056004 106737 042430      MFPS     @#PSWORD      ;SAVE PS
14549
14550 056010 122737 000010 042430      CMPB     #10,@#PSWORD      ;IS PS = 10
14551 056016 001403      BEQ      .+10
14552 056020 004767 002660      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14553                                ;PS IS WRONG
14554 056024 000332      332      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14555                                ;BY (013746 000172 000207)
14556
14557
14558 056026 022702 152525      CMP      #152525,%2      ;IS QUOTIENT = 152525
14559 056032 001403      BEQ      .+10
14560 056034 004767 002644      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14561                                ;QUOTIENT IS WRONG
14562 056040 000333      333      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14563                                ;BY (013746 000172 000207)
14564
14565
14566 056042 022703 000000      CMP      #0,%2+1      ;IS REMAINDER = 0
14567 056046 001403      BEQ      .+10
14568 056050 004767 002630      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14569                                ;WRONG REMAINDER
14570 056054 000334      334      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14571                                ;BY (013746 000172 000207)
14572
14573 056056 021527 000264      CMP      (R5),#264
14574 056062 001403      BEQ      .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT
14575 056064 004767 002614      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14576                                ;TEST IS IN WRONG SEQUENCE
14577 056070 000335      335      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14578                                ;BY (013746 000172 000207)
14579
14580 056072 005215      INC      (R5)
14581
14582 :*****
14583 :TEST:265      DIV      -1 -1 / #-1 = 1      REM - 0      PS      0
14584 :*****
14585
```



```
14642
14643
14644 056252 022700 100000      CMP      #100000,%0      ;IS QUOTIENT = 100000
14645 056256 001403      BEQ      .+10
14646 056260 004767 002420      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14647                                     ;QUOTIENT IS WRONG
14648 056264 000343      343      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14649                                     ;BY (013746 000172 000207)
14650
14651
14652 056266 022701 000001      CMP      #1,%0+1      ;IS REMAINDER = 1
14653 056272 001403      BEQ      .+10
14654 056274 004767 002404      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14655                                     ;WRONG REMAINDER
14656 056300 000344      344      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14657                                     ;BY (013746 000172 000207)
14658
14659 056302 021527 000266      CMP      (R5),#266
14660 056306 001403      BEQ      .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT
14661 056310 004767 002370      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14662                                     ;TEST IS IN WRONG SEQUENCE
14663 056314 000345      345      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14664                                     ;BY (013746 000172 000207)
14665
14666 056316 005215      INC      (R5)
14667
14668
14669 :*****
14670 ;TEST:267      DIV      37777 77777 / #77777 = 77777      REM - 77776      PS = 0
14671 :*****
14672 056320 010767 121646      TST267: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
14673 056324 012702 037777      MOV      #37777,%2      ;LOAD HIGH ORDER WITH 37777
14674 056330 012703 077777      MOV      #77777,%2+1      ;LOAD LOW ORDER WITH 77777
14675 056334 071227 077777      DIV      #77777,%2      ;DIVIDE BY #77777
14676 056340 106737 042430      MFPS      @#PSWORD      ;SAVE PS
14677
14678 056344 122737 000000 042430      CMPB     #0,@#PSWORD      ;IS PS = 0
14679 056352 001403      BEQ      .+10
14680 056354 004767 002324      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14681                                     ;PS IS WRONG
14682 056360 000346      346      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14683                                     ;BY (013746 000172 000207)
14684
14685
14686 056362 022702 077777      CMP      #77777,%2      ;IS QUOTIENT = 77777
14687 056366 001403      BEQ      .+10
14688 056370 004767 002310      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14689                                     ;QUOTIENT IS WRONG
14690 056374 000347      347      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14691                                     ;BY (013746 000172 000207)
14692
14693
14694 056376 022703 077776      CMP      #77776,%2+1      ;IS REMAINDER = 77776
14695 056402 001403      BEQ      .+10
14696 056404 004767 002274      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14697                                     ;WRONG REMAINDER
```

```
14698 056410 000350          350          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14699                          ;BY (013746 000172 000207)
14700
14701 056412 021527 000267      CMP      (R5),#267
14702 056416 001403          BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
14703 056420 004767 002260      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14704                          ;TEST IS IN WRONG SEQUENCE
14705 056424 000351          351          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14706                          ;BY (013746 000172 000207)
14707
14708 056426 005215          INC      (R5)
14709
14710
14711                          :*****
14712                          ;TEST:270      DIV      0 100000 / #2 = 40000      REM      0      PS      0
14713                          :*****
14714 056430 010767 121536      TST270: MOV    PC,LPADR          ;STORE ERROR LOOP ADDRESS
14715 056434 010501          MOV    R5,R1          ;SAVE R5
14716 056436 012704 000000      MOV    #0,%4          ;LOAD HIGH ORDER WITH 0
14717 056442 012705 100000      MOV    #100000,%4+1    ;LOAD LOW ORDER WITH 100000
14718 056446 071427 000002      DIV    #2,%4          ;DIVIDE BY #2
14719 056452 106737 042430      MFPS   @#PSWORD        ;SAVE PS
14720
14721 056456 122737 000000 042430  CMPB   #0,@#PSWORD      ;IS PS = 0
14722 056464 001403          BEQ    .+10
14723 056466 004767 002212      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14724                          ;PS IS WRONG
14725 056472 000352          352          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14726                          ;BY (013746 000172 000207)
14727
14728
14729 056474 022704 040000          CMP    #40000,%4      ;IS QUOTIENT = 40000
14730 056500 001403          BEQ    .+10
14731 056502 004767 002176      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14732                          ;QUOTIENT IS WRONG
14733 056506 000353          353          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14734                          ;BY (013746 000172 000207)
14735
14736
14737 056510 022705 000000          CMP    #0,%4+1        ;IS REMAINDER = 0
14738 056514 001403          BEQ    .+10
14739 056516 004767 002162      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14740                          ;WRONG REMAINDER
14741 056522 000354          354          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14742                          ;BY (013746 000172 000207)
14743
14744 056524 010105          MOV    R1,R5          ;RESTORE R5
14745 056526 021527 000270      CMP    (R5),#270
14746 056532 001403          BEQ    .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
14747 056534 004767 002144      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14748                          ;TEST IS IN WRONG SEQUENCE
14749 056540 000355          355          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14750                          ;BY (013746 000172 000207)
14751
14752 056542 005215          INC    (R5)
14753
```

```
14754 :*****
14755 :TEST:271 DIV 177777 77777 / #177776 = 40000 REM - 177777 PS
14756 :*****
14757
14758 056544 010767 121422 TST271: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
14759 056550 012700 177777 MOV #177777,%0 ;LOAD HIGH ORDER WITH 177777
14760 056554 012701 077777 MOV #77777,%0+1 ;LOAD LOW ORDER WITH 77777
14761 056560 071027 177776 DIV #177776,%0 ;DIVIDE BY #177776
14762 056564 106737 042430 MFPS @#PSWORD ;SAVE PS
14763
14764 056570 122737 000000 042430 CMPB #0,@#PSWORD ;IS PS = 0
14765 056576 001403 BEQ .+10
14766 056600 004767 002100 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14767 ;PS IS WRONG
14768 056604 000356 356 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14769 ;BY (013746 000172 000207)
14770
14771
14772 056606 022700 040000 CMP #40000,%0 ;IS QUOTIENT - 40000
14773 056612 001403 BEQ .+10
14774 056614 004767 002064 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14775 ;QUOTIENT IS WRONG
14776 056620 000357 357 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14777 ;BY (013746 000172 000207)
14778
14779
14780 056622 022701 177777 CMP #177777,%0+1 ;IS REMAINDER = 177777
14781 056626 001403 BEQ .+10
14782 056630 004767 002050 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14783 ;WRONG REMAINDER
14784 056634 000360 360 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14785 ;BY (013746 000172 000207)
14786
14787 056636 021527 000271 CMP (R5),#271
14788 056642 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
14789 056644 004767 002034 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14790 ;TEST IS IN WRONG SEQUENCE
14791 056650 000361 361 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14792 ;BY (013746 000172 000207)
14793
14794 056652 005215 INC (R5)
14795
14796 :*****
14797 :TEST:272 DIV 0 52525 / #52525 - 1 REM 0 PS 0
14798 :*****
14799
14800 056654 010767 121312 TST272: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
14801 056660 012702 000000 MOV #0,%2 ;LOAD HIGH ORDER WITH 0
14802 056664 012703 052525 MOV #52525,%2+1 ;LOAD LOW ORDER WITH 52525
14803 056670 071227 052525 DIV #52525,%2 ;DIVIDE BY #52525
14804 056674 106737 042430 MFPS @#PSWORD ;SAVE PS
14805
14806 056700 122737 000000 042430 CMPB #0,@#PSWORD ;IS PS - 0
14807 056706 001403 BEQ .+10
14808 056710 004767 001770 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14809 ;PS IS WRONG
```



14866 057054 005215  
14867  
14868  
14869  
14870  
14871  
14872 057056 010767 121110  
14873 057062 012700 077777  
14874 057066 012701 177777  
14875 057072 071027 000002  
14876 057076 106737 042430  
14877 057102 042737 000014 042430  
14878  
14879 057110 122737 000002 042430  
14880 057116 001403  
14881 057120 004767 001560  
14882  
14883 057124 000370  
14884  
14885  
14886  
14887 057126 021527 000274  
14888 057132 001403  
14889 057134 004767 001544  
14890  
14891 057140 000371  
14892  
14893  
14894 057142 005215  
14895  
14896 057144 012702 000002  
14897 057150 012703 042472  
14898 057154 012704 042474  
14899  
14900  
14901  
14902  
14903  
14904 057160 010767 121006  
14905 057164 012700 000000  
14906 057170 012701 052525  
14907 057174 071067 163272  
14908 057200 106737 042430  
14909  
14910 057204 122737 000000 042430  
14911 057212 001403  
14912 057214 004767 001464  
14913  
14914 057220 000372  
14915  
14916  
14917  
14918 057222 022700 025252  
14919 057226 001403  
14920 057230 004767 001450  
14921

INC (R5)  
:\*\*\*\*\*  
: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)  
370  
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)  
371  
INC (R5)  
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  
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
372  
CMP #25252,%0 ;IS QUOTIENT = 25252  
BEQ .+10  
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;QUOTIENT IS WRONG

```

14922 057234 000373          373          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14923                               :BY (013746 000172 000207)
14924
14925
14926 057236 022701 000001      CMP      #1,%0+1          :IS REMAINDER = 1
14927 057242 001403          BEQ      .+10
14928 057244 004767 001434      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14929                               :WRONG REMAINDER
14930 057250 000374          374          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14931                               :BY (013746 000172 000207)
14932
14933 057252 021527 000275      CMP      (R5),#275
14934 057256 001403          BEQ      .+10          :IF IN WRONG SEQUENCE GO TO THE HLT
14935 057260 004767 001420      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14936                               :TEST IS IN WRONG SEQUENCE
14937 057264 000375          375          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14938                               :BY (013746 000172 000207)
14939
14940 057266 005215          INC      (R5)
14941
14942
14943
14944
14945
14946 057270 010767 120676      TST276: MOV      PC,LPADR          :STORE ERROR LOOP ADDRESS
14947 057274 012700 000000      MOV      #0,%0          :LOAD HIGH ORDER WITH 0
14948 057300 012701 052525      MOV      #52525,%0+1    :LOAD LOW ORDER WITH 52525
14949 057304 071077 163164      DIV      @S10,%0        :DIVIDE BY @S10
14950 057310 106737 042430      MFPS     @#PSWORD       :SAVE PS
14951
14952 057314 122737 000000 042430  CMPB     #0,@#PSWORD     :IS PS = 0
14953 057322 001403          BEQ      .+10
14954 057324 004767 001354      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14955                               :PS IS WRONG
14956 057330 000376          376          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14957                               :BY (013746 000172 000207)
14958
14959
14960 057332 022700 025252      CMP      #25252,%0      :IS QUOTIENT - 25252
14961 057336 001403          BEQ      .+10
14962 057340 004767 001340      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14963                               :QUOTIENT IS WRONG
14964 057344 000377          377          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14965                               :BY (013746 000172 000207)
14966
14967
14968 057346 022701 000001      CMP      #1,%0+1          :IS REMAINDER - 1
14969 057352 001403          BEQ      .+10
14970 057354 004767 001324      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14971                               :WRONG REMAINDER
14972 057360 000400          400          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14973                               :BY (013746 000172 000207)
14974
14975 057362 021527 000276      CMP      (R5),#276
14976 057366 001403          BEQ      .+10          :IF IN WRONG SEQUENCE GO TO THE HLT
14977 057370 004767 001310      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE

```

```

:*****
:TEST:276      DIV      0 52525 / @S10 = 25252      REM - 1      PS 0
:*****

```



```
14978
14979 057374 000401          401          ;TEST IS IN WRONG SEQUENCE
14980                                     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14981                                     ;BY (013746 000172 000207)
14982 057376 005215          INC      (R5)
14983
14984 :*****
14985 :TEST:277      DIV      0 52525 / @#S9 = 25252      REM 1      PS 0
14986 :*****
14987
14988 057400 010767 120566      TST277: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
14989 057404 012700 000000      MOV      #0,%0              ;LOAD HIGH ORDER WITH 0
14990 057410 012701 052525      MOV      #52525,%0+1        ;LOAD LOW ORDER WITH 52525
14991 057414 071037 042472      DIV      @#S9,%0            ;DIVIDE BY @#S9
14992 057420 106737 042430      MFPS     @#PSWORD           ;SAVE PS
14993
14994 057424 122737 000000 042430  CMPB     #0,@#PSWORD         ;IS PS = 0
14995 057432 001403          BEQ      .+10
14996 057434 004767 001244          JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14997                                     ;PS IS WRONG
14998 057440 070402          402          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14999                                     ;BY (013746 000172 000207)
15000
15001
15002 057442 022700 025252          CMP      #25252,%0          ;IS QUOTIENT = 25252
15003 057446 001403          BEQ      .+10
15004 057450 004767 001230          JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15005                                     ;QUOTIENT IS WRONG
15006 057454 000403          403          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15007                                     ;BY (013746 000172 000207)
15008
15009
15010 057456 022701 000001          CMP      #1,%0+1           ;IS REMAINDER = 1
15011 057462 001403          BEQ      .+10
15012 057464 004767 001214          JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15013                                     ;WRONG REMAINDER
15014 057470 000404          404          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15015                                     ;BY (013746 000172 000207)
15016
15017 057472 021527 000277          CMP      (R5),#277
15018 057476 001403          BEQ      .+10              ;IF IN WRONG SEQUENCE GO TO THE HLT
15019 057500 004767 001200          JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15020                                     ;TEST IS IN WRONG SEQUENCE
15021 057504 000405          405          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15022                                     ;BY (013746 000172 000207)
15023
15024 057506 005215          INC      (R5)
15025
15026 :*****
15027 :TEST:300      DIV      0 52525 / %2 = 25252      REM = 1      PS 0
15028 :*****
15029
15030 057510 010767 120456      TST300: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
15031 057514 012700 000000      MOV      #0,%0              ;LOAD HIGH ORDER WITH 0
15032 057520 012701 052525      MOV      #52525,%0+1        ;LOAD LOW ORDER WITH 52525
15033 057524 071002          DIV      %2,%0              ;DIVIDE BY %2
```

```

15034 057526 106737 042430 MFPS @#PSWORD ;SAVE PS
15035
15036 057532 122737 000000 042430 CMPB #0,@#PSWORD ;IS PS = 0
15037 057540 001403 BEQ .+10
15038 057542 004767 001136 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15039 ;PS IS WRONG
15040 057546 000406 406 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15041 ;BY (013746 000172 000207)
15042
15043
15044 057550 022700 025252 CMP #25252,%0 ;IS QUOTIENT - 25252
15045 057554 001403 BEQ .+10
15046 057556 004767 001122 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15047 ;QUOTIENT IS WRONG
15048 057562 000407 407 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15049 ;BY (013746 000172 000207)
15050
15051
15052 057564 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
15053 057570 001403 BEQ .+10
15054 057572 004767 001106 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15055 ;WRONG REMAINDER
15056 057576 000410 410 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15057 ;BY (013746 000172 000207)
15058
15059 057600 021527 000300 CMP (R5),#300
15060 057604 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
15061 057606 004767 001072 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15062 ;TEST IS IN WRONG SEQUENCE
15063 057612 000411 411 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15064 ;BY (013746 000172 000207)
15065
15066 057614 005215 INC (R5)
15067
15068 ;*****
15069 ;TEST:301 DIV 0 52525 / (3)+ = 25252 REM - 1 PS 0
15070 ;*****
15071
15072 057616 010767 120350 TST301: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
15073 057622 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
15074 057626 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
15075 057632 071023 DIV (3)+,%0 ;DIVIDE BY (3)+
15076 057634 106737 042430 MFPS @#PSWORD ;SAVE PS
15077
15078 057640 122737 000000 042430 CMPB #0,@#PSWORD ;IS PS = 0
15079 057646 001403 BEQ .+10
15080 057650 004767 001030 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15081 ;PS IS WRONG
15082 057654 000412 412 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15083 ;BY (013746 000172 000207)
15084
15085
15086 057656 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
15087 057662 001403 BEQ .+10
15088 057664 004767 001014 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15089 ;QUOTIENT IS WRONG

```

```

15090 057670 000413          413          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15091                                     :BY (013746 000172 000207)
15092
15093
15094 057672 022701 000001    CMP      #1,%0+1          :IS REMAINDER 1
15095 057676 001403          BEQ      .+10
15096 057700 004767 001000    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15097                                     :WRONG REMAINDER
15098 057704 000414          414          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15099                                     :BY (013746 000172 000207)
15100
15101 057706 021527 000301    CMP      (R5),#301
15102 057712 001403          BEQ      .+10          :IF IN WRONG SEQUENCE GO TO THE HLT
15103 057714 004767 000764    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15104                                     :TEST IS IN WRONG SEQUENCE
15105 057720 000415          415          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15106                                     :BY (013746 000172 000207)
15107
15108 057722 005215          INC      (R5)
15109
15110 :*****
15111 :TEST:302 DIV 0 52525 / -(3) = 25252 REM = 1 PS = 0
15112 :*****
15113
15114 057724 010767 120242    TST302: MOV    PC,LPADR          :STORE ERROR LOOP ADDRESS
15115 057730 012700 000000    MOV      #0,%0          :LOAD HIGH ORDER WITH 0
15116 057734 012701 052525    MOV      #52525,%0+1    :LOAD LOW ORDER WITH 52525
15117 057740 071043          DIV      -(3),%0        :DIVIDE BY -(3)
15118 057742 106737 042430    MFPS    @#PSWORD        :SAVE PS
15119
15120 057746 122737 000000 042430 CMPB     #0,@#PSWORD     :IS PS = 0
15121 057754 001403          BEQ      .+10
15122 057756 004767 000722    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15123                                     :PS IS WRONG
15124 057762 000416          416          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15125                                     :BY (013746 000172 000207)
15126
15127
15128 057764 022700 025252    CMP      #25252,%0      :IS QUOTIENT = 25252
15129 057770 001403          BEQ      .+10
15130 057772 004767 000706    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15131                                     :QUOTIENT IS WRONG
15132 057776 000417          417          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15133                                     :BY (013746 000172 000207)
15134
15135
15136 060000 022701 000001    CMP      #1,%0+1          :IS REMAINDER = 1
15137 060004 001403          BEQ      .+10
15138 060006 004767 000672    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15139                                     :WRONG REMAINDER
15140 060012 000420          420          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15141                                     :BY (013746 000172 000207)
15142
15143 060014 021527 000302    CMP      (R5),#302
15144 060020 001403          BEQ      .+10          :IF IN WRONG SEQUENCE GO TO THE HLT
15145 060022 004767 000656    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE

```

```

15146                                     :TEST IS IN WRONG SEQUENCE
15147 060026 000421                       421 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15148                                     ;BY (013746 000172 000207)
15149
15150 060030 005215                       INC (R5)
15151
15152 :*****
15153 ;TEST:303 DIV 0 52525 / 2(4) = 25252 REM - 1 PS 0
15154 :*****
15155
15156 060032 010767 120134 TST303: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
15157 060036 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
15158 060042 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
15159 060046 071064 000002 DIV 2(4),%0 ;DIVIDE BY 2(4)
15160 060052 106737 042430 MFPS @#PSWORD ;SAVE PS
15161
15162 060056 122737 000000 042430 CMPB #0,@#PSWORD ;IS PS = 0
15163 060064 001403 BEQ .+10
15164 060066 004767 000612 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15165 ;PS IS WRONG
15166 060072 000422 422 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15167 ;BY (013746 000172 000207)
15168
15169
15170 060074 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
15171 060100 001403 BEQ .+10
15172 060102 004767 000576 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15173 ;QUOTIENT IS WRONG
15174 060106 000423 423 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15175 ;BY (013746 000172 000207)
15176
15177
15178 060110 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
15179 060114 001403 BEQ .+10
15180 060116 004767 000562 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15181 ;WRONG REMAINDER
15182 060122 000424 424 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15183 ;BY (013746 000172 000207)
15184
15185 060124 021527 000303 CMP (R5),#303
15186 060130 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
15187 060132 004767 000546 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15188 ;TEST IS IN WRONG SEQUENCE
15189 060136 000425 425 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15190 ;BY (013746 000172 000207)
15191
15192 060140 005215 INC (R5)
15193
15194 :*****
15195 ;TEST:304 DIV 0 52525 / @ (4) = 25252 REM = 1 PS C
15196 :*****
15197
15198 060142 010767 120024 TST304: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
15199 060146 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
15200 060152 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
15201 060156 071074 000000 DIV @ (4),%0 ;DIVIDE BY @ (4)

```

```
15202 060162 106737 042430 MFPS @#PSWORD ;SAVE PS
15203
15204 060166 122737 000000 042430 CMPB #0,@#PSWORD ;IS PS = 0
15205 060174 001403 BEQ .+10
15206 060176 004767 000502 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15207 ;PS IS WRONG
15208 060202 000426 426 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15209 ;BY (013746 000172 000207)
15210
15211
15212 060204 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
15213 060210 001403 BEQ .+10
15214 060212 004767 000466 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15215 ;QUOTIENT IS WRONG
15216 060216 000427 427 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15217 ;BY (013746 000172 000207)
15218
15219
15220 060220 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
15221 060224 001403 BEQ .+10
15222 060226 004767 000452 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15223 ;WRONG REMAINDER
15224 060232 000430 430 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15225 ;BY (013746 000172 000207)
15226
15227 060234 021527 000304 CMP (R5),#304
15228 060240 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
15229 060242 004767 000436 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15230 ;TEST IS IN WRONG SEQUENCE
15231 060246 000431 431 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15232 ;BY (013746 000172 000207)
15233
15234 060250 005215 INC (R5)
15235
15236 ;*****
15237 ;TEST:305 DIV 0 52525 / @ (4)+ = 25252 RFM 1 PS - 0
15238 ;*****
15239
15240 060252 010767 117714 TST305: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
15241 060256 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
15242 060262 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
15243 060266 071034 DIV @ (4)+,%0 ;DIVIDE BY @ (4)+
15244 060270 106737 042430 MFPS @#PSWORD ;SAVE PS
15245
15246 060274 122737 000000 042430 CMPB #0,@#PSWORD ;IS PS = 0
15247 060302 001403 BEQ .+10
15248 060304 004767 000374 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15249 ;PS IS WRONG
15250 060310 000432 432 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15251 ;BY (013746 000172 000207)
15252
15253
15254 060312 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
15255 060316 001403 BEQ .+10
15256 060320 004767 000360 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15257 ;QUOTIENT IS WRONG
```

```
15258 060324 000433          433          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15259                               ;BY (013746 000172 000207)
15260
15261
15262 060326 022701 000001      CMP      #1,%0+1          ;IS REMAINDER = 1
15263 060332 001403          BEQ      .+10
15264 060334 004767 000344      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15265                               ;WRONG REMAINDER
15266 060340 000434          434          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15267                               ;BY (013746 000172 000207)
15268
15269 060342 021527 000305      CMP      (R5),#305
15270 060346 001403          BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
15271 060350 004767 000330      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15272                               ;TEST IS IN WRONG SEQUENCE
15273 060354 000435          435          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15274                               ;BY (013746 000172 000207)
15275
15276 060356 005215          INC      (R5)
15277
15278 ;*****
15279 ;TEST:306      DIV      0 52525 / @-(4) = 25252      REM - 1      PS      0
15280 ;*****
15281
15282 060360 010767 117606      TST306: MOV      PC,LPADR          ;STORE ERROR LOOP ADDRESS
15283 060364 012700 000000      MOV      #0,%0          ;LOAD HIGH ORDER WITH 0
15284 060370 012701 052525      MOV      #52525,%0+1    ;LOAD LOW ORDER WITH 52525
15285 060374 071054          DIV      @-(4),%0        ;DIVIDE BY @-(4)
15286 060376 106737 042430      MFPS     @#PSWORD        ;SAVE PS
15287
15288 060402 122737 000000 042430  CMPB     #0,@#PSWORD      ;IS PS = 0
15289 060410 001403          BEQ      .+10
15290 060412 004767 000266      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15291                               ;PS IS WRONG
15292 060416 000436          436          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15293                               ;BY (013746 000172 000207)
15294
15295
15296 060420 022700 025252      CMP      #25252,%0        ;IS QUOTIENT = 25252
15297 060424 001403          BEQ      .+10
15298 060426 004767 000252      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15299                               ;QUOTIENT IS WRONG
15300 060432 000437          437          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15301                               ;BY (013746 000172 000207)
15302
15303
15304 060434 022701 000001      CMP      #1,%0+1          ;IS REMAINDER = 1
15305 060440 001403          BEQ      .+10
15306 060442 004767 000236      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15307                               ;WRONG REMAINDER
15308 060446 000440          440          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15309                               ;BY (013746 000172 000207)
15310
15311 060450 021527 000306      CMP      (R5),#306
15312 060454 001403          BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
15313 060456 004767 000222      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
```

15314  
15315 060462 000441  
15316  
15317  
15318 060464 005215  
15319

441  
  
INC (R5)

:TEST IS IN WRONG SEQUENCE  
:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
:BY (013746 000172 000207)

```

15320
15321
15322
15323
15324
15325
15326
15327
15328 060466 005227 177777
15329 060472 001002
15330 060474 104401 060642
15331 060500 005267 117602
15332 060504 042767 100000 117574
15333 060512 005327
15334 060514 000001
15335 060516 003037
15336 060520 012737
15337 060522 000016
15338 060524 060514
15339 060526 104401 060625
15340 060532 016746 117550
15341 060536 104405
15342 060540 104401 060622
15343 060544 013700 000042
15344 060550 001405
15345 060552 000005
15346 060554 004710
15347 060556 000240
15348 060560 000240
15349 060562 000240
15350 060564 013746 000004
15351 060570 012737 060606 000004
15352 060576 012737 000001 164000
15353 060604 000402
15354 060606 062706 000004
15355 060612 012637 000004
15356 060616 000137
15357 060620 001024
15358 060622 377 377 000
15359 060625 015 042412 042116
15360 060632 050040 051501 020123
15361 060640 000043
15362 060642 005015 045103 042113
15363 060650 042102 020060 041504
15364 060656 030506 026461 040501
15365 060664 041440 052520 042040
15366 060672 040511 047107 051517
15367 060700 044524 000103
15368

```

```

.SBTTL END OF PASS ROUTINE
:*****
:*INCREMENT THE PASS NUMBER ($PASS) IN APT MAILBOX
:*TYPE 'END PASS #XXXXX' (WHERE XXXXX IS A DECIMAL NUMBER)
:*IF THERE IS A MONITOR GO TO IT
:*IF THERE ISN'T JUMP TO RESTART
:*****
$EOP: INC #-1 ;TYPE ID ONLY ON FIRST PASS
      BNE SKPMSG ;BRANCH AROUND AFTER FIRST PASS
      TYPE ,MSG1 ;TYPE TITLE
SKPMSG: INC $PASS ;INCREMENT THE PASS NUMBER
        BIC #100000,$PASS ;DON'T LET COUNT GO NEGATIVE
        DEC (PC)+ ;LOOP?
$EOPCT: .WORD 1
        BGT $DOAGN ;YES
        MOV (PC)+,@(PC)+ ;RESTORE LOOP COUNTER
$ENDCT: .WORD 16
        $EOPCT
        TYPE ,SENDMG ;TYPE 'END PASS #'
        MOV $PASS,-(SP) ;SAVE PASS COUNT FOR TYPEOUT
        TYPDS ;GO TYPE PASS COUNT IN DECIMAL
        TYPE ,SENULL ;TYPE A FEW NULL CHARACTERS
$GET42: MOV @#42,RO ;GET MONITOR ADDRESS
        BEQ DOAGIN ;BRANCH IF NO MONITOR
        RESET ;CLEAR THE WORLD
$ENDAD: JSR PC,(RO) ;GO TO MONITOR
        NOP ;SAVE ROOM
        NOP ;FOR
        NOP ;ACT11
DOAGIN: MOV @#4,-(SP) ;SAVE CONTENTS OF LOCATION 4
        MOV #1$,@#4 ;SET UP IN CASE OF TRAP
        MOV #1,@#164000 ;NOTIFY MULTI-TESTER WE'RE DONE
        BR 2$ ;NO TRAP SO DON'T RESET STACK
1$: ADD #4,SP ;RESET STACK AFTER TRAP
2$: MOV (SP)+,@#4 ;RESTORE LOCATION 4 FOR TESTING
$DOAGN: JMP @(PC)+ ;RETURN TO TEST
$RTNAD: .WORD RESTR
$ENULL: .BYTE -1,-1,0 ;NULL CHARACTER STRING
$SENDMG: .ASCIZ <15><12>/END PASS #/
MSG1: .ASCIZ <15><12>/CJKDBD0 DCF11-AA CPU DIAGNOSTIC/
.EVEN

```



```
15369 .SBTTL HALT ROUTINE
15370
15371 ;* HALT ROUTINE
15372 ;* -----
15373 ;*
15374 ;*
15375 ;* PROGRAM COMES HERE ON ENCOUNTERING ANY ERROR
15376 ;*
15377
15378 060704 017637 000000 000302 $HLT: MOV @($P),@($FATAL ;PLACE THE ERROR NUMBER AT LOCATION $FATAL
15379 060712 011637 061072 MOV ($P),@($CONTIN ;SAVE ERROR NUMBER ADDRESS
15380 060716 032777 020000 146530 BIT #20000,@($SWR ;HAS THE OPERATOR ASKED TO SUPRESS ERROR TYPE OUTS
15381 060724 001021 BNE 6$
15382 060726 104401 042506 TYPE ,($CRLF ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15383 060732 104401 061032 TYPE ,MSGERR ;INFORM ERROR
15384 060736 104401 042506 TYPE ,($CRLF ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15385 060742 013746 061072 MOV @($CONTIN,-($P) ;RETRIVE ERROR NUMBER ADDR
15386 060746 162716 000004 SUB #4,($P) ;CALCULATE ERROR PC
15387 060752 104402 TYPCC ;TYPE PC
15388 060754 104401 042506 TYPE ,($CRLF ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15389 060760 013746 000302 MOV @($FATAL,-($P) ;RETRIVE ERROR NUMBER
15390 060764 104403 TYPOS ;TYPE ERROR NUMBER
15391 060766 003 .BYTE 3
15392 060767 000 .BYTE 0
15393 060770 105767 117324 6$: TST $ENV ;IF WE ARE NOT UNDER APT. THEN GO TO
15394 060774 001403 BEQ 8$ ;8$
15395 060776 005237 000300 INC @($MSGTY ;OTHERWISE INFORM APT. ABOUT SEEING THE ERROR
15396 061002 000777 BR ;AND LOOP
15397 061004 104401 042506 8$: TYPE ,($CRLF ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15398 061010 005777 146440 TST @($SWR ;IS IT REQUIRED TO HALT ON ERROR ?
15399 061014 100001 BPL 10$ ;IF NOT THEN GO TO 10$
15400 061016 000000 HALT
15401 061020 013746 061072 10$: MOV @($CONTIN,-($P) ;
15402 061024 062716 000002 ADD #2,($P) ;CALCULATE RETURN ADDRESS
15403 061030 000207 RTS PC ;RETURN
15404 061032 042440 051122 051117 MSGERR: .ASCIZ / ERROR! PC, AND ERROR # ARE: /
15405 061040 020041 020040 041520
15406 061046 020054 047101 020104
15407 061054 051105 047522 020122
15408 061062 020043 051101 035105
15409 061070 000040
15410 .EVEN
15411 061072 000000 CONTIN: .WORD 0
15412
15413 .SBTTL POWER FAIL ROUTINE
15414 061074 012767 061104 116722 PWRDN: MOV #PWRUP,24
15415 061102 000000 HALT
15416
15417 061104 012767 061074 116712 PWRUP: MOV #PWRDN,24
15418 061112 012706 001000 MOV #BUFF,$P
15419 061116 132767 000040 117175 BITB #40,$ENVM ;WILL APT ALLOW PRINTING?
15420 061124 001013 BNE PFRES ;NO
15421 061126 012700 061160 MOV #MSGPWF,$R0 ;GET MSG ADDR.
15422 061132 105737 177564 PWAIT: TSTB @($TPS ;TTY READY
15423 061136 100375 BPL PWAIT ;NO WAIT
15424 061140 112037 177566 MOVB ($R0)+,@($TPB ;PRINT CHARACTER
```

15425 061144 001372  
 15426 061146 105737 177564  
 15427 061152 100375  
 15428 061154 000167 117644  
 15429 061160 005015 047520  
 15430 061166 020122 040506  
 15431 061174 042105 000041

PWAIT1: TSTB @WTPS ;NEXT IF NOT DONE.  
 BNE PWAIT  
 BPL PWAIT1  
 PFRES: JMP RESTRT  
 MSGPWF: .ASCIZ <15><12>.POWER FAILED..

.EVEN  
 .SBTTL TYPE ROUTINE

15432  
15433  
15434  
15435  
15436  
15437  
15438  
15439  
15440  
15441  
15442  
15443  
15444  
15445  
15446  
15447  
15448  
15449

\*\*\*\*\*  
 ;\*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  
 ;\*

15450 061200 105767 000337  
 15451 061204 100002  
 15452 061206 000000  
 15453 061210 000430  
 15454 061212 010046  
 15455 061214 017600 000002  
 15456 061220 122767 000001 117072  
 15457 061226 001011  
 15458 061230 132767 000100 117063  
 15459 061236 001405  
 15460 061240 010067 000004  
 15461 061244 004767 000306  
 15462 061250 000000  
 15463 061252 132767 000040 117041  
 15464 061260 001003  
 15465 061262 112046  
 15466 061264 001005  
 15467 061266 005726  
 15468 061270 012600  
 15469 061272 062716 000002  
 15470 061276 000002  
 15471 061300 122716 000011  
 15472 061304 001430  
 15473 061306 122716 000200  
 15474 061312 001006  
 15475 061314 005726  
 15476 061316 104401  
 15477 061320 042506  
 15478 061322 105067 000202  
 15479 061326 000755  
 15480 061330 004767 000056

\$TYPE: TSTB \$TPFLG ;; IS THERE A TERMINAL?  
 BPL 1\$ ;; BR IF YES  
 BR HALT ;; HALT HERE IF NO TERMINAL  
 BR 3\$ ;; LEAVE  
 1\$: MOV RO,-(SP) ;; SAVE RO  
 MOV @2(SP),RO ;; 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 RO,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 (RO)+,-(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)+,RO ;; RESTORE RO  
 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

```

15481 061334 126726 000207      6$:  CMPB  $FILLC,(SP)+  ;; IS IT TIME FOR FILLER CHARS.?
15482 061340 001350              BNE  2$                ;; IF NO GO GET NEXT CHAR.
15483 061342 016746 000172      MOV  $NULL,-(SP)      ;; GET # OF FILLER CHARS. NEEDED.
15484                                ;; AND THE NULL CHAR.
15485 061346 105366 000001      7$:  DECB  1(SP)        ;; DOES A NULL NEED TO BE TYPED?
15486 061352 002770              BLT  6$                ;; BR IF NO--GO POP THE NULL OFF OF STACK
15487 061354 004767 000032      JSR  PC,$TYPEPC      ;; GO TYPE A NULL
15488 061360 105367 000144      DECB  $CHARCNT       ;; DO NOT COUNT AS A COUNT
15489 061364 000770              BR   7$                ;; LOOP
15490
15491                                ;HORIZONTAL TAB PROCESSOR
15492
15493 061366 112716 000040      8$:  MOVB  #' ,(SP)    ;; REPLACE TAB WITH SPACE
15494 061372 004767 000014      9$:  JSR  PC,$TYPEPC   ;; TYPE A SPACE
15495 061376 132767 000007 000124 BITB  #7,$CHARCNT    ;; BRANCH IF NOT AT
15496 061404 001372              BNE  9$                ;; TAB STOP
15497 061406 005726              TST  (SP)+            ;; POP SPACE OFF STACK
15498 061410 000724              BR   2$                ;; GET NEXT CHARACTER
15499 061412
15500 061412 105777 000116      $TYPEPC: TSTB @ $TKS      ;; CHAR IN KYBD BUFFER? ;MJD001
15501 061416 100022              BPL  10$              ;; BR IF NOT ;MJD001
15502 061420 017746 000112      MOV  @ $TKB,-(SP)    ;; GET CHAR ;MJD001
15503 061424 042716 177600      BIC  #177600,(SP)   ;; STRIP EXTRANEIOUS BITS ;MJD001
15504 061430 122716 000023      CMPB # $XOFF,(SP)   ;; WAS CHAR XOFF ;MJD001
15505 061434 001012              BNE  102$             ;; BR IF NOT ;MJD001
15506 061436
15507 061436 105777 000072      101$: TSTB @ $TKS      ;; WAIT FOR CHAR ;MJD001
15508 061442 100375              BPL  101$            ;; ;MJD001
15509 061444 117716 000066      MOVB @ $TKB,(SP)    ;; GET CHAR ;MJD001
15510 061450 042716 177600      BIC  #177600,(SP)   ;; STRIP IT ;MJD001
15511 061454 122716 000021      CMPB # $XON,(SP)    ;; WAS IT XON? ;MJD001
15512 061460 001366              BNE  101$            ;; BR IF NOT ;MJD001
15513 061462
15514 061462 005726              102$: TST  (SP)+      ;; FIX STACK ;MJD001
15515 061464
15516 061464 105777 161014      10$:  TSTB @ $TPS      ;; WAIT UNTIL PRINTER IS READY ;MJD001
15517 061470 100375              BPL  10$              ;; ;MJD001
15518 061472 116677 000002 161002 MOVB  2(SP),@ $TPB   ;; LOAD CHAR TO BE TYPED INTO DATA REG.
15519 061500 122766 000015 000002 CMPB  #CR,2(SP)      ;; IS CHARACTER A CARRIAGE RETURN?
15520 061506 001003              BNE  1$                ;; BRANCH IF NO
15521 061510 105067 000014      CLRB $CHARCNT       ;; YES--CLEAR CHARACTER COUNT
15522 061514 000406              BR   $TYPEX           ;; EXIT
15523 061516 122766 000012 000002 1$:  CMPB  #LF,2(SP)    ;; IS CHARACTER A LINE FEED?
15524 061524 001402              BEQ  $TYPEX           ;; BRANCH IF YES
15525 061526 105227              INCB (PC)+           ;; COUNT THE CHARACTER
15526 061530 000000      $CHARCNT: .WORD 0   ;; CHARACTER COUNT STORAGE
15527 061532 000207      $TYPEX: RTS  PC
15528
15529 061534 177560      $TKS: .WORD 177560  ;; TTY KDB STATUS ;MJD001
15530 061536 177562      $TKB: .WORD 177562  ;; TTY KDB BUFFER ;MJD001
15531 061540 000          $NULL: .BYTE 0      ;; CONTAINS NULL CHARACTER FOR FILLS
15532 061541 002          $FILLS: .BYTE 2     ;; CONTAINS # OF FILLER CHARACTERS REQUIRED
15533 061542 012          $FILLC: .BYTE 12   ;; INSERT FILL CHARS. AFTER A 'LINE FEED'
15534 061543 000          $TPFLG: .BYTE 0    ;; 'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
15535 061544 077          $QUES: .ASCII '??' ;; QUESTION MARK
15536 061545 012 000      $LF: .ASCII <12>  ;; LINEFEED

```

```
15537          061550          .EVEN
15538          .SBTTL  APT COMMUNICATIONS ROUTINE
15539
15540          ;*****
15541 061550 112767 000001 000236 $ATY1:  MOVB  #1,$FFLG      ;;TO REPORT FATAL ERROR
15542 061556 112767 000001 000226 $ATY3:  MOVB  #1,$MFLG      ;;TO TYPE A MESSAGE
15543 061564 000403          BR          $ATYC
15544 061566 112767 000001 000220 $ATY4:  MOVB  #1,$FFLG      ;;TO ONLY REPORT FATAL FROR
15545 061574          $ATYC:
15546 061574 010046          MOV  R0,-(SP)      ;;PUSH R0 ON STACK
15547 061576 010146          MOV  R1,-(SP)      ;;PUSH R1 ON STACK
15548 061600 105767 000206          TSTB  $MFLG      ;;SHOULD TYPE A MESSAGE?
15549 061604 001450          BEQ   5$          ;;IF NOT: BR
15550 061606 122767 000001 116504          CMPB  #APTENV,$ENV      ;;OPERATING UNDER APT?
15551 061614 001031          BNE   3$          ;;IF NOT: BR
15552 061616 132767 000100 116475          BITB  #APTSPOOL,$ENVM  ;;SHOULD SPOOL MESSAGES?
15553 061624 001425          BEQ   3$          ;;IF NOT: BR
15554 061626 017600 000004          MOV  @4(SP),R0      ;;GET MESSAGE ADDR.
15555 061632 062766 000007 000004          ADD  #2,4(SP)      ;;BUMP RETURN ADDR.
15556 061640 005767 116434          1$:  TST  $MSGTYPE      ;;SEE IF DONE W/ LAST XMISSION?
15557 061644 001375          BNE   1$          ;;IF NOT: WAIT
15558 061646 010067 116442          MOV  R0,$MSGAD      ;;PUT ADDR IN MAILBOX
15559 061652 105720          2$:  TSTB  (R0)+      ;;FIND END OF MESSAGE
15560 061654 001376          BNE   2$
15561 061656 166700 116432          SUB  $MSGAD,R0      ;;SUB START OF MESSAGE
15562 061662 006200          ASR  R0            ;;GET MESSAGE LGTH IN WORDS
15563 061664 010067 116426          MOV  R0,$MSGGLGT    ;;PUT LENGTH IN MAILBOX
15564 061670 012767 000004 116402          MOV  #4,$MSGTYPE    ;;TELL APT TO TAKE MSG.
15565 061676 000413          BR   5$
15566 061700 017667 000004 000016 3$:  MOV  @4(SP),4$      ;;PUT MSG ADDR IN JSR LINKAGE
15567 061706 062766 000002 000004          ADD  #2,4(SP)      ;;BUMP RETURN ADDRESS
15568 061714 016746 116056          MOV  177776,-(SP)  ;;PUSH 177776 ON STACK
15569 061720 004767 177254          JSR  PC,$TYPE      ;;CALL TYPE MACRO
15570 061724 000000          4$:  .WORD  0
15571 061726          5$:
15572 061726 105767 000062          10$: TSTB  $FFLG      ;;SHOULD REPORT FATAL ERROR?
15573 061732 001416          BEQ   12$          ;;IF NOT: BR
15574 061734 005767 116360          TST  $ENV          ;;RUNNING UNDER APT?
15575 061740 001413          BEQ   12$          ;;IF NOT: BR
15576 061742 005767 116332          11$: TST  $MSGTYPE      ;;FINISHED LAST MESSAGE?
15577 061746 001375          BNE   11$          ;;IF NOT: WAIT
15578 061750 017667 000004 116324          MOV  @4(SP),$FATAL  ;;GET ERROR #
15579 061756 062766 000002 000004          ADD  #2,4(SP)      ;;BUMP RETURN ADDR.
15580 061764 005267 116310          INC  $MSGTYPE      ;;TELL APT TO TAKE ERROR
15581 061770 105067 000020          12$: CLRB  $FFLG      ;;CLEAR FATAL FLAG
15582 061774 105067 000013          CLRB  $LFLG      ;;CLEAR LOG FLAG
15583 062000 105067 000006          CLRB  $MFLG      ;;CLEAR MESSAGE FLAG
15584 062004 012601          MOV  (SP)+,R1      ;;POP STACK INTO R1
15585 062006 012600          MOV  (SP)+,R0      ;;POP STACK INTO R0
15586 062010 000207          RTS   PC          ;;RETURN
15587 062012          000          $MFLG: .BYTE  0      ;;MESSG. FLAG
15588 062013          000          $LFLG: .BYTE  0      ;;LOG FLAG
15589 062014          000          $FFLG: .BYTE  0      ;;FATAL FLAG
15590          062016          .EVEN
15591          000200          APTSIZE=200
15592          000001          APTENV=001
```

```

15593      000100      APTSPool-100
15594      000040      APTCSUP=040
15595                                     .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
15596
15597
15598 .....
15599 *THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
15600 *SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
15601 *NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
15602 *BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
15603 *REPLACED WITH SPACES.
15604 *CALL:
15605 *      MOV      NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
15606 *      TYPDS      ;;GO TO THE ROUTINE
15607
15608 $TYPDS:
15609      MOV      R0,-(SP)      ;;PUSH R0 ON STACK
15610      MOV      R1,-(SP)      ;;PUSH R1 ON STACK
15611      MOV      R2,-(SP)      ;;PUSH R2 ON STACK
15612      MOV      R3,-(SP)      ;;PUSH R3 ON STACK
15613      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
15614      MOV      #20200,-(SP)  ;;SET BLANK SWITCH AND SIGN
15615      MOV      20(SP),R5     ;;GET THE INPUT NUMBER
15616      BPL      R5           ;;BR IF INPUT IS POS.
15617      NEG      R5           ;;MAKE THE BINARY NUMBER POS.
15618      MOV      #'-,1(SP)    ;;MAKE THE ASCII NUMBER NEG.
15619      CLR      R0           ;;ZERO THE CONSTANTS INDEX
15620      MOV      #SDBLK,R3    ;;SETUP THE OUTPUT POINTER
15621      MOV      #' ,(R3)+    ;;SET THE FIRST CHARACTER TO A BLANK
15622      CLR      R2           ;;CLEAR THE BCD NUMBER
15623      MOV      $DTBL(R0),R1  ;;GET THE CONSTANT
15624      SUB      R1,R5       ;;FORM THIS BCD DIGIT
15625      BLT      4$          ;;BR IF DONE
15626      INC      R2         ;;INCREASE THE BCD DIGIT BY 1
15627      BR      3$
15628      ADD      R1,R5       ;;ADD BACK THE CONSTANT
15629      TST      R2         ;;CHECK IF BCD DIGIT 0
15630      BNE      5$         ;;FALL THROUGH IF 0
15631      TSTB   (SP)        ;;STILL DOING LEADING 0'S?
15632      BMI      7$         ;;BR IF YES
15633      ASLB   (SP)        ;;MSD?
15634      BCC      6$         ;;BR IF NO
15635      MOV      1(SP),-1(R3)  ;;YES--SET THE SIGN
15636      BIS      #'0,R2      ;;MAKE THE BCD DIGIT ASCII
15637      BIS      #' ,R2     ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
15638      MOV      R2,(R3)+    ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
15639      TST      (R0)+      ;;JUST INCREMENTING
15640      CMP      R0,#10     ;;CHECK THE TABLE INDEX
15641      RLT      2$         ;;GO DO THE NEXT DIGIT
15642      BGT      8$         ;;GO TO EXIT
15643      MOV      R5,R2      ;;GET THE LSD
15644      BR      6$         ;;GO CHANGE TO ASCII
15645      TSTB   (SP)+      ;;WAS THE LSD THE FIRST NON-ZERO?
15646      SPL      9$         ;;BR IF NO
15647      MOV      -1(SP),-2(R3)  ;;YES--SET THE SIGN FOR TYPING
15648      CLRB   (R3)        ;;SET THE TERMINATOR
15649      MOV      (SP)+,R5    ;;POP STACK INTO R5

```

```

15649 062174 012603      MOV      (SP)+,R3      ;;POP STACK INTO R3
15650 062176 012602      MOV      (SP)+,R2      ;;POP STACK INTO R2
15651 062200 012601      MOV      (SP)+,R1      ;;POP STACK INTO R1
15652 062202 012600      MOV      (SP)+,R0      ;;POP STACK INTO R0
15653 062204 104401 062232      TYPE      $DBLK        ;;NOW TYPE THE NUMBER
15654 062210 016666 000002 0G0004      MOV      2(SP),4(SP)   ;;ADJUST THE STACK
15655 062216 012616      MOV      (SP)+,(SP)
15656 062220 0C0002      RTI                          ;;RETURN TO USER
15657 062222 023420      $DTBL: 10000.
15658 062224 001750      1000.
15659 062226 000144      100.
15660 062230 000012      10.
15661 062232 00C004      $DBLK: .BLKW 4
15662      .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
15663
15664      ;*****
15665      ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
15666      ;*OCTAL (ASCII) NUMBER AND TYPE IT.
15667      ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
15668      ;*CALL:
15669      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
15670      ;*      TYPOS      ;;CALL FOR TYPEOUT
15671      ;*      .BYTE  N      ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
15672      ;*      .BYTE  M      ;;M=1 OR 0
15673      ;*      ;;1=TYPE LEADING ZEROS
15674      ;*      ;;0=SUPPRESS LEADING ZEROS
15675      ;*
15676      ;*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
15677      ;*$TYPOS OR $TYPOC
15678      ;*CALL:
15679      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
15680      ;*      TYPON      ;;CALL FOR TYPEOUT
15681      ;*
15682      ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
15683      ;*CALL:
15684      ;*      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
15685      ;*      TYPOC      ;;CALL FOR TYPEOUT
15686      ;*
15687 062242 017646 000000      $TYPOS: MOV      @ (SP),-(SP)      ;;PICKUP THE MODE
15688 062246 116667 000001 000211      MOVVB   1(SP),$OFILL      ;;LOAD ZERO FILL SWITCH
15689 062254 112667 000207      MOVVB   (SP)+,$OMODE+1    ;;NUMBER OF DIGITS TO TYPE
15690 062260 062716 000002      ADD     #2,(SP)          ;;ADJUST RETURN ADDRESS
15691 062264 000406      BR      $TYPON
15692 062266 112767 000001 000171      $TYPOC: MOVVB   #1,$OFILL      ;;SET THE ZERO FILL SWITCH
15693 062274 112767 000006 000165      MOVVB   #6,$OMODE+1      ;;SET FOR SIX(6) DIGITS
15694 062302 112767 000005 000154      $TYPON: MOVVB   #5,$OCNT      ;;SET THE ITERATION COUNT
15695 062310 010346      MOV     R3,-(SP)        ;;SAVE R3
15696 062312 010446      MOV     R4,-(SP)        ;;SAVE R4
15697 062314 010546      MOV     R5,-(SP)        ;;SAVE R5
15698 062316 116704 000145      MOVVB   $OMODE+1,R4      ;;GET THE NUMBER OF DIGITS TO TYPE
15699 062322 005404      NEG     R4
15700 062324 062704 000006      ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
15701 062330 110467 000132      MOVVB   R4,$OMODE        ;;SAVE IT FOR USE
15702 062334 116704 000125      MOVVB   $OFILL,R4        ;;GET THE ZERO FILL SWITCH
15703 062340 016605 000012      MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
15704 062344 005003      CLR     R3              ;;CLEAR THE OUTPUT WORD

```

```

15705 062346 006105      1$:  ROL    R5          ;;ROTATE MSB INTO 'C'
15706 062350 000404      BR     3$             ;;GO DO MSB
15707 062352 006105      2$:  ROL    R5          ;;FORM THIS DIGIT
15708 062354 006105      ROL    R5
15709 062356 006105      ROL    R5
15710 062360 010503      MOV    R5,R3
15711 062362 006103      3$:  ROL    R3          ;;GET LSB OF THIS DIGIT
15712 062364 105367 000076  DECB   $OMODE        ;;TYPE THIS DIGIT?
15713 062370 100016      BPL    7$            ;;BR IF NO
15714 062372 042703 177770  BIC    #177770,R3    ;;GET RID OF JUNK
15715 062376 001002      BNE    4$            ;;TEST FOR 0
15716 062400 005704      TST    R4            ;;SUPPRESS THIS 0?
15717 062402 001403      BEQ    5$            ;;BR IF YES
15718 062404 005204      4$:  INC    R4          ;;DON'T SUPPRESS ANYMORE 0'S
15719 062406 052703 000060  BIS    #'0,R3        ;;MAKE THIS DIGIT ASCII
15720 062412 052703 000040  5$:  BIS    #' ,R3     ;;MAKE ASCII IF NOT ALREADY
15721 062416 110367 000040  MOV    R3,8$         ;;SAVE FOR TYPING
15722 062422 104401 062462  TYPE   8$            ;;GO TYPE THIS DIGIT
15723 062426 105367 000032  7$:  DECB   $OCNT      ;;COUNT BY 1
15724 062432 003347      BGT    2$            ;;BR IF MORE TO DO
15725 062434 002402      BLT    6$            ;;BR IF DONE
15726 062436 005204      INC    R4            ;;INSURE LAST DIGIT ISN'T A BLANK
15727 062440 000744      BR     2$            ;;GO DO THE LAST DIGIT
15728 062442 012605      6$:  MOV    (SP)+,R5    ;;RESTORE R5
15729 062444 012604      MOV    (SP)+,R4    ;;RESTORE R4
15730 062446 012603      MOV    (SP)+,R3    ;;RESTORE R3
15731 062450 016666 000002 000004  MOV    2(SP),4(SP)  ;;SET THE STACK FOR RETURNING
15732 062456 012616      MOV    (SP)+,(SP)
15733 062460 000002      RTI                    ;;RETURN
15734 062462 000          8$:  .BYTE  0            ;;STORAGE FOR ASCII DIGIT
15735 062463 000          .BYTE  0            ;;TERMINATOR FOR TYPE ROUTINE
15736 062464 000          $OCNT: .BYTE  0      ;;OCTAL DIGIT COUNTER
15737 062465 000          $OFILL: .BYTE  0    ;;ZERO FILL SWITCH
15738 062466 000000      $OMODE: .WORD  0    ;;NUMBER OF DIGITS TO TYPE
15739 .SBTTL TRAP DECODER
15740
15741 .....
15742 ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE 'TRAP' INSTRUCTION
15743 ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
15744 ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
15745 ;*GO TO THAT ROUTINE.
15746
15747 062470 010046      $TRAP: MOV    R0,-(SP)  ;;SAVE R0
15748 062472 016600 000002  MOV    2(SP),R0      ;;GET TRAP ADDRESS
15749 062476 005740      TST    -(R0)         ;;BACKUP BY 2
15750 062500 111000      MOV    (R0),R0      ;;GET RIGHT BYTE OF TRAP
15751 062502 006300      ASL    R0            ;;POSITION FOR INDEXING
15752 062504 016000 062524  MOV    $TRPAD(R0),R0 ;;INDEX TO TABLE
15753 062510 000200      RTS    R0            ;;GO TO ROUTINE
15754
15755
15756 ;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO
15757
15758 062512 011646      $TRAP2: MOV   (SP),-(SP) ;;MOVE THE PC DOWN
15759 062514 016666 000004 000002  MOV   4(SP),2(SP)    ;;MOVE THE PSW DOWN
15760 062522 000002      RTI                    ;;RESTORE THE PSW

```

15761  
15762  
15763  
15764  
15765  
15766  
15767  
15768  
15769 062524 062512  
15770 062526 061200  
15771 062530 062266  
15772 062532 062242  
15773 062534 062302  
15774 062536 062016  
15775  
15776  
15777 062540 177777  
15778 000001

.SBTTL TRAP TABLE

::\*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED  
::\*BY THE "TRAP" INSTRUCTION.

	ROUTINE		
\$TRPAD:	.WORD	\$TRAP2	
	\$TYPE	::CALL=TYPE	TRAP+1(104401) TTY TYPEOUT ROUTINE
	\$TYPOC	::CALL=TYPOC	TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)
	\$TYPOS	::CALL=TYPOS	TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)
	\$TYPON	::CALL=TYPON	TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)
	\$TYPDS	::CALL=TYPDS	TRAP+5(104405) TYPE DECIMAL NUMBER (WITH SIGN)

PASSPT: -1  
.END







BRMFPD	027120	8473	8477#											
BRN1	003230	1256	1262#											
BRN2	003240	1257	1267#											
BRN3	003250	1269	1275#											
BRTAB	027214	7937	8503#											
BRV1	003300	1298	1304#											
BRV2	003310	1299	1309#											
BRV3	003320	1311	1317#											
BRZ1	003160	1209	1215#											
BRZ2	003170	1210	1220#											
BRZ3	003200	1222	1228#											
BR1	027552	8664	8673#											
BR10	027766	8752	8761#											
BR11	030072	8785	8794#											
BR12	030140	8800	8809#											
BR13	030206	8815	8824#											
BR14	030254	8830	8839#											
BR15	030352	8861	8870#											
BR16	030372	8871	8880#											
BR17	030412	8881	8890#											
BR2	027600	8676	8685#											
BR20	030432	8891	8899#											
BR21	030452	8900	8909#											
BR22	030472	8910	8919#											
BR23	030512	8920	8929#											
BR3	027622	8688	8697#											
BR33	030562	8949	8957#											
BR34	030574	8958	8966#											
BR35	030606	8967	8975#											
BR36	030620	8976	8984#											
BR37	030642	8987	8995#											
BR4	027646	8701	8710#											
BR40	030654	8996	9004#											
BR41	030666	9005	9013#											
BR45	032214	9418	9424#											
BR46	032762	9524	9622#											
BR46A	032772	9617	9626#											
BR47	033552	9825	9831#											
BR5	027672	8714	8723#											
BR51	034320	9933	10031#											
BR51A	034330	10026	10035#											
BR6	027716	8727	8735#											
BR7	027742	8739	8748#											
BR70	037306	10798	10813#											
BR71	040236	10995	11007	11016#										
B' CON	025624	8047	8052#											
B'TERR	025610	8042	8048#											
BTRAP	037644	10906	10924#											
BUFF =	001000	8626#	9029	9043	9046	9062	9065	9081	9086	9095	9100	9116	9165	9227
		9240	9243	9259	9262	9278	9283	9292	9296	9312	9361	9422	9439	9452
		9455	9471	9474	9490	9495	9504	9509	9525	9574	9634	9647	9650	9666
		9669	9685	9690	9699	9704	9720	9769	9829	9847	9860	9863	9879	9882
		9898	9903	9913	9918	9934	9983	10050	10063	10066	10082	10085	10101	10106
		10115	10120	10136	10185	10242	10255	10258	10274	10277	10294	10299	10308	10313
		10329	10378	10650	10666	10675	10691	10697	10716	10741	10771	10800	10825	10831
		10866	10888	10907	10934	10950	11003	11036	11097	11126	11141	11171	11195	11216



DNMB2E	011352	3892	3901#						
DNMB2F	011370	3903	3911#						
DNMB3A	011452	3937	3938	3939	3945#				
DNMB3B	011462	3940	3949#						
DNMB3C	011500	3950	3958#						
DNMB3D	011516	3961	3962	3968#					
DNMB3E	011526	3963	3972#						
DNMB4A	011716	4040	4041	4042	4048#				
DNMB4B	011726	4043	4052#						
DNMB4C	011744	4053	4061#						
DNMB4D	011754	4062	4068#						
DNMB4E	011764	4063	4072#						
DNMB4F	012000	4073	4081#						
DNM03A	010136	3366	3367	3368	3374#				
DNM03B	010146	3369	3378#						
DNM03C	010156	3379	3386#						
DNM1	010010	3300	3308#						
DNM1A	011122	3790	3791	3792	3798#				
DNM1B	011132	3793	3802#						
DNM2	C10024	3309	3317#						
DNM2A	011200	3823	3824	3830#					
DNM2B	011210	3825	3834#						
DNM2C	011216	3841#							
DNM2D	011226	3836	3845#						
DNM3	010042	3319	3327#						
DNM4	010064	3331	3339#						
DNM4A	011610	3996	3997	3998	4004#				
DNM4B	011620	3999	4008#						
DNM4C	011634	4009	4017#						
DNM5A	012060	4104	4105	4106	4112#				
DNM5B	012070	4107	4116#						
DNM5C	012106	4117	4125#						
DNM6A	012166	4147	4148	4149	4155#				
DNM6B	012176	4150	4159#						
DNM6C	012214	4160	4168#						
DNM7A	012276	4190	4191	4192	4198#				
DNM7B	012306	4193	4202#						
DNM7C	012324	4203	4211#						
DOAGIN	060564	15344	15350#						
DOPB2A	010550	3610	3618#						
DOPB2B	010626	3648	3656#						
DOP0A	007552	3184	3192#						
DOP0B	007576	3197	3205#						
DOP0C	007616	3208	3216#						
DOP0D	007646	3223	3231#						
DOP03A	007724	3261	3262	3263	3269#				
DOP03B	007734	3264	3273#						
DOP1	010362	3502	3510#						
DOP2	010474	3570	3578#						
DOP4	014000	4781	4793#						
DOP5	014064	4819	4831#						
DTRAP1	037626	10910#	10934						
DUMMY =	000000	11566#	12533	12567	12602	12637	12671	12706	12741
END1	027204	8484	8497#						
ENT176	047022	12511	12518#						
ENT51	044362	11938	11942#						









MDM6A	013452	4646	4647	4653#		
MDM6B	013462	4648	4657#			
MDM6C	01350C	4658	4666#			
MDM6D	013520	4667	4675#			
MDM6E	013550	4678	4686#			
MDM7A	013624	4714	4715	4721#		
MDM7B	013634	4716	4725#			
MDM7C	013652	4726	4734#			
MDM7D	013672	4735	4743#			
MDM7E	013716	4745	4753#			
MFP10	025006	7865	7870#			
MFP10A	025034	7872	7876#			
MFP51	023532	7476	7485#			
MFP52A	023622	7510	7511	7512	7518#	
MFP52B	023632	7513	7522#			
MFP52C	023652	7523	7531#			
MFP53A	023730	7552	7553	7554	7560#	
MFP53B	023740	7555	7564#			
MFP53C	023760	7565	7573#			
MFP54A	024036	7594	7595	7596	7602#	
MFP54B	024046	7597	7606#			
MFP54C	024066	7607	7615#			
MFP55A	024144	7636	7637	7638	7644#	
MFP55B	024154	7639	7648#			
MFP55C	024174	7649	7657#			
MFP56A	024254	7678	7679	7680	7686#	
MFP56B	024264	7681	7690#			
MFP56C	024304	7691	7699#			
MFP57A	024364	7720	7721	7722	7728#	
MFP57B	024374	7723	7732#			
MFP57C	024414	7733	7741#			
MFP1 -	000007	11354#	11359			
MOR0	026106	8147	8158#			
MOR1	026146	8170	8179#			
MOR2	026226	8203	8212#			
MOR3	026320	8238	8247#			
MOR4	026340	8248	8257#			
MOR5	026360	8259	8268#			
MOR6	026450	8292	8301#			
MOR7	026470	8303	8313#			
MOR8	026510	8314	8323#			
MOV1	017166	5912	5913	5919#		
MOV2	017176	5914	5924#			
MOV3	017214	5927	5928	5934#		
MRK1	022572	7188	7195#			
MRK2	022614	7195	7196	7197	7199	7206#
MRK3	022624	7201	7210#			
MRK4	022646	7213	7215#			
MRK5	022660	7214	7219#			
MRK6	022674	7220	7227#			
MSGERR	061032	15383	15404#			
MSGPWF	061160	15421	15429#			
MSG1	060642	15330	15362#			
MTP10	025122	7891	7896#			
MTP5*	022744	7252	7260#			
M*P5*A	022764	7264	7265	7266	7272#	





REG3E	002314	911	919#	
REG4	002416	959#	962	
REG4A	002462	983#	986	
REG4E	002430	960	968#	
REG45	046034	12367	12370#	
REG5	002532	1009#	1012	
REG5A	002576	1033#	1036	
REG5E	002544	1010	1018#	
REG6	002654	1059#	1062	
REG6A	002720	1083#	1086	
REG6E	002666	1060	1068#	
RESET2	040632	11098	11108#	
RESET3	040622	11092	11104#	
REST	024472	7778#		
RESTR1	001024	444#	15357	15428
RET	042204	11448	11456#	
RETA	030724	9028	9032#	
RETAH	030734	9030	9036#	
RETAI	036624	10651	10659#	
RETA1	031500	9228	9233#	
RETA2	032306	9440	9445#	
RETA3	033040	9635	9640#	
RETA4	033644	9848	9853#	
RETA5	034376	10051	10056#	
RETB	030760	9044	9046#	
RETB1	036672	10667	10675#	
RETB1	031524	9241	9243#	
RETB2	032332	9453	9455#	
RETB3	033064	9648	9650#	
RETB4	033670	9861	9863#	
RETB5	034422	10064	10066#	
RETC	031022	9063	9065#	
RETC1	036744	10692	10697#	
RETC1	031566	9260	9262#	
RETC2	032374	9472	9474#	
RETC3	033126	9667	9669#	
RETC4	033732	9880	9882#	
RETC5	034464	10083	10085#	
RETD	031074	9082	9086#	
RETD1	031640	9279	9283#	
RETD2	032446	9491	9495#	
RETD3	033200	9686	9690#	
RETD4	034004	9899	9903#	
RETD5	034536	10102	10106#	
RETE	031140	9096	9100#	
RETE1	031702	9293	9296#	
RETE2	032512	9505	9509#	
RETE3	033244	9700	9704#	
RETE4	034050	9914	9918#	
RETE5	034602	10116	10120#	
RETF	031210	9117	9120#	
RETF1	031752	9313	9316#	
RETF2	032562	9526	9529#	
RETF3	033314	9721	9724#	
RETF4	034120	9935	9938#	
RETF5	034652	10137	10140#	





SHLE	001516	656	664#		
SHR	001616	699#	702		
SHRE	001632	700	708#		
SKPMSG	060500	15329	15331#		
SKP104	037760	10958#			
SKTST2	040650	11096	11111#		
SNMBOA	006300	2599	2600	2606#	
SNMB1A	006404	2664	2665	2671#	
SNMB1B	006414	2666	2675#		
SNMB1C	006436	2680	2681	2682	2688#
SNMB2A	006560	2756	2757	2763#	
SNMB2B	006570	2758	2767#		
SNMB2C	006604	2768	2776#		
SNMB2D	006624	2780	2781	2782	2788#
SNMB2E	006634	2783	2792#		
SNMB3A	006776	2872	2873	2879#	
SNMB3B	007006	2874	2883#		
SNMB3C	007024	2886	2887	2888	2894#
SNMB3D	007034	2889	2898#		
SNM0A	006240	2566	2567	2568	2574#
SNM1A	006342	2631	2632	2633	2639#
SNM2A	006500	2712	2713	2714	2720#
SNM2B	006510	2715	2724#		
SNM3A	006710	2824	2825	2826	2832#
SNM3B	006720	2827	2836#		
SNM4A	007104	2929	2930	2936#	
SNM4B	007114	2931	2940#		
SNM5A	007166	2972	2973	2979#	
SNM5B	007176	2974	2983#		
SNM6A	007252	3015	3016	3022#	
SNM6B	007262	3017	3026#		
SNM7A	007334	3057	3058	3064#	
SNM7B	007344	3059	3068#		
SOB1	022454	7144#	7157		
SOB2	022462	7144	7145	7151#	
SOB3	022472	7146	7155#		
SOB4	022512	7158	7159	7160	7167#
SOPA	006156	2524	2532#		
SOPB	006176	2521	2534	2541#	
SOPBOA	003566	1492	1500#		
SOPBOB	003576	1501	1508#		
SOPB1A	003710	1570	1578#		
SOPB1B	003726	1579	1582	1589#	
SOPB1C	003772	1616	1624#		
SOPB1D	004012	1629	1636#		
SOPB2A	004146	1712	1720#		
SOPB2B	004166	1724	1732#		
SOPB2C	004236	1756	1764#		
SOPB2D	004262	1770	1778#		
SOPB3A	004750	2022	2030#		
SOPB3B	004774	2033	2035	2044#	
SOPB3C	005042	2075	2083#		
SOPB3D	005064	2089	2096#		
SOPX	006142	2523#	2532*	2533*	2546
SOPXAD	006206	2535*	2546#		
SOPZA	004060	1664	1672#		









TST233	052604	13595#							
TST234	052702	13630#							
TST235	053000	13665#							
TST236	053076	13700#							
TST237	053200	13737#							
TST240	053276	13772#							
TST241	053374	13807#							
TST242	053512	13848#							
TST243	053610	13883#							
TST244	053706	13918#							
TST245	054004	13953#							
TST246	054100	13988#							
TST247	054174	14023#							
TST250	054270	14058#							
TST251	054366	14093#							
TST252	054464	14128#							
TST253	054560	14163#							
TST254	054654	14204#							
TST255	054764	14246#							
TST256	055074	14288#							
TST257	055210	14332#							
TST260	055320	14374#							
TST261	055430	14416#							
TST262	055544	14460#							
TST263	055654	14502#							
TST264	055764	14544#							
TST265	056074	14586#							
TST266	056210	14630#							
TST267	056320	14672#							
TST270	056430	14714#							
TST271	056544	14758#							
TST272	056654	14800#							
TST273	056764	14842#							
TST274	057056	14872#							
TST275	057160	14904#							
TST276	057270	14946#							
TST277	057400	14988#							
TST300	057510	15030#							
TST301	057616	15072#							
TST302	057724	15114#							
TST303	060032	15156#							
TST304	060142	15198#							
TST305	060252	15240#							
TST306	060360	15282#							
TST37	044010	11684	11722	11760	11798	11838	11878	11884#	
TST40	044044	11885	11891#						
TST41	044060	11892	11895#						
TST42	044076	11896	11899#						
TST43	044132	11900	11906#						
TST44	044164	11907	11912#						
TST45	044216	11913	11918#						
TST46	044250	11919	11924#						
TST47	044306	11925	11931#						
TST50	044336	11932	11937#						
TST51	044414	11958#							
TST52	044476	11987#							

TST53	044560	12016#			
TST54	044642	12045#			
TST55	044722	12074#			
TST56	045002	12103#			
TST57	045062	12132#			
TST60	045144	12161#			
TST61	045226	12190#			
TST62	045306	12219#			
TS1	001056	452#			
TS10	001456	622	626	636	649#
TS100	006446	2657	2683	2704#	
TS101	006526	2706	2726	2747#	
TS102	006652	2749	2794	2814#	
TS103	006736	2816	2838	2860#	
TS104	007052	2862	2900	2920#	
TS105	007130	2922	2941	2961#	
TS106	007214	2963	2985	3005#	
TS107	007276	3007	3027	3047#	
TS11	001526	651	659	673#	
TS110	007360	3049	3069	3088#	
TS111	007414	3090	3096	3113#	
TS112	007450	3115	3121	3138#	
TS113	007524	3140	3158	3179#	
TS114	007666	3181	3234	3252#	
TS115	007756	3254	3275	3293#	
TS116	010102	3295	3341	3358#	
TS117	010166	3360	3381	3401#	
TS12	001572	675	680	693#	
TS120	010224	3403	3410	3429#	
TS121	010262	3431	3438	3458#	
TS122	010320	3460	3467	3491#	
TS123	010400	3493	3512	3532#	
TS124	010440	3534	3542	3562#	
TS125	010512	3564	3580	3602#	
TS126	010566	3604	3619	3638#	
TS127	010644	3640	3658	3678#	
TS13	001642	695	703	737#	
TS130	010706	3680	3685	3704#	
TS131	010750	3706	3711	3730#	
TS132	011012	3732	3737	3750#	
TS133	011070	3752	3770	3783#	
TS134	011146	3785	3803	3816#	
TS135	011244	3818	3846	3859#	
TS136	011406	3861	3912	3926#	
TS137	011546	3928	3974	3987#	
TS14	001672	739	743	756#	
TS140	011654	3989	4018	4031#	
TS141	012016	4033	4082	4095#	
TS142	012126	4097	4126	4139#	
TS143	012234	4141	4169	4182#	
TS144	012344	4184	4212	4231#	
TS145	012420	4233	4251	4270#	
TS146	012514	4272	4301	4319#	
TS147	012650	4321	4367	4385#	
TS15	001724	758	761	774#	
TS150	013026	4387	4433	4452#	

TS151	013122	4454	4480	4501#
TS152	013256	4503	4548	4570#
TS153	013420	4572	4618	4639#
TS154	013570	4641	4687	4707#
TS155	013736	4709	4754	4779#
TS156	014022	4788	4817#	
TS157	014106	4826	4854#	
TS16	001756	776	779	792#
TS160	014172	4856	4863	4885#
TS161	014256	4887	4894	4914#
TS162	014352	4916	4938	4961#
TS163	014514	4963	5003	5024#
TS164	014672	5026	5072	5093#
TS165	015034	5095	5129	5151#
TS166	015110	5153	5162	5186#
TS167	015174	5198	5220#	
TS17	002010	794	797	810#
TS170	015242	5222	5228	5249#
TS171	015320	5258	5281#	
TS172	C15366	5283	5295	5315#
TS173	015430	5317	5322	5343#
TS174	015510	5345	5359	5380#
TS175	015552	5382	5386	5407#
TS176	015632	5409	5423	5446#
TS177	015720	5462	5488#	
TS2	001146	454	487	511#
TS20	002054	812	820	834#
TS200	015770	5495	5521#	
TS201	016042	5528	5576#	
TS202	016406	5681	5711#	
TS203	017064	5845	5867#	
TS204	017140	5869	5880	5906#
TS205	017224	5908	5929	5942#
TS206	017314	5944	5966	5979#
TS207	017404	5981	6002	6015#
TS21	002124	836	845	858#
TS210	017476	6017	6040	6065#
TS211	017614	6067	6105	6119#
TS212	017756	6121	6172	6196#
TS213	020014	6198	6205	6219#
TS214	020100	6221	6243	6256#
TS215	020166	6258	6280	6303#
TS216	020352	6305	6366	6380#
TS217	020470	6382	6419	6443#
TS22	002200	882#		
TS220	020614	6445	6485	6499#
TS221	020774	6501	6553	6567#
TS222	021034	6569	6576	6601#
TS223	021206	6603	6655	6668#
TS224	021354	6670	6723	6747#
TS225	021522	6749	6802	6815#
TS226	021664	6817	6869	6882#
TS227	022034	6884	6938	6951#
TS23	002260	884	904#	
TS230	022214	6953	7010	7034#
TS231	022322	7036	7066	7087#

TS232	022434	7089	7118	7138#
TS233	022522	7140	7162	7182#
TS234	022704	7184	7222	7245#
TS235	022774	7247	7267	7281#
TS236	023056	7283	7298	7311#
TS237	023146	7313	7329	7342#
TS24	002324	906	914	928#
TS240	023232	7344	7359	7372#
TS241	023314	7374	7389	7402#
TS242	023404	7404	7419	7432#
TS243	023474	7434	7449	7470#
TS244	023566	7472	7490	7503#
TS245	023670	7505	7532	7545#
TS246	023776	7547	7574	7587#
TS247	024104	7589	7616	7629#
TS25	002374	930	939	953#
TS250	024212	7631	7658	7671#
TS251	024322	7673	7700	7713#
TS252	024432	7715	7742	7763#
TS253	024472	7765	7769	7788#
TS254	024556	7790	7815#	
TS255	024730	7857#		
TS256	025034	7859	7881#	
TS257	025146	7883	7898	7934#
TS26	002440	955	963	977#
TS260	025406	7991#		
TS261	025554	8040#		
TS262	025630	8063#		
TS263	025670	8065	8072	8099#
TS264	026106	8162#		
TS265	026166	8164	8182	8195#
TS266	026246	8197	8215	8228#
TS267	026376	8230	8269	8282#
TS27	002510	979	988	1003#
TS270	026526	8284	8324	8344#
TS271	026562	8346	8350	8363#
TS272	026622	8365	8369	8382#
TS273	026670	8384	8389	8402#
TS274	026732	8404	8408	8421#
TS275	026772	8423	8427	8442#
TS276	027040	8444	8450	8463#
TS277	027120	8465	8482#	
TS3	001202	513	517	530#
TS30	002554	1005	1013	1027#
TS300	027516	8658#		
TS301	030014	8660	8764	8776#
TS302	030322	8778	8845	8857#
TS303	030532	8859	8930	8944#
TS304	030700	8946	9014	9026#
TS305	030734	9040#		
TS306	030776	9042	9047	9059#
TS307	031042	9061	9066	9078#
TS31	002624	1029	1038	1052#
TS310	031160	9080	9101	9113#
TS311	031432	9222#		
TS312	031500	9224	9237#	

TS313	031542	9239	9244	9256#
TS314	031606	9258	9263	9275#
TS315	031722	9277	9297	9309#
TS316	032162	9311	9404	9416#
TS317	032252	9436#		
TS32	002676	1054	1063	1077#
TS320	032306	9438	9449#	
TS321	032350	9451	9456	9468#
TS322	032414	9470	9475	9487#
TS323	032532	9489	9510	9522#
TS324	033004	9631#		
TS325	033040	9633	9644#	
TS326	033102	9646	9651	9663#
TS327	033146	9665	9670	9682#
TS33	002746	1079	1088	1116#
TS330	033264	9684	9705	9717#
TS331	033520	9719	9811	9823#
TS332	033610	9844#		
TS333	033644	9846	9857#	
TS334	033706	9859	9864	9876#
TS335	033752	9878	9883	9895#
TS336	034070	9897	9919	9931#
TS337	034342	10047#		
TS34	003006	1118	1122	1135#
TS340	034376	10049	10060#	
TS341	034440	10062	10067	10079#
TS342	034504	10081	10086	10098#
TS343	034622	10100	10121	10133#
TS344	035056	10135	10227	10239#
TS345	035112	10241	10252#	
TS346	035154	10254	10259	10271#
TS347	035220	10273	10278	10291#
TS35	003044	1137	1140	1153#
TS350	035336	10293	10314	10326#
TS351	035570	10328	10420	10433#
TS352	035624	10435	10447#	
TS353	035666	10449	10454	10467#
TS354	036010	10469	10502#	
TS355	036060	10517#		
TS356	036130	10532#		
TS357	036200	10547#		
TS36	003102	1155	1158	1171#
TS360	036250	10562#		
TS361	036320	10577#		
TS362	036406	10590	10602#	
TS363	036456	10618#		
TS364	036556	10647#		
TS365	036624	10649	10663#	
TS366	036710	10665	10676	10688#
TS367	036764	10690	10698	10712#
TS37	003140	1173	1176	1203#
TS370	037036	10714	10723	10737#
TS371	037136	10739	10768#	
TS372	037222	10770	10783	10796#
TS373	037332	10822#		
TS374	037420	10856#		







SENV	000320	355#	3648	10967	10998	11031	11066	11093	11120	11165	15393	15456	15550	15574
SEVM	000321	356#	15419	15458	15463	15552								
SEOP	060466	15328#												
SEOPCT	060514	15334#	15338											
SERN =	001163	316#	463	464#	472	473#	483	484#	492	493#	522	523#	541	542#
		560	561#	578	579#	612	613#	631	632#	641	642#	665	666#	685
		686#	709	710#	748	749#	766	767#	784	785#	802	803#	826	827#
		851	852#	874	875#	896	897#	920	921#	945	946#	969	970#	994
		995#	1019	1020#	1044	1045#	1069	1070#	1094	1095#	1127	1128#	1145	1146#
		1163	1164#	1181	1182#	1216	1217#	1229	1230#	1263	1264#	1276	1277#	1305
		1306#	1318	1319#	1347	1348#	1360	1361#	1408	1409#	1419	1420#	1428	1429#
		1456	1457#	1474	1475#	1497	1498#	1509	1510#	1535	1536#	1548	1549#	1575
		1576#	1590	1591#	1621	1622#	1637	1638#	1669	1670#	1685	1686#	1717	1718#
		1753	1734#	1761	1762#	1779	1780#	1807	1808#	1817	1818#	1832	1833#	1841
		1842#	1864	1865#	1874	1875#	1888	1889#	1897	1898#	1920	1921#	1934	1935#
		1943	1944#	1978	1979#	1992	1993#	2027	2028#	2045	2046#	2080	2081#	2097
		2098#	2124	2125#	2133	2134#	2144	2145#	2153	2154#	2164	2165#	2192	2193#
		2208	2209#	2248	2249#	2262	2263#	2290	2291#	2302	2303#	2332	2333#	2344
		2345#	2368	2369#	2377	2378#	2386	2387#	2413	2414#	2422	2423#	2432	2433#
		2458	2459#	2467	2468#	2491	2492#	2503	2504#	2529	2530#	2542	2543#	2575
		2576#	2607	2608#	2640	2641#	2672	2673#	2689	2690#	2721	2722#	2731	2732#
		2764	2765#	2773	2774#	2789	2790#	2799	2800#	2833	2834#	2843	2844#	2880
		2881#	2895	2896#	2905	2906#	2937	2938#	2946	2947#	2980	2981#	2990	2991#
		3023	3024#	3032	3033#	3065	3066#	3074	3075#	3101	3102#	3126	3127#	3154
		3155#	3163	3164#	3189	3190#	3202	3203#	3213	3214#	3228	3229#	3239	3240#
		3270	3271#	3280	3281#	3305	3306#	3314	3315#	3324	3325#	3336	3337#	3346
		3347#	3375	3376#	3387	3388#	3415	3416#	3443	3444#	3472	3473#	3507	3508#
		3517	3518#	3547	3548#	3575	3576#	3585	3586#	3615	3616#	3624	3625#	3653
		3654#	3663	3664#	3690	3691#	3716	3717#	3742	3743#	3766	3767#	3775	3776#
		3799	3800#	3808	3809#	3831	3832#	3842	3843#	3851	3852#	3876	3877#	3885
		3886#	3898	3899#	3908	3909#	3917	3918#	3946	3947#	3955	3956#	3969	3970#
		3979	3980#	4005	4006#	4014	4015#	4023	4024#	4049	4050#	4058	4059#	4069
		4070#	4078	4079#	4087	4088#	4113	4114#	4122	4123#	4131	4132#	4156	4157#
		4165	4166#	4174	4175#	4199	4200#	4208	4209#	4217	4218#	4247	4248#	4256
		4257#	4286	4287#	4297	4298#	4306	4307#	4333	4334#	4342	4343#	4354	4355#
		4363	4364#	4372	4373#	4400	4401#	4409	4410#	4418	4419#	4428	4429#	4438
		4439#	4467	4468#	4476	4477#	4485	4486#	4514	4515#	4523	4524#	4535	4536#
		4544	4545#	4553	4554#	4585	4586#	4594	4595#	4603	4604#	4614	4615#	4623
		4624#	4654	4655#	4663	4664#	4672	4673#	4683	4684#	4692	4693#	4722	4723#
		4731	4732#	4740	4741#	4750	4751#	4759	4760#	4794	4795#	4832	4833#	4868
		4869#	4899	4900#	4929	4930#	4944	4945#	4977	4978#	4992	4993#	5009	5010#
		5042	5043#	5059	5060#	5078	5079#	5107	5108#	5121	5122#	5135	5136#	5168
		5169#	5204	5205#	5234	5235#	5264	5265#	5291	5292#	5300	5301#	5327	5328#
		5355	5356#	5364	5365#	5391	5392#	5419	5420#	5428	5429#	5458	5459#	5468
		5469#	5501	5502#	5534	5535#	5588	5589#	5597	5598#	5611	5612#	5623	5624#
		5632	5633#	5645	5646#	5658	5659#	5672	5673#	5686	5687#	5726	5727#	5745
		5746#	5764	5765#	5782	5783#	5800	5801#	5816	5817#	5832	5833#	5851	5852#
		5876	5877#	5885	5886#	5920	5921#	5935	5936#	5957	5958#	5972	5973#	5994
		5995#	6008	6009#	6031	6032#	6046	6047#	6080	6081#	6096	6097#	6111	6112#
		6134	6135#	6148	6149#	6162	6163#	6178	6179#	6211	6212#	6234	6235#	6249
		6250#	6271	6272#	6286	6287#	6317	6318#	6331	6332#	6345	6346#	6357	6358#
		6372	6373#	6395	6396#	6410	6411#	6425	6426#	6459	6460#	6475	6476#	6491
		6492#	6514	6515#	6529	6530#	6545	6546#	6559	6560#	6582	6583#	6616	6617#
		6631	6632#	6646	6647#	6661	6662#	6684	6685#	6699	6700#	6714	6715#	6729
		6730#	6763	6764#	6778	6779#	6792	6793#	6808	6809#	6831	6832#	6846	6847#
		6861	6862#	6875	6876#	6898	6899#	6913	6914#	6927	6928#	6944	6945#	6967

6968#	6983	6984#	6998	6999#	7016	7017#	7052	7053#	7072	7073#	7106	7107#
7124	7125#	7152	7153#	7168	7169#	7192	7193#	7207	7208#	7216	7217#	7228
7229#	7257	7258#	7273	7274#	7294	7295#	7303	7304#	7325	7326#	7334	7335#
7355	7356#	7364	7365#	7385	7386#	7394	7395#	7415	7416#	7424	7425#	7445
7446#	7454	7455#	7481	7482#	7495	7496#	7519	7520#	7528	7529#	7537	7538#
7561	7562#	7570	7571#	7579	7580#	7603	7604#	7612	7613#	7621	7622#	7645
7646#	7654	7655#	7663	7664#	7687	7688#	7696	7697#	7705	7706#	7729	7730#
7738	7739#	7747	7748#	7774	7775#	7799	7800#	7824	7825#	7834	7835#	7842
7843#	7867	7868#	7873	7874#	7893	7894#	7903	7904#	7966	7967#	8026	8027#
8049	8050#	8078	8079#	8117	8118#	8140	8141#	8175	8176#	8187	8188#	8208
8209#	8220	8221#	8243	8244#	8253	8254#	8264	8265#	8274	8275#	8297	8298#
8308	8309#	8319	8320#	8329	8330#	8355	8356#	8374	8375#	8394	8395#	8413
8414#	8432	8433#	8455	8456#	8474	8475#	8494	8495#	8498	8499#	8548	8549#
8552	8553#	8556	857#	8560	8561#	8564	8565#	8568	8569#	8572	8573#	8576
8577#	8669	8670#	866	8682#	8693	8694#	8706	8707#	8719	8720#	8732	8733#
8744	8745#	8757	875#	8769	8770#	8790	8791#	8805	8806#	8820	8821#	8835
8836#	8850	8851#	8866	8867#	8876	8877#	8886	8887#	8896	8897#	8905	8906#
8915	8916#	8925	8926#	8935	8936#	8954	8955#	8963	8964#	8972	8973#	8981
8982#	8992	8993#	9001	9002#	9010	9011#	9019	9020#	9033	9034#	9052	9053#
9071	9072#	9092	9093#	9106	9107#	9126	9127#	9135	9136#	9144	9145#	9153
9154#	9162	9163#	9175	9176#	9184	9185#	9193	9194#	9202	9203#	9214	9215#
9230	9231#	9249	9250#	9268	9269#	9289	9290#	9302	9303#	9322	9323#	9331
9332#	9340	9341#	9349	9350#	9358	9359#	9371	9372#	9380	9381#	9389	9390#
9398	9399#	9409	9410#	9425	9426#	9442	9443#	9461	9462#	9480	9481#	9501
9502#	9515	9516#	9535	9536#	9544	9545#	9553	9554#	9562	9563#	9571	9572#
9584	9585#	9593	9594#	9602	9603#	9611	9612#	9623	9624#	9637	9638#	9656
9657#	9675	9676#	9696	9697#	9710	9711#	9730	9731#	9739	9740#	9748	9749#
9757	9758#	9766	9767#	9779	9780#	9788	9789#	9797	9798#	9806	9807#	9816
9817#	9832	9833#	9850	9851#	9869	9870#	9888	9889#	9910	9911#	9924	9925#
9944	9945#	9953	9954#	9962	9963#	9971	9972#	9980	9981#	9993	9994#	10002
10003#	10011	10012#	10020	10021#	10032	10033#	10053	10054#	10072	10073#	10091	10092#
10112	10113#	10126	10127#	10146	10147#	10155	10156#	10164	10165#	10173	10174#	10182
10183#	10195	10196#	10204	10205#	10213	10214#	10222	10223#	10232	10233#	10245	10246#
10264	10265#	10283	10284#	10305	10306#	10319	10320#	10339	10340#	10348	10349#	10357
10358#	10366	10367#	10375	10376#	10388	10389#	10397	10398#	10406	10407#	10415	10416#
10425	10426#	10439	10440#	10459	10460#	10480	10481#	10487	10488#	10494	10495#	10510
10511#	10525	10526#	10540	10541#	10555	10556#	10570	10571#	10585	10586#	10595	10596#
10610	10611#	10639	10640#	10656	10657#	10672	10673#	10681	10682#	10703	10704#	10728
10729#	10750	10751#	10761	10762#	10778	10779#	10788	10789#	10810	10811#	10814	10815#
10838	10839#	10874	10875#	10885	10886#	10894	10895#	10918	10919#	10931	10932#	10941
10942#	10979	10980#	10983	10984#	11013	11014#	11017	11018#	11047	11048#	11079	11080#
11083	11084#	11105	11106#	11137	11138#	11153	11154#	11183	11184#	11192	11193#	11202
11203#	11219	11220#	11228	11229#	11253	11254#	11272	11273#	11281	11282#	11306	11307#
11310	11311#	11319	11320#	11328	11329#	11337	11338#	11369	11370#	11392	11393#	11458
11459#	11463	11464#	11473	11474#								

\$ERROR 000302  
 \$ETABL 000320  
 \$ETEND 000330  
 \$FATAL 000302  
 \$FFLG 062014  
 \$FILLC 061542  
 \$FILLS 061541  
 \$GET42 060544  
 \$GTSWR= \*\*\*\*\* U  
 \$HIBTS 000330  
 \$HLT 060704

427# 447#  
 354#  
 366# 389  
 347# 427 15378\* 15389 15578\*  
 15541\* 15544\* 15572 15581\* 15589#  
 15481 15533#  
 15532#  
 15343#  
 15776  
 384#  
 11657 11666 11696 11705 11734 11743 11772 11781 11811 11820 11851 11860 11944

11964	11972	11993	12001	12022	12030	12051	12059	12080	12088	12109	12117	12138
12146	12167	12175	12196	12204	12225	12233	12276	12286	12294	12302	12331	12339
12347	12354	12384	12392	12400	12407	12520	12540	12547	12554	12574	12581	12588
12609	12616	12624	12644	12651	12658	12678	12685	12692	12713	12720	12728	12748
12755	12762	12782	12789	12796	12803	12828	12835	12843	12865	12872	12880	12902
12909	12917	12939	12946	12954	12976	12983	12991	13013	13020	13028	13050	13057
13065	13087	13094	13102	13124	13131	13139	13161	13168	13176	13210	13217	13225
13245	13252	13260	13280	13287	13295	13316	13323	13331	13352	13359	13367	13387
13394	13402	13423	13430	13438	13459	13466	13474	13494	13501	13509	13529	13536
13544	13565	13572	13580	13501	13608	13616	13636	13643	13651	13671	13678	13686
13707	13714	13722	13743	13750	13758	13778	13785	13793	13814	13821	13829	13854
13861	13869	13889	13896	13904	13924	13931	13939	13959	13966	13974	13994	14001
14009	14029	14036	14044	14064	14071	14079	14099	14106	14114	14134	14141	14149
14169	14176	14184	14212	14220	14228	14235	14254	14262	14270	14277	14297	14305
14313	14321	14340	14348	14356	14363	14382	14390	14398	14405	14425	14433	14441
14449	14468	14476	14484	14491	14510	14518	14526	14533	14552	14560	14568	14575
14595	14603	14611	14619	14638	14646	14654	14661	14680	14688	14696	14703	14723
14731	14739	14747	14766	14774	14782	14789	14808	14816	14824	14831	14852	14861
14881	14889	14912	14920	14928	14935	14954	14962	14970	14977	14996	15004	15012
15019	15038	15046	15054	15061	15080	15088	15096	15103	15122	15130	15138	15145
15164	15172	15180	15187	15206	15214	15222	15229	15248	15256	15264	15271	15290
15298	15306	15313	15378#									

\$LF 061545  
\$LFLG 062013  
\$MAIL 000300  
\$MBADR 000332  
\$MFLG 062012  
\$MSGAD 000314  
\$MSGLG 000316  
\$MSGTY 000300  
\$NULL 061540  
\$OCNT 062464  
\$OMODE 062466  
\$PASS 000306  
\$PASTM 000336  
\$QUES 061544  
\$RDCHR= \*\*\*\*\* U  
\$RDDEC= \*\*\*\*\* U  
\$RDLIN= \*\*\*\*\* U  
\$RDOCT= \*\*\*\*\* U  
\$RTNAD 060620  
\$R2A = \*\*\*\*\* U  
\$SAVRE= \*\*\*\*\* U  
\$SETUP= 000020  
\$STUP = 177777  
\$SVPC = 000400  
\$SWR = 000000  
\$SWREG 000322  
\$TESTN 000304  
\$TKB 061536  
\$TKS 061534  
\$TN - 000413

15536#												
15582*	15588#											
345#	385	389	15456									
385#												
15542*	15548	15583*	15587#									
352#	15558*	15561										
353#	15563*											
346#	448*	15395*	15556	15564*	15576	15580*						
15483	15531#											
15694*	15723*	15736#										
15689*	15693*	15698	15701*	15712*	15738#							
349#	442*	10969	11000	11033	11068	11095	11122	11167	11647	11687	11725	11763
11802	11842	12269	12322	12375	15331*	15332*	15340					
387#												
15535#												
15777												
15777												
15777												
15777												
15357#												
15777												
15777												
11610#												
11610#												
334#	339											
316#												
357#	8650											
348#	428	435	445	873	877	894	899	7965	8029	11619		
15502	15509	15530#										
15500	15507	15529#										
316#	449	455#	487	508	514#	517	527	533#	536	546	552#	555
565	571#	573	597	603#	607	617	623#	626	636	646	652#	659
670	676#	680	690	696#	703	734	740#	743	753	759#	761	771
777#	779	789	795#	797	807	813#	820	831	837#	845	855	861#

879	885#	901	907#	914	925	931#	939	950	956#	963	974	980#
988	1000	1006#	1013	1024	1030#	1038	1049	1055#	1063	1074	1.80#	1088
1113	1119#	1122	1132	1138#	1140	1150	1156#	1158	1168	1174#	1176	1200
1206#	1223	1247	1253#	1270	1289	1295#	1312	1331	1337#	1354	1396	1402#
1423	1443	1449#	1468	1485	1491#	1503	1522	1528#	1542	1560	1566#	1584
1605	1611#	1631	1652	1658#	1679	1700	1706#	1727	1743	1749#	1773	1789
1795#	1836	1845	1851#	1892	1901	1907#	1938	1962	1968#	1986	2009	2015#
2039	2063	2069#	2091	2101	2107#	2159	2177	2183#	2202	2228	2234#	2256
2275	2281#	2296	2316	2322#	2338	2349	2355#	2381	2390	2396#	2427	2436
2442#	2462	2471	2477#	2498	2516	2522#	2536	2556	2562#	2569	2588	2594#
2601	2620	2626#	2634	2652	2658#	2683	2701	2707#	2726	2744	2750#	2794
2811	2817#	2838	2857	2863#	2900	2917	2923#	2941	2958	2964#	2985	3002
3008#	3027	3044	3050#	3069	3085	3091#	3096	3110	3116#	3121	3135	3141#
3158	3176	3182#	3234	3249	3255#	3275	3290	3296#	3341	3355	3361#	3381
3398	3404#	3410	3426	3432#	3438	3455	3461#	3467	3488	3494#	3512	3529
3535#	3542	3559	3565#	3580	3599	3605#	3619	3635	3641#	3658	3675	3681#
3685	3701	3707#	3711	3727	3733#	3737	3747	3753#	3770	3780	3786#	3803
3813	3819#	3846	3856	3862#	3912	3923	3929#	3974	3984	3990#	4018	4028
4034#	4082	4092	4098#	4126	4136	4142#	4169	4179	4185#	4212	4228	4234#
4251	4267	4273#	4301	4316	4322#	4367	4382	4388#	4433	4449	4455#	4480
4498	4504#	4548	4567	4573#	4618	4636	4642#	4687	4704	4710#	4754	4776
4782#	4788	4814	4820#	4826	4851	4857#	4863	4882	4888#	4894	4911	4917#
4938	4958	4964#	5003	5021	5027#	5072	5090	5096#	5129	5148	5154#	5162
5183	5189#	5198	5217	5223#	5228	5246	5252#	5258	5278	5284#	5295	5312
5318#	5322	5340	5346#	5359	5377	5383#	5386	5404	5410#	5423	5443	5449#
5462	5485	5491#	5495	5518	5524#	5528	5573	5579#	5681	5708	5714#	5845
5864	5870#	5880	5903	5909#	5929	5939	5945#	5966	5976	5982#	6002	6012
6018#	6040	6062	6068#	6105	6116	6122#	6172	6193	6199#	6205	6216	6222#
6243	6253	6259#	6280	6300	6306#	6366	6377	6383#	6419	6440	6446#	6485
6496	6502#	6553	6564	6570#	6576	6598	6604#	6655	6665	6671#	6723	6744
6750#	6802	6812	6818#	6869	6879	6885#	6938	6948	6954#	7010	7031	7037#
7066	7084	7090#	7118	7135	7141#	7162	7179	7185#	7222	7242	7248#	7267
7278	7284#	7298	7308	7314#	7329	7339	7345#	7359	7369	7375#	7389	7399
7405#	7419	7429	7435#	7449	7467	7473#	7490	7500	7506#	7532	7542	7548#
7574	7584	7590#	7616	7626	7632#	7658	7668	7674#	7700	7710	7716#	7742
7760	7766#	7769	7785	7791#	7812	7818#	7854	7860#	7878	7884#	7898	7931
7937#	7988	7994#	8037	8043#	8060	8066#	8072	8096	8102#	8159	8165#	8182
8192	8198#	8215	8225	8231#	8269	8279	8285#	8324	8341	8347#	8350	8360
8366#	8369	8379	8385#	8389	8399	8405#	8408	8418	8424#	8427	8439	8445#
8450	8460	8466#	8479	8485#	8655	8661#	8764	8773	8779#	8845	8854	8860#
8930	8941	8947#	9014	9023	9029#	9037	9043#	9047	9056	9062#	9066	9075
9081#	9101	9110	9116#	9219	9225#	9234	9240#	9244	9253	9259#	9263	9272
9278#	9297	9306	9312#	9404	9413	9419#	9433	9439#	9446	9452#	9456	9465
9471#	9475	9484	9490#	9510	9519	9525#	9628	9634#	9641	9647#	9651	9660
9666#	9670	9679	9685#	9705	9714	9720#	9811	9820	9826#	9841	9847#	9854
9860#	9864	9873	9879#	9883	9892	9898#	9919	9928	9934#	10044	10050#	10057
10063#	10067	10076	10082#	10086	10095	10101#	10121	10130	10136#	10227	10236	10242#
10249	10255#	10259	10268	10274#	10278	10288	10294#	10314	10323	10329#	10420	10430
10436#	10444	10450#	10454	10464	10470#	10499	10505#	10514	10520#	10529	10535#	10544
10550#	10559	10565#	10574	10580#	10590	10599	10605#	10615	10621#	10644	10650#	10660
10666#	10676	10685	10691#	10698	10709	10715#	10723	10734	10740#	10765	10771#	10783
10793	10799#	10819	10825#	10853	10859#	10959	10965#	10990	10996#	11023	11029#	11058
11064#	11074	11087	11093#	11112	11118#	11159	11165#	11206	11212#	11223	11236	11242#
11247	11263	11269#	11276	11294	11300#	11345	11351#	11382	11388#	11397	11403#	
11604#	15518*	15531										
11590#												

\$TPB 042502  
\$PCNT 042447

\$TPFLG 061543  
 \$TPS 042504  
 \$TRAP 062470  
 \$TRAP2 062512  
 \$TRP = 000006  
 \$TRPAD 062524  
 \$STSM 000334  
 \$STSM= 000304  
 \$STPBN= \*\*\*\*\* U  
 \$STPDS 062016  
 \$TYPE 061200  
 \$TYPEC 061412  
 \$TYPEX 061532  
 \$TYPOC 062266  
 \$TYPON 062302  
 \$TYPOS 062242  
 \$UNIT 000312  
 \$UNITM 000340  
 \$USWR 000324  
 \$X - C41750

15450	15534#												
11605#	15516	15531											
11627	15747#												
15758#	15769												
15762#	15771#	15772#	15773#	15774#	15775#								
15752	15769#												
386#													
428#	446*												
15775													
15607#	15774												
15450#	15569	15762	15770										
15480	15487	15494	15499#										
15522	15524	15527#											
15692#	15771												
15691	15694#	15773											
15687#	15772												
351#													
388#													
358#													
455#	470	490	514#	520	533#	539	552#	558	571#	576	603#	610	
623#	629	639	652#	662	676#	683	696#	706	740#	746	759#	764	
777#	782	795#	800	813#	823	837#	848	861#	885#	907#	917	931#	
942	956#	966	980#	991	1006#	1016	1030#	1041	1055#	1066	1080#	1091	
1119#	1125	1138#	1143	1156#	1161	1174#	1179	1206#	1213	1226	1253#	1260	
1273	1295#	1302	1315	1337#	1344	1357	1402#	1406	1417	1426	1449#	1454	
1471	1491#	1495	1506	1528#	1533	1545	1566#	1573	1587	1611#	1619	1634	
1658#	1667	1682	1706#	1715	1730	1749#	1759	1776	1795#	1804	1815	1829	
1839	1851#	1861	1872	1885	1895	1907#	1917	1932	1941	1968#	1976	1989	
2015#	2025	2042	2069#	2078	2094	2107#	2121	2131	2142	2151	2162	2183#	
2190	2205	2234#	2246	2259	2281#	2288	2299	2322#	2330	2341	2355#	2365	
2375	2384	2396#	2410	2420	2430	2442#	2455	2465	2477#	2488	2501	2522#	
2527	2539	2562#	2572	2594#	2604	2626#	2637	2658#	2669	2686	2707#	2718	
2729	2750#	2761	2771	2786	2797	2817#	2830	2841	2863#	2877	2892	2903	
2923#	2934	2944	2964#	2977	2988	3008#	3020	3030	3050#	3062	3072	3091#	
3099	3116#	3124	3141#	3151	3161	3182#	3187	3200	3211	3226	3237	3255#	
3267	3278	3296#	3303	3312	3322	3334	3344	3361#	3372	3384	3404#	3413	
3432#	3441	3461#	3470	3494#	3505	3515	3535#	3545	3565#	3573	3583	3605#	
3613	3622	3641#	3651	3661	3681#	3688	3707#	3714	3733#	3740	3753#	3763	
3773	3786#	3796	3806	3819#	3828	3839	3849	3862#	3873	3883	3895	3906	
3915	3929#	3943	3953	3966	3977	3990#	4002	4012	4021	4034#	4046	4056	
4066	4076	4085	4098#	4110	4120	4129	4142#	4153	4163	4172	4185#	4196	
4206	4215	4234#	4244	4254	4273#	4283	4294	4304	4322#	4330	4340	4351	
4361	4370	4388#	4397	4407	4416	4426	4436	4455#	4464	4474	4483	4504#	
4512	4521	4532	4542	4551	4573#	4582	4592	4601	4612	4621	4642#	4651	
4661	4670	4681	4690	4710#	4719	4729	4738	4748	4757	4782#	4791	4820#	
4829	4857#	4866	4888#	4897	4917#	4926	4941	4964#	4974	4989	5006	5027#	
5039	5056	5075	5096#	5104	5118	5132	5154#	5165	5189#	5201	5223#	5231	
5252#	5261	5284#	5289	5298	5318#	5325	5346#	5353	5362	5383#	5389	5410#	
5417	5426	5449#	5456	5465	5491#	5498	5524#	5531	5579#	5586	5595	5609	
5621	5630	5643	5656	5670	5684	5714#	5742	5761	5779	5797	5813	5829	
5848	5870#	5883	5909#	5917	5932	5945#	5954	5969	5982#	5991	6005	6018#	
6028	6043	6068#	6077	6093	6108	6122#	6131	6145	6159	6175	6199#	6208	
6222#	6231	6246	6259#	6268	6283	6306#	6314	6328	6342	6354	6369	6383#	
6392	6407	6422	6446#	6456	6472	6488	6502#	6511	6526	6542	6556	6570#	
6579	6604#	6613	6628	6643	6658	6671#	6681	6696	6711	6726	6750#	6760	
6775	6789	6805	6818#	6828	6843	6858	6872	6885#	6895	6910	6924	6941	

6954#	6964	6980	6995	7013	7037#	7049	7069	7090#	7103	7121	7141#	7149
7165	7185#	7204	7225	7248#	7255	7270	7284#	7292	7301	7314#	7323	7332
7345#	7353	7362	7375#	7383	7392	7405#	7413	7422	7435#	7443	7452	7473#
7479	7493	7506#	7516	7526	7535	7548#	7558	7568	7577	7590#	7600	7610
7619	7632#	7642	7652	7661	7674#	7684	7694	7703	7716#	7726	7736	7745
7766#	7772	7791#	7818#	7832	7860#	7884#	7901	7937#	7994#	8023	8043#	8066#
8075	8102#	8115	8137	8165#	8173	8185	8198#	8206	8218	8231#	8241	8251
8262	8272	8285#	8295	8306	8317	8327	8347#	8353	8366#	8372	8385#	8392
8405#	8411	8424#	8430	8445#	8453	8466#	8485#	8661#	8667	8679	8691	8704
8717	8730	8742	8755	8767	8779#	8788	8803	8818	8833	8848	8860#	8864
8874	8884	8894	8903	8913	8923	8933	8947#	8952	8961	8970	8979	8990
8999	9008	9017	9029#	9043#	9050	9062#	9069	9081#	9090	9104	9116#	9124
9133	9142	9151	9160	9173	9182	9191	9200	9211	9225#	9240#	9247	9259#
9266	9278#	9287	9300	9312#	9320	9329	9338	9347	9356	9369	9378	9387
9396	9407	9419#	9439#	9452#	9459	9471#	9478	9490#	9499	9513	9525#	9533
9542	9551	9560	9569	9582	9591	9600	9609	9620	9634#	9647#	9654	9666#
9673	9685#	9694	9708	9720#	9728	9737	9746	9755	9764	9777	9786	9795
9804	9814	9826#	9847#	9860#	9867	9879#	9886	9898#	9908	9922	9934#	9942
9951	9960	9969	9978	9991	10000	10009	10018	10029	10050#	10063#	10070	10082#
10089	10101#	10110	10124	10136#	10144	10153	10162	10171	10180	10193	10202	10211
10220	10230	10242#	10255#	10262	10274#	10281	10294#	10303	10317	10329#	10337	10346
10355	10364	10373	10386	10395	10404	10413	10423	10436#	10450#	10457	10470#	10478
10505#	10520#	10535#	10550#	10565#	10580#	10593	10605#	10621#	10636	10650#	10666#	10679
10691#	10701	10715#	10726	10740#	10759	10771#	10786	10799#	10825#	10835	10859#	10883
10892	10916	10929	10938	10965#	10996#	11029#	11064#	11077	11093#	11118#	11135	11150
11165#	11190	11199	11212#	11226	11242#	11250	11269#	11279	11300#	11317	11326	11335
11351#	11366	11388#	11403#	11470								
15504	15531											
15511	15531											
470#	490#	520#	539#	558#	576#	610#	629#	639#	662#	683#	706#	746#
764#	782#	800#	823#	848#	917#	942#	966#	991#	1016#	1041#	1066#	1091#
1125#	1143#	1161#	1179#	1213#	1226#	1260#	1273#	1302#	1315#	1344#	1357#	1406#
1417#	1426#	1454#	1471#	1495#	1506#	1533#	1545#	1573#	1587#	1619#	1634#	1667#
1682#	1715#	1730#	1759#	1776#	1804#	1815#	1829#	1839#	1861#	1872#	1885#	1895#
1917#	1932#	1941#	1976#	1989#	2025#	2042#	2078#	2094#	2121#	2131#	2142#	2151#
2162#	2190#	2205#	2246#	2259#	2288#	2299#	2330#	2341#	2365#	2375#	2384#	2410#
2420#	2430#	2455#	2465#	2488#	2501#	2527#	2539#	2572#	2604#	2637#	2669#	2686#
2718#	2729#	2761#	2771#	2786#	2797#	2830#	2841#	2877#	2892#	2903#	2934#	2944#
2977#	2988#	3020#	3030#	3062#	3072#	3099#	3124#	3151#	3161#	3187#	3200#	3211#
3226#	3237#	3267#	3278#	3303#	3312#	3322#	3334#	3344#	3372#	3384#	3413#	3441#
3470#	3505#	3515#	3545#	3573#	3583#	3613#	3622#	3651#	3661#	3688#	3714#	3740#
3763#	3775#	3796#	3806#	3828#	3839#	3849#	3873#	3883#	3895#	3906#	3915#	3943#
3953#	3966#	3977#	4002#	4012#	4021#	4046#	4056#	4066#	4076#	4085#	4110#	4120#
4129#	4153#	4163#	4172#	4196#	4206#	4215#	4244#	4254#	4283#	4294#	4304#	4330#
4340#	4351#	4361#	4370#	4397#	4407#	4416#	4426#	4436#	4464#	4474#	4483#	4512#
4521#	4532#	4542#	4551#	4582#	4592#	4601#	4612#	4621#	4651#	4661#	4670#	4681#
4690#	4719#	4729#	4738#	4748#	4757#	4791#	4829#	4866#	4897#	4926#	4941#	4974#
4989#	5006#	5039#	5056#	5075#	5104#	5118#	5132#	5165#	5201#	5231#	5261#	5289#
5298#	5325#	5353#	5362#	5389#	5417#	5426#	5456#	5465#	5498#	5531#	5586#	5595#
5609#	5621#	5630#	5643#	5656#	5670#	5684#	5742#	5761#	5779#	5797#	5813#	5829#
5848#	5883#	5917#	5932#	5954#	5969#	5991#	6005#	6028#	6043#	6077#	6093#	6108#
6131#	6145#	6159#	6175#	6208#	6231#	6246#	6268#	6283#	6314#	6328#	6342#	6354#
6369#	6392#	6407#	6422#	6456#	6472#	6488#	6511#	6526#	6542#	6556#	6579#	6613#
6628#	6643#	6658#	6681#	6696#	6711#	6726#	6760#	6775#	6789#	6805#	6828#	6843#
6858#	6872#	6895#	6910#	6924#	6941#	6964#	6980#	6995#	7013#	7049#	7069#	7103#
7121#	7149#	7165#	7204#	7225#	7255#	7270#	7292#	7301#	7323#	7332#	7353#	7362#

\$XOFF = 000023  
 \$XON = 000021  
 \$XX = 177640



7383#	7392#	7413#	7422#	7443#	7452#	7479#	7493#	7516#	7526#	7535#	7558#	7568#
7577#	7600#	7610#	7619#	7642#	7652#	7661#	7684#	7694#	7703#	7726#	7736#	7745#
7772#	7832#	7901#	8023#	8075#	8115#	8137#	8173#	8185#	8206#	8218#	8241#	8251#
8262#	8272#	8295#	8306#	8317#	8327#	8353#	8372#	8392#	8411#	8430#	8453#	8667#
8679#	8691#	8704#	8717#	8730#	8742#	8755#	8767#	8788#	8803#	8818#	8833#	8848#
8864#	8874#	8884#	8894#	8903#	8913#	8923#	8933#	8952#	8961#	8970#	8979#	8990#
8999#	9008#	9017#	9050#	9063#	9090#	9104#	9124#	9133#	9142#	9151#	9160#	9173#
9182#	9191#	9200#	9211#	9247#	9266#	9287#	9300#	9320#	9329#	9338#	9347#	9356#
9369#	9378#	9387#	9396#	9407#	9459#	9478#	9499#	9513#	9533#	9542#	9551#	9560#
9569#	9582#	9591#	9600#	9609#	9620#	9654#	9673#	9694#	9708#	9728#	9737#	9746#
9755#	9764#	9777#	9786#	9795#	9804#	9814#	9867#	9886#	9908#	9922#	9942#	9951#
9960#	9969#	9978#	9991#	10000#	10009#	10018#	10029#	10070#	10089#	10110#	10124#	10144#
10153#	10162#	10171#	10180#	10193#	10202#	10211#	10220#	10230#	10262#	10281#	10303#	10317#
10337#	10346#	10355#	10364#	10373#	10386#	10395#	10404#	10413#	10423#	10457#	10478#	10593#
10636#	10679#	10701#	10726#	10759#	10786#	10835#	10883#	10892#	10916#	10929#	10938#	11077#
11135#	11150#	11190#	11199#	11226#	11250#	11279#	11317#	11326#	11335#	11366#	11470#	
470#	490#	520#	539#	558#	576#	610#	629#	639#	662#	683#	706#	746#
764#	782#	800#	823#	848#	917#	942#	966#	991#	1016#	1041#	1066#	1091#
1125#	1143#	1161#	1179#	1213#	1226#	1260#	1273#	1302#	1315#	1344#	1357#	1406#
1417#	1426#	1454#	1471#	1495#	1506#	1533#	1545#	1573#	1587#	1619#	1634#	1667#
1682#	1715#	1730#	1759#	1776#	1804#	1815#	1829#	1839#	1861#	1872#	1885#	1895#
1917#	1932#	1941#	1976#	1989#	2025#	2042#	2078#	2094#	2121#	2131#	2142#	2151#
2162#	2190#	2205#	2246#	2259#	2288#	2299#	2330#	2341#	2365#	2375#	2384#	2410#
2420#	2430#	2455#	2465#	2488#	2501#	2527#	2539#	2572#	2604#	2637#	2669#	2686#
2718#	2729#	2761#	2771#	2786#	2797#	2830#	2841#	2877#	2892#	2903#	2934#	2944#
2977#	2988#	3020#	3030#	3062#	3072#	3099#	3124#	3151#	3161#	3187#	3200#	3211#
3226#	3237#	3267#	3278#	3303#	3312#	3322#	3334#	3344#	3372#	3384#	3413#	3441#
3470#	3505#	3515#	3545#	3573#	3583#	3613#	3622#	3651#	3661#	3688#	3714#	3740#
3763#	3773#	3796#	3806#	3828#	3839#	3849#	3873#	3883#	3895#	3906#	3915#	3943#
3953#	3966#	3977#	4002#	4012#	4021#	4046#	4056#	4066#	4076#	4085#	4110#	4120#
4129#	4153#	4163#	4172#	4196#	4206#	4215#	4244#	4254#	4283#	4294#	4304#	4330#
4340#	4351#	4361#	4370#	4397#	4407#	4416#	4426#	4436#	4464#	4474#	4483#	4512#
4521#	4532#	4542#	4551#	4582#	4592#	4601#	4612#	4621#	4651#	4661#	4670#	4681#
4690#	4719#	4729#	4738#	4748#	4757#	4791#	4829#	4866#	4897#	4926#	4941#	4974#
4989#	5006#	5039#	5056#	5075#	5104#	5118#	5132#	5165#	5201#	5231#	5261#	5289#
5298#	5325#	5353#	5362#	5389#	5417#	5426#	5456#	5465#	5498#	5531#	5586#	5595#
5609#	5621#	5630#	5643#	5656#	5670#	5684#	5742#	5761#	5779#	5797#	5813#	5829#
5848#	5883#	5917#	5932#	5954#	5969#	5991#	6005#	6028#	6043#	6077#	6093#	6108#
6131#	6145#	6159#	6175#	6208#	6231#	6246#	6268#	6283#	6314#	6328#	6342#	6354#
6369#	6392#	6407#	6422#	6456#	6472#	6488#	6511#	6526#	6542#	6556#	6579#	6613#
6628#	6643#	6658#	6681#	6696#	6711#	6726#	6760#	6775#	6789#	6805#	6828#	6843#
6858#	6872#	6895#	6910#	6924#	6941#	6964#	6980#	6995#	7013#	7049#	7069#	7103#
7121#	7149#	7165#	7204#	7225#	7255#	7270#	7292#	7301#	7323#	7332#	7353#	7362#
7383#	7392#	7413#	7422#	7443#	7452#	7479#	7493#	7516#	7526#	7535#	7558#	7568#
7577#	7600#	7610#	7619#	7642#	7652#	7661#	7684#	7694#	7703#	7726#	7736#	7745#
7772#	7832#	7901#	8023#	8075#	8115#	8137#	8173#	8185#	8206#	8218#	8241#	8251#
8262#	8272#	8295#	8306#	8317#	8327#	8353#	8372#	8392#	8411#	8430#	8453#	8667#
8679#	8691#	8704#	8717#	8730#	8742#	8755#	8767#	8788#	8803#	8818#	8833#	8848#
8864#	8874#	8884#	8894#	8903#	8913#	8923#	8933#	8952#	8961#	8970#	8979#	8990#
8999#	9008#	9017#	9050#	9063#	9090#	9104#	9124#	9133#	9142#	9151#	9160#	9173#
9182#	9191#	9200#	9211#	9247#	9266#	9287#	9300#	9320#	9329#	9338#	9347#	9356#
9369#	9378#	9387#	9396#	9407#	9459#	9478#	9499#	9513#	9533#	9542#	9551#	9560#
9569#	9582#	9591#	9600#	9609#	9620#	9654#	9673#	9694#	9708#	9728#	9737#	9746#
9755#	9764#	9777#	9786#	9795#	9804#	9814#	9867#	9886#	9908#	9922#	9942#	9951#
9960#	9969#	9978#	9991#	10000#	10009#	10018#	10029#	10070#	10089#	10110#	10124#	10144#
10153#	10162#	10171#	10180#	10193#	10202#	10211#	10220#	10230#	10262#	10281#	10303#	10317#

EXX 000637



SO FILL 062465  
- 062542

10337#	10346#	10355#	10364#	10373#	10386#	10395#	10404#	10413#	10423#	10457#	10478#	10593#
10636#	10679#	10701#	10726#	10759#	10786#	10835#	10883#	10892#	10916#	10929#	10938#	11077#
11135#	11150#	11190#	11199#	11226#	11250#	11279#	11317#	11326#	11335#	11366#	11470#	
15688#	15692#	15702	15737#									
330#	334	335#	337#	339#	340#	373	374#	376#	378#	393#	399#	404#
407#	413#	421#	429#	432#	439#	455	470	490	514	520	533	539
552	558	571	576	603	610	623	629	639	652	662	676	683
696	706	740	746	759	764	777	782	795	800	813	823	837
848	861	885	907	917	931	942	956	966	980	991	1006	1016
1030	1041	1055	1066	1080	1091	1119	1125	1138	1143	1156	1161	1174
1179	1206	1213	1226	1253	1260	1273	1295	1302	1315	1337	1344	1357
1402	1406	1417	1426	1449	1454	1471	1491	1495	1506	1528	1533	1545
1566	1573	1587	1611	1619	1634	1658	1667	1682	1706	1715	1730	1749
1759	1776	1795	1804	1815	1829	1839	1851	1861	1872	1885	1895	1907
1917	1932	1941	1968	1976	1989	2015	2025	2042	2069	2078	2094	2107
2121	2131	2142	2151	2162	2183	2190	2205	2234	2246	2259	2281	2288
2299	2322	2330	2341	2355	2365	2375	2384	2396	2410	2420	2430	2442
2455	2465	2477	2488	2501	2522	2527	2539	2562	2572	2594	2604	2626
2637	2658	2669	2686	2707	2718	2729	2750	2761	2771	2786	2797	2817
2830	2841	2863	2877	2892	2903	2923	2934	2944	2964	2977	2988	3008
3020	3030	3050	3062	3072	3091	3099	3116	3124	3141	3151	3161	3182
3187	3200	3211	3226	3237	3255	3267	3278	3296	3303	3312	3322	3334
3344	3361	3372	3384	3404	3413	3432	3441	3461	3470	3494	3505	3515
3535	3545	3565	3573	3583	3605	3613	3622	3641	3651	3661	3681	3688
3707	3714	3733	3740	3753	3763	3773	3786	3796	3806	3819	3828	3839
3849	3862	3873	3883	3895	3906	3915	3929	3943	3953	3966	3977	3990
4002	4012	4021	4034	4046	4056	4066	4076	4085	4098	4110	4120	4129
4142	4153	4163	4172	4185	4196	4206	4215	4234	4244	4254	4273	4283
4294	4304	4322	4330	4340	4351	4361	4370	4388	4397	4407	4416	4426
4436	4455	4464	4474	4483	4504	4512	4521	4532	4542	4551	4573	4582
4592	4601	4612	4621	4642	4651	4661	4670	4681	4690	4710	4719	4729
4738	4748	4757	4782	4791	4820	4829	4857	4866	4888	4897	4917	4926
4941	4964	4974	4989	5006	5027	5039	5056	5075	5096	5104	5118	5132
5154	5165	5189	5201	5223	5231	5252	5261	5284	5289	5298	5318	5325
5346	5353	5362	5383	5389	5410	5417	5426	5449	5456	5465	5491	5498
5524	5531	5579	5582	5586	5595	5609	5621	5630	5643	5656	5670	5684
5714	5742	5761	5779	5797	5813	5829	5848	5870	5883	5909	5917	5932
5945	5954	5969	5982	5991	6005	6018	6028	6043	6068	6077	6093	6108
6122	6131	6145	6159	6175	6199	6208	6222	6231	6246	6259	6268	6283
6306	6314	6328	6342	6354	6369	6383	6392	6407	6422	6446	6456	6472
6488	6502	6511	6526	6542	6556	6570	6579	6604	6613	6628	6643	6658
6671	6681	6696	6711	6726	6750	6760	6775	6789	6805	6818	6828	6843
6858	6872	6885	6895	6910	6924	6941	6954	6964	6980	6995	7013	7037
7049	7069	7090	7103	7121	7141	7149	7165	7185	7204	7225	7248	7255
7270	7284	7292	7301	7314	7323	7332	7345	7353	7362	7375	7383	7392
7405	7413	7422	7435	7443	7452	7473	7479	7493	7506	7516	7526	7535
7548	7558	7568	7577	7590	7600	7610	7619	7632	7642	7652	7661	7674
7684	7694	7703	7716	7726	7736	7745	7766	7772	7791	7818	7832	7860
7884	7901	7937	7948	7973	7978	7994	8023	8043	8066	8075	8102	8115
8137	8165	8173	8185	8198	8206	8218	8231	8241	8251	8262	8272	8285
8295	8306	8317	8327	8347	8353	8366	8372	8385	8392	8405	8411	8424
8430	8445	8453	8466	8485	8503	8504	8505	8506	8507	8508	8509	8510
8511	8512	8513	8514	8515	8516	8517	8539#	8661	8667	8679	8691	8704
8717	8730	8742	8755	8767	8779	8788	8803	8818	8833	8848	8860	8864
8874	8884	8894	8903	8913	8923	8933	8947	8952	8961	8970	8979	8990
8999	9008	9017	9029	9043	9050	9062	9065	9069	9081	9090	9104	9116

9124	9133	9142	9151	9160	9173	9182	9191	9200	9211	9225	9240	9247
9259	9262	9266	9278	9287	9300	9312	9320	9329	9338	9347	9356	9369
9378	9387	9396	9407	9419	9439	9452	9459	9471	9474	9478	9490	9499
9513	9525	9533	9542	9551	9560	9569	9582	9591	9600	9609	9620	9634
9647	9654	9666	9669	9673	9685	9694	9708	9720	9728	9737	9746	9755
9764	9777	9786	9795	9804	9814	9826	9847	9860	9867	9879	9882	9886
9898	9908	9922	9934	9942	9951	9960	9969	9978	9991	10000	10009	10018
10029	10050	10063	10070	10082	10085	10089	10101	10110	10124	10136	10144	10153
10162	10171	10180	10193	10202	10211	10220	10230	10242	10255	10262	10274	10281
10294	10303	10317	10329	10337	10346	10355	10364	10373	10386	10395	10404	10413
10423	10436	10450	10457	10470	10478	10505	10520	10535	10550	10565	10580	10593
10605	10621	10636	10650	10653	10666	10669	10679	10691	10694	10697	10701	10715
10726	10740	10759	10771	10786	10799	10806	10825	10835	10859	10883	10892	10916
10929	10938	10965	10996	11029	11064	11077	11093	11118	11135	11150	11165	11190
11199	11212	11226	11242	11250	11269	11279	11300	11317	11326	11335	11351	11366
11388	11403	11470	11520	11576#	11578#	11580#	11582#	11584#	11586#	11656	11664	11695
11703	11733	11741	11771	11779	11810	11818	11850	11858	11943	11963	11970	11992
11999	12021	12028	12050	12057	12079	12086	12108	12115	12137	12144	12166	12173
12195	12202	12224	12231	12277	12285	12292	12301	12330	12338	12345	12353	12383
12391	12398	12406	12519	12539	12546	12553	12573	12580	12587	12608	12615	12623
12643	12650	12657	12677	12684	12691	12712	12719	12727	12747	12754	12761	12781
12788	12795	12802	12827	12834	12841	12864	12871	12878	12901	12908	12915	12938
12945	12952	12975	12982	12989	13012	13019	13026	13049	13056	13063	13086	13093
13100	13123	13130	13137	13160	13167	13174	13209	13216	13223	13244	13251	13258
13279	13286	13293	13315	13322	13329	13351	13358	13365	13386	13393	13400	13422
13429	13436	13458	13465	13472	13493	13500	13507	13528	13535	13542	13564	13571
13578	13600	13607	13614	13635	13642	13649	13670	13677	13684	13706	13713	13720
13742	13749	13756	13777	13784	13791	13813	13820	13827	13853	13860	13867	13888
13895	13902	13923	13930	13937	13958	13965	13972	13993	14000	14007	14028	14035
14042	14063	14070	14077	14098	14105	14112	14133	14140	14147	14168	14175	14182
14211	14219	14227	14234	14253	14261	14269	14276	14296	14304	14312	14320	14339
14347	14355	14362	14381	14389	14397	14404	14424	14432	14440	14448	14467	14475
14483	14490	14509	14517	14525	14532	14551	14559	14567	14574	14594	14602	14610
14618	14637	14645	14653	14660	14679	14687	14695	14702	14722	14730	14738	14746
14765	14773	14781	14788	14807	14815	14823	14830	14851	14860	14880	14888	14911
14919	14927	14934	14953	14961	14969	14976	14995	15003	15011	15018	15037	15045
15053	15060	15079	15087	15095	15102	15121	15129	15137	15144	15163	15171	15179
15186	15205	15213	15221	15228	15247	15255	15263	15270	15289	15297	15305	15312
15396	15529	15530	15531	15532	15533	15534	15535	15536	15537#	15590#	15661#	
15542	15545											
373#	378											

.SA,TA \*\*\*\*\* U  
.SY 000330

FR R	316#	462	466	482	486	517	536	555	573	607	626	636	659	680	703
	743	761	779	797	820	845	874	895	914	939	963	988	1013	1038	1063
	1088	1122	1140	1158	1176	1210	1223	1257	1270	1299	1312	1341	1354	1403	1414
	1423	1451	1468	1492	1503	1530	1542	1570	1584	1616	1631	1664	1679	1712	1727
	1756	1773	1801	1812	1826	1836	1858	1869	1882	1892	1914	1929	1938	1973	1986
	2022	2039	2075	2091	2118	2128	2139	2148	2159	2187	2202	2243	2256	2285	2296
	2327	2338	2362	2372	2381	2407	2417	2427	2452	2462	2485	2498	2524	2536	2569
	2601	2634	2666	2683	2715	2726	2758	2768	2783	2794	2827	2838	2874	2889	2900
	2931	2941	2974	2985	3017	3027	3059	3069	3096	3121	3148	3158	3184	3197	3208
	3223	3234	3264	3275	3300	3309	3319	3331	3341	3369	3381	3410	3438	3467	3502
	3512	3542	3570	3580	3610	3619	3648	3658	3685	3711	3737	3760	3770	3793	3803
	3825	3836	3846	3870	3880	3892	3903	3912	3940	3950	3963	3974	3999	4009	4018
	4043	4053	4063	4073	4082	4107	4117	4126	4150	4160	4169	4193	4203	4212	4241
	4251	4280	4291	4301	4327	4337	4348	4358	4367	4394	4404	4413	4423	4433	4461
	4471	4480	4509	4518	4529	4539	4548	4579	4589	4598	4609	4618	4648	4658	4667
	4678	4687	4716	4726	4735	4745	4754	4788	4826	4863	4894	4923	4938	4971	4986
	5003	5036	5053	5072	5101	5115	5129	5162	5198	5228	5258	5286	5295	5322	5350
	5359	5386	5414	5423	5453	5462	5495	5528	5583	5592	5606	5618	5627	5640	5653
	5667	5681	5725	5739	5758	5776	5794	5810	5826	5845	5876	5880	5914	5929	5951
	5966	5988	6002	6025	6040	6074	6090	6105	6128	6142	6156	6172	6205	6228	6243
	6265	6280	6311	6325	6339	6351	6366	6389	6404	6419	6453	6469	6485	6508	6523
	6539	6553	6576	6610	6625	6640	6655	6678	6693	6708	6723	6757	6772	6786	6802
	6825	6840	6855	6869	6892	6907	6921	6938	6961	6977	6992	7010	7046	7066	7100
	7118	7146	7162	7192	7201	7216	7222	7252	7267	7289	7298	7320	7329	7350	7359
	7380	7389	7410	7419	7440	7449	7476	7490	7513	7523	7532	7555	7565	7574	7597
	7607	7616	7639	7649	7658	7681	7691	7700	7723	7733	7742	7769	7799	7824	7829
	7842	7867	7873	7893	7898	7966	8020	8049	8072	8112	8134	8170	8182	8203	8215
	8238	8248	8259	8269	8292	8303	8314	8324	8350	8369	8389	8408	8427	8450	8474
	8494	8497	8547	8551	8555	8559	8563	8567	8571	8575	8664	8676	8688	8701	8714
	8727	8739	8752	8764	8785	8800	8815	8830	8845	8861	8871	8881	8891	8900	8910
	8920	8930	8949	8957	8966	8975	8987	8995	9004	9013	9032	9047	9066	9087	9101
	9121	9129	9138	9147	9157	9170	9178	9187	9196	9208	9222	9244	9263	9284	9297
	9317	9325	9334	9343	9353	9366	9374	9383	9392	9404	9424	9442	9456	9475	9496
	9510	9530	9538	9547	9556	9566	9579	9587	9596	9605	9617	9637	9651	9670	9691
	9705	9725	9733	9742	9751	9761	9774	9782	9791	9800	9811	9831	9850	9864	9883
	9905	9919	9939	9947	9956	9965	9975	9988	9996	10005	10014	10026	10053	10067	10086
	10107	10121	10141	10149	10158	10167	10177	10190	10198	10207	10216	10227	10245	10259	10278
	10300	10314	10334	10342	10351	10360	10370	10383	10391	10400	10409	10420	10439	10454	10475
	10487	10493	10509	10524	10539	10554	10569	10584	10590	10609	10633	10656	10672	10676	10698
	10723	10750	10756	10778	10783	10809	10813	10832	10873	10880	10889	10913	10926	10935	10979
	10982	11012	11016	11046	11074	11083	11104	11132	11147	11183	11187	11196	11219	11223	11247
	11272	11276	11306	11309	11314	11323	11332	11363	11392	11458	11463	11467			
ERROR2	11615#	11657	11666	11696	11705	11734	11743	11772	11781	11811	11820	11851	11860	11944	11964
	11972	11993	12001	12022	12030	12051	12059	12080	12088	12109	12117	12138	12146	12167	12175
	12196	12204	12225	12233	12278	12286	12294	12302	12331	12339	12347	12354	12384	12392	12400
	12407	12520	12540	12547	12554	12574	12581	12588	12609	12616	12624	12644	12651	12658	12678
	12685	12692	12713	12720	12728	12748	12755	12762	12782	12789	12796	12803	12828	12835	12843
	12865	12872	12880	12902	12909	12917	12939	12946	12954	12976	12983	12991	13013	13020	13028
	13050	13057	13065	13087	13094	13102	13124	13131	13139	13161	13168	13176	13210	13217	13225
	13245	13252	13260	13280	13287	13295	13316	13323	13331	13352	13359	13367	13387	13394	13402
	13423	13430	13438	13459	13466	13474	13494	13501	13509	13529	13536	13544	13565	13572	13580
	13601	13608	13616	13636	13643	13651	13671	13678	13686	13707	13714	13722	13743	13750	13758
	13778	13785	13793	13814	13821	13829	13854	13861	13869	13889	13896	13904	13924	13931	13939
	13959	13966	13974	13994	14001	14009	14029	14036	14044	14064	14071	14079	14099	14106	14114
	14134	14141	14149	14169	14176	14184	14212	14220	14228	14235	14254	14262	14270	14277	14297
	14305	14313	14321	14340	14348	14356	14363	14382	14390	14398	14405	14425	14433	14441	14449

14468	14476	14484	14491	14510	14518	14526	14533	14552	14560	14568	14575	14595	14603	14611
14619	14638	14646	14654	14661	14680	14688	14696	14703	14723	14731	14739	14747	14766	14774
14782	14789	14808	14816	14824	14831	14852	14861	14881	14889	14912	14920	14928	14935	14954
14962	14970	14977	14996	15004	15012	15019	15038	15046	15054	15061	15080	15088	15096	15103
15122	15130	15138	15145	15164	15172	15180	15187	15206	15214	15222	15229	15248	15256	15264
15271	15290	15298	15306	15313										
11613#	11657	11665	11696	11704	11734	11742	11772	11780	11811	11819	11851	11859	11944	11964
11971	11993	12000	12022	12029	12051	12058	12080	12087	12109	12116	12138	12145	12167	12174
12196	12203	12225	12232	12278	12286	12294	12302	12331	12339	12347	12357	12384	12392	12400
12407	12520	12540	12547	12554	12574	12581	12588	12609	12616	12624	12644	12651	12658	12678
12685	12692	12713	12720	12728	12748	12755	12762	12782	12789	12796	12803	12828	12835	12842
12865	12872	12879	12902	12909	12916	12939	12946	12953	12976	12983	12990	13013	13020	13027
13050	13057	13064	13087	13094	13101	13124	13131	13138	13161	13168	13175	13210	13217	13224
13245	13252	13259	13280	13287	13294	13316	13323	13330	13352	13359	13366	13387	13394	13401
13423	13430	13437	13459	13466	13473	13494	13501	13508	13529	13536	13543	13565	13572	13579
13601	13608	13615	13636	13643	13650	13671	13678	13685	13707	13714	13721	13743	13750	13757
13778	13785	13792	13814	13821	13828	13854	13861	13868	13889	13896	13903	13924	13931	13938
13959	13966	13973	13994	14001	14008	14029	14036	14043	14064	14071	14078	14099	14106	14113
14134	14141	14148	14169	14176	14183	14212	14220	14228	14235	14254	14262	14270	14277	14297
14305	14313	14321	14340	14348	14356	14363	14382	14390	14398	14405	14425	14433	14441	14449
14468	14476	14484	14491	14510	14518	14526	14533	14552	14560	14568	14575	14595	14603	14611
14619	14638	14646	14654	14661	14680	14688	14696	14703	14723	14731	14739	14747	14766	14774
14782	14789	14808	14816	14824	14831	14852	14861	14881	14889	14912	14920	14928	14935	14954
14962	14970	14977	14996	15004	15012	15019	15038	15046	15054	15061	15080	15088	15096	15103
15122	15130	15138	15145	15164	15172	15180	15187	15206	15214	15222	15229	15248	15256	15264
15271	15290	15298	15306	15313										
JNE LOOP	7020#	7946	7971	7976										
	316#	470	490	520	539	558	576	610	629	639	662	683	706	746
	782	800	823	848	917	942	966	991	1016	1041	1066	1091	1125	1143
	1179	1213	1226	1260	1273	1302	1315	1344	1357	1406	1417	1426	1454	1471
	1506	1533	1545	1573	1587	1619	1634	1667	1682	1715	1730	1759	1776	1804
	1829	1839	1861	1872	1885	1895	1917	1932	1941	1976	1989	2025	2042	2078
	2121	2131	2142	2151	2162	2190	2205	2246	2259	2288	2299	2330	2341	2365
	2384	2410	2420	2430	2455	2465	2488	2501	2527	2539	2572	2604	2637	2669
	2718	2729	2761	2771	2786	2797	2830	2841	2877	2892	2903	2934	2944	2977
	3020	3030	3062	3072	3099	3124	3151	3161	3187	3200	3211	3226	3237	3267
	3303	3312	3322	3334	3344	3372	3384	3413	3441	3470	3505	3515	3545	3573
	3613	3622	3651	3661	3688	3714	3740	3763	3773	3796	3806	3828	3839	3849
	3883	3895	3906	3915	3943	3953	3966	3977	4002	4012	4021	4046	4056	4066
	4085	4110	4120	4129	4153	4163	4172	4196	4206	4215	4244	4254	4283	4294
	4330	4340	4351	4361	4370	4397	4407	4416	4426	4436	4464	4474	4483	4512
	4532	4542	4551	4582	4592	4601	4612	4621	4651	4661	4670	4681	4690	4719
	4738	4748	4757	4791	4829	4866	4897	4926	4941	4974	4989	5006	5039	5056
	5104	5118	5132	5165	5201	5231	5261	5289	5298	5325	5353	5362	5389	5417
	5456	5465	5498	5531	5586	5595	5609	5621	5630	5643	5656	5670	5684	5742
	5779	5797	5813	5829	5848	5883	5917	5932	5954	5969	5991	6005	6028	6043
	6093	6108	6131	6145	6159	6175	6208	6231	6246	6268	6283	6314	6328	6342
	6369	6392	6407	6422	6456	6472	6488	6511	6526	6542	6556	6579	6613	6628
	6658	6681	6696	6711	6726	6760	6775	6789	6805	6828	6843	6858	6872	6895
	6924	6941	6964	6980	6995	7013	7049	7069	7103	7121	7149	7165	7204	7225
	7270	7292	7301	7323	7332	7353	7362	7383	7392	7413	7422	7443	7452	7479
	7516	7526	7535	7558	7568	7577	7600	7610	7619	7642	7652	7661	7684	7694
	7726	7736	7745	7772	7832	7901	8023	8075	8115	8137	8173	8185	8206	8218
	8251	8262	8272	8295	8306	8317	8327	8353	8372	8392	8411	8430	8453	8667
	8691	8704	8717	8730	8742	8755	8767	8788	8803	8818	8833	8848	8864	8874
	8894	8903	8913	8923	8933	8952	8961	8970	8979	8990	8999	9008	9017	9050

	9190	9104	9124	9133	9142	9151	9160	9173	9182	9191	9200	9211	9247	9266	9287
	9300	9320	9329	9338	9347	9356	9369	9378	9387	9396	9407	9459	9478	9499	9513
	9533	9542	9551	9560	9569	9582	9591	9600	9609	9620	9654	9673	9694	9708	9728
	9737	9746	9755	9764	9777	9786	9795	9804	9814	9867	9886	9908	9927	9942	9951
	9960	9969	9978	9991	10000	10009	10018	10029	10070	10089	10110	10124	10144	10153	10162
	10171	10180	10193	10202	10211	10220	10230	10262	10281	10303	10317	10337	10346	10355	10364
	10373	10386	10395	10404	10413	10423	10457	10478	10593	10636	10679	10701	10726	10759	10786
	10835	10883	10892	10916	10929	10938	11077	11135	11150	11190	11199	11226	11250	11279	11317
	11326	11335	11366	11470											
	316#	449	508	527	546	565	597	617	646	670	690	734	753	771	789
	807	831	855	879	901	925	950	974	1000	1024	1049	1074	1113	1132	1150
	1168	1200	1247	1289	1331	1396	1443	1485	1522	1560	1605	1652	1700	1743	1789
	1845	1901	1962	2009	2063	2101	2177	2228	2275	2316	2349	2390	2436	2471	2516
	2556	2588	2620	2652	2701	2744	2811	2857	2917	2958	3002	3044	3085	3110	3135
	3176	3249	3290	3355	3398	3426	3455	3488	3529	3559	3599	3635	3675	3701	3727
	3747	3780	3813	3856	3923	3984	4028	4092	4136	4179	4228	4267	4316	4382	4449
	4498	4567	4636	4704	4776	4814	4851	4882	4911	4958	5021	5090	5148	5183	5217
	5246	5278	5312	5340	5377	5404	5443	5485	5518	5573	5708	5864	5903	5939	5976
	6012	6062	6116	6193	6216	6253	6300	6377	6440	6496	6564	6598	6665	6744	6812
	6879	6948	7031	7084	7135	7179	7242	7278	7308	7339	7369	7399	7429	7467	7500
	7542	7584	7626	7668	7710	7760	7785	7812	7854	7878	7931	7988	8037	8060	8096
	8158	8192	8225	8279	8341	8360	8379	8399	8418	8439	8460	8479	8654	8773	8854
	8941	9023	9037	9056	9075	9110	9219	9234	9253	9272	9306	9413	9433	9446	9465
	9484	9519	9628	9641	9660	9679	9714	9820	9841	9854	9873	9892	9928	10044	10057
	10076	10095	10130	10236	10249	10268	10288	10323	10430	10444	10464	10499	10514	10529	10544
	10559	10574	10599	10615	10644	10660	10685	10709	10734	10765	10793	10819	10853	10958	10990
	11023	11058	11087	11112	11159	11205	11236	11263	11294	11345	11382	11397			
POP	330#	15584	15585	15648											
PUSH	330#	15545	15547	15568	15607										
SETRA	15762#	15771	15772	15773	15774										
STARS	316#	332	343	370	372	379	390	392	418	420	425	449	451	497	508
	510	527	529	546	548	565	567	583	597	599	617	619	646	648	670
	672	690	692	714	734	736	753	755	771	773	789	791	807	809	831
	833	855	857	879	881	901	903	925	927	950	952	974	976	1000	1002
	1024	1026	1049	1051	1074	1076	1099	1113	1115	1132	1134	1150	1152	1168	1170
	1187	1200	1202	1234	1247	1249	1281	1289	1291	1323	1331	1333	1365	1380	1384
	1396	1398	1434	1443	1445	1479	1485	1487	1514	1522	1524	1553	1560	1562	1596
	1605	1607	1642	1652	1654	1690	1700	1702	1738	1743	1745	1784	1789	1791	1845
	1847	1901	1903	1948	1962	1964	1997	2009	2011	2050	2063	2065	2101	2103	2169
	2177	2179	2213	2228	2230	2267	2275	2277	2307	2316	2318	2349	2351	2390	2392
	2436	2438	2471	2473	2508	2516	2518	2548	2556	2558	2580	2588	2590	2612	2620
	2622	2645	2652	2654	2694	2701	2703	2736	2744	2746	2804	2811	2813	2848	2857
	2859	2909	2917	2919	2951	2958	2960	2995	3002	3004	3037	3044	3046	3079	3085
	3087	3106	3110	3112	3131	3135	3137	3168	3176	3178	3244	3249	3251	3284	3290
	3292	3350	3355	3357	3391	3398	3400	3420	3426	3428	3448	3455	3457	3477	3488
	3490	3522	3529	3531	3552	3559	3561	3590	3599	3601	3628	3635	3637	3667	3675
	3677	3694	3701	3703	3720	3727	3729	3747	3749	3780	3782	3813	3815	3856	3858
	3923	3925	3984	3986	4028	4030	4092	4094	4136	4138	4179	4181	4222	4228	4230
	4261	4267	4269	4311	4316	4318	4377	4382	4384	4442	4449	4451	4490	4498	4500
	4558	4567	4569	4628	4636	4638	4697	4704	4706	4764	4776	4778	4805	4814	4816
	4841	4851	4853	4872	4882	4884	4903	4911	4913	4949	4958	4960	5014	5021	5023
	5083	5090	5092	5140	5148	5150	5173	5183	5185	5210	5217	5219	5239	5246	5248
	5271	5278	5280	5305	5312	5314	5333	5340	5342	5370	5377	5379	5397	5404	5406
	5434	5443	5445	5476	5485	5487	5508	5518	5520	5542	5573	5575	5692	5708	5710
	5857	5864	5866	5889	5903	5905	5939	5941	5976	5978	6012	6014	6051	6062	6064
	6116	6118	6184	6193	6195	6216	6218	6253	6255	6290	6300	6302	6377	6379	6430

	6442	6496	6498	6564	6566	6588	6598	6600	6665	6667	6734	6744	6746	6812
	6874	6879	6881	6948	6950	7023	7031	7033	7076	7084	7128	7135	7137	7172
	7173	7181	7233	7242	7244	7278	7280	7308	7310	7339	7341	7369	7371	7399
	7429	7431	7459	7467	7469	7500	7502	7542	7544	7584	7586	7626	7628	7668
	7710	7712	7752	7760	7762	7780	7785	7787	7805	7812	7814	7849	7854	7856
	7880	7909	7931	7933	7981	7988	7990	8031	8037	8039	8054	8060	8062	8082
	8078	8148	8159	8161	8192	8194	8225	8227	8279	8281	8334	8341	8343	8360
	8379	8381	8399	8401	8418	8420	8437	8439	8441	8460	8462	8479	8481	8541
	8579#	8655	8657	8773	8775	8854	8856	8941	8943	9023	9025	9037	9039	9056
	9075	9077	9110	9112	9219	9221	9234	9236	9253	9255	9272	9274	9306	9308
	9415	9433	9435	9446	9448	9465	9467	9484	9486	9519	9521	9628	9630	9641
	9660	9662	9679	9681	9714	9716	9820	9822	9841	9843	9854	9856	9873	9875
	9894	9928	9930	10044	10046	10057	10059	10076	10078	10095	10097	10130	10132	10236
	10249	10251	10268	10270	10288	10290	10323	10325	10430	10432	10444	10446	10464	10466
	10501	10514	10516	10529	10531	10544	10546	10559	10561	10574	10576	10599	10601	10615
	10644	10646	10660	10662	10685	10687	10709	10711	10734	10736	10765	10767	10793	10795
	10821	10853	10855	10959	10961	10990	10992	11023	11025	11058	11060	11087	11089	11112
	11159	11161	11206	11208	11236	11238	11263	11265	11294	11296	11345	11347	11382	11384
	11399	15321	15326	15435	15540	15597	15664	15741						11397
TRMTRP	15762#													
TYPDEC	330#													
VTRP	10498#	10499	10514	10529	10544	10559	10574	10599						
SASHC	11619#	12528	12562	12596	12632	12666	12700	12736	12770					
SASHCS	11619#	12816	12853	12890	12927	12964	13001	13038	13075	13112	13149			
SASHS	11619#	11954	11983	12012	12041	12070	12099	12128	12157	12186	12215			
SDIV	13193#	14200	14242	14284	14328	14370	14412	14456	14498	14540	14582	14626	14668	14710
	14796	14838	14868	14900	14942	14984	15026	15068	15110	15152	15194	15236	15278	14754
SMUL	13193#	13200	13235	13270	13305	13342	13377	13412	13449	13484	13519	13554	13591	13626
	13696	13733	13768	13803	13844	13879	13914	13949	13984	14019	14054	14089	14124	14159
SSERCD	316#	463	472	483	492	522	541	560	578	612	631	641	665	685
	748	766	784	802	826	851	874	896	920	945	969	994	1019	1044
	1094	1127	1145	1163	1181	1216	1229	1263	1276	1305	1318	1347	1360	1408
	1428	1456	1474	1497	1509	1535	1548	1575	1590	1621	1637	1669	1685	1717
	1761	1779	1807	1817	1832	1841	1864	1874	1888	1897	1920	1934	1943	1978
	2027	2045	2080	2097	2124	2133	2144	2153	2164	2192	2208	2248	2262	2290
	2332	2344	2368	2377	2386	2413	2422	2432	2458	2467	2491	2503	2529	2542
	2607	2640	2672	2689	2721	2731	2764	2773	2789	2799	2833	2843	2880	2895
	2937	2946	2980	2990	3023	3032	3065	3074	3101	3126	3154	3163	3189	3202
	3228	3239	3270	3280	3305	3314	3324	3336	3346	3375	3387	3415	3443	3472
	3517	3547	3575	3585	3615	3624	3653	3663	3690	3716	3742	3766	3775	3799
	3831	3842	3851	3876	3885	3898	3908	3917	3946	3955	3969	3979	4005	4014
	4049	4058	4069	4078	4087	4113	4122	4131	4156	4165	4174	4199	4208	4217
	4256	4286	4297	4306	4333	4342	4354	4363	4372	4400	4409	4418	4428	4438
	4476	4485	4514	4523	4535	4544	4553	4585	4594	4603	4614	4623	4654	4663
	4683	4692	4722	4731	4740	4750	4759	4794	4832	4868	4899	4929	4944	4977
	5009	5042	5059	5078	5107	5121	5135	5168	5204	5234	5264	5291	5300	5327
	5364	5391	5419	5428	5458	5468	5501	5534	5588	5597	5611	5623	5632	5645
	5672	5686	5726	5745	5764	5782	5800	5816	5832	5851	5876	5885	5920	5935
	5972	5994	6008	6031	6046	6080	6096	6111	6134	6148	6162	6178	6211	6234
	6271	6286	6317	6331	6345	6357	6372	6395	6410	6425	6459	6475	6491	6514
	6545	6559	6582	6616	6631	6646	6661	6684	6699	6714	6729	6763	6778	6792
	6831	6846	6861	6875	6898	6913	6927	6944	6967	6983	6998	7016	7052	7072
	7124	7152	7168	7192	7207	7216	7228	7257	7273	7294	7303	7325	7334	7355
	7385	7394	7415	7424	7445	7454	7481	7495	7519	7528	7537	7561	7570	7579
	7612	7621	7645	7654	7663	7687	7696	7705	7729	7738	7747	7774	7799	7824
	7842	7867	7873	7893	7903	7966	8026	8049	8078	8117	8140	8175	8187	8208



\$\$\$RNU

8243	8253	8264	8274	8297	8308	8319	8329	8355	8374	8394	8413	8432	8455	8474
8494	8498	8548	8552	8556	8560	8564	8568	8572	8576	8669	8681	8693	8706	8719
8732	8744	8757	8769	8790	8805	8820	8835	8850	8866	8876	8886	8896	8905	8915
8925	8935	8954	8963	8972	8981	8992	9001	9010	9019	9033	9052	9071	9092	9106
9126	9135	9144	9153	9162	9175	9184	9193	9202	9214	9230	9249	9268	9289	9302
9322	9331	9340	9349	9358	9371	9380	9389	9398	9409	9425	9442	9461	9480	9501
9515	9535	9544	9553	9562	9571	9584	9593	9602	9611	9623	9637	9656	9675	9696
9710	9730	9739	9748	9757	9766	9779	9788	9797	9806	9816	9832	9850	9869	9888
9910	9924	9944	9953	9962	9971	9980	9993	10002	10011	10020	10032	10053	10072	10091
10112	10126	10146	10155	10164	10173	10182	10195	10204	10213	10222	10232	10245	10264	10283
10305	10319	10339	10348	10357	10366	10375	10388	10397	10406	10415	10425	10439	10459	10480
10487	10494	10510	10525	10540	10555	10570	10585	10595	10610	10639	10656	10672	10681	10703
10728	10750	10761	10778	10788	10810	10814	10838	10874	10885	10894	10918	10931	10941	10979
10983	11013	11017	11047	11079	11083	11105	11137	11153	11183	11192	11202	11219	11228	11253
11272	11281	11306	11310	11319	11328	11337	11369	11392	11458	11463	11473			
3164	463	472	483	492	522	541	560	578	612	631	641	665	685	709
748	766	784	802	826	851	874	896	920	945	969	994	1019	1044	1069
1094	1127	1145	1163	1181	1216	1229	1263	1276	1305	1318	1347	1360	1408	1419
1428	1456	1474	1497	1509	1535	1548	1575	1590	1621	1637	1669	1685	1717	1733
1761	1779	1807	1817	1832	1841	1864	1874	1888	1897	1920	1934	1943	1978	1992
2027	2045	2080	2097	2124	2133	2144	2153	2164	2192	2208	2248	2262	2290	2302
2332	2344	2368	2377	2386	2413	2422	2432	2458	2467	2491	2503	2529	2542	2575
2607	2640	2672	2689	2721	2731	2764	2773	2789	2799	2833	2843	2880	2895	2905
2937	2946	2980	2990	3023	3032	3065	3074	3101	3126	3154	3163	3189	3202	3213
3228	3239	3270	3280	3305	3314	3324	3336	3346	3375	3387	3415	3443	3472	3507
3517	3547	3575	3585	3615	3624	3653	3663	3690	3716	3742	3766	3775	3799	3808
3831	3842	3851	3876	3885	3898	3908	3917	3946	3955	3969	3979	4005	4014	4023
4049	4058	4069	4078	4087	4113	4122	4131	4156	4165	4174	4199	4208	4217	4247
4256	4286	4297	4306	4333	4342	4354	4363	4372	4400	4409	4418	4428	4438	4467
4476	4485	4514	4523	4535	4544	4553	4585	4594	4603	4614	4623	4654	4663	4672
4683	4692	4722	4731	4740	4750	4759	4794	4832	4868	4899	4929	4944	4977	4992
5009	5042	5059	5078	5107	5121	5135	5168	5204	5234	5264	5291	5300	5327	5355
5364	5391	5419	5428	5458	5468	5501	5534	5588	5597	5611	5623	5632	5645	5658
5672	5686	5726	5745	5764	5782	5800	5816	5832	5851	5876	5885	5920	5935	5957
5972	5994	6008	6031	6046	6080	6096	6111	6134	6148	6162	6178	6211	6234	6249
6271	6286	6317	6331	6345	6357	6372	6395	6410	6425	6459	6475	6491	6514	6529
6545	6559	6582	6616	6631	6646	6661	6684	6699	6714	6729	6763	6778	6792	6808
6831	6846	6861	6875	6898	6913	6927	6944	6967	6983	6998	7016	7052	7072	7106
7124	7152	7168	7192	7207	7216	7228	7257	7273	7294	7303	7325	7334	7355	7364
7385	7394	7415	7424	7445	7454	7481	7495	7519	7528	7537	7561	7570	7579	7603
7612	7621	7645	7654	7663	7687	7696	7705	7729	7738	7747	7774	7799	7824	7834
7842	7867	7873	7893	7903	7966	8026	8049	8078	8117	8140	8175	8187	8208	8220
8243	8253	8264	8274	8297	8308	8319	8329	8355	8374	8394	8413	8432	8455	8474
8494	8498	8548	8552	8556	8560	8564	8568	8572	8576	8669	8681	8693	8706	8719
8732	8744	8757	8769	8790	8805	8820	8835	8850	8866	8876	8886	8896	8905	8915
8925	8935	8954	8963	8972	8981	8992	9001	9010	9019	9033	9052	9071	9092	9106
9126	9135	9144	9153	9162	9175	9184	9193	9202	9214	9230	9249	9268	9289	9302
9322	9331	9340	9349	9358	9371	9380	9389	9398	9409	9425	9442	9461	9480	9501
9515	9535	9544	9553	9562	9571	9584	9593	9602	9611	9623	9637	9656	9675	9696
9710	9730	9739	9748	9757	9766	9779	9788	9797	9806	9816	9832	9850	9869	9888
9910	9924	9944	9953	9962	9971	9980	9993	10002	10011	10020	10032	10053	10072	10091
10112	10126	10146	10155	10164	10173	10182	10195	10204	10213	10222	10232	10245	10264	10283
10305	10319	10339	10348	10357	10366	10375	10388	10397	10406	10415	10425	10439	10459	10480
10487	10494	10510	10525	10540	10555	10570	10585	10595	10610	10639	10656	10672	10681	10703
10728	10750	10761	10778	10788	10810	10814	10838	10874	10885	10894	10918	10931	10941	10979
10983	11013	11017	11047	11079	11083	11105	11137	11153	11183	11192	11202	11219	11228	11253

	11272	11281	11306	11310	11319	11328	11337	11369	11392	11458	11463	11473			
\$\$\$	316#	487	517	536	555	573	607	626	636	659	680	703	743	761	779
	797	820	845	914	939	963	988	1013	1038	1063	1088	1122	1140	1158	1176
	1223	1270	1312	1354	1423	1468	1503	1542	1584	1631	1679	1727	1773	1836	1892
	1938	1986	2039	2091	2159	2202	2256	2296	2338	2381	2427	2462	2498	2536	2569
	2601	2634	2683	2726	2794	2838	2900	2941	2985	3027	3069	3096	3121	3158	3234
	3275	3341	3381	3410	3438	3467	3512	3542	3580	3619	3658	3685	3711	3737	3770
	3803	3846	3912	3974	4018	4082	4126	4169	4212	4251	4301	4367	4433	4480	4548
	4618	4687	4754	4788	4826	4863	4894	4938	5003	5072	5129	5162	5198	5228	5258
	5295	5322	5359	5386	5423	5462	5495	5528	5681	5845	5880	5929	5966	6002	6040
	6105	6172	6205	6243	6280	6366	6419	6485	6553	6576	6655	6723	6802	6869	6938
	7010	7066	7118	7162	7222	7267	7298	7329	7359	7389	7419	7449	7490	7532	7574
	7616	7658	7700	7742	7769	7898	8072	8182	8215	8269	8324	8350	8369	8389	8408
	8427	8450	8764	8845	8930	9014	9047	9066	9101	9244	9263	9297	9404	9456	9475
	9510	9651	9670	9705	9811	9864	9883	9919	10067	10086	10121	10227	10259	10278	10314
	10420	10454	10590	10676	10698	10723	10783	11074	11223	11247	11276				
\$\$\$LOOP	316#	470	490	520	539	558	576	610	629	639	662	683	706	746	764
	782	800	823	848	877	942	966	991	1016	1041	1066	1091	1125	1143	1161
	1179	1213	1226	1260	1273	1302	1315	1344	1357	1406	1417	1426	1454	1471	1495
	1506	1533	1545	1573	1587	1619	1634	1667	1682	1715	1730	1759	1776	1804	1815
	1829	1839	1861	1872	1885	1895	1917	1932	1941	1976	1989	2025	2042	2078	2094
	2121	2131	2142	2151	2162	2190	2205	2246	2259	2288	2299	2330	2341	2365	2375
	2384	2410	2420	2430	2455	2465	2488	2501	2527	2539	2572	2604	2637	2669	2686
	2718	2729	2761	2771	2786	2797	2830	2841	2877	2892	2903	2934	2944	2977	2988
	3020	3030	3062	3072	3099	3124	3151	3161	3187	3200	3211	3226	3237	3267	3278
	3303	3312	3322	3334	3344	3372	3384	3413	3441	3470	3505	3515	3545	3573	3583
	3613	3622	3651	3661	3688	3714	3740	3763	3773	3796	3806	3828	3839	3849	3873
	3883	3895	3906	3915	3943	3953	3966	3977	4002	4012	4021	4046	4056	4066	4076
	4085	4110	4120	4129	4153	4163	4172	4196	4206	4215	4244	4254	4283	4294	4304
	4330	4340	4351	4361	4370	4397	4407	4416	4426	4436	4464	4474	4483	4512	4521
	4532	4542	4551	4582	4592	4601	4612	4621	4651	4661	4670	4681	4690	4719	4729
	4738	4748	4757	4791	4829	4866	4897	4926	4941	4974	4989	5006	5039	5056	5075
	5104	5118	5132	5165	5201	5231	5261	5289	5298	5325	5353	5362	5389	5417	5426
	5456	5465	5498	5531	5586	5595	5609	5621	5630	5643	5656	5670	5684	5742	5761
	5779	5797	5813	5829	5848	5883	5917	5932	5954	5969	5991	6005	6028	6043	6077
	6093	6108	6131	6145	6159	6175	6208	6231	6246	6268	6283	6314	6328	6342	6354
	6369	6392	6407	6422	6456	6472	6488	6511	6526	6542	6556	6579	6613	6628	6643
	6658	6681	6696	6711	6726	6760	6775	6789	6805	6828	6843	6858	6872	6895	6910
	6924	6941	6964	6980	6995	7013	7049	7069	7103	7121	7149	7165	7204	7225	7255
	7270	7292	7301	7323	7332	7353	7362	7383	7392	7413	7422	7443	7452	7479	7493
	7516	7526	7535	7558	7568	7577	7600	7610	7619	7642	7652	7661	7684	7694	7703
	7726	7736	7745	7772	7832	7901	8023	8075	8115	8137	8173	8185	8206	8218	8241
	8251	8262	8272	8295	8306	8317	8327	8353	8372	8392	8411	8430	8453	8667	8679
	8691	8704	8717	8730	8742	8755	8767	8788	8803	8818	8833	8848	8864	8874	8884
	8894	8903	8913	8923	8933	8952	8961	8970	8979	8990	8999	9008	9017	9050	9069
	9090	9104	9124	9133	9142	9151	9160	9173	9182	9191	9200	9211	9247	9266	9287
	9300	9320	9329	9338	9347	9356	9369	9378	9387	9396	9407	9459	9478	9499	9513
	9533	9542	9551	9560	9569	9582	9591	9600	9609	9620	9654	9673	9694	9708	9728
	9737	9746	9755	9764	9777	9786	9795	9804	9814	9867	9886	9908	9922	9942	9951
	9960	9969	9978	9991	10000	10009	10018	10029	10070	10089	10110	10124	10144	10153	10162
	10171	10180	10193	10202	10211	10220	10230	10262	10281	10303	10317	10337	10346	10355	10364
	10373	10386	10395	10404	10413	10423	10457	10478	10593	10636	10679	10701	10726	10759	10786
	10835	10883	10892	10916	10929	10938	11077	11135	11150	11190	11199	11226	11250	11279	11317
	11326	11335	11366	11470											
\$\$\$NEXT	316#	449	508	527	546	565	597	617	646	670	690	734	753	771	789
	807	831	855	879	901	925	950	974	1000	1024	1049	1074	1113	1132	1150



1168	1200	1247	1289	1331	1396	1443	1485	1522	1560	1605	1652	1700	1743	1787
1845	1901	1962	2009	2063	2101	2177	2228	2275	2316	2349	2390	2436	2471	2516
2556	2588	2620	2652	2701	2744	2811	2857	2917	2958	3002	3044	3085	3110	3135
3176	3249	3290	3355	3398	3426	3455	3488	3529	3559	3599	3635	3675	3701	3727
3747	3780	3813	3856	3923	3984	4028	4092	4136	4179	4228	4267	4316	4382	4449
4498	4567	4636	4704	4776	4814	4851	4882	4917	4958	5021	5090	5148	5183	5217
5246	5278	5312	5340	5377	5404	5443	5485	5518	5573	5708	5864	5903	5939	5976
6012	6062	6116	6193	6216	6253	6300	6377	6440	6496	6564	6598	6665	6744	6812
6879	6948	7031	7084	7135	7179	7242	7278	7308	7339	7367	7399	7429	7467	7500
7542	7584	7626	7668	7710	7760	7785	7812	7854	7878	7931	7988	8037	8060	8096
8159	8192	8225	8279	8341	8360	8379	8399	8418	8439	8460	8479	8655	8773	8854
8941	9023	9037	9056	9075	9110	9219	9234	9253	9272	9306	9413	9433	9446	9465
9484	9519	9628	9641	9660	9679	9714	9820	9841	9854	9873	9892	9928	10044	10057
10076	10095	10130	10236	10249	10268	10288	10323	10430	10444	10464	10499	10514	10529	10544
10559	10574	10599	10615	10644	10660	10685	10709	10734	10765	10793	10819	10853	10959	10990
11023	11058	11087	11112	11159	11206	11236	11263	11294	11345	11382	11397			
\$\$SET	15762#	15771	15772	15773	15774									
.SETUP	330#	11610												
.SACT1	330#													
.SAPT8	330#	341												
.SAPTH	330#	368												
.SAPTY	330#	15538												
.SEOP	330#													
.STRAP	330#	15739												
.STYPD	330#	15595												
.STYPE	330#	15433												
.STYPO	330#	15662												

. ABS. 062542 000

ERRORS DETECTED: 0

DSKZ:CJKDBD, DSKZ:CJKDBD/CRF/SOL/NL:TOC=CJKDBD.P11  
 RUN-TIME: 136 173 18 SECONDS  
 RUN-TIME RATIO: 437/329=1.3  
 CORE USED: 25K (49 PAGES)