

DCF11-AA

DCF11-AA DIAG
CJKDBCO

AH-F140C-MC
FICHE 1 OF 2

MAY 1980
COPYRIGHT © 1979
MADE IN USA



A large grid of technical data, likely a diagnostic chart or table, covering the majority of the page. The text is extremely faint and illegible due to the low resolution and high contrast of the scan. The grid appears to be organized into multiple columns and rows, with some larger, more prominent text blocks interspersed.



DCF11-AA

DCF11-AA DIAG
CJKDBCO

AH-F140C-MC
FICHE 2 OF 2

MAY 1980
COPYRIGHT © 1979
MADE IN USA



The main body of the document is a microfiche card containing a grid of approximately 10 columns and 20 rows of data. Each cell in the grid contains a small, high-contrast image, likely a scan of a document page or a specific data point. The images are too small to read clearly but appear to be organized in a structured manner.



000000

.REPT 0

IDENTIFICATION

PRODUCT CODE: AC-F141C-MC

PRODUCT NAME: CJKDBC0 DCF11-AA DIAG

DATE: NOV-79

MAINTAINER: DIAGNOSTIC ENGINEERING

COPYRIGHT (C) 1979 DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

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

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

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

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 CHECK OUT THE BASIC PDP-11 INSTRUCTIONS IN EVERY ADDRESSING MODES WITH VARIOUS TYPES OF DATA PATTERNS.

PART TWO:

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

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

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

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

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

IF A HALT OCCURS IN THE TRAP OR INTERRUPT VECTOR AREA, REGISTER SIX SHOULD BE EXAMINED TO DETERMINE ITS CONTENTS, THEN USE REGISTER SIX CONTENTS AS AN ADDRESS TO DETERMINE WHERE THE PROGRAM WAS. WHEN THE INTERRUPT OR TRAP OCCURRED; MEMORY AS SPECIFIED BY R6 CONTAINS THE PC OF THE INSTRUCTION FOLLOWING THE INSTRUCTION WHERE THE TRAP OCCURRED.

THE CONTENTS OF LOCATION '\$TESTN'(304) CONTAINS THE TEST NUMBER THAT IT WAS DOING BEFORE IT TRAPPED.

6.1 ERROR HANDLING IN PART ONE AND PART TWO

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

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

EXAMPLE

```
TSTA:  INC      (R2)           ;INCREMENT THE TEST NUMBER  
        CMP     #A,(R2)       ;COMPARE FOR THE RIGHT TEST  
        BNF    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:

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! PC AND ERROR # ARE:
PC #
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
1	000002.....	CIS CHIP SET PRESENT
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. THE COMMENTS BESIDE THE CALL TO THE HALT SUBROUTINE

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

TELLS WHAT WAS BEING TESTED AND WHAT WAS EXPECTED. ALL
PRINTOUTS WILL BE SUPPRESSED WHEN BIT 5 OF LOCATION
\$ENVN IS HIGH.
WHILE RUNNING UNDER APT THE DIAGNOSTIC WILL NOT
SUPPORT SPOOLING OF CONSOLE OUTPUTS.

9.2 POWER FAIL ROUTINE

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

.ENDR

312
313
314
315
316
317
318
319
320
321 000240
322 000007
323 000006
324 177776
325 177560
326 177562
327 177564
328 177566
329 140000
330 030000
331
332
333
334
335 000400
336 000046
337 000046 060462
338 000052
339 000052 000000
340 000400
341 000300
342
343
344
345
346 000300
347 000300 000000
348 000302 000000
349 000304 000000
350 000306 000000
351 000310 000000
352 000312 000000
353 000314 000000
354 000316 000000
355 000320
356 000320 000
357 000321 000
358 000322 000000
359 000324 000000
360 000326 000000
361
362
363
364
365
366
367 000330

;PROGRAMMER: KIN C. LEE

.TITLE CJKDB-C DCF11-AA CPU DIAG.
.ENABLE ABS
.NLIST CND,MC,MD
.LIST ME
SCOPE=NOP
R7=%7
R6=%6
PS=177776
TKS=177560
TKB=177562
TPS=177564
TPB=177566
USRM=140000
PUSRM=30000
.SBTTL ACT11 HOOKS

;HOOKS REQUIRED BY ACT11

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

.SBTTL APT MAILBOX-ETABLE

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

```
368 .MEXIT
369 .SBTTL APT PARAMETER BLOCK
370
371 ;*****
372 ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
373 ;*****
374 000330 .$.X- ;;SAVE CURRENT LOCATION
375 000024 -24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM
376 000200 200 ;;FOR APT START UP
377 000044 -44 ;;POINT TO APT INDIRECT ADDRESS PNTR.
378 000330 $APTHDR ;;POINT TO APT HEADER BLOCK
379 000330 .=$.X ;;RESET LOCATION COUNTER
380 ;*****
381 ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
382 ;INTERFACE SPEC.
383
384 000330 $APTHD:
385 000330 000000 $HIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
386 000332 000300 $MBADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
387 000334 000013 $TSTM: .WORD 13 ;;RUN TIM OF LONGEST TEST
388 000336 000020 $PASTM: .WORD 20 ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
389 000340 000005 $UNITM: .WORD 5 ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
390 000342 000014 .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
391 ;*****
392 ;SOME POINTERS TO CPU TRAP HANDLERS
393 ;*****
394 000004 .=4
395 000004 T04
396 000006 0
397 000010 T010
398 000012 0
399 000014 T014
400 000030 .=30
401 000030 T030
402 000032 0
403 000034 T034
404 000036 0
405 000114 .=114
406 000114 T0114
407 000116 0
408 000244 .=244
409 000244 T0244
410 000246 0
411 000250 T0250
412 000252 0
413
414 000172 .=172
415 000172 000000 LPADR: 0 ;;LOOP ADDRESS (EIS TEST)
416 000174 000000 DISPREG: 0 ;;SOFTWARE DISPLAY REGISTER
417 000176 000000 SWREG: 0 ;;SOFTWARE SWITCH REGISTER
418
419 ;*****
420 ;DATA TABLE FOR USE IN ADDRESSING MODE TESTS
421 ;*****
422 000370 000370 000000 000000 .=370
423 000370 000000 000000 000000 0,0,0,0,0,0
```

```
424 000376 000000 000000 000000
425 000404 000001 000001 177777      1,1,-1
426
427 ;*****
428 ;SET UP STARTING ADDRESS
429 $ERROR=$FATAL
430 $STSTM=$TESTN
431 001000 000000      .=1000
432 STBOT: .WORD 0      ;STACK POINTER
433
434 000200 000167 000576      .=200
435 000204 012706 001000      JMP START
436 000210 012702 000304      MOV #STBOT,R6      ;SET STACK POINTER
437 000214 000137      MOV #STESTN,R2     ;SET MAILBOX POINTER
438 000216 000000      JMP @PC+           ;JUMP TO SUBTEST
439
440
441      .=1002
442 001002 012737 061002 000024      SBTTL **STARTING OF CPU TEST **
443 001010 012737 000000 000306      START: MOV #PWRDN,@#24      ;SET UP FOR POWER FAIL
444 001016 012737 000016 062370      MOV #0,@SPASS      ;CLEAR PASS COUNT
445 001024 012706 001000      MOV #16,@PASSPT    ;SET PRINT COUNTER
446 001030 012702 000304      RESTRT: MOV #STBOT,R6 ;INITIALIZE STACK POINTER
447 001034 012737 000000 000304      MOV #STESTN,R2     ;SET UP POINTER TO MESSAGE TYPE
448 001042 012737 000000 000302      MOV #0,@STSTM      ;CLEAR TEST NUMBER
449 001050 012737 000000 000300      MOV #0,@$ERROR     ;CLEAR ERROR NUMBER
450
451      MOV #0,@$MSGTY   ;CLEAR MESSAGE TYPE(FOR APT)
452 ;*****
453 ;TEST 1 CHECK BRANCHES ON Z BIT
454 ;*****
455 TS1: INC (R2)      ;UPDATE TEST NUMBER
456 CMP #1,(R2)      ;SEQUENCE ERROR?
457 BNE TS2-10 ;BR TO ERROR HALT ON SEQ ERROR
458 CCC             ;CLEAR ALL CONDITION CODES
459 BEQ BRA1        ;SHOULD BRANCH
460 BR BRA2         ;BAD BRANCH OF Z-BIT
461
462 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
463 ; BRANCH INSTRUCTION AND <====
464 ; REPLACE THE MOVE INSTRUCTION <====
465 ; FOLLOWING W/ 774 <====
466
467 BRA1: MOV #1,-(R2) ;MOVE TO MAILBOX # ***** 1 *****
468 INC -(R2)        ;SET MSGTYP TO FATAL ERROR
469 HALT             ;SHOULD HAVE BRANCHED: Z=0
470
471 BRA2: BNE BRA3
472
473 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
474 ; CONDITIONAL BRANCH INST. AND <====
475 ; REPLACE THE MOVE INSTRUCTION <====
476 ; WHICH FOLLOWS W/ 767 <====
477
478 BRA3: MOV #2,-(R2) ;MOVE TO MAILBOX # ***** 2 *****
479 INC -(R2)        ;SET MSGTYP TO FATAL ERROR
480 HALT
481 SEZ
482 BNE BRA4
483 BR BRA5
484
485 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
```

```
480  
481  
482  
483 001124  
484 001124 012742 000003  
485 001130 005242  
486 001132 000000  
487 001134  
488 001134 001404  
489  
490  
491  
492  
493 001136 012742 000004  
494 001142 005242  
495 001144 000000  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512 001146 005212  
513 001150 022712 000002  
514 001154 001006  
515 001156 012737 000000 000000  
516  
517 001164 005737 000000  
518 001170 001404  
519  
520  
521  
522  
523 001172 012742 000005  
524 001176 005242  
525 001200 000000  
526  
527  
528  
529  
530  
531 001202 005212  
532 001204 022712 000003  
533 001210 001007  
534 001212 012737 125252 000000  
535
```

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

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
```

```
536 001220 022737 125252 000000    CMP    #125252,@#0    ;SUCCESSFUL
537 001226 001404                BEQ    TS4
538                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
539                                ; CONDITIONAL BRANCH INST. AND <====
540                                ; REPLACE THE MOVE INSTRUCTION <====
541                                ; WHICH FOLLOWS W/ 770 <====
542 001230 012742 000006    MOV    #6,-(R2)      ;MOVE TO MAILBOX # ***** 6 *****
543 001234 005242                INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
544 001236 000000                HALT                ;DATA INCORRECT
545                                ; OR SEQUENCE ERROR
```

```
547
548 ;*****
549 ;TEST 4 TEST OF PATTERN 052525 IN DATA PATH
550 ;*****
```

```
550 001240 005212                TS4:  INC    (R2)      ;UPDATE TEST NUMBER
551 001242 022712 000004                CMP    #4,(R2)      ;SEQUENCE ERROR?
552 001246 001007                BNE    TS5-10 ;BR TO ERROR HALT ON SEQ ERROR
553 001250 012737 052525 000000    MOV    #052525,@#0 ;MOVE ALTERNATING ZEROES AND ONES
554                                ;THRU DATA PATH
555 001256 022737 052525 000000    CMP    #052525,@#0 ;SUCCESSFUL?
556 001264 001404                BEQ    TS5
557                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
558                                ; CONDITIONAL BRANCH INST. AND <====
559                                ; REPLACE THE MOVE INSTRUCTION <====
560                                ; WHICH FOLLOWS W/ 770 <====
561 001266 012742 000007    MOV    #7,-(R2)      ;MOVE TO MAILBOX # ***** 7 *****
562 001272 005242                INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
563 001274 000000                HALT                ;DATA INCORRECT
564                                ; OR SEQUENCE ERROR
```

```
566 ;*****
567 ;TEST 5 TEST OF ALL ONES IN DATA PATH
568 ;*****
```

```
569 001276 005212                TS5:  INC    (R2)      ;UPDATE TEST NUMBER
570 001300 022712 000005                CMP    #5,(R2)      ;SEQUENCE ERROR?
571 001304 001007                BNE    TS6-10 ;BR TO ERROR HALT ON SEQ ERROR
572 001306 012737 177777 000000    MOV    #177777,@#0 ;MOVE ONES THRU DATA PATH
573 001314 022737 177777 000000    CMP    #177777,@#0 ;SUCCESSFUL
574 001322 001404                BEQ    TS6
575                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
576                                ; CONDITIONAL BRANCH INST. AND <====
577                                ; REPLACE THE MOVE INSTRUCTION <====
578                                ; WHICH FOLLOWS W/ 770 <====
579 001324 012742 000010    MOV    #10,-(R2)     ;MOVE TO MAILBOX # ***** 10 *****
580 001330 005242                INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
581 001332 000000                HALT                ;DATA INCORRECT
582                                ; OR SEQUENCE ERROR
```

```
584 ;*****
585 ;SBTTL B-REGISTER TEST
586 ;
587 ; THE B-REGISTER (LOCATION 0) SHIFTING LOGIC TESTS ARE USED
588 ; TO TEST THAT THE B-REGISTER CAN HOLD VARIOUS DATA PATTERNS AND THAT
589 ; THE ASSOCIATED LOGIC SUPPORTS THE SHIFTING FUNCTIONS WITHIN THE
590 ; B-REGISTER AND C-BIT.
591 ; A ONE IS SHIFTED THROUGH EVERY BIT IN THE B-REGISTER AND C-BIT IN
```

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

```
; BOTH DIRECTIONS.  
; THE B-REGISTER ITSELF IS TESTED IN ITS ABILITY AS A BUFFER AND AS  
; A SHIFT REGISTER. DATA IS ALSO PASSED THROUGH THE DATA PATH AND ALU,  
; IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0) TO SEE  
; WHICH BITS OF THE B-REGISTER MAY BE FAILING.  
;*****  
; TEST 6 SHIFT BIT 0 TO BIT 1  
;*****  
TS6:  INC      (R2)          ; UPDATE TEST NUMBER  
      CMP      #6,(R2)     ; SEQUENCE ERROR?  
      BNE     TS7-10      ; BR TO ERROR HALT ON SEQ ERROR  
      CLC                    ; CLEAR CARRY BIT  
      MOV     #1,@#0      ; LOAD A 1  
      ROL     @#0         ; SHIFT LEFT  
      CMP     #2,@#0      ; SUCCESSFUL  
      BEQ     TS7  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
;          CONDITIONAL BRANCH INST. AND <=====  
;          REPLACE THE MOVE INSTRUCTION <=====  
;          WHICH FOLLOWS W/ 765 <=====  
      MOV     #11,-(R2)    ; MOVE TO MAILBOX # ***** 11 *****  
      INC     -(R2)        ; SET MSGTYP TO FATAL ERROR  
      HALT                    ; BIT 1 NOT SET  
;          OR SEQUENCE ERROR  
;*****  
; TEST 7 SHIFT CARRY INTO BIT 0  
;*****  
TS7:  INC      (R2)          ; UPDATE TEST NUMBER  
      CMP     #7,(R2)     ; SEQUENCE ERROR?  
      BNE     TS10-10     ; BR TO ERROR HALT ON SEQ ERROR  
      MOV     #0,@#0      ; CLEAR LOCATION  
      SEC                    ; SET CARRY  
      ROL     @#0         ; ROTATE CARRY BIT TO BIT 0  
      BCC     TS10  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
;          CONDITIONAL BRANCH INST. AND <=====  
;          REPLACE THE MOVE INSTRUCTION <=====  
;          WHICH FOLLOWS W/ 770 <=====  
      MOV     #12,-(R2)   ; MOVE TO MAILBOX # ***** 12 *****  
      INC     -(R2)        ; SET MSGTYP TO FATAL ERROR  
      HALT                    ; CARRY CLEAR  
;          OR SEQUENCE ERROR  
      CMP     #1,@#0      ; BIT 0 SET  
      BEQ     TS10  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
;          CONDITIONAL BRANCH INST. AND <=====  
;          REPLACE THE MOVE INSTRUCTION <=====  
;          WHICH FOLLOWS W/ 760 <=====  
      MOV     #13,-(R2)   ; MOVE TO MAILBOX # ***** 13 *****  
      INC     -(R2)        ; SET MSGTYP TO FATAL ERROR  
      HALT                    ; BIT 0 NOT SET  
;          OR SEQUENCE ERROR  
;*****
```

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


704 001630 001404 BEQ TS13
705 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
706 ; CONDITIONAL BRANCH INST. AND <====
707 ; REPLACE THE MOVE INSTRUCTION <====
708 ; WHICH FOLLOWS W/ 763 <====

709 001632 SHRE:
710 001632 012742 000016 MOV #16,-(R2) ;MOVE TO MAILBOX # ***** 16 *****
711 001636 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
712 001640 000000 HALT ;RIGHT SHIFT LOGIC FAILED
713 ; OR SEQUENCE ERROR

714
715
716 :*****
717 :SBTTL SCRATCH PAD TESTS

718 :
719 : THE SCRATCH PAD TESTS ARE USED TO VERIFY THAT VARIOUS
720 : DATA PATTERNS CAN BE SUCCESSFULLY HELD IN THE SCRATCH PAD
721 : CIRCUITRY. MOVE AND COMPARE INSTRUCTIONS ARE USED TO TEST THAT
722 : R0 CAN HOLD VARIOUS DATA PATTERNS. EACH DATA PATTERN IS
723 : MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR SCOPING. THE
724 : SUCCESSFUL COMPLETION OF THESE TESTS SHOULD VERIFY THE CIRCUITRY EXTERNAL
725 : TO THE SCRATCH PAD ITSELF.
726 : THE REMAINDER OF THE GENERAL REGISTERS ARE TESTED BY MOVING
727 : A BIT INTO BIT 0 OF THE REGISTER AND SHIFTING IT LEFT ONE
728 : BIT AT A TIME INTO THE CARRY BIT. THE RESULT IS THEN CHECKED TO INSURE THAT
729 : NO BITS WERE PICKED. THE PROCEDURE IS THEN REPEATED UNDER OPPOSITE
730 : CONDITIONS. THE GENERAL REGISTER AND THE CARRY BIT ARE SET TO
731 : ALL ONES, AND A ZERO IS SHIFTED LEFT FROM BIT 0 INTO THE CARRY BIT.
732 : THE RESULT IS THEN CHECKED TO INSURE THAT NO ZEROES WERE PICKED.
733 : AT THIS POINT ALL OF THE GENERAL REGISTERS HAVE BEEN EXERCISED
734 : AS WELL AS REGISTER 11.

735 :*****
736 :TEST 13 TEST IF R0 CAN HOLD ALL ZEROES
737 :*****

738 001642 005212 TS13: INC (R2) ;UPDATE TEST NUMBER
739 001644 022712 000013 CMP #13,(R2) ;SEQUENCE ERROR?
740 001650 001004 BNE TS14-10 ;BR TO ERROR HALT ON SEQ ERROR
741
742 001652 012700 000000 MOV #0,R0 ;MOVE ZEROES TO R0
743 001656 005700 TST R0 ;SUCCESSFUL?
744 001660 001404 BEQ TS14

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

749 001662 012742 000017 MOV #17,-(R2) ;MOVE TO MAILBOX # ***** 17 *****
750 001666 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
751 001670 000000 HALT ;R0 NOT 0
752 ; OR SEQUENCE ERROR

753 :*****
754 :TEST 14 TEST IF R0 CAN HOLD ONES AND ZEROES
755 :*****

756
757 001672 005212 TS14: INC (R2) ;UPDATE TEST NUMBER
758 001674 022712 000014 CMP #14,(R2) ;SEQUENCE ERROR?
759 001700 001005 BNE TS15-10 ;BR TO ERROR HALT ON SEQ ERROR

```
760 001702 012700 125252      MOV      #125252,R0      ;MOVE ALTERNATING ONES AND ZEROS TO R0
761 001706 020027 125252      CMP      R0,#125252     ;SUCCESSFUL?
762 001712 001404              BEQ      TS15
763                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
764                                ;          CONDITIONAL BRANCH INST. AND <====
765                                ;          REPLACE THE MOVE INSTRUCTION <====
766                                ;          WHICH FOLLOWS W/ 772 <====
767 001714 012742 000020      MOV      #20,-(R2)      ;MOVE TO MAILBOX # ***** 20 *****
768 001720 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
769 001722 000000              HALT                    ;R0 NOT 125252
770                                ; OR SEQUENCE ERROR
```

```
*****
;TEST 15      TEST IF R0 CAN HOLD ZEROS AND ONES
*****
```

```
774 TS15: INC      (R2)          ;UPDATE TEST NUMBER
775 001724 005212              CMP      #15,(R2)      ;SEQUENCE ERROR?
776 001726 022712 000015      BNE     TS16-10        ;BR TO ERROR HALT ON SEQ ERROR
777 001732 001005              MOV      #052525,R0    ;MOVE ALTERNATING ZEROS AND ONES TO R0
778 001734 012700 052525      CMP      R0,#052525   ;SUCCESSFUL?
779 001740 020027 052525      BEQ      TS16
780 001744 001404              ; 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                    ;R0 NOT 52525
787                                ; OR SEQUENCE ERROR
```

```
*****
;TEST 16      TEST IF R0 CAN HOLD ALL ONES
*****
```

```
792 TS16: INC      (R2)          ;UPDATE TEST NUMBER
793 001756 005212              CMP      #16,(R2)      ;SEQUENCE ERROR?
794 001760 022712 000016      BNE     TS17-10        ;BR TO ERROR HALT ON SEQ ERROR
795 001764 001005              MOV      #177777,R0    ;MOVE ALL ONES TO R0
796 001766 012700 177777      CMP      R0,#177777   ;SUCCESSFUL?
797 001772 020027 177777      BEQ      TS17
798 001776 001404              ; 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                    ;R0 NOT 177777
805                                ; OR SEQUENCE ERROR
```

```
*****
;TEST 17      TEST IF R1 CAN HOLD A ONE IN ALL BITS
*****
```

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

```
816 002030 000241
817 002032 005200
818 002034 001403
819 002036 006101
820 002040 103374
821 002042 001404
822
823
824
825
826 002044
827 002044 012742 000023
828 002050 005242
829 002052 000000
830
831
832
833
834
835 002054 005212
836 002056 022712 000020
837 002062 001014
838 002064 012701 177776
839 002070 012700 177757
840 002074 000261
841 002076 005200
842 002100 001405
843 002102 006101
844 002104 103774
845 002106 022701 177777
846 002112 001404
847
848
849
850
851 002114
852 002114 012742 000024
853 002120 005242
854 002122 000000
855
856
857
858
859 002124 005212
860 002126 022712 000021
861 002132 001012
862 002134 012702 000001
863 002140 012700 177757
864 002144 000241
865 002146 005200
866 002150 001403
867 002152 006102
868 002154 103374
869 002156 001406
870
871
```

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

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

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

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

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

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

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

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; BRANCH INSTRUCTION AND <---
```

```
872
873
874 002160 012702 000304      MOV    # $TESTN,R2      ;
875 002164 012742 000025      MOV    #25,-(R2)        ;REPLACE THE MOVE INSTRUCTION <==--
876 002170 005242              INC    -(R2)            ;FOLLOWING W/ 771 <====
877 002172 000000              HALT                   ;
878 002174 012702 000304      REG2A: MOV # $TESTN,R2  ;RESTORE POINTER
879
880
881 ;*****
882 ;TEST 22      TEST IF R2 CAN HOLD A ZERO IN ALL BITS
883 002200 005212              TS22: INC (R2)          ;UPDATE TEST NUMBER
884 002202 022712 000022      CMP    #22,(R2)        ;SEQUENCE ERROR?
885 002206 001020              BNE    TS23-10         ;BR TO ERROR HALT ON SEQ ERROR
886 002210 012702 177776      MOV    #-2,R2          ;SET ALL ONES IN R2 EXCEPT FOR BIT 0
887 002214 012700 177757      MOV    #-21,R0         ;SET BIT COUNTER
888 002220 000261              SEC                   ;SET C-BIT
889 002222 005200              REG2B: INC R0           ;INCREMENT BIT COUNTER
890 002224 001407              BEQ    R2ERR           ;BR TO ERROR HALT IF COUNTER=0
891 002226 006102              ROL    R2              ;ROTATE 1 POSITION
892 002230 103774              BCS    REG2B           ;CONTINUE UNTIL C-BIT IS CLEAR
893 002232 022702 177777      CMP    #-1,R2          ;CHECK DATA IN R2
894 002236 001406              BEQ    REG2C           ;
895 002240 012702 000304      REG2C: MOV # $TESTN,R2  ;RESTORE POINTER
896 002244
897 002244 012742 000026      MOV    #26,-(R2)       ;MOVE TO MAILBOX # ***** 26 *****
898 002250 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
899 002252 000000              HALT                   ;FAILURE WITH R2
900 002254 012702 000304      REG2C: MOV # $TESTN,R2  ;RESTORE POINTER
901
902
903 ;*****
904 ;TEST 23      TEST IF R3 CAN HOLD A ONE IN ALL BITS
905 002260 005212              TS23: INC (R2)          ;UPDATE TEST NUMBER
906 002262 022712 000023      CMP    #23,(R2)        ;SEQUENCE ERROR?
907 002266 001012              BNE    TS24-10         ;BR TO ERROR HALT ON SEQ ERROR
908 002270 012703 000001      MOV    #1,R3           ;SET BIT 0
909 002274 012700 177757      MOV    #-21,R0         ;SET BIT COUNTER
910 002300 000241              CLC                   ;CLEAR C-BIT
911 002302 000200              REG3: INC R0            ;INCREMENT BIT COUNTER
912 002304 001403              BEQ    REG3E           ;BR TO ERROR HALT IF BIT IS LOST
913 002306 006103              ROL    R3              ;ROTATE 1 POSITION
914 002310 103374              BCC    REG3            ;ALL DONE
915 002312 001404              BEQ    TS24            ;
916
917 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-- --
918 ; CONDITIONAL BRANCH INST. AND <-- --
919 ; REPLACE THE MOVE INSTRUCTION <== --
920 ; WHICH FOLLOWS W/ 765 <-- --
921 002314 012742 000027      REG3E: MOV #27,-(R2)     ;MOVE TO MAILBOX # ***** 27 *****
922 002320 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
923 002322 000000              HALT                   ;FAILURE WITH R3
924 ; OR SEQUENCE ERROR
925
926 ;*****
927 ;TEST 24      TEST IF R3 CAN HOLD A ZERO IN ALL BITS
```

```
928
929 002324 005212
930 002326 022712 000024
931 002332 001014
932 002334 012703 177776
933 002340 012700 177757
934 002344 000261
935 002346 005200
936 002350 001405
937 002352 006103
938 002354 103774
939 002356 022703 177777
940 002362 001404
941
942
943
944
945 002364
946 002364 012742 000030
947 002370 005242
948 002372 000000
949
950
951
952
953
954 002374 005212
955 002376 022712 000025
956 002402 001012
957 002404 012704 000001
958 002410 012700 177757
959 002414 000241
960 002416 005200
961 002420 001403
962 002422 006104
963 002424 103374
964 002426 001404
965
966
967
968
969 002430
970 002430 012742 000031
971 002434 005242
972 002436 000000
973
974
975
976
977
978 002440 005212
979 002442 022712 000026
980 002446 001014
981 002450 012704 177776
982 002454 012700 177757
983 002460 000261

;*****
TS24: INC (R2) ;UPDATE TEST NUMBER
      CMP #24,(R2) ;SEQUENCE ERROR?
      BNE TS25-10 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #-2,R3 ;SET ALL ONES IN R3 EXCEPT FOR BIT 0
      MOV #-21,R0 ;SET BIT COUNTER
      SEC ;SET C-BIT
REG3A: INC R0 ;INCREMENT BIT COUNTER
      BEQ R3ERR ;BR TO ERROR HALT IF COUNTER=0
      ROL R3 ;ROTATE 1 POSITION
      BCS REG3A ;CONTINUE UNTIL C-BIT IS CLEAR
      CMP #-1,R3 ;CHECK DATA
      BEQ TS25

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 763
R3ERR: MOV #30,-(R2) ;MOVE TO MAILBOX # ***** 30 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;FAILURE WITH R3
      ; OR SEQUENCE ERROR

;*****
;TEST 25 TEST IF R4 CAN HOLD A ONE IN ALL BITS
;*****
TS25: INC (R2) ;UPDATE TEST NUMBER
      CMP #25,(R2) ;SEQUENCE ERROR?
      BNE TS26-10 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #1,R4 ;SET BIT 0
      MOV #-21,R0 ;SET BIT COUNTER
      CLC ;CLEAR C-BIT
REG4: INC R0 ;INCREMENT BIT COUNTER
      BEQ REG4E ;BR TO ERROR HALT IF BIT IS LOST
      ROL R4 ;ROTATE 1 POSITION
      BCC REG4 ;ALL DONE
      BEQ TS26

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 765
REG4E: MOV #31,-(R2) ;MOVE TO MAILBOX # ***** 31 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;FAILURE WITH R4
      ; OR SEQUENCE ERROR

;*****
;TEST 26 TEST IF R4 CAN HOLD A ZERO IN ALL BITS
;*****
TS26: INC (R2) ;UPDATE TEST NUMBER
      CMP #26,(R2) ;SEQUENCE ERROR?
      BNE TS27-10 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #-2,R4 ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
      MOV #-21,R0 ;SET BIT COUNTER
      SEC ;SET C-BIT
```

```
984 002462 005200 REG4A: INC R0 ;INCREMENT BIT COUNTER
985 002464 001405 BEQ R4ERR ;BR TO ERROR HALT IF COUNTER=0
986 002466 006104 ROL R4 ;ROTATE 1 POSITION
987 002470 103774 BCS REG4A ;CONTINUE UNTIL C-BIT IS CLEAR
988 002472 022704 177777 CMP #-1,R4 ;CHECK DATA
989 002476 001404 BEQ TS27
990 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
991 ; CONDITIONAL BRANCH INST. AND <====
992 ; REPLACE THE MOVE INSTRUCTION <====
993 ; WHICH FOLLOWS W/ 763 <====
994 002500 R4ERR:
995 002500 012742 000032 MOV #32,-(R2) ;MOVE TO MAILBOX # ***** 32 *****
996 002504 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
997 002506 000000 HALT ;FAILURE WITH R4
998 ; OR SEQUENCE ERROR
999
JO
```

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

```
1001
1002
1003 TS27: INC (R2) ;UPDATE TEST NUMBER
1004 002510 005212 CMP #27,(R2) ;SEQUENCE ERROR?
1005 002512 022712 000027 BNE TS30-10 ;BR TO ERROR HALT ON SEQ ERROR
1006 002516 001012 MOV #1,R5 ;SET BIT 0
1007 002520 012705 000001 MOV #-21,R0 ;SET BIT COUNTER
1008 002524 012700 177757 CLC ;CLEAR C-BIT
1009 002530 000241 REG5: INC R0 ;INCREMENT BIT COUNTER
1010 002532 005200 BEQ REG5E ;BR TO ERROR HALT IF BIT IS LOST
1011 002534 001403 ROL R5 ;ROTATE 1 POSITION
1012 002536 006105 BCC REG5 ;ALL DONE
1013 002540 103374 BEQ TS30
1014 002542 001404
1015 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1016 ; CONDITIONAL BRANCH INST. AND <====
1017 ; REPLACE THE MOVE INSTRUCTION <====
1018 ; WHICH FOLLOWS W/ 765 <====
```

```
1019 002544 REG5E:
1020 002544 012742 000033 MOV #33,-(R2) ;MOVE TO MAILBOX # ***** 33 *****
1021 002550 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1022 002552 000000 HALT ;FAILURE WITH R5
1023 ; OR SEQUENCE ERROR
1024
1025
```

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

```
1026
1027 TS30: INC (R2) ;UPDATE TEST NUMBER
1028 002554 005212 CMP #30,(R2) ;SEQUENCE ERROR?
1029 002556 022712 000030 BNE TS31-10 ;BR TO ERROR HALT ON SEQ ERROR
1030 002562 001014 MOV #-2,R5 ;SET ALL ONES IN R5 EXCEPT FOR BIT 0
1031 002564 012705 177776 MOV #-21,R0 ;SET BIT COUNTER
1032 002570 012700 177757 SEC ;SET C-BIT
1033 002574 000261 REG5A: INC R0 ;INCREMENT BIT COUNTER
1034 002576 005200 BEQ R5ERR ;BR TO ERROR HALT IF COUNTER=0
1035 002600 001405 ROL R5 ;ROTATE 1 POSITION
1036 002602 006105 BCS REG5A ;CONTINUE UNTIL C-BIT IS CLEAR
1037 002604 103774 CMP #-1,R5 ;CHECK DATA
1038 002606 022705 177777 BEQ TS31
1039 002612 001404
```

```
1040                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1041                                     ; CONDITIONAL BRANCH INST. AND <====
1042                                     ; REPLACE THE MOVE INSTRUCTION <----
1043                                     ; WHICH FOLLOWS W/ 763 <====
1044 002614                                RSERR:
1045 002614 012742 000034                MOV #34,-(R2) ;MOVE TO MAILBOX # ***** 34 *****
1046 002620 005242                        INC -(R2) ;SET MSGTYP TO FATAL ERROR
1047 002622 000000                        HALT ;FAILURE WITH R5
1048                                     ; OR SEQUENCE ERROR
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
1066                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1067                                     ; CONDITIONAL BRANCH INST. AND <====
1068                                     ; REPLACE THE MOVE INSTRUCTION <====
1069                                     ; WHICH FOLLOWS W/ 762 <====
1070 002666 012742 000035                                REG6E: MOV #35,-(R2) ;MOVE TO MAILBOX # ***** 35 *****
1071 002672 005242                                INC -(R2) ;SET MSGTYP TO FATAL ERROR
1072 002674 000000                                HALT ;FAILURE WITH R6
1073                                     ; OR SEQUENCE ERROR
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 001015                                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
1091                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=
1092                                     ; CONDITIONAL BRANCH INST. AND <==
1093                                     ; REPLACE THE MOVE INSTRUCTION <--
1094                                     ; WHICH FOLLOWS W/ 763 <--
1095 002736 012742 000036                                R6ERR: MOV #36,-(R2) ;MOVE TO MAILBOX # ***** 36 *****
```

1096 002742 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1097 002744 000000 HALT ;FAILURE WITH R6
1098 ; OR SEQUENCE ERROR
1099

1100 :*****
1101 :SBTTL PSW TESTS
1102 :
1103 : THE PSW TESTS ARE USED TO VERIFY THAT VARIOUS DATA
1104 : PATTERNS CAN BE SUCCESSFULLY HELD IN THE PSW AND THAT THE
1105 : PSW ADDRESSING LOGIC IS FUNCTIONING. MOVE AND COMPARE INSTRUCTIONS
1106 : ARE USED TO TEST THAT THE PSW CAN HOLD VARIOUS DATA PATTERNS.
1107 : EACH DATA PATTERN IS MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR
1108 : SCOPING.
1109 : THE PSW REGISTER IS TESTED, THE CC INPUTS ARE TESTED
1110 : LATER IN THE MICROCODE TESTS. SETTING OF THE T-BIT BY THE
1111 : TEST PATTERNS IS PURPOSELY AVOIDED. TESTING OF THE
1112 : T-BIT TRAP CIRCUITRY IS LEFT FOR THE TRAP TEST.
1113 :
1114 :*****

1115 :TEST 33 TEST IF PSW WILL HOLD ZEROES
1116 :*****

1117 002746 005212 TS33: INC (R2) ;UPDATE TEST NUMBER
1118 002750 022712 000033 CMP #33,(R2) ;SEQUENCE ERROR?
1119 002754 001010 BNE TS34-10 ;BR TO ERROR HALT ON SEQ ERROR
1120 002756 012706 001000 MOV #STBOT,R6
1121 002762 012737 000000 177776 MOV #0,@#PS ;SET PSW TO ZERO
1122 002770 005737 177776 TST @#PS ;SUCCESSFUL
1123 002774 001404 BEQ TS34

1124 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
1125 : CONDITIONAL BRANCH INST. AND <---
1126 : REPLACE THE MOVE INSTRUCTION <---
1127 : WHICH FOLLOWS W/ 767 <---

1128 002776 012742 000037 MOV #37, -(R2) ;MOVE TO MAILBOX # ***** 37 *****
1129 003002 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1130 003004 000000 HALT ;PSW NOT 0
1131 ; OR SEQUENCE ERROR
1132

1133 :*****
1134 :TEST 34 TEST IF PSW WILL HOLD ONES AND ZEROES
1135 :*****

1136 003006 005212 TS34: INC (R2) ;UPDATE TEST NUMBER
1137 003010 022712 000034 CMP #34,(R2) ;SEQUENCE ERROR?
1138 003014 001007 BNE TS35-10 ;BR TO ERROR HALT ON SEQ ERROR
1139 003016 012737 000252 177776 MOV #252,@#PS ;MOVE ALT. ONES AND ZEROES TO PSW
1140 003024 023727 177776 000252 CMP @#PS,#252 ;SUCCESSFUL?
1141 003032 001404 BEQ TS35

1142 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
1143 : CONDITIONAL BRANCH INST. AND <===
1144 : REPLACE THE MOVE INSTRUCTION <===
1145 : WHICH FOLLOWS W/ 770 <---

1146 003034 012742 000040 MOV #40, -(R2) ;MOVE TO MAILBOX # ***** 40 *****
1147 003040 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1148 003042 000000 HALT ;PSW NOT 252
1149 ; OR SEQUENCE ERROR
1150

1151 :*****

1152 ;TEST 35 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES
1153 :*****
1154 003044 005212 TS35: INC (R2) ;UPDATE TEST NUMBER
1155 003046 022712 000035 CMP #35,(R2) ;SEQUENCE ERROR?
1156 003052 001007 BNE TS36-10 ;BR TO ERROR HALT ON SEQ ERROR
1157 003054 012737 000105 177776 MOV #105,@#PS ;MOVE ALT. ONES AND ZEROES TO PSW
1158 003062 023727 177776 000105 CMP @#PS,#105 ;SUCCESSFUL?
1159 003070 001404 BEQ TS36

1160 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
1161 ; CONDITIONAL BRANCH INST. AND <====
1162 ; REPLACE THE MOVE INSTRUCTION <====
1163 ; WHICH FOLLOWS W/ 770 <====
1164 003072 012742 000041 MOV #41,-(R2) ;MOVE TO MAILBOX # ***** 41 *****
1165 003076 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1166 003100 000000 HALT ;PSW NOT 105
1167 ; OR SEQUENCE ERROR
1168

1169 :*****
1170 ;TFST 36 TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES
1171 :*****
1172 003102 005212 TS36: INC (R2) ;UPDATE TEST NUMBER
1173 003104 022712 000036 CMP #36,(R2) ;SEQUENCE ERROR?
1174 003110 001007 BNE TS37-10 ;BR TO ERROR HALT ON SEQ ERROR
1175 003112 012737 000357 177776 MOV #357,@#PS ;MOVE ONES TO PSW
1176 003120 023727 177776 000357 CMP @#PS,#357 ;SUCCESSFUL
1177 003126 001404 BEQ TS37

1178 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1179 ; CONDITIONAL BRANCH INST. AND <====
1180 ; REPLACE THE MOVE INSTRUCTION <====
1181 ; WHICH FOLLOWS W/ 770 <====
1182 003130 012742 000042 MOV #42,-(R2) ;MOVE TO MAILBOX # ***** 42 *****
1183 003134 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
1184 003136 000000 HALT ;PSW NOT 357
1185 ; OR SEQUENCE ERROR
1186

1187 .SBTTL CONDITION CODE TEST
1188 :*****
1189 :
1190 : THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE Z-BIT.
1191 : THE Z-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1192 : BEQ AND BNE ARE TESTED FOR PROPER EXECUTION. THEN THE Z-BIT IS
1193 : SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1194 : AGAIN FOR PROPER OPERATION.
1195 : THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
1196 : CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
1197 : BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
1198 : LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY
1199 : USED IN THE TEST ARE VERIFIED HERE.
1200 :
1201 :*****

1202 ;TEST 37 TEST BRANCHES AROUND Z-BIT
1203 :*****
1204 003140 005212 TS37: INC (R2) ;UPDATE TEST NUMBER
1205 003142 022712 000037 CMP #37,(R2) ;SEQUENCE ERROR?
1206 003146 001014 BNE TS40-10 ;BR TO ERROR HALT ON SEQ ERROR
1207 ;FIRST WITH Z-BIT ON

```
1208 003150 000257      CCC      ;CC=0100: JUST Z-BIT
1209 003152 000264      SEZ
1210 003154 001001      BNE      BRZ1      ;CHECK OPPOSITE CONDITION
1211 003156 001404      BEQ      BRZ2
1212      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1213      ;          CONDITIONAL BRANCH INST. AND <====
1214      ;          REPLACE THE MOVE INSTRUCTION <====
1215      ;          WHICH FOLLOWS W/ 773 <====
1216 003160      BRZ1:
1217 003160 012742 000043      MOV      #43,-(R2)      ;MOVE TO MAILBOX # ***** 43 *****
1218 003164 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
1219 003166 000000      HALT          ;IMPROPER BR W/ Z=1
1220      ;CHECK WITH Z-BIT OFF
1221 003170 000277      BRZ2:      SCC          ;CC=1011: ALL BUT Z-BIT
1222 003172 000244      CLZ
1223 003174 001401      BEQ      BRZ3
1224 003176 001004      BNE      TS40
1225      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1226      ;          CONDITIONAL BRANCH INST. AND <====
1227      ;          REPLACE THE MOVE INSTRUCTION <====
1228      ;          WHICH FOLLOWS W/ 763 <====
1229 003200      BRZ3:
1230 003200 012742 000044      MOV      #44,-(R2)      ;MOVE TO MAILBOX # ***** 44 *****
1231 003204 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
1232 003206 000000      HALT          ;IMPROPER BR W/ Z=0
1233      ; OR SEQUENCE ERROR
1234
1235      ;*****
1236      ;
1237      ; THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE N-BIT.
1238      ; THE N-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
1239      ; BMI AND BPL ARE TESTED FOR PROPER EXECUTION. THEN THE N-BIT IS
1240      ; SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
1241      ; AGAIN FOR PROPER OPERATION.
1242      ; THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
1243      ; CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
1244      ; BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
1245      ; LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY
1246      ; USED IN THE TEST ARE VERIFIED HERE.
1247      ;
1248      ;*****
1249      ;TEST 40      TEST BRANCHES AROUND N-BIT
1250      ;*****
1251 003210 005212      TS40:      INC      (R2)          ;UPDATE TEST NUMBER
1252 003212 022712 000040      CMP      #40,(R2)      ;SEQUENCE ERROR?
1253 003216 001014      BNE      TS41-10      ;BR TO ERROR HALT ON SEQ ERROR
1254      ;FIRST WITH N-BIT ON
1255 003220 000257      CCC          ;CC=1000: JUST N-BIT
1256 003222 000270      SEN
1257 003224 100001      BPL      BRN1      ;CHECK OPPOSITE CONDITION
1258 003226 100404      BMI      BRN2
1259      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
1260      ;          CONDITIONAL BRANCH INST. AND <--
1261      ;          REPLACE THE MOVE INSTRUCTION <---
1262      ;          WHICH FOLLOWS W/ 773 <---
1263 003230      BRN1:
```

```
1264 003230 012742 000045      MOV    #45,-(R2)      ;MOVE TO MAILBOX # ***** 45 *****
1265 003234 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
1266 003236 000000      HALT                ;IMPROPER BR W/ N=1
1267                                ;CHECK WITH N-BIT OFF
1268 003240 000277      BRN2:  SCC           ;CC=0111
1269 003242 000250      CLN
1270 003244 100401      BMI    BRN3         ;CHECK OPPOSITE CONDITION
1271 003246 100004      BPL    TS41
1272                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1273                                ;          CONDITIONAL BRANCH INST. AND <====
1274                                ;          REPLACE THE MOVE INSTRUCTION <====
1275                                ;          WHICH FOLLOWS W/ 763 <====
1276 003250
1277 003250 012742 000046      BRN3:  MOV    #46,-(R2) ;MOVE TO MAILBOX # ***** 46 *****
1278 003254 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
1279 003256 000000      HALT                ;IMPROPER BR W/ N=0
1280                                ; OR SEQUENCE ERROR
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298
1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
```

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

```
1293 003260 005212 000041      TS41:  INC    (R2)      ;UPDATE TEST NUMBER
1294 003262 022712      CMP    #41,(R2)     ;SEQUENCE ERROR?
1295 003266 001014      BNE    TS42-10     ;BR TO ERROR HALT ON SEQ ERROR
1296                                ;FIRST WITH V-BIT ON
1297 003270 000257      CCC           ;CC=0010: JUST V-BIT
1298 003272 000262      SEV
1299 003274 102001      BVC    BRV1       ;CHECK OPPOSITE CONDITION
1300 003276 102404      BVS    BRV2
1301                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1302                                ;          CONDITIONAL BRANCH INST. AND <====
1303                                ;          REPLACE THE MOVE INSTRUCTION <====
1304                                ;          WHICH FOLLOWS W/ 773 <====
1305 003300
1306 003300 012742 000047      BRV1:  MOV    #47,-(R2) ;MOVE TO MAILBOX # ***** 47 *****
1307 003304 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
1308 003306 000000      HALT                ;IMPROPER BR W/ V-1
1309                                ;CHECK WITH V-BIT OFF
1310 003310 000277      BRV2:  SCC           ;CC=1101: ALL BVT V-BIT
1311 003312 000242      CLV
1312 003314 102401      BVS    BRV3       ;CHECK OPPOSITE CONDITION
1313 003316 102004      BVC    TS42
1314                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1315                                ;          CONDITIONAL BRANCH INST. AND <====
1316                                ;          REPLACE THE MOVE INSTRUCTION <====
1317                                ;          WHICH FOLLOWS W/ 763 <====
1318 003320
1319 003320 012742 000050      BRV3:  MOV    #50,-(R2) ;MOVE TO MAILBOX # ***** 50 *****
```

```
1320 003324 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
1321 003326 000000          HALT                ;IMPROPER BR W/ V=0
1322                                     ; OR SEQUENCE ERROR
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334
1335 003330 005212          TS42: INC    (R2)          ;UPDATE TEST NUMBER
1336 003332 022712 000042    CMP    #42,(R2)      ;SEQUENCE ERROR?
1337 003336 001014          BNE    TS43-10       ;BR TO ERROR HALT ON SEQ ERROR
1338                                     ;FIRST WITH C-BIT ON
1339 003340 000257          CCC                ;CC=0001: JUST C-BIT
1340 003342 000261          SEC
1341 003344 103001          BCC    BRC1          ;CHECK OPPOSITE CONDITION
1342 003346 103404          BCS    BRC2
1343                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1344                                     ; CONDITIONAL BRANCH INST. AND <====
1345                                     ; REPLACE THE MOVE INSTRUCTION <====
1346                                     ; WHICH FOLLOWS W/ 773 <====
1347 003350
1348 003350 012742 000051    BRC1: MOV    #51,-(R2)   ;MOVE TO MAILBOX # ***** 51 *****
1349 003354 005242          INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
1350 003356 000000          HALT                ;IMPROPER BR W/ C=1
1351                                     ;CHECK WITH C-BIT OFF
1352 003360 000277          BRC2: SCC                ;CC=1110
1353 003362 000241          CLC
1354 003364 103401          BCS    BRC3          ;CHECK OPPOSITE CONDITION
1355 003366 100404          BMI    TS43
1356                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1357                                     ; CONDITIONAL BRANCH INST. AND <====
1358                                     ; REPLACE THE MOVE INSTRUCTION <====
1359                                     ; WHICH FOLLOWS W/ 763 <====
1360 003370
1361 003370 012742 000052    BRC3: MOV    #52,-(R2)   ;MOVE TO MAILBOX # ***** 52 *****
1362 003374 005242          INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
1363 003376 000000          HALT                ;IMPROPER BR W/ C=0
1364                                     ; OR SEQUENCE ERROR
1365
1366
1367
1368
1369
1370
1371
1372
1373
1374
1375
```

: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

```
1376 :ONLY INSTRUCTIONS AND ADDRESSING MODES WHICH HAVE BEEN PREVIOUSLY
1377 :VERIFIED.
1378 : IF THESE TESTS FAIL, CHECK THE RESULTS FOR A CLUE TO THE
1379 :FAULT.
1380 :
1381 :*****
1382 :
1383 :
1384 :
1385 :*****
1386 :
1387 : THE CLR INSTRUCTION IS USED TO INTRODUCE EACH ADDRESSING
1388 :MODE WITH THE SINGLE OPERAND INSTRUCTION. FOLLOWING THE SEQUENCE CHECK,
1389 :THE CLR INSTRUCTION IS EXECUTED AND A BRANCH TEST IS EXECUTED WHICH
1390 :CHECKS THAT THE Z-BIT WAS PROPERLY SET. THIS TEST CAN CHECK IR DECODE
1391 :AND MICROCODE FOR SOP INSTRUCTIONS WITH MODE 0. FOLLOWING THIS TEST
1392 :SEVERAL OTHER SOP INSTRUCTIONS ARE INTRODUCED WITH MODE 0. THESE
1393 :INSTRUCTIONS MAINPULATE DATA AND SERVE TO CHECK THE DATA RESULTS
1394 :OF THE SOP INSTRUCTIONS IN THIS TEST. THE DATA IN THIS TEST IS
1395 :OPERATED ON BY EACH INSTRUCTION WITHOUT REINITIALIZING.
1396 :
1397 :*****
1398 :TEST 43 TEST MODE 0 USING SOP INST.
1399 :*****
1400 003400 005212
1401 003402 022712 000043
1402 003406 001020
1403 003410 005000
1404 003412 001404
1405
1406 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1407 : CONDITIONAL BRANCH INST. AND <====
1408 : REPLACE THE MOVE INSTRUCTION <====
1409 : WHICH FOLLOWS W/ 775 <====
1410 003414 012742 000053
1411 003420 005242
1412 003422 000000
1413 003424 005200
1414 003426 005100
1415 003430 005200
1416 003432 100404
1417
1418 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1419 : CONDITIONAL BRANCH INST. AND <====
1420 : REPLACE THE MOVE INSTRUCTION <====
1421 : WHICH FOLLOWS W/ 765 <====
1422 003434 012742 000054
1423 003440 005242
1424 003442 000000
1425 003444 005100
1426 003446 001404
1427
1428 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1429 : CONDITIONAL BRANCH INST. AND <====
1430 : REPLACE THE MOVE INSTRUCTION <====
1431 : WHICH FOLLOWS W/ 757 <====
1432 003450 012742 000055
1433 003454 005242
1434 003456 000000
```

```
TS43: INC (R2) ;UPDATE TEST NUMBER
      CMP #43,(R2) ;SEQUENCE ERROR?
      BNE TS44-10 ;BR TO ERROR HALT ON SEQ ERROR
      CLR R0 ;TRY THE CLEAR INST.
      BEQ SOPOA
;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 775 <====
; MOVE TO MAILBOX # ***** 53 *****
; SET MSGTYP TO FATAL ERROR
; CLR DID NOT SET Z-BIT
; TRY THE INCREMENT INST.
; TRY COMPLEMENT
SOPOA: INC R0
      COM R0
      INC R0
      BMI SOPOB
;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 765 <====
; MOVE TO MAILBOX # ***** 54 *****
; SET MSGTYP TO FATAL ERROR
; NEGATE DID NOT SET N-BIT
; TRY COMPLEMENT INST.
SOPOB: COM R0
      BEQ TS44
;
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 757 <====
; MOVE TO MAILBOX # ***** 55 *****
; SET MSGTYP TO FATAL ERROR
; CUMMULATIVE RESULT OF CLR,INC,NEG AND COM INSTS. FAILED
```

```
1432                                     ; OR SEQUENCE ERROR
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447 003460 005212
1448 003462 022712 000044
1449 003466 001021
1450 003470 005000
1451 003472 005300
1452 003474 100404
1453
1454
1455
1456
1457 003476 012742 000056
1458 003502 005242
1459 003504 000000
1460 003506 000261
1461 003510 005500
1462 003512 001007
1463 003514 000261
1464 003516 005600
1465 003520 100004
1466 003522 005100
1467 003524 005200
1468 003526 005300
1469 003530 001404
1470
1471
1472
1473
1474 003532
1475 003532 012742 000057
1476 003536 005242
1477 003540 000000
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
```

```

:*****
: 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 <--
SOPOC: MOV #56,-(R2) ;MOVE TO MAILBOX # ***** 56 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;N-BIT NOT SET ON DEC
      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
:*****
```

```
1488
1489 003542 005212
1490 003544 022712 000045
1491 003550 001012
1492 003552 105000
1493 003554 001404
1494
1495
1496
1497
1498 003556 012742 000060
1499 003562 005242
1500 003564 000000
1501 003566 105100
1502 003570 100002
1503 003572 105200
1504 003574 001404
1505
1506
1507
1508
1509 003576
1510 003576 012742 000061
1511 003602 005242
1512 003604 000000
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526 003606 005212
1527 003610 022712 000046
1528 003614 001014
1529 003616 005000
1530 003620 005010
1531 003622 001404
1532
1533
1534
1535
1536 003624 012742 000062
1537 003630 005242
1538 003632 000000
1539 003634 005310
1540 003636 100003

:*****
TS45:  INC      (R2)          ;UPDATE TEST NUMBER
      CMP      #45,(R2)      ;SEQUENCE ERROR?
      BNE     TS46-10        ;BR TO ERROR HALT ON SEQ ERROR
      CLRB    R0            ;TRY CLEARING EVEN BYTE OF REGISTER
      BEQ     SOPBOA
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ;          CONDITIONAL BRANCH INST. AND <====
      ;          REPLACE THE MOVE INSTRUCTION <====
      ;          WHICH FOLLOWS W/ 775 <====
      MOV     #60,-(R2)      ;MOVE TO MAILBOX # ***** 60 *****
      INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
      HALT
SOPBOA: COMB    R0          ;CLRB DID NOT SET Z-BIT
      BPL    SOPBOB        ;TRY SETTING EVEN BYTE OF REGISTER
      INCB   R0            ;TRY INCREMENTING EVEN BYTE OF REGISTER>>
      BEQ     TS46
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ;          CONDITIONAL BRANCH INST. AND <====
      ;          REPLACE THE MOVE INSTRUCTION <====
      ;          WHICH FOLLOWS W/ 765 <====
SOPBOB: MOV     #61,-(R2)      ;MOVE TO MAILBOX # ***** 61 *****
      INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
      HALT                ;TEST CUMMULATIVE RESULT OF ABOVE BYTE INST.
                        ; OR SEQUENCE ERROR

:*****
:
: THIS TEST USES THE CLR INSTRUCTION TO INTRODUCE AND TEST
: SINGLE OPERAND MODE 1 INSTRUCTIONS. AGAIN, THE CLR INSTRUCTION
: IS USED TO INTRODUCE THE MICROCODE AND TO TEST THAT THE PROPER
: CONDITION CODES ARE SET. OTHER SOP INSTRUCTIONS ARE USED TO MANIPULATE
: COMMON DATA TO VERIFY THAT THE CORRECT DATA IS PRODUCED.
:
:*****
: TEST 46 TEST MODE 1 USING SOP INST.
:*****
TS46:  INC      (R2)          ;UPDATE TEST NUMBER
      CMP      #46,(R2)      ;SEQUENCE ERROR?
      BNE     TS47-10        ;BR TO ERROR HALT ON SEQ ERROR
      CLR     R0            ;INITIALIZE R0
      CLR     (R0)          ;TRY CLEAR INST W/MODE 1
      BEQ     SOP1A
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < ---
      ;          CONDITIONAL BRANCH INST. AND < ---
      ;          REPLACE THE MOVE INSTRUCTION < ---
      ;          WHICH FOLLOWS W/ 774 < ---
      MOV     #62,-(R2)      ;MOVE TO MAILBOX # ***** 62 *****
      INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
      HALT                ;CLRB DID NOT SET Z-BIT
SOP1A: DEC     (R0)          ;TRY DECREMENT INST W/MODE 1
      BPL    SOP1B
```


1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609 003736 005212
1610 003740 022712 000050
1611 003744 001022
1612 003746 005000
1613 003750 005010
1614 003752 005110
1615 003754 005200
1616 003756 105010
1617 003760 001404
1618
1619
1620
1621
1622 003762 012742 000066
1623 003766 005242
1624 003770 000000
1625 003772 005300
1626 003774 005210
1627 003776 005200
1628 004000 105110
1629 004002 105210
1630 004004 100002
1631 004006 105210
1632 004010 001404
1633
1634
1635
1636
1637 004012
1638 004012 012742 000067
1639 004016 005242
1640 004020 000000
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652

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

TEST 50 TEST MODE 1 ODD BYTE USING SOP INST

TS50: INC (R2) ;UPDATE TEST NUMBER
CMP #50,(R2) ;SEQUENCE ERROR?
BNE TS51-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;INITIALIZE R0
CLR (R0) ;INITIALIZE LOC. 0
COM (R0)
INC R0 ;R0=ODD BYTE
CLRB (R0) ;TRY TO CLEAR BYTE 1
BEQ SOPB1C

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

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

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

SOPB1D: MOV #67,-(R2) ;MOVE TO MAILBOX # ***** 67 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST 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 VRFIFIES THE PROPER INCREMENTING OF THE
REGISTER.

1653
1654
1655
1656 004022 005212
1657 004024 022712 000051
1658 004030 001023
1659 004032 005000
1660 004034 105100
1661 004036 005200
1662 004040 005010
1663 004042 005110
1664 004044 005020
1665 004046 001404
1666
1667
1668
1669
1670 004050 012742 000070
1671 004054 005242
1672 004056 000000
1673 004060 005300
1674 004062 005300
1675 004064 005120
1676 004066 100004
1677 004070 005300
1678 004072 005300
1679 004074 005220
1680 004076 001404
1681
1682
1683
1684
1685 004100
1686 004100 012742 000071
1687 004104 005242
1688 004106 000000
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704 004110 005212
1705 004112 022712 000052
1706 004116 001023
1707 004120 005000
1708 004122 105100

:TEST 51 TEST MODE 2 USING SOP INST.

TS51: INC (R2) ;UPDATE TEST NUMBER
CMP #51,(R2) ;SEQUENCE ERROR?
BNE TS52-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;SET R0=400
COMB R0
INC R0
CLR (R0) ;CLEAR 400
COM (R0) ;INITIALIZE: 400 -1
CLR (R0)+ ;TRY CLEARING WITH MODE 2
BEQ SOPZA

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

MOV #70,-(R2) ;MOVE TO MAILBOX # ***** 70 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR INST DID NOT SET Z-BIT
SOPZA: DEC R0 ;RESET R0
DEC R0
COM (R0)+ ;TRY COMPLEMENTING WITH MODE 2
BPL SOP2B
DEC R0 ;RESET R0
DEC R0
INC (R0)+ ;TRY INCREMENTING WITH MODE 2
BEQ TS52

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

SOP2B: MOV #71,-(R2) ;MOVE TO MAILBOX # ***** 71 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CHECK CUMMULATIVE RESULT OF ABOVE INST
; OR SEQUENCE ERROR

: THIS TEST VERIFIES MODE 2 SINGLE OPERAND INSTRUCTIONS WHICH
: ADDRESS EVEN BYTES. 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.

TS52: INC (R2) ;UPDATE TEST NUMBER
CMP #52,(R2) ;SEQUENCE ERROR?
BNE TS53-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;SET R0=400
COMB R0

```

1709 004124 005200          INC      R0
1710 004126 005010          CLR      (R0)          ;CLEAR 400
1711 004130 005110          COM      (R0)          ;INITIALIZE: 400=-1
1712 004132 105020          CLRB    (R0)+         ;TRY TO CLEAT 400 W/MODE 2
1713 004134 001404          BEQ     SOPB2A
1714
1715                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=---
1716                          ;          CONDITIONAL BRANCH INST. AND <=---
1717                          ;          REPLACE THE MOVE INSTRUCTION <=---
1718                          ;          WHICH FOLLOWS W/ 770 <=---
1718 004136 012742 000072      MOV     #72,-(R2)      ;MOVE TO MAILBOX # ***** 72 *****
1719 004142 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
1720 004144 000000          HALT
1721 004146 005300          SOPB2A: DEC     R0      ;CLR DID NOT SET Z-BIT
1722 004150 005210          INC     (R0)         ;RESULT R0=400
1723 004152 105110          COMB    (R0)         ;INC 400 TO TEST WORD
1724 004154 105220          INCB   (R0)+         ;TRY TO INC EVEN BYTE
1725 004156 100003          BPL     SOPB2B
1726 004160 005300          DEC     R0           ;RESET R0=400
1727 004162 105220          INCB   (R0)+         ;TRY INCREMENT OF EVEN BYTE
1728 004164 001404          BEQ     TS53
1729
1730                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=---
1731                          ;          CONDITIONAL BRANCH INST. AND <=---
1732                          ;          REPLACE THE MOVE INSTRUCTION <=---
1733                          ;          WHICH FOLLOWS W/ 754 <=---
1733 004166          SOPB2B:
1734 004166 012742 000073      MOV     #73,-(R2)      ;MOVE TO MAILBOX # ***** 73 *****
1735 004172 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
1736 004174 000000          HALT                 ;TEST CUMMULATIVE RESULT OF ABOVE INST.
1737
1738                          ; OR SEQUENCE ERROR
1739
1740 :*****
1741 :
1742 :          THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS
1743 :TEST.  HERE,  THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.
1744 :
1745 :*****
1746 :TEST 53          TEST MODE 2 ODD BYTE USING SOP INST.
1747 :*****
1747 004176 005212          TS53:  INC     (R2)      ;UPDATE TEST NUMBER
1748 004200 022712 000053      CMP     #53,(R2)      ;SEQUENCE ERROR?
1749 004204 001026          BNE     TS54-10      ;BR TO ERROR HALT ON SEQ ERROR
1750 004206 005000          CLR     R0           ;SET R0=400
1751 004210 105100          COMB    R0
1752 004212 005200          INC     R0
1753 004214 005010          CLR     (R0)         ;CLEAR LOC 400
1754 004216 005110          COM     (R0)         ;INITIALIZE: 400=-1
1755 004220 005200          INC     R0           ;R0=ODD BYTE
1756 004222 105020          CLRB   (R0)+         ;TRY TO CLEAR ODD BYTE
1757 004224 001404          BEQ     SOPB2C
1758
1759                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=---
1760                          ;          CONDITIONAL BRANCH INST. AND <=---
1761                          ;          REPLACE THE MOVE INSTRUCTION <=---
1762                          ;          WHICH FOLLOWS W/ 767 <=---
1762 004226 012742 000074      MOV     #74,-(R2)      ;MOVE TO MAILBOX # ***** 74 *****
1763 004232 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
1764 004234 000000          HALT                 ;CLRB DID NOT SET Z-BIT
    
```

```
1765 004236 005300 SOPB2C: DEC R0 ;RO=WORD ADDR.  
1766 004240 005300 DEC R0  
1767 004242 005220 INC (R0)+ ;INCREMENT WORD  
1768 004244 005300 DEC R0 ;POINT TO ODD BYTE  
1769 004246 105110 COMB (R0) ;COMPLEMENT ODD BYTE  
1770 004250 105220 INCB (R0)+ ;TRY TO INCREMENT ODD BYTE  
1771 004252 100003 BPL SOPB2D  
1772 004254 005300 DEC R0 ;RESET RO TO ODD BYTE  
1773 004256 105220 INCB (R0)+ ;TRY TO INCREMENT ODD BYTE  
1774 004260 001404 BEQ TS54
```

```
1775  
1776 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
1777 ; CONDITIONAL BRANCH INST. AND <====  
1778 ; REPLACE THE MOVE INSTRUCTION <====  
1779 ; WHICH FOLLOWS W/ 751 <====
```

```
1779 004262 SOPB2D: MOV #75,-(R2) ;MOVE TO MAILBOX # ***** 75 *****  
1780 004262 012742 000075 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1781 004266 005242 HALT ;TEST CUMMULATIVE RESULT OF ABOVE INST.  
1782 004270 000000 ; OR SEQUENCE ERROR  
1783  
1784
```

```
1785 :*****  
1786 :  
1787 : THESE TESTS CHECK THE NEGATE INSTRUCTION IN ALL MODES. PREVIOUSLY  
1788 : TESTED SINGLE-OPERAND INSTRUCTIONS ARE USED TO TEST THE NEGATE INSTRUCTION.  
1789 :  
1790 :*****
```

```
1791 :TEST 54 TEST MODE 0 USING NEGATE INSTRUCTION  
1792 :*****
```

```
1793 004272 005212 TS54: INC (R2) ;UPDATE TEST NUMBER  
1794 004274 022712 000054 CMP #54,(R2) ;SEQUENCE ERROR?  
1795 004300 001035 BNE TS55-10 ;BR TO ERROR HA T ON SEQ ERROR  
1796 004302 005000 CLR R0 ;SET RO=0  
1797 004304 005200 INC R0 ; RO=1  
1798 004306 005400 NEG R0 ;TRY NEGATE MODE 0: RO--1  
1799 004310 100003 BPL NEG00 ;CC=1001?  
1800 004312 001404 BEQ NEG00  
1801 004314 102401 BVS NEG00  
1802 004316 103404 BCS NEG01
```

```
1803 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---  
1804 ; CONDITIONAL BRANCH INST. AND <---  
1805 ; REPLACE THE MOVE INSTRUCTION <---  
1806 ; WHICH FOLLOWS W/ 770 <---
```

```
1807 004320 NEG00: MOV #76,-(R2) ;MOVE TO MAILBOX # ***** 76 *****  
1808 004320 012742 000076 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1809 004324 005242 HALT ;NEGATE DID NOT SET CC'S CORRECTLY  
1810 004326 000000
```

```
1811 NEG01: INC R0 ;TEST DATA RESULT  
1812 004330 005200 BEQ NEG02  
1813 004332 001404
```

```
1814 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---  
1815 ; CONDITIONAL BRANCH INST. AND <---  
1816 ; REPLACE THE MOVE INSTRUCTION <---  
1817 ; WHICH FOLLOWS W/ 762 <---
```

```
1818 004334 012742 000077 MOV #77,-(R2) ;MOVE TO MAILBOX # ***** 77 *****  
1819 004340 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1820 004342 000000 HALT ;DATA RESULT OF NEGATE INCORRECT
```

```
1821
1822 004344 105100      NEG02:  COMB    R0          ;R0=377
1823 004346 105400      NEG02:  NEGB   R0          ;R0=1
1824 004350 100403      NEG02:  BMI    NEG03      ;CC=0001?
1825 004352 001402      NEG02:  BEQ    NEG03
1826 004354 102401      NEG02:  BVS    NEG03
1827 004356 103404      NEG02:  BCS    NEG04
1828
1829                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1830                      ; CONDITIONAL BRANCH INST. AND <====
1831                      ; REPLACE THE MOVE INSTRUCTION <====
1832                      ; WHICH FOLLOWS W/ 750 <====
1832 004360      NEG03:
1833 004360 012742 000100      NEG03:  MOV    #100,-(R2)    ;MOVE TO MAILBOX # ***** 100 *****
1834 004364 005242      NEG03:  INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
1835 004366 000000      NEG03:  HALT                   ;NEGB DID NOT SET CC'S CORRECTLY
1836 004370 005300      NEG04:  DEC    R0            ;TEST DATA RESULT
1837 004372 001404      NEG04:  BEQ    TS55
1838
1839                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1840                      ; CONDITIONAL BRANCH INST. AND <====
1841                      ; REPLACE THE MOVE INSTRUCTION <====
1842                      ; WHICH FOLLOWS W/ 742 <====
1842 004374 012742 000101      NEG04:  MOV    #101,-(R2)    ;MOVE TO MAILBOX # ***** 101 *****
1843 004400 005242      NEG04:  INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
1844 004402 000000      NEG04:  HALT                   ;DATA RESULT OF NEGB INCORRECT
1845                      ; OR SEQUENCE ERROR
1846
1847 *****
1848 :TEST 55 TEST MODE 1 USING NEGATE INST.
1849 *****
1849 004404 005212      TS55:  INC    (R2)          ;UPDATE TEST NUMBER
1850 004406 022712 000055      TS55:  CMP    #55,(R2)      ;SEQUENCE ERROR?
1851 004412 001040      TS55:  BNE    TS56-10      ;BR TO ERROR HALT ON SEQ ERROR
1852 004414 005000      TS55:  CLR    R0            ;POINT TO LOC. 0
1853 004416 005010      TS55:  CLR    (R0)         ;CLEAR LOC. 0
1854 004420 005210      TS55:  INC    (R0)         ;LOC. 0=1
1855 004422 005410      TS55:  NEG    (R0)         ;TRY NEG. LOC. 0=-1
1856 004424 100003      TS55:  BPL    NEG10        ;CC=1001
1857 004426 001402      TS55:  BEQ    NEG10
1858 004430 102401      TS55:  BVS    NEG10
1859 004432 103404      TS55:  BCS    NEG11
1860
1861                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1862                      ; CONDITIONAL BRANCH INST. AND <====
1863                      ; REPLACE THE MOVE INSTRUCTION <====
1864                      ; WHICH FOLLOWS W/ 767 <====
1864 004434      NEG10:
1865 004434 012742 000102      NEG10:  MOV    #102,-(R2)    ;MOVE TO MAILBOX # ***** 102 *****
1866 004440 005242      NEG10:  INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
1867 004442 000000      NEG10:  HALT                   ;NEGATE DID NOT SET CC'S CORRECTLY
1868
1869 004444 005237 000000      NEG11:  INC    @#0
1870 004450 001404      NEG11:  BEQ    NEG12        ;TEST DATA RESULT
1871
1872                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <= =
1873                      ; CONDITIONAL BRANCH INST. AND <= -
1874                      ; REPLACE THE MOVE INSTRUCTION <=-==
1875                      ; WHICH FOLLOWS W/ 760 <
1875 004452 012742 000103      NEG11:  MOV    #103,-(R2)    ;MOVE TO MAILBOX # ***** 103 *****
1876 004456 005242      NEG11:  INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
```

```
1877 004460 000000
1878 004462 105110
1879 004464 105410
1880 004466 100403
1881 004470 001402
1882 004472 102401
1883 004474 103404
1884
1885
1886
1887
1888 004476
1889 004476 012742 000104
1890 004502 005242
1891 004504 000000
1892 004506 005337 000000
1893 004512 001404
1894
1895
1896
1897
1898 004514 012742 000105
1899 004520 005242
1900 004522 000000
1901
1902
1903
1904
1905 004524 005212
1906 004526 022712 000056
1907 004532 001032
1908 004534 005000
1909 004536 005010
1910 004540 005210
1911 004542 005420
1912 004544 100003
1913 004546 001402
1914 004550 102401
1915 004552 103404
1916
1917
1918
1919
1920 004554
1921 004554 012742 000106
1922 004560 005242
1923 004562 000000
1924 004564 105300
1925 004566 105300
1926 004570 105420
1927 004572 105420
1928 004574 105340
1929 004576 005300
1930 004600 001404
1931
1932
```

```

NEG12: HALT ;DATA RESULT OF NEGATE INCORRECT
        COMB (R0) ;LOC. 0=377
        NEGB (R0) ;TRY NEGB LOC. 0=1
        BMI NEG13 ;CC=0001?
        BEQ NEG13
        BVS NEG13
        BCS NEG14

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

NEG13: MOV #104,-(R2) ;MOVE TO MAILBOX # ***** 104 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;NEGB DID NOT SET CC'S CORRECTLY
NEG14: DEC @R0 ;TEST DATA RESULT
        BEQ TS56

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

        MOV #105,-(R2) ;MOVE TO MAILBOX # ***** 105 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;DATA RESULT OF NEGB INCORRECT
        ; OR SEQUENCE ERROR

;*****
;TEST 56 TEST MODE 2 USING NEGATE INSTRUCTION
;*****
TS56: INC (R2) ;UPDATE TEST NUMBER
      CMP #56,(R2) ;SEQUENCE ERROR?
      BNE TS57-10 ;BR TO ERROR HALT ON SEQ ERROR
      CLR R0 ;POINT TO LOC. 0
      CLR (R0) ;CLEAR LOC. 0
      INC (R0) ;LOC. 0=1
      NEG (R0)+ ;TRY NEG.: LOC. 0 -1
      BPL NEG20 ;CC=1001?
      BEQ NEG20
      BVS NEG20
      BCS NEG21

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

NEG20: MOV #106,-(R2) ;MOVE TO MAILBOX # ***** 106 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;NEGATE DID NOT SET CC'S CORRECTLY
NEG21: DECB R0 ;R0=LOC. 0
        DECB R0
        NEGB (R0)+ ;BYTE 0=1 R0=1
        NEGB (R0)+ ;BYTE 1=1 R0=2
        DECB -(R0) ;R0=1 LOC. 0=01
        DEC R0 ;R0=0
        BEQ NEG22

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

```
1933                                     :           REPLACE THE MOVE INSTRUCTION <=====  
1934                                     :           WHICH FOLLOWS W/ 754           <=====  
1935 004602 012742 000107                MOV    #107,-(R2)  ;MOVE TO MAILBOX # ***** 107 *****  
1936 004606 005242                       INC    -(R2)       ;SET MSGTYP TO FATAL ERROR  
1937 004610 000000                       HALT                               ;REGISTER NOT INCREMENTED CORRECTLY  
1938 004612 005337 000000                NEG22: DEC    @#0   ;LOC. 0=0  
1939 004616 001404                       BEQ    TS57  
1940                                     :           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
1941                                     :           CONDITIONAL BRANCH INST. AND <=====  
1942                                     :           REPLACE THE MOVE INSTRUCTION <=====  
1943                                     :           WHICH FOLLOWS W/ 745           <=====  
1944 004620 012742 000110                MOV    #110,-(R2) ;MOVE TO MAILBOX # ***** 110 *****  
1945 004624 005242                       INC    -(R2)       ;SET MSGTYP TO FATAL ERROR  
1946 004626 000000                       HALT                               ;NEG BYTE INSTRUCTIONS FAILED  
1947                                     :           ; OR SEQUENCE ERROR  
1948  
1949  
1950
```

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

```
1951  
1952  
1953  
1954  
1955  
1956  
1957  
1958  
1959  
1960  
1961  
1962  
1963  
1964  
1965  
1966 004630 005212 000057                TS57:  INC    (R2)       ;UPDATE TEST NUMBER  
1967 004632 022712                       CMP    #57,(R2)    ;SEQUENCE ERROR?  
1968 004636 001020                       BNE    TS60-10     ;BR TO ERROR HALT ON SEQ ERROR  
1969 004640 005000                       CLR    R0          ;SET R0=400  
1970 004642 105100                       COMB   R0  
1971 004644 005200                       INC    R0  
1972 004646 005010                       CLR    (R0)        ;CLEAR LOC 400  
1973 004650 005030                       CLR    @(R0)+      ;TRY TO CLEAR LOC 0 USING MODE 3 ;R0=402  
1974 004652 001404                       BEQ    SOP3A  
1975  
1976                                     :           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
1977                                     :           CONDITIONAL BRANCH INST. AND <=====  
1978                                     :           REPLACE THE MOVE INSTRUCTION <=====  
1979 004654 012742 000111                MOV    #111,-(R2) ;MOVE TO MAILBOX # ***** 111 *****  
1980 004660 005242                       INC    -(R2)       ;SET MSGTYP TO FATAL ERROR  
1981 004662 000000                       HALT                               ;CLR DID NOT SET Z-BIT  
1982 004664 005300                SOP3A: DEC    R0          ;RESET R0=400  
1983 004666 005300                       DEC    R0  
1984 004670 005130                       COM    @(R0)+      ;TRY TO COMPLEMENT LOC 0 OF MODE 3 ;R0=402  
1985 004672 100002                       BPL    SOP3B  
1986 004674 005230                       INC    @(R0)+      ;TRY TO INCREMENT LOC 0 W/MODE 3 ;R0=404  
1987 004676 001404                       BEQ    TS60  
1988                                     :           ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
1989
```

```
1989  
1990  
1991  
1992 004700 SOP3B: :  
1993 004700 012742 000112 MOV #112,-(R2) ;MOVE TO MAILBOX # ***** 112 *****  
1994 004704 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
1995 004706 000000 HALT ;CUMMULATIVE RESULT OF ABOVE INST FAILED  
1996  
1997 ; OR SEQUENCE ERROR  
1998  
1999
```

```
2000 :*****  
2001 : THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS  
2002 : WHICH ADDRESS EVEN BYTES. AGAIN, THE TARGET LOCATION 0 IS USED  
2003 : AND THE SAME TABLE AT 400 IS EMPLOYED.  
2004 : AFTER POINTING R4 TO THE TABLE (400) AND SETTING LOCATION  
2005 : 0 TO -1, A CLRB INSTRUCTION IS USED TO CLEAR BYTE 0.  
2006 : SEVERAL OTHER MODE 3 INSTRUCTIONS ARE THEN USED WITH THE TABLE  
2007 : TO VERIFY THE DATA RESULTS AND THE PROPER INCREMENTING OF THE REGISTER.  
2008 : IF A FAILURE IS DETECTED, BE SURE THAT THE TABLE (LOCATION 400-402) HAS  
2009 : THE PROPER VALUES (0).  
2010 :*****
```

```
2011 :TEST 60 TEST MODE 3 EVEN BYTE USING SOP INST.  
2012 :*****
```

```
2013 004710 005212 TS60: INC (R2) ;UPDATE TEST NUMBER  
2014 004712 022712 000060 CMP #60,(R2) ;SEQUENCE ERROR?  
2015 004716 001026 BNE TS61-10 ;BR TO ERROR HALT ON SEQ ERROR  
2016 004720 005004 CLR R4 ;SET R4=400  
2017 004722 105104 COMB R4  
2018 004724 005204 INC R4  
2019 004726 005000 CLR R0 ;INITIALIZE LOC. 0--1  
2020 004730 005010 CLR (R0)  
2021 004732 005110 COM (R0) ;LOC. 0=-1  
2022 004734 105034 CLRB @(R4)+ ;TRY TO CLEAR EVEN BYTE ;LOC. 0=177400 R4=402  
2023 004736 001404 BEQ SOPB3A
```

```
2024 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
2025 : CONDITIONAL BRANCH INST. AND <====  
2026 : REPLACE THE MOVE INSTRUCTION <---  
2027 : WHICH FOLLOWS W/ 767 <---
```

```
2028 004740 012742 000113 SOPB3A: MOV #113,-(R2) ;MOVE TO MAILBOX # ***** 113 *****  
2029 004744 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
2030 004746 000000 HALT ;CLRB DID NOT SET Z-BIT  
2031 004750 005304 DEC R4 ;RESET POINTER R4=400  
2032 004752 005304 DEC R4  
2033 004754 005234 INC @(R4)+ ;TRY INCREMENTING WORD LOC.0=177401 R4=402  
2034 004756 100006 BPL SOPB3B  
2035 004760 105434 NEGB @(R4)+ ;TRY TO NEGATE EVEN BYTE ;LOC.0=-1 R4 404  
2036 004762 100004 BPL SOPB3B  
2037 004764 005304 DEC R4 ;R4-402  
2038 004766 005304 DEC R4  
2039 004770 105234 INCB @(R4)+ ;TRY TO INCREMENT EVEN BYTE ;LOC. 0=17400  
2040 004772 001404 BEQ TS61
```

```
2041 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < ---  
2042 : CONDITIONAL BRANCH INST. AND < -  
2043 : REPLACE THE MOVE INSTRUCTION <= -  
2044 : WHICH FOLLOWS W/ 751 <
```


2045 004774
2046 004774 012742 000114
2047 005000 005242
2048 005002 000000
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067 005004 005212
2068 005006 022712 000061
2069 005012 001024
2070 005014 005000
2071 005016 105100
2072 005020 005200
2073 005022 005030
2074 005024 005130
2075 005026 105030
2076 005030 001404
2077
2078
2079
2080
2081 005032 012742 000115
2082 005036 005242
2083 005040 000000
2084 005042 005300
2085 005044 005300
2086 005046 005300
2087 005050 005300
2088 005052 005230
2089 005054 105430
2090 005056 100002
2091 005060 105230
2092 005062 001404
2093
2094
2095
2096
2097 005064
2098 005064 012742 000116
2099 005070 005242
2100 005072 000000

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

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

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

TS61:

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

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

SOPB3C:

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

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

SOPB3D:

MOV #116,-(R2) ;MOVE TO MAILBOX # ***** 116 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CUMMULATIVE RESULT OF ABOVE INSTS FAILED

```
2101                                     ; OR SEQUENCE ERROR
2102                                     ;*****
2103 :TEST 62 TEST MODE 3 USING NEGATE INSTRUCTION
2104 :*****
2105 005074 005212 TS62: INC (R2) ;UPDATE TEST NUMBER
2106 005076 022712 000062 CMP #62,(R2) ;SEQUENCE ERROR?
2107 005102 001054 BNE TS63-10 ;BR TO ERROR HALT ON SEQ ERROR
2108 005104 005090 CLR R0 ;R0=400
2109 005106 105100 COMB R0
2110 005110 005200 INC R0
2111 005112 005010 CLR (R0) ;LOC. 400=0
2112 005114 005004 CLR R4 ;R4=0
2113 005116 005014 CLR (R4) ;LOC. 0=0
2114 005120 005214 INC (R4) ;LOC. 0=1
2115 005122 005430 NEG @(R0)+ ;TRY NEGATE LOC. 0--1 R0=402
2116 005124 100003 BPL NEG30 ;CC=1001?
2117 005126 001402 BEQ NEG30
2118 005130 102401 BVS NEG30
2119 005132 103404 BCS NEG31
2120                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2121                                     ; CONDITIONAL BRANCH INST. AND <====
2122                                     ; REPLACE THE MOVE INSTRUCTION <====
2123                                     ; WHICH FOLLOWS W/ 763 <====
2124 005134 NEG30:
2125 005134 012742 000117 MOV #117,-(R2) ;MOVE TO MAILBOX # ***** 117 *****
2126 005140 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2127 005142 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
2128 005144 005214 NEG31: INC (R4) ;LOC. 0=0
2129 005146 001404 BEQ NEG32
2130                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2131                                     ; CONDITIONAL BRANCH INST. AND <====
2132                                     ; REPLACE THE MOVE INSTRUCTION <====
2133                                     ; WHICH FOLLOWS W/ 755 <====
2134 005150 012742 000120 MOV #120,-(R2) ;MOVE TO MAILBOX # ***** 120 *****
2135 005154 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2136 005156 000000 HALT ;DATA RESULT OF NEG INCORRECT
2137 005160 105137 000001 NEG32: COMB @#1 ;LOC 0=177400
2138 005164 005237 000000 INC @#0 ;LOC. 0=177401
2139 005170 105430 NEGB @(R0)+ ;TRY NEGB LOC. 0=177777 R0=404
2140 005172 100404 BMI NEG33
2141                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2142                                     ; CONDITIONAL BRANCH INST. AND <====
2143                                     ; REPLACE THE MOVE INSTRUCTION <====
2144                                     ; WHICH FOLLOWS W/ 743 <====
2145 005174 012742 000121 MOV #121,-(R2) ;MOVE TO MAILBOX # ***** 121 *****
2146 005200 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2147 005202 000000 HALT ;NEGB FAILED WITH EVEN BYTE
2148 005204 105430 NEG33: NEGB @(R0)+ ;TRY NEGB LOC.0=777 R0=406
2149 005206 100004 BPL NEG34
2150                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2151                                     ; CONDITIONAL BRANCH INST. AND <====
2152                                     ; REPLACE THE MOVE INSTRUCTION <====
2153                                     ; WHICH FOLLOWS W/ 735 <====
2154 005210 012742 000122 MOV #122,-(R2) ;MOVE TO MAILBOX # ***** 122 *****
2155 005214 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2156 005216 000000 HALT ;NEGB FAILED WITH ODD BYTE
```

```

2157 005220 105137 000001      NEG34: COMB  @#1      ;LOC. 0=177377
2158 005224 105237 000001      INCB  @#1      ;LOC. 0=177777
2159 005230 005214              INC   (R4)     ;LOC. 0=0
2160 005232 001404              BEQ   TS63
2161
2162
2163
2164
2165 005234 012742 000123      MOV   #123,-(R2) ;MOVE TO MAILBOX # ***** 123 *****
2166 005240 005242              INC   -(R2)    ;SET MSGTYP TO FATAL ERROR
2167 005242 000000              HALT          ;DATA RESULT OF NEGB'S INCORRECT
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
    
```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---=
;          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.
:
:R0 IS SET TO 400. A CLR INSTRUCTION IS EXECUTED IN MODE 4 TO CLEAR
:LOC. 376. R0 IS RESET TO 400 AND A COM INSTRUCTION USING MODE 4
:COMPLEMENTS LOC.376.
: TWO INC INSTRUCTIONS AND A MODE 4 INSTRUCTION ARE EXECUTED
:TO COMPLETE THE TEST.
:*****
    
```

```

:TEST 63 TEST MODE 4 USING SOP INSTS
:*****
    
```

```

2182 005244 005212 000063      TS63:  INC   (R2)      ;UPDATE TEST NUMBER
2183 005246 022712              CMP   #63,(R2)   ;SEQUENCE ERROR?
2184 005252 001021              BNE  TS64-10    ;BR TO ERROR HALT ON SEQ ERROR
2185 005254 005000              CLR  R0         ;SET R0=400
2186 005256 105100              COMB R0
2187 005260 005200              INC  R0
2188 005262 005040              CLR  -(R0)     ;TRY TO CLEAR USING MODE 4
2189 005264 001404              BEQ  SOP4A
2190
2191
2192
2193
2194 005266 012742 000124      MOV   #124,-(R2) ;MOVE TO MAILBOX # ***** 124 *****
2195 005272 005242              INC  -(R2)     ;SET MSGTYP TO FATAL ERROR
2196 005274 000000              HALT          ;CLR DID NOT SET Z-BIT
2197 005276 005200      SOP4A: INC  R0       ;RESET R0
2198 005300 005200              INC  R0
2199 005302 005140              COM  -(R0)    ;TRY TO COMPLEMENT USING MODE 4
2200 005304 100004              BPL  SOP4B
2201 005306 005200              INC  R0       ;MOVE POINTER
2202 005310 005200              INC  R0
2203 005312 005240              INC  -(R0)
2204 005314 001404              BEQ  TS64
2205
2206
2207
2208
2209
2210 005316 012742 000125      SOP4B: MOV   #125,-(R2) ;MOVE TO MAILBOX # ***** 125 *****
2211 005322 005242              INC  -(R2)    ;SET MSGTYP TO FATAL ERROR
2212 005324 000000              HALT          ;CHECK CUMMULATIVE RESULT OF ABOVE INST.
    
```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 772 <====
;MOVE TO MAILBOX # ***** 124 *****
;SET MSGTYP TO FATAL ERROR
;CLR DID NOT SET Z-BIT
;RESET R0
;TRY TO COMPLEMENT USING MODE 4
;MOVE POINTER
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 756 <====
;MOVE TO MAILBOX # ***** 125 *****
;SET MSGTYP TO FATAL ERROR
;CHECK CUMMULATIVE RESULT OF ABOVE INST.
    
```

```
2213 ; OR SEQUENCE ERROR
2214
2215 :*****
2216 :
2217 : THIS TEST VERIFIES MODE 5 SINGLE OPERAND INSTRUCTIONS. IT
2218 : USES LOCATION 0 AS ITS TARGET DATA. A TABLE LOCATED AT LOC. 372
2219 : THRU 374 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
2220 : INSTRUCTIONS UNDER TEST.
2221 : R0 IS SET TO 376, (THE START OF THE ADDRESS TABLE) +2,
2222 : AND A CLR INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR
2223 : LOC. 0. THEN R0 IS INCREMENTED BY TWO AND TWO OTHER MODE 3
2224 : INSTRUCTIONS OPERATE ON LOC. 0 TO VERIFY THE DATA RESULTS OF
2225 : THE TEST. THE PROPER DECREMENTING OF THE REGISTER IS ALSO
2226 : VERIFIED IN THIS MANNER.
2227 : IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
2228 : (LOC. 372 THRU 374) HAS THE PROPER VALUES (0).
2229 :
2230 :*****
2231 :TEST 64 TEST MODE 5 USING SOP INSTS
2232 :*****
2233 005326 005212 TS64: INC (R2) ;UPDATE TEST NUMBER
2234 005330 022712 000064 CMP #64,(R2) ;SEQUENCE ERROR?
2235 005334 001025 BNE TS65-10 ;BR TO ERROR HALT ON SEQ ERROR
2236 005336 012700 000370 MOV #370,R0 ;CLEAR LOCATION 370-376
2237 005342 005020 CLR (R0)+ ;370
2238 005344 005020 CLR (R0)+ ;372
2239 005346 005020 CLR (R0)+ ;374
2240 005350 005010 CLR (R0) ;376
2241 005352 005000 CLR R0 ;SET R0-376 (LOW BYTE)
2242 005354 005020 CLR (R0)+
2243 005356 105400 NEGB R0
2244 005360 005050 CLR @-(R0) ;TRY TO CLEAR LOC 0 W/MODE 5
2245 005362 001404 BEQ SOP5A
2246
2247 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
2248 ; CONDITIONAL BRANCH INST. AND <
2249 ; REPLACE THE MOVE INSTRUCTION <
2250 ; WHICH FOLLOWS W/ 764 <
2251 005364 012742 000126 MOV #126,-(R2) ;MOVE TO MAILBOX # ***** 126 *****
2252 005370 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2253 005372 000000 HALT ;CLR DID NOT SET Z-BIT
2254 005374 005200 SOP5A: INC R0 ;RESET R0
2255 005376 005200 INC R0
2256 005400 005150 COM @-(R0) ;TRY TO COMPLEMENT LOC. 0 W/MODE 5
2257 005402 100002 BPL SOP5B
2258 005404 005250 INC @-(R0) ;TRY TO INCREMENT LOC. 0 W/MODE 5
2259 005406 001404 BEQ TS65
2260 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
2261 ; CONDITIONAL BRANCH INST. AND <===
2262 ; REPLACE THE MOVE INSTRUCTION <= ==
2263 ; WHICH FOLLOWS W/ 752 <= -
2264 005410 012742 000127 SOP5B: MOV #127,-(R2) ;MOVE TO MAILBOX # ***** 127 *****
2265 005414 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2266 005416 000000 HALT ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
2267 ; OR SEQUENCE ERROR
2268
```

2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280 005420 005212
2281 005422 022712 000065
2282 005426 001020
2283 005430 005000
2284 005432 105100
2285 005434 005200
2286 005436 005060 177400
2287 005442 001404
2288
2289
2290
2291
2292 005444 012742 000130
2293 005450 005242
2294 005452 000000
2295 005454 005160 177400
2296 005460 100003
2297 005462 005260 177400
2298 005466 001404
2299
2300
2301
2302
2303 005470
2304 005470 012742 000131
2305 005474 005242
2306 005476 000000
2307
2308
2309
2310
2311
2312
2313
2314
2315
2316
2317
2318
2319
2320
2321 005500 005212
2322 005502 022712 000066
2323 005506 001021
2324 005510 005000

: 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
BEQ SOP6A
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
: CONDITIONAL BRANCH INST. AND <- -
: REPLACE THE MOVE INSTRUCTION <- -
: WHICH FOLLOWS W/ 771 <==
MOV #130,-(R2) ;MOVE TO MAILBOX # ***** 130 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR DID NOT SET Z-BIT
SOP6A: COM -400(R0) ;TRY TO COMPLEMENT LOCATION 0 W/MODE 6
BPL SOP6B
INC -400(R0) ;TRY TO INCREMENT LOCATION 0 W/MODE 6
BEQ TS66
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
: CONDITIONAL BRANCH INST. AND <===
: REPLACE THE MOVE INSTRUCTION <===
: WHICH FOLLOWS W/ 757 <===
SOP6B: MOV #131,-(R2) ;MOVE TO MAILBOX # ***** 131 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
: OR SEQUENCE ERROR

: THIS TEST VERIFIES MODE 7 SINGLE OPERAND INSTRUCTIONS. IT USES
: THE POINTER TO LOC. 0 WHICH IS STORED AT LOC. 402.
: R0 IS SET TO 400 AND A MODE 7 CLR INSTRUCTION IS
: EXECUTED WITH A +2 OFFSET TO CLEAR LOC. 0.
: SEVERAL OTHER MODE 7 INSTRUCTIONS ARE THEN USED ON THE COMMON
: LOCATION TO VERIFY THE DATA RESULTS.

: TEST 66 TEST MODE 7 USING SOP INST.

TS66: INC (R2) ;UPDATE TEST NUMBER
CMP #66,(R2) ;SEQUENCE ERROR?
BNE TS67-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;SET R0=400

```
2325 005512 105100          COMB  R0
2326 005514 005200          INC   R0
2327 005516 005210          INC   (R0)
2328 005520 005070 000002   CLR   @2(R0)
2329 005524 001404          BEQ   SOP7A
2330
2331                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
2332                          ;          CONDITIONAL BRANCH INST. AND
2333                          ;          REPLACE THE MOVE INSTRUCTION
2334                          ;          WHICH FOLLOWS W/ 770
2334 005526 012742 000132   MOV   #132,-(R2)
2335 005532 005242          INC   -(R2)
2336 005534 000000          HALT
2337 005536 005170 000002   SOP7A: COM @2(R0)
2338 005542 100003          BPL   SOP7B
2339 005544 005270 000002   INC   @2(R0)
2340 005550 001404          BEQ   TS67
2341
2342                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
2343                          ;          CONDITIONAL BRANCH INST. AND
2344                          ;          REPLACE THE MOVE INSTRUCTION
2345                          ;          WHICH FOLLOWS W/ 756
2345 005552          SOP7B:
2346 005552 012742 000133   MOV   #133,-(R2)
2347 005556 005242          INC   -(R2)
2348 005560 000000          HALT
2349
2350                          ; OR SEQUENCE ERROR
2351
2352                          ;*****
2353                          ;TEST 67          TEST MODE 4 WITH NEGATE INSTRUCTION
2354                          ;*****
2354 005562 005212          TS67: INC   (R2)
2355 005564 022712 000067   CMP   #67,(R2)
2356 005570 001024          BNE   TS70-10
2357 005572 005000          CLR   R0
2358 005574 005010          CLR   (R0)
2359 005576 005120          COM   (R0)+
2360 005600 005440          NEG   -(R0)
2361 005602 100403          BMI   NEG40
2362 005604 001402          BEQ   NEG40
2363 005606 102401          BVS   NEG40
2364 005610 103404          BCS   NEG41
2365
2366                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
2367                          ;          CONDITIONAL BRANCH INST. AND
2368                          ;          REPLACE THE MOVE INSTRUCTION
2369                          ;          WHICH FOLLOWS W/ 767
2369 005612          NEG40:
2370 005612 012742 000134   MOV   #134,-(R2)
2371 005616 005242          INC   -(R2)
2372 005620 000000          HALT
2373 005622 005400          NEG41: NEG   R0
2374 005624 001404          BEQ   NEG42
2375
2376                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
2377                          ;          CONDITIONAL BRANCH INST. AND
2378                          ;          REPLACE THE MOVE INSTRUCTION
2379                          ;          WHICH FOLLOWS W/ 761
2379 005626 012742 000135   MOV   #135,-(R2)
2380 005632 005242          INC   -(R2)
2380                          ; SET MSGTYP TO FATAL ERROR
```

```
2381 005634 000000
2382 005636 005310
2383 005640 001404
2384
2385
2386
2387
2388 005642 012742 000136
2389 005646 005242
2390 005650 000000
2391
2392
2393
2394
2395 005652 005212
2396 005654 022712 000070
2397 005660 001031
2398 005662 005000
2399 005664 005010
2400 005666 105100
2401 005670 005200
2402 005672 005010
2403 005674 005004
2404 005676 005314
2405 005700 005450
2406 005702 100403
2407 005704 001402
2408 005706 102401
2409 005710 103404
2410
2411
2412
2413
2414 005712
2415 005712 012742 000137
2416 005716 005242
2417 005720 000000
2418 005722 005314
2419 005724 001404
2420
2421
2422
2423
2424 005726 012742 000140
2425 005732 005242
2426 005734 000000
2427 005736 105100
2428 005740 005300
2429 005742 001404
2430
2431
2432
2433
2434 005744 012742 000141
2435 005750 005242
2436 005752 000000
```

NEG42: HALT ;RO NOT DECREMENTED PROPERLY
DEC (R0) ;TEST DTA RESULT OF NEG
BEQ TS70

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

MOV #136,-(R2) ;MOVE TO MAILBOX # ***** 136 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DATA RESULT OF NEG INCORRECT
OR SEQUENCE ERROR

;TEST 70 TEST MODE 5 WITH NEGATE INSTRUCTION

TS70: INC (R2) ;UPDATE TEST NUMBER
CMP #70,(R2) ;SEQUENCE ERROR?
BNE TS71-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC. 0=0
COMB R0 ;R0=377
INC R0 ;R0=400
CLR (R0) ;SET 400 = 0
CLR R4 ;R4=0
DEC (R4) ;LOC. 0=177777
NEG @-(R0) ;TRY NEGATE: LOC. 0-1
BMI NEG50 ;CC=0001?
BEQ NEG50
BVS NEG50
BCS NEG51

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

NEG50: MOV #137,-(R2) ;MOVE TO MAILBOX # ***** 137 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;NEG DID NOT SET CC'S CORRECTLY

NEG51: DEC (R4)
BEQ NEG52

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

NEG52: MOV #140,-(R2) ;MOVE TO MAILBOX # ***** 140 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DATA RESULT OF NEG INCORRECT
COMB R0
DEC R0
BEQ TS71

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

MOV #141,-(R2) ;MOVE TO MAILBOX # ***** 141 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;REGISTER NOT DECREMENTED PROPERLY

```
2437 ; OR SEQUENCE ERROR
2438 :*****
2439 :TEST 71 TEST MODE 6 WITH NEGATE
2440 :*****
2441 TS71: INC (R2) ;UPDATE TEST NUMBER
2442 005754 005212 000071 CMP #71,(R2) ;SEQUENCE ERROR?
2443 005756 022712 BNE TS72-10 ;BR TO ERROR HALT ON SEQ ERROR
2444 005762 001022 CLR R0 ;R0=0
2445 005764 005000 CLR R4 ;R4=0
2446 005766 005004 COMB R0 ;R0=377
2447 005770 105100 CLR (R4) ;LOC. 0=0
2448 005772 005014 CLR (R4)+ ;LOC. 0=177777, R4-1
2449 005774 105024 COMB (R4) ;LOC. 0=177400
2450 006000 005460 177401 NEG -377(R0) ;LOC. 0=400
2451 006004 100403 BMI NEG60 ;CC=0001
2452 006006 001402 BEQ NEG60
2453 006010 102401 BVS NEG60
2454 006012 103404 BCS NEG61
2455 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
2456 ; CONDITIONAL BRANCH INST. AND <
2457 ; REPLACE THE MOVE INSTRUCTION <
2458 ; WHICH FOLLOWS W/ 763 <
2459 006014 NEG60:
2460 006014 012742 000142 MOV #142,-(R2) ;MOVE TO MAILBOX # ***** 142 *****
2461 006020 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2462 006022 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
2463 006024 105314 NEG61: DECB (R4)
2464 006026 001404 BEQ TS72
2465 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
2466 ; CONDITIONAL BRANCH INST. AND <
2467 ; REPLACE THE MOVE INSTRUCTION <
2468 ; WHICH FOLLOWS W/ 755 <
2469 006030 012742 000143 MOV #143,-(R2) ;MOVE TO MAILBOX # ***** 143 *****
2470 006034 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2471 006036 000000 HALT ;DATA RESULT OF NEG INCORRECT
2472 ; OR SEQUENCE ERROR
2473 :*****
2474 :TEST 72 TEST MODE 7 W/ NEGATE
2475 :*****
2476 TS72: INC (R2) ;UPDATE TEST NUMBER
2477 006040 005212 000072 CMP #72,(R2) ;SEQUENCE ERROR?
2478 006042 022712 BNE TS73-10 ;BR TO ERROR HALT ON SEQ ERROR
2479 006046 001024 CLR R0 ;R0=0
2480 006050 005000 CLR (R0) ;LOC. 0=0
2481 006052 005010 COM (R0) ;LOC. 0=177777
2482 006054 005110 COMB R0 ;R0=377
2483 006056 105100 000005 NEGB @5(R0) ;R0+5-404, 404-1, LOC. 0=777
2484 006060 105470 BMI NEG70 ;CC=0001?
2485 006064 100403 BEQ NEG70
2486 006066 001402 BVS NEG70
2487 006070 102401 BCS NEG71
2488 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2489 ; CONDITIONAL BRANCH INST. AND <====
2490 ; REPLACE THE MOVE INSTRUCTION <====
2491 ; WHICH FOLLOWS W/ 765 <====
2492 006074 NEG70:
```



```
2493 006074 012742 000144      MOV      #144,-(R2)      ;MOVE TO MAILBOX # ***** 144 *****
2494 006100 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2495 006102 000000              HALT                    ;NEG DID NOT SET CC'S CORRECTLY
2496 006104 105100      NEG71:  COMB      R0      ;R0=0
2497 006106 105120              COMB      (R0)+        ;LOC. 0=400, R0=1
2498 006110 105310              DECB     (R0)          ;LOC. 0=0
2499 006112 005467 171662      NEG      0              ;USE NEG MODE 67 TO TST FOR ZERO
2500 006116 001404              BEQ      TS73
2501
2502                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
2503                          ;          CONDITIONAL BRANCH INST. AND <---
2504                          ;          REPLACE THE MOVE INSTRUCTION <---
2505                          ;          WHICH FOLLOWS W/ 753 <---
2505 006120 012742 000145      MOV      #145,-(R2)      ;MOVE TO MAILBOX # ***** 145 *****
2506 006124 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2507 006126 000000              HALT                    ;DATA RESULT OF NEG WAS INCORRECT
2508
2509                          ; OR SEQ ENCE ERROR
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531 006146 012742 000146      MOV      #146,-(R2)      ;MOVE TO MAILBOX # ***** 146 *****
2532 006152 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2533 006154 000000              HALT                    ;CLR DID NOT SET Z-BIT
2534 006156 005237 006142      SOPA:   INC      @#SOPX    ;INC SOPX W/MODE 37
2535 006162 005467 177754      NEG      SOPX           ;NEGATE SOPX W/MODE 67
2536 006166 100003              BPL      SOPB
2537 006170 005277 000012      INC      @SOPXAD        ;INC SOPX W/MODE 77
2538 006174 001405              BEQ      TS74
2539
2540                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
2541                          ;          CONDITIONAL BRANCH INST. AND <---
2542                          ;          REPLACE THE MOVE INSTRUCTION <---
2543                          ;          WHICH FOLLOWS W/ 760 <---
2543 006176
2544 006176 012742 000147      MOV      #147,-(R2)      ;MOVE TO MAILBOX # ***** 147 *****
2545 006202 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2546 006204 000000              HALT                    ;INC DID NOT SET Z-BIT
2547
2548      SOPXAD: SOPX          ; OR SEQUENCE ERROR
                          ;INDIRECT ADDRESS OF SOPX
```

2549
 2550
 2551
 2552
 2553
 2554
 2555
 2556
 2557
 2558
 2559
 2560
 2561 006210 005212
 2562 006212 022712 000074
 2563 006216 001010
 2564 006220 005000
 2565 006222 000277
 2566 006224 000244
 2567 006226 005700
 2568 006230 102403
 2569 006232 100402
 2570 006234 103401
 2571 006236 001404
 2572
 2573
 2574
 2575
 2576 006240
 2577 006240 012742 000150
 2578 006244 005242
 2579 006246 000000
 2580
 2581
 2582
 2583
 2584
 2585
 2586
 2587
 2588
 2589
 2590
 2591
 2592
 2593 006250 005212
 2594 006252 022712 000075
 2595 006256 001010
 2596 006260 005000
 2597 006262 105100
 2598 006264 000277
 2599 006266 000250
 2600 006270 105700
 2601 006272 102402
 2602 006274 101401
 2603 006276 100404
 2604

```

*****
:
:   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     CC=1011     ;SET CC=1011
        CLZ
        TST     R0          ;TRY TST W/ MODE 0
        BVS     SNMOA       ;CHECK THAT CC=0100
        BMI     SNMOA
        BCS     SNMOA
        BEQ     TS75
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
:             CONDITIONAL BRANCH INST. AND <===
:             REPLACE THE MOVE INSTRUCTION <===
:             WHICH FOLLOWS W/ 767        <===

```

```

SNMOA:  MOV     #150,-(R2)    ;MOVE TO MAILBOX # ***** 150 *****
        INC     -(R2)
        HALT
:
:   OR SEQUENCE ERROR

```

```

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

```

```

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

```

```

*****
TS75:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #75,(R2)    ;SEQUENCE ERROR?
        BNE     TS76-10     ;BR TO ERROR HALT ON SEQ ERROR
        CLR     R0          ;INITIALIZE
        COMB    R0          ;R0=377
        SCC     CC=0111     ;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 <===

```

2605 :
2606 :
2607 :
2608 :
2609 :
2610 :
2611 :
2612 :
2613 :
2614 :
2615 :
2616 :
2617 :
2618 :
2619 :
2620 :
2621 :
2622 :
2623 :
2624 :
2625 :
2626 :
2627 :
2628 :
2629 :
2630 :
2631 :
2632 :
2633 :
2634 :
2635 :
2636 :
2637 :
2638 :
2639 :
2640 :
2641 :
2642 :
2643 :
2644 :
2645 :
2646 :
2647 :
2648 :
2649 :
2650 :
2651 :
2652 :
2653 :
2654 :
2655 :
2656 :
2657 :
2658 :
2659 :
2660 :

006300
006300 012742 000151
006304 005242
006306 000000

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

: 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.

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 RO ;POINT TO LOC 0
CLR (RO) ;CLEAR LOC 0
SCC ;INITIALIZE
CLZ ;CC=1011
TST (RO) ;TRY TST W/ MODE 1
BVS SNM1A ;CHECK CC=0100
BCS SNM1A
BMI SNM1A
BEQ TS77

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

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 RO ;POINT TO LOC 0

```
2661 006364 005010 CLR (R0) ;CLEAR LOC 0
2662 006366 105110 COMB (R0) ;COMPLEMENT BYTE 0
2663 006370 000277 SCC ;SET CC=0111
2664 006372 000250 CLN
2665 006374 105710 TSTB (R0) ;TRY TST ON EVEN BYTE
2666 006376 102402 BVS SNMB1A
2667 006400 101401 BLOS SNMB1A
2668 006402 100404 BMI SNMB1B
2669 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=--
2670 ; CONDITIONAL BRANCH INST. AND < --
2671 ; REPLACE THE MOVE INSTRUCTION <===
2672 ; WHICH FOLLOWS W/ 766 <===
2673 006404 SNMB1A:
2674 006404 012742 000153 MOV #153,-(R2) ;MOVE TO MAILBOX # ***** 153 *****
2675 006410 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2676 006412 000000 HALT ;CC'S NOT CORRECT
2677 006414 005000 SNMB1B: CLR R0
2678 006416 005200 INC R0
2679 006420 000277 SCC ;SET CC=1011
2680 006422 000244 CLZ
2681 006424 105710 TSTB (R0) ;TRY TO TST AN ODD BYTE
2682 006426 102403 BVS SNMB1C
2683 006430 103402 BCS SNMB1C
2684 006432 100401 BMI SNMB1C
2685 006434 001404 BEQ TS100
2686 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2687 ; CONDITIONAL BRANCH INST. AND <====
2688 ; REPLACE THE MOVE INSTRUCTION <====
2689 ; WHICH FOLLOWS W/ 751 <====
2690 006436 SNMB1C:
2691 006436 012742 000154 MOV #154,-(R2) ;MOVE TO MAILBOX # ***** 154 *****
2692 006442 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2693 006444 000000 HALT ;CC'S NOT CORRECT
2694 ; OR SEQUENCE ERROR
2695
2696 :*****
2697 :
2698 : THIS TEST VERIFIES THE SINGLE-OPERAND NON-MODIFYING INSTRUCTIONS
2699 : USING MODE 2. IT USES THE IDENTICAL PROCEDURE EMPLOYED IN THE
2700 : MODE 1 TESTS. ADDITIONALLY, THE REGISTER IS CHECKED TO ASSURE THAT
2701 : IT IS INCREMENTED PROPERLY.
2702 :
2703 :*****
2704 :TEST 100 TEST MODE 2 WITH SOP NON-MODIFYING
2705 :*****
2706 006446 005212 000100 TS100: INC (R2) ;UPDATE TEST NUMBER
2707 006450 022712 000100 CMP #100,(R2) ;SEQUENCE ERROR?
2708 006454 001020 BNE TS101-10 ;BR TO ERROR HALT ON SEQ ERROR
2709 006456 005000 CLR R0 ;INITIALIZE R0=0
2710 006460 005010 CLR (R0) ;CLEAR LOC 0
2711 006462 000277 SCC ;SET CC=1011
2712 006464 000244 CLZ
2713 006466 005720 TST (R0)+ ;TRY TS; W/ MODE 2
2714 006470 102403 BVS SNM2A ;CHECK CC=0100
2715 006472 103402 BCS SNM2A
2716 006474 100401 BMI SNM2A
```

```
2717 006476 001404          BEQ      SNM2B
2718
2719                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2720                          ;          CONDITIONAL BRANCH INST. AND <====
2721                          ;          REPLACE THE MOVE INSTRUCTION <====
2722                          ;          WHICH FOLLOWS W/ 766 <====
2722 006500          SNM2A:
2723 006500 012742 000155      MOV      #155,-(R2)      ;MOVE TO MAILBOX # ***** 155 *****
2724 006504 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2725 006506 000000          HALT
2726 006510 00530C          SNM2B: DEC      R0          ;CC'S NOT CORRECT
2727 006512 005300          DEC      R0          ;RESET R0
2728 006514 001404          BEQ      TS101
2729
2730                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2731                          ;          CONDITIONAL BRANCH INST. AND <====
2732                          ;          REPLACE THE MOVE INSTRUCTION <====
2733                          ;          WHICH FOLLOWS W/ 757 <====
2733 006516 012742 000156      MOV      #156,-(R2)      ;MOVE TO MAILBOX # ***** 156 *****
2734 006522 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2735 006524 000000          HALT          ;MODE 2 DID NOT INC REQ CORRECTLY
2736
2737                          ; OR SEQUENCE ERROR
2738
2739 *****
2740                          ; THIS TEST VERIFIES MODE 2 SINGLE OPERAND NON-MODIFYING BYTE
2741                          ; INSTRUCTIONS IT USES R0 TO POINT TO LOC. 0. WITH LOCATION 0
2742                          ; SET TO 377, THE EVEN AND ODD BYTE IS TESTED WITH TSTB INSTRUCTIONS
2743                          ; TO VERIFY THE CORRECT CC ARE SET. THE REGISTER IS CHECKED FOR
2744                          ; PROPER INCREMENTING.
2745
2746 *****
2747                          ; TEST 101 TEST MODE 2 - BYTE W/ SOP NON-MODIFYING
2748 *****
2749 006526 005212 000101      TS101: INC      (R2)          ;UPDATE TEST NUMBER
2750 006530 022712          CMP      #101,(R2)      ;SEQUENCE ERROR?
2751 006534 001042          BNE     TS102-10      ;BR TO ERROR HALT ON SEQ ERROR
2752 006536 005000          CLR      R0          ;CLEAR R0
2753 006540 005010          CLR      (R0)         ;CLEAR LOC 0
2754 006542 105110          COMB   (R0)         ;SET LOC 0=377
2755 006544 000277          SCC
2756 006546 000250          CLN
2757 006550 105720          TSTB   (R0)+        ;TRY TST OF EVEN BYTE
2758 006552 102402          BVS     SNMB2A
2759 006554 101401          BLOS   SNMB2A
2760 006556 100404          BMI     SNMB2B
2761
2762                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2763                          ;          CONDITIONAL BRANCH INST. AND <====
2764                          ;          REPLACE THE MOVE INSTRUCTION <====
2765                          ;          WHICH FOLLOWS W/ 766 <====
2765 006560          SNMB2A:
2766 006560 012742 000157      MOV      #157,-(R2)      ;MOVE TO MAILBOX # ***** 157 *****
2767 006564 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2768 006566 000000          HALT
2769 006570 005300          SNMB2B: DEC      R0          ;CC'S NOT SET CORRECTLY
2770 006572 001404          BEQ      SNMB2C
2771
2772                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
2773                          ;          CONDITIONAL BRANCH INST. AND <==
```

```
2773
2774
2775 006574 012742 000160          MOV    #160,-(R2)      ;
2776 006600 005242                INC    -(R2)          ; REPLACE THE MOVE INSTRUCTION <====
2777 006602 000000                HALT                    ; WHICH FOLLOWS W/ 760 <====
2778 006604 005200          SNMB2C: INC    R0      ;:MOVE TO MAILBOX # ***** 160 *****
2779 006606 000277                SCC                    ;:SET MSGTYP TO FATAL ERROR
2780 006610 000244                CLZ                    ;:MODE 2 DID NOT INC REG CORRECTLY
2781 006612 105720                TSTB   (R0)+          ;:POINT TO ODD BYTE
2782 006614 102403                BVS    SNMB2D         ;:SET CC=1011
2783 006616 103402                BCS    SNMB2D
2784 006620 100401                BMI    SNMB2D
2785 006622 001404                BEQ    SNMB2E
2786
2787
2788
2789
2790 006624
2791 006624 012742 000161          SNMB2D: MOV    #161,-(R2) ;
2792 006630 005242                INC    -(R2)          ;:MOVE TO MAILBOX # ***** 161 *****
2793 006632 000000                HALT                    ;:SET MSGTYP TO FATAL ERROR
2794 006634 005300          SNMB2E: DEC    R0      ;:CC'S NOT CORRECT
2795 006636 005300                DEC    R0
2796 006640 001404                BEQ    TS102
2797
2798
2799
2800
2801 006642 012742 000162          MOV    #162,-(R2)      ;
2802 006646 005242                INC    -(R2)          ;:MOVE TO MAILBOX # ***** 162 *****
2803 006650 000000                HALT                    ;:SET MSGTYP TO FATAL ERROR
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816 006652 005212 000102          TS102: INC    (R2)      ;
2817 006654 022712                CMP    #102,(R2)      ;:UPDATE TEST NUMBER
2818 006660 001022                BNE    TS103-10       ;:SEQUENCE ERROR?
2819 006662 005000                CLR    R0              ;:BR TO ERROR HALT ON SEQ ERROR
2820 006664 005010                CLR    (R0)           ;:RO=0
2821 006666 105100                COMB   R0              ;:CLEAR LOC 0
2822 006670 005300                DEC    R0              ;:RO=376
2823 006672 000277                SCC                    ;:SET CC=1011
2824 006674 000244                CLZ
2825 006676 005730                TST    @ (R0)+        ;:TRY TST W/ MODE 3
2826 006700 102403                BVS    SNM3A         ;:CHECK CC=0100
2827 006702 103402                BCS    SNM3A
2828 006704 100401                BMI    SNM3A
```

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

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

```
2829 006706 001404          BEQ      SNM3B          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2830                                     ;          CONDITIONAL BRANCH INST. AND <====
2831                                     ;          REPLACE THE MOVE INSTRUCTION <====
2832                                     ;          WHICH FOLLOWS W/ 764 <====
2833
2834 006710          SNM3A:
2835 006710 012742 000163      MOV      #163,-(R2)      ;MOVE TO MAILBOX # ***** 163 *****
2836 006714 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2837 006716 000000          HALT
2838 006720 005300          SNM3B: DEC      R0      ;CC'S NOT CORRECT
2839 006722 105100          COMB    R0              ;R0=377
2840 006724 001404          BEQ      TS103         ;R0=0
2841
2842                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2843                                     ;          CONDITIONAL BRANCH INST. AND <====
2844                                     ;          REPLACE THE MOVE INSTRUCTION <====
2845                                     ;          WHICH FOLLOWS W/ 755 <====
2846 006726 012742 000164      MOV      #164,-(R2)      ;MOVE TO MAILBOX # ***** 164 *****
2847 006732 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
2848 006734 000000          HALT
2849                                     ;MODE 3 DID NOT INC REG CORRECTLY
2850                                     ; OR SEQUENCE ERROR
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
```

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

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

```
TS103: INC      (R2)          ;UPDATE TEST NUMBER
      CMP      #103,(R2)      ;SEQUENCE ERROR?
      BNE     TS104-10        ;BR TO ERROR HALT ON SEQ ERROR
      CLR     R0              ;R0=0
      CLR     (R0)            ;CLEAR LOC 0
      COMB    (R0)            ;LOC. 0 =377
      COMB    R0
      INC     R0
      TST     (R0)+           ;R0=402
      SCC     ;CC=0111
      CLN
      TSTB    @ (R0)+         ;TRY TST OF EVEN BYTE
      BVS     SNM3A           ;CHECK CC=1000
      BLOS    SNM3A
      BMI     SNM3B
2881
2882
2883
2884          SNM3A:
          MOV      #165,-(R2)      ;MOVE TO MAILBOX # ***** 165 *****
          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
          HALT                    ;CC'S NOT CORRECT
```

```
2885 007006 000277          SNMB3B: SCC                ;SET CC=1011
2886 007010 000244          CLZ
2887 007012 105730          TSTB    @(R0)+            ;TRY TST OF ODD BYTE
2888 007014 102403          BVS     SNMB3C            ;CHECK CC=0100
2889 007016 103402          BCS     SNMB3C
2890 007020 100401          BMI     SNMB3C
2891 007022 001404          BEQ     SNMB3D
2892
2893                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2894                          ;          CONDITIONAL BRANCH INST. AND <====
2895                          ;          REPLACE THE MOVE INSTRUCTION <====
2896                          ;          WHICH FOLLOWS W/ 750 <====
2896 007024          SNMB3C:
2897 007024 012742 000166      MOV     #166,-(R2)        ;MOVE TO MAILBOX # ***** 166 *****
2898 007030 005242          INC     -(R2)            ;SET MSGTYP TO FATAL ERROR
2899 007032 000000          HALT
2900 007034 005720          SNMB3D: TST    (R0)+      ;CC'S NOT CORRECT
2901 007036 005710          TST    (R0)             ;RO=410
2902 007040 100404          BMI     TS104
2903
2904                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2905                          ;          CONDITIONAL BRANCH INST. AND <====
2906                          ;          REPLACE THE MOVE INSTRUCTION <====
2907                          ;          WHICH FOLLOWS W/ 741 <====
2907 007042 012742 000167      MOV     #167,-(R2)        ;MOVE TO MAILBOX # ***** 167 *****
2908 007046 005242          INC     -(R2)            ;SET MSGTYP TO FATAL ERROR
2909 007050 000000          HALT                    ;TSTB DID NOT INCREMENT RO CORRECTLY
2910
2911                          ; OR SEQUENCE ERROR
2912
2913          :*****
2914          : THIS TEST VERIFIES MODE 4 SOP NON-MODIFYING INSTRUCTIONS.
2915          : LOC. 0 IS SET TO -1 AND THE CC'S ARE SET TO THE COMPLEMENT OF THE
2916          : EXPECTED RESULTS. RO AND SET TO 2 AND A TST MODE 4 IS EXECUTED.
2917          : THE CC'S ARE CHECKED WITH CONDITIONAL BRANCH INSTRUCTIONS AND THE REGISTER
2918          : IS CHECKED FOR PROPER DECREMENTING.
2919          :*****
2920          :TEST 104          TEST MODE 4 W/ SOP NON-MODIFYING INSTS
2921          :*****
2922 007052 005212 000104      TS104: INC     (R2)          ;UPDATE TEST NUMBER
2923 007054 022712          CMP     #104,(R2)        ;SEQUENCE ERROR?
2924 007060 001017          BNE     TS105-10        ;BR TO ERROR HALT ON SEQ ERROR
2925 007062 005000          CLR     R0              ;R0=0
2926 007064 005010          CLR     (R0)            ;LOC 0=0
2927 007066 005120          COM    (R0)+           ;LOC 0=-1
2928 007070 000277          SCC
2929 007072 000244          CLZ
2930 007074 005740          TST    -(R0)            ;TRY TST W/ MODE 4
2931 007076 102402          BVS     SNM4A            ;CHECK CC=0100
2932 007100 101401          BLOS   SNM4A
2933 007102 100404          BMI     SNM4B
2934
2935                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2936                          ;          CONDITIONAL BRANCH INST. AND <====
2937                          ;          REPLACE THE MOVE INSTRUCTION <====
2938                          ;          WHICH FOLLOWS W/ 766 <====
2938 007104          SNM4A:
2939 007104 012742 000170      MOV     #170,-(R2)        ;MOVE TO MAILBOX # ***** 170 *****
2940 007110 005242          INC     -(R2)            ;SET MSGTYP TO FATAL ERROR
```



```

2941 007112 000000
2942 007114 005700
2943 007116 001404
2944
2945
2946
2947
2948 007120 012742 000171
2949 007124 005242
2950 007126 000000
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963 007130 005212 000105
2964 007132 022712
2965 007136 001022
2966 007140 005000
2967 007142 005010
2968 007144 005110
2969 007146 105100
2970 007150 005200
2971 007152 000277
2972 007154 000250
2973 007156 005750
2974 007160 102402
2975 007162 101401
2976 007164 100404
2977
2978
2979
2980
2981 007166
2982 007166 012742 000172
2983 007172 005242
2984 007174 000000
2985 007176 005200
2986 007200 105100
2987 007202 001404
2988
2989
2990
2991
2992 007204 012742 000173
2993 007210 005242
2994 007212 000000
2995
2996
    
```

```

SNM4B: HALT ;CC'S NOT CORRECT
        TST R0
        BEQ TS105
; 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 R0 CORRECTLY
; OR SEQUENCE ERROR
    
```

```

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

```

:*****
: 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 R0 ;R0=0
        CLR (R0) ;LOC 0=0
        COM (R0) ;LOC 0=-1
        COMB R0 ;R0=377
        INC R0 ;R0=400
        SCC ;SET CC=0111
        CLN
        TST @-(R0) ;TRY TST W/ MODE 5
        BVS SNM5A ;CHECK CC=1000
        BLGS 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) ;SET MSGTYP TO FATAL ERROR
        HALT ;CC'S NOT SET PROPEPLY
SNM5B: INC R0 ;R0=377
        COMB R0 ;R0=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) ;SET MSGTYP TO FATAL ERROR
        HALT ;MODE 5 DID NOT DEC R0 CORRECTLY
; OR SEQUENCE ERROR
    
```

2997
2998
2999
3000
3001
3002
3003
3004
3005
3006
3007 007214 005212
3008 007216 022712 000106
3009 007222 001021
3010 007224 005000
3011 007226 005010
3012 007230 005110
3013 007232 105100
3014 007234 000277
3015 007236 000250
3016 007240 005760 177401
3017 007244 102402
3018 007246 101401
3019 007250 100404
3020
3021
3022
3023
3024 007252
3025 007252 012742 000174
3026 007256 005242
3027 007260 000000
3028 007262 105100
3029 007264 001404
3030
3031
3032
3033
3034 007266 012742 000175
3035 007272 005242
3036 007274 000000
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050 007276 005212
3051 007300 022712 000107
3052 007304 001021

: 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 R0 ;R0=0
CLR (R0) ;LOC 0=0
COM (R0) ;LOC 0=-1
COMB R0 ;R0=377
SCC ;SET CC=0111
CLN
TST -377(R0) ;TRY TST W/ MODE 6
BVS SNM6A ;CHECK CC=1000
BLOS SNM6A
BMI SNM6B

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

SNM6A: MOV #174,-(R2) ;MOVE TO MAILBOX # ***** 174 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S INCORRECT
SNM6B: COMB R0 ;R0=0
BEQ TS107

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

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

: 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

```
3053 007306 005000 CLR R0 ;R0=0
3054 007310 005010 CLR (R0) ;LOC 0=0
3055 007312 005110 COM (R0) ;LOC 0=-1
3056 007314 105100 COMB R0 ;R0=377
3057 007316 000277 SCC ;CC=0111
3058 007320 000250 CLN
3059 007322 005770 000001 TST @1(R0) ;TRY TST W/ MODE 7
3060 007326 102402 BVS SNM7A ;CHECK CC=1000
3061 007330 101401 BLOS SNM7A
3062 007332 100404 BMI SNM7B
3063 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3064 ; CONDITIONAL BRANCH INST. AND <====
3065 ; REPLACE THE MOVE INSTRUCTION <====
3066 ; WHICH FOLLOWS W/ 764 <====
3067 007334 SNM7A:
3068 007334 012742 000176 MOV #176,-(R2) ;MOVE TO MAILBOX # ***** 176 *****
3069 007340 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3070 007342 000000 HALT ;CC'S NOT CORRECT
3071 007344 105100 SNM7B: COMB R0 ;R0=0
3072 007346 001404 BEQ TS110
3073 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3074 ; CONDITIONAL BRANCH INST. AND <====
3075 ; REPLACE THE MOVE INSTRUCTION <====
3076 ; WHICH FOLLOWS W/ 756 <====
3077 007350 012742 000177 MOV #177,-(R2) ;MOVE TO MAILBOX # ***** 177 *****
3078 007354 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3079 007356 000000 HALT ;TST MODE 7 INCORRECTLY CHANGED R0
3080 ; OR SEQUENCE ERROR
```

```
*****
: 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.
```

```
3091 *****
3092 007360 005212 TS110: INC (R2) ;UPDATE TEST NUMBER
3093 007362 022712 000110 CMP #110,(R2) ;SEQUENCE ERROR?
3094 007366 001006 BNE TS111-10 ;BR TO ERROR HALT ON SEQ ERROR
3095 007370 005000 CLR R0 ;R0=0
3096 007372 005100 COM R0 ;R0=-1
3097 007374 005004 CLR R4 ;R4=0
3098 007376 060004 ADD R0,R4 ;TRY ADD: R4 -1
3099 007400 005204 INC R4 ;R4=0
3100 007402 001404 BEQ TS111
3101 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--
3102 ; CONDITIONAL BRANCH INST. AND <--
3103 ; REPLACE THE MOVE INSTRUCTION <--
3104 ; WHICH FOLLOWS W/ 771 <--
3105 007404 012742 000200 MOV #200,-(R2) ;MOVE TO MAILBOX # ***** 200 *****
3106 007410 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3107 007412 000000 HALT ;ADD INST. FAILED W/ MODE 0
3108 ; OR SEQUENCE ERROR
```

3109
 3110
 3111
 3112
 3113
 3114
 3115
 3116
 3117 007414 005212
 3118 007416 022712 000111
 3119 007422 001006
 3120 007424 005000
 3121 007426 005004
 3122 007430 005100
 3123 007432 010004
 3124 007434 005204
 3125 007436 001404
 3126
 3127
 3128
 3129
 3130 007440 012742 000201
 3131 007444 005242
 3132 007446 000000
 3133
 3134
 3135
 3136
 3137
 3138
 3139
 3140
 3141
 3142 007450 005212
 3143 007452 022712 000112
 3144 007456 001016
 3145 007460 005000
 3146 007462 005004
 3147 007464 005204
 3148 007466 160400
 3149 007470 100003
 3150 007472 001402
 3151 007474 102401
 3152 007476 103404
 3153
 3154
 3155
 3156
 3157 007500
 3158 007500 012742 000202
 3159 007504 005242
 3160 007506 000000
 3161 007510 005200
 3162 007512 001404
 3163
 3164

```

*****
                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.
*****
: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             <= --

```

```
3165  
3166  
3167 007514 012742 000203      MOV    #203,-(R2)      ;  
3168 007520 005242              INC    -(R2)          ;REPLACE THE MOVE INSTRUCTION <===  
3169 007522 000000              HALT                   ;WHICH FOLLOWS W/ 761 <===  
3170                                ;MOVE TO MAILBOX # ***** 203 *****  
3171                                ;SET MSGTYP TO FATAL ERROR  
3172                                ;DATA RESULT OF SUB FAILED  
3173                                ;OR SEQUENCE ERROR
```

```
3174  
3175  
3176  
3177  
3178  
3179  
3180  
3181  
3182  
3183  
3184  
3185  
3186  
3187  
3188  
3189  
3190  
3191  
3192  
3193  
3194  
3195  
3196  
3197  
3198  
3199  
3200  
3201  
3202  
3203  
3204  
3205  
3206  
3207  
3208  
3209  
3210  
3211  
3212  
3213  
3214  
3215  
3216  
3217  
3218  
3219  
3220
```

```
*****  
: 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 <
```

```
3194 007542 012742 000204      MOV    #204,-(R2)      ;MOVE TO MAILBOX # ***** 204 *****  
3195 007546 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR  
3196 007550 000000              HALT                   ;Z-BIT NOT SET  
3197 007552 005200      DOP0A: INC    R0      ;R0=1  
3198 007554 005100      C.M    R0      ;R0=177776  
3199 007556 005104      C.M    R4      ;R4=177777  
3200 007560 040004      BIC   R0,R4      ;TRY BIC: R4=1  
3201 007562 005304      DEC   R4          ;R4=0  
3202 007564 001404      BEQ   DOP0B
```

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

```
3207 007566 012742 000205      MOV    #205,-(R2)      ;MOVE TO MAILBOX # ***** 205 *****  
3208 007572 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR  
3209 007574 000000              HALT                   ;BIC CLEAR RESULT INCORRECT  
3210 007576 050004      DOP0B: BIS   R0,R4      ;TRY BIS: R4=177777  
3211 007600 005204              INC    R4  
3212 007602 005204              INC    R4  
3213 007604 001404      BEQ   DOP0C      ;R4=0
```

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

```
3218 007606 012742 000206      MOV    #206,-(R2)      ;MOVE TO MAILBOX # ***** 206 *****  
3219 007612 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR  
3220 007614 000000              HALT                   ;RESULT OF BIS INCORRECT
```

3221 007616 005000
3222 007620 105100
3223 007622 005004
3224 007624 005104
3225 007626 040004
3226 007630 060004
3227 007632 005204
3228 007634 001404

DOPOC: CLR R0 ;R0=0
COMB R0 ;R0=377
CLR R4 ;R4=0
COM R4 ;R4=177777
BIC R0,R4 ;R4=177400
ADD R0,R4 ;TRY ADD: R4=177777
INC R4 ;R4=0
BEQ DOPOD

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 736

3229
3230
3231
3232
3233 007636 012742 000207
3234 007642 005242
3235 007644 000000
3236 007646 160004
3237 007650 105404
3238 007652 005204
3239 007654 001404

DOPOD: MOV #207,-(R2) ;MOVE TO MAILBOX # ***** 207 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULT OF ADD INCORRECT
SJB R0,R4 ;177401=R4
NEGB R4 ;R4=177777
INC R4 ;RD=0
BEQ TS114

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

3240
3241
3242
3243
3244 007656 012742 000210
3245 007662 005242
3246 007664 000000
3247
3248
3249

DOPOD: MOV #210,-(R2) ;MOVE TO MAILBOX # ***** 210 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULT OF SUB INCORRECT
; OR SEQUENCE ERROR

3250
3251
3252
3253
3254
3255

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

3256
3257
3258 007666 005212
3259 007670 022712 000114
3260 007674 001024
3261 007676 005000
3262 007700 005010
3263 007702 105110
3264 007704 005220
3265 007706 005400
3266 007710 060037 000000
3267 007714 100403
3268 007716 001402
3269 007720 102401
3270 007722 103404

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

3271
3272
3273
3274
3275 007724
3276 007724 012742 000211

DOP03A: MOV #211,-(R2) ;MOVE TO MAILBOX # ***** 211 *****

```
3277 007730 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
3278 007732 000000          HALT                    ;CC'S NOT SET CORRECTLY
3279 007734 105137 000000    DOP03B: COMB          @#0          ;LOC. 0=1
3280 007740 005337 000000    DEC      @#0          ;LOC. 0=0
3281 007744 001404          BEQ      TS115
3282
3283                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3284                          ;          CONDITIONAL BRANCH INST. AND <====
3285                          ;          REPLACE THE MOVE INSTRUCTION <====
3286 007746 012742 000212    MOV      #212,-(R2)    ;MOVE TO MAILBOX # ***** 212 ***** <====
3287 007752 005242          INC      -(R2)
3288 007754 000000          HALT                    ;SET MSGTYP TO FATAL ERROR
3289                          ;DATA RESULT INCORRECT
3290                          ; OR SEQUENCE ERROR
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300 007756 005212 000115    TS115: INC      (R2)          ;UPDATE TEST NUMBER
3301 007760 022712          CMP      #115,(R2)    ;SEQUENCE ERROR?
3302 007764 001042          BNE     TS116-10      ;BR TO ERROR HALT ON SEQ ERROR
3303 007766 005000          CLR     R0           ;R0=0
3304 007770 005004          CLR     R4           ;R4=0
3305 007772 005204          INC     R4           ;R4=1
3306 007774 020400          CMP     R4,R0        ;TRY COMPARE R4 TO R0
3307 007776 003004          BGT     DNM1
3308
3309                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3310                          ;          CONDITIONAL BRANCH INST. AND <====
3311                          ;          REPLACE THE MOVE INSTRUCTION <====
3312                          ;          WHICH FOLLOWS W/ 772 <====
3312 010000 012742 000213    MOV      #213,-(R2)    ;MOVE TO MAILBOX # ***** 213 *****
3313 010004 005242          INC     -(R2)
3314 010006 000000          HALT                    ;SET MSGTYP TO FATAL ERROR
3315 010010 020004          DNM1: CMP     R0,R4    ;CC'S NOT CORRECT FOR CMP
3316 010012 002404          BLT     DNM2          ;TRY COMPARE R0 TO R4
3317
3318                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3319                          ;          CONDITIONAL BRANCH INST. AND <====
3320                          ;          REPLACE THE MOVE INSTRUCTION <====
3321                          ;          WHICH FOLLOWS W/ 764 <====
3321 010014 012742 000214    MOV      #214,-(R2)    ;MOVE TO MAILBOX # ***** 214 *****
3322 010020 005242          INC     -(R2)
3323 010022 000000          HALT                    ;SET MSGTYP TO FATAL ERROR
3324 010024 005200          DNM2: INC     R0       ;CC'S NOT CORRECT FOR CMP
3325 010026 020400          CMP     R4,R0        ;R0=1
3326 010030 001404          BEQ     DNM3          ;TRY COMPARE R4=1 TO R0=1
3327
3328                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3329                          ;          CONDITIONAL BRANCH INST. AND <====
3330                          ;          REPLACE THE MOVE INSTRUCTION <====
3331                          ;          WHICH FOLLOWS W/ 755 <====
3331 010032 012742 000215    MOV      #215,-(R2)    ;MOVE TO MAILBOX # ***** 215 *****
3332 010036 005242          INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
```

```
3333 010040 000000
3334 010042 005000
3335 010044 005100
3336 010046 005004
3337 010050 030004
3338 010052 001404
3339
3340
3341
3342
3343 010054 012742 000216
3344 010060 005242
3345 010062 000000
3346 010064 005304
3347 010066 030004
3348 010070 100404
3349
3350
3351
3352
3353 010072 012742 000216
3354 010076 005242
3355 010100 000000
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365 010102 005212
3366 010104 022712 000116
3367 010110 001022
3368 010112 005000
3369 010114 005010
3370 010116 005110
3371 010120 005200
3372 010122 020037 000000
3373 010126 100403
3374 010130 001402
3375 010132 102401
3376 010134 103404
3377
3378
3379
3380
3381 010136
3382 010136 012742 000220
3383 010142 005242
3384 010144 000000
3385 010146 005300
3386 010150 001002
3387 010152 005210
3388 010154 001404

DNM3:  HALT
      CLR R0
      COM R0
      CLR R4
      BIT R0,R4
      BEQ DNM4
      ;CC'S NOT CORRECT (Z=1) FOR CMP
      ;R0=0
      ;R0=177777
      ;R4=0
      ;TRY BIT R0 TO R4
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
      ; CONDITIONAL BRANCH INST. AND
      ; REPLACE THE MOVE INSTRUCTION
      ; WHICH FOLLOWS W/ 744
      ; MOVE TO MAILBOX # ***** 216 *****
      ; SET MSGTYP TO FATAL ERROR
      ;CC'S NOT CORRECT FOR BIT
      ;R4=177777
      ;TRY BIT AGAIN
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
      ; CONDITIONAL BRANCH INST. AND
      ; REPLACE THE MOVE INSTRUCTION
      ; WHICH FOLLOWS W/ 735
      ; MOVE TO MAILBOX # ***** 217 *****
      ; SET MSGTYP TO FATAL ERROR
      ;CC'S NOT CORRECT FOR BIT
      ; 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.
      ;*****
TS116: INC (R2) ;UPDATE TEST NUMBER
      CMP #116,(R2) ;SEQUENCE ERROR?
      BNE TS117-10 ;BR TO ERROR HALT ON SEQ ERROR
      CLR R0 ;R0=0
      CLR (R0) ;LOC. 0=0
      COM (R0) ;LOC. 0=177777
      INC R0 ;R0=1
      CMP R0,#0 ;TRY CMP MODE 0,3
      BMI DNM03A ;CC=0001
      BEQ DNM03A
      BVS DNM03A
      BCS DNM03B
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
      ; CONDITIONAL BRANCH INST. AND
      ; REPLACE THE MOVE INSTRUCTION
      ; WHICH FOLLOWS W/ 765
      ; MOVE TO MAILBOX # ***** 220 *****
      ; SET MSGTYP TO FATAL ERROR
      ;CC'S NOT SET CORRECTLY
DNM03A: MOV #220,-(R2)
      INC -(R2)
DNM03B: DEC R0
      BNE DNM03C
      INC (R0)
      BEQ TS117
```



```
3389 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
3390 ; CONDITIONAL BRANCH INST. AND <====  
3391 ; REPLACE THE MOVE INSTRUCTION <====  
3392 ; WHICH FOLLOWS W/ 755 <====  
3393 010156 DNM03C:  
3394 010156 012742 000221 MOV #221,-(R2) ;MOVE TO MAILBOX # ***** 221 *****  
3395 010162 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3396 010164 000000 HALT ;DATA INCORRECTLY MODIFIED BY CMP  
3397 ; OR SEQUENCE ERROR  
3398  
3399
```

```
3400 :*****  
3401 : THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS. R0 IS SET TO -1  
3402 : AND LOC 0 TO 1. R4 IS THEN CLEARED AND USED TO POINT TO LOC 0.  
3403 : IN THE ADD MODE 1 INSTRUCTION, LOC 0 IS ADDED TO R0 AND THE  
3404 : RESULTS VERIFIED.  
3405 :*****
```

```
3406 :TEST 117 TEST MODE 1 W/ DOP INST.  
3407 :*****
```

```
3408 TS117: INC (R2) ;UPDATE TEST NUMBER  
3409 010166 005212 000117 CMP #117,(R2) ;SEQUENCE ERROR?  
3410 010170 022712 BNE TS120-10 ;BR TO ERROR HALT ON SEQ ERROR  
3411 010174 001007 CLR R0 ;R0=0  
3412 010176 005000 COM R0 ;R0=177777  
3413 010200 005100 CLR R4 ;R4=0  
3414 010202 005004 CLR (R4) ;LOC 0=0  
3415 010204 005014 INC (R4) ;LOC 0=1  
3416 010206 005214 ADD (R4),R0 ;TRY ADD SOURCE MODE 1  
3417 010210 061400 BEQ TS120
```

```
3418 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-- -  
3419 ; CONDITIONAL BRANCH INST. AND <---=  
3420 ; REPLACE THE MOVE INSTRUCTION <=====  
3421 ; WHICH FOLLOWS W/ 770 <=====  
3422
```

```
3423 010214 012742 000222 MOV #222,-(R2) ;MOVE TO MAILBOX # ***** 222 *****  
3424 010220 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
3425 010222 000000 HALT ;RESULT OF ADD INCORRECT  
3426 ; OR SEQUENCE ERROR  
3427  
3428
```

```
3429 :*****  
3430 : THIS TEST VERIFIES MODE 1 DOP BYTE INSTRUCTIONS WHICH ADDRESS  
3431 : EVEN BYTES. LOC. 0 IS SET TO -1 AND R4 IS CLEARED. THEN R4 IS  
3432 : SET TO -1 USING A BISB THRU R0 WITH MODE 1.  
3433 :*****
```

```
3434 :TEST 120 TEST MODE 1 - EVEN BYTE W/ DOP INSTS.  
3435 :*****
```

```
3436 TS120: INC (R2) ;UPDATE TEST NUMBER  
3437 010224 005212 000120 CMP #120,(R2) ;SEQUENCE ERROR?  
3438 010226 022712 BNE TS121-10 ;BR TO ERROR HALT ON SEQ ERROR  
3439 010232 001007 CLR R0 ;R0=0  
3440 010234 005000 CLR (R0) ;LOC. 0=0  
3441 010236 005010 COM (R0) ;LOC. 0=177777  
3442 010240 005110 CLR R4 ;R4=0  
3443 010242 005004
```

```

3445 010244 151004      BISB   (R0),R4      ;TRY MODE 1- EVEN BYTE W/ DOP
3446 010246 105104      COMB   R4           ;R4=0
3447 010250 001404      BEQ    TS121
3448
3449
3450
3451
3452 010252 012742 000223  MOV    #223,-(R2)   ;MOVE TO MAILBOX # ***** 223 *****
3453 010256 005242      INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
3454 010260 000000      HALT
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468 010262 005212 000121  TS121: INC    (R2)           ;UPDATE TEST NUMBER
3469 010264 022712      CMP    #121,(R2)   ;SEQUENCE ERROR?
3470 010270 001007      BNE   TS122-10    ;BR TO ERROR HALT ON SEQ ERROR
3471 010272 005000      CLR   R0          ;R0=0
3472 010274 005010      CLR   (R0)        ;LOC 0=0
3473 010276 005110      COM   (R0)        ;LOC 0=177777
3474 010300 005004      CLR   R4          ;R4=0
3475 010302 105104      COMB  R4          ;R4=377
3476 010304 121004      CMPB  (R0),R4     ;TRY MODE 1 - EVEN BYTE W/ DOP NON-MODIFYING
3477 010306 001404      BEQ   TS122
3478
3479
3480
3481
3482 010310 012742 000224  MOV    #224,-(R2)   ;MOVE TO MAILBOX # ***** 224 *****
3483 010314 005242      INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
3484 010316 000000      HALT
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
    
```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 77C <====
    
```

```

*****
: 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
    
```

```

*****
: THIS TEST VERIFIES MODE 1,0 MOVVB INSTRUCTIONS
: WHICH ADDRESS EVEN BYTES. LOC. 0 IS SET TO 177400, R0 IS CLEARED AND
: R4 IS SET TO -1. MOVVB 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
*****
    
```

```
3501
3502 010320 005212
3503 010322 022712 000122
3504 010326 001020
3505 010330 005000
3506 010332 005010
3507 010334 105110
3508 010336 005110
3509 010340 005004
3510 010342 005104
3511 010344 111004
3512 010346 005704
3513 010350 001404
3514
3515
3516
3517
3518 010352 012742 000225
3519 010356 005242
3520 010360 000000
3521 010362 005110
3522 010364 111004
3523 010366 100404
3524
3525
3526
3527
3528 010370 012742 000226
3529 010374 005242
3530 010376 000000
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544 010400 005212
3545 010402 022712 000123
3546 010406 001010
3547 010410 005000
3548 010412 005010
3549 010414 005004
3550 010416 005204
3551 010420 105114
3552 010422 151410

:*****
TS122: INC (R2) ;UPDATE TEST NUMBER
      CMP #122,(R2) ;SEQUENCE ERROR?
      BNE TS123-10 ;BR TO ERROR HALT ON SEQ ERROR
      CLR R0 ;R0=0
      CLR (R0) ;LOC 0=0
      COMB (R0) ;LOC 0=177400
      COM (R0)
      CLR R4 ;R4=0
      COM R4 ;R4=177777
      MOVB (R0),R4 ;R4=0
      TST R4 ;CHECK SIGN OF WORD
      BEQ DOP1
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 766 <====
      MOV #225,-(R2) ;MOVE TO MAILBOX # ***** 225 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;MOVB SHOULD SIGN X-TEND
DOP1: COM (R0) ;LOC 0=177777
      MOVB (R0),R4 ;DO MOVB W/ EVEN BYTE
      BMI TS123
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 757 <====
      MOV #226,-(R2) ;MOVE TO MAILBOX # ***** 226 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;MOVB SHOULD SIGN X-TEND
      ; OR SEQUENCE ERROR

:*****
: 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
```

```

3553 010424 005210      INC      (R0)      ;CHECK RESULT
3554 010426 001404      BEQ      TS124
3555                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3556                                     ;          CONDITIONAL BRANCH INST. AND <====
3557                                     ;          REPLACE THE MOVE INSTRUCTION <= --
3558                                     ;          WHICH FOLLOWS W/ 767 <====
3559 010430 012742 000227  MOV      #227,-(R2) ;MOVE TO MAILBOX # ***** 227 *****
3560 010434 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3561 010436 000000      HALT           ;RESULT OF BISB INCORRECT
3562                                     ; OR SEQUENCE ERROR
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574

```

```

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

```

```

3573 :TEST 124 TEST MODE 2 W/ DOP INSTS.
3574 :*****
3575 010440 005212      TS124: INC      (R2)      ;UPDATE TEST NUMBER
3576 010442 022712 000124  CMP      #124,(R2) ;SEQUENCE ERROR?
3577 010446 001015      BNE      TS125-10 ;BR TO ERROR HALT ON SEQ ERROR
3578 010450 005000      CLR      R0        ;R0=0
3579 010452 005010      CLR      (R0)      ;LOC. 0=0
3580 010454 005110      COM      (R0)      ;LOC. 0=177777
3581 010456 012004      MOV      (R0)+,R4  ;TRY MOVE MODE 2,0
3582 010460 005204      INC      R4        ;CHECK R4
3583 010462 001404      BEQ      DOP2
3584
3585
3586
3587

```

```

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

```

```

3588 010464 012742 000230  MOV      #230,-(R2) ;MOVE TO MAILBOX # ***** 230 *****
3589 010470 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3590 010472 000000      HALT           ;RESULT OF MOV INST INCORRECT
3591 010474 005300      DOP2: DEC      R0        ;TEST R0 AFTER MODE 2
3592 010476 005300      DEC      R0
3593 010500 001404      BEQ      TS125
3594
3595
3596
3597

```

```

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

```

```

3598 010502 012742 000231  MOV      #231,-(R2) ;MOVE TO MAILBOX # ***** 231 *****
3599 010506 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
3600 010510 000000      HALT           ;REGISTER NOT INCREMENTED IN MODE 2
3601                                     ; OR SEQUENCE ERROR
3602
3603
3604
3605
3606
3607
3608

```

```

:*****
: 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
:*****

```

```

3609
3610
3611
3612
3613
3614
3615
3616 010512 005212
3617 010514 022712 000125
3618 010520 001016
3619 010522 005000
3620 010524 010010
3621 010526 005110
3622 010530 142010
3623 010532 105737 000001
3624 010536 001404
3625
3626
3627
3628
3629 010540 012742 000232
3630 010544 005242
3631 010546 000000
3632 010550 105137 000000
3633 010554 001404
3634
3635
3636
3637
3638 010556 012742 000233
3639 010562 005242
3640 010564 000000
3641
3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653 010566 005212
3654 010570 022712 000126
3655 010574 001017
3656 010576 005000
3657 010600 005004
3658 010602 005010
3659 010604 005110
3660 010606 105120
3661 010610 112004
3662 010612 005204
3663 010614 001404
3664

```

```

;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 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;BICB SOURCE INCORRECTLY CHANGED
; OR SEQUENCE ERROR

```

```

*****
; 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 MOVSB USES R0 TO MOVE BYTE 1 TO R4. AN INCREMENT
; IS USED TO CHECK THAT THE PROPER BYTE WAS MOVED AND SIGN X-TENDED.

```

```

*****
;TEST 126 TEST MODE 2 - ODD BYTE W/ DOP INST.
*****

```

```

TS126: INC (R2) ;UPDATE TEST NUMBER
CMP #126,(R2) ;SEQUENCE ERROR?
BNE TS127-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR R4 ;R4=0
CLR (R0) ;LOC. 0=0
COM (R0) ;LOC. 0=177777
COMB (R0)+ ;LOC 0=177400; R0=1
MOVSB (R0)+,R4 ;TRY DOP MODE 2 W/ ODD BYTE
INC R4 ;CHECK RESULT OF MOVSB
BEQ DOPB2B

```

```

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

```

```

3665
3666
3667
3668 010616 012742 000234      MOV    #234,-(R2)      ; MOVE TO MAILBOX # ***** 234 *****
3669 010622 005242              INC    -(R2)          ; SET MSGTYP TO FATAL ERROR
3670 010624 000000              HALT                    ; RESULT OF MOV B INCORRECT
3671 010626 005740      DOPB2B: TST    -(R0)    ; BUMP R0 DOWN BY 2
3672 010630 005700              TST    R0              ; CHECK R0
3673 010632 001404              BEQ    TS127
    
```

```

3674
3675
3676
3677
3678 010634 012742 000235      MOV    #235,-(R2)    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
3679 010640 005242              INC    -(R2)          ; CONDITIONAL BRANCH INST. AND
3680 010642 000000              HALT                    ; REPLACE THE MOVE INSTRUCTION
    ; WHICH FOLLOWS W/ 760
    ; MOVE TO MAILBOX # ***** 235 *****
    ; SET MSGTYP TO FATAL ERROR
    ; MODE 2 BYTE DID NOT INCREMENT REG. CORRECTLY
    ; OR SEQUENCE ERROR
    
```

```

3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
    
```

```

:*****
: 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.
:*****
    
```

```

3706 010644 005212              TS127: INC    (R2)          ; UPDATE TEST NUMBER
3707 010646 022712 000127      CMP    #127,(R2)      ; SEQUENCE ERROR?
3708 010652 001011              BNE    TS130-10       ; BR TO ERROR HALT ON SEQ ERROR
3709 010654 012737 052525 000000  MOV    #052525,a#0    ; MOVE 52525 TO LOC. 0
3710 010662 012700 125252              MOV    #125252,R0     ; SET ALT. ONE AND ZERO IN R0
3711 010666 053700 000000              BIS    a#0,R0         ; TRY TO SET ALL OTHER BITS W/ MODE 3
3712 010672 005200              INC    R0              ; TEST RESULT
3713 010674 001404              BEQ    TS130
    
```

```

3714
3715
3716
3717
3718
3719
3720 010676 012742 000236      MOV    #236,-(R2)    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
3721 010702 005242              INC    -(R2)          ; CONDITIONAL BRANCH INST. AND
3722 010704 000000              HALT                    ; REPLACE THE MOVE INSTRUCTION
    ; WHICH FOLLOWS W/ 766
    ; MOVE TO MAILBOX # ***** 236 *****
    ; SET MSGTYP TO FATAL ERROR
    ; BIS W/ MODE 3 INCORRECT RESULT
    ; OR SEQUENCE ERROR
    
```

```

3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810
3811
3812
3813
3814
3815
3816
3817
3818
3819
3820
3821
3822
3823
3824
3825
3826
3827
3828
3829
3830
3831
3832
3833
3834
3835
3836
3837
3838
3839
3840
3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869
3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889
3890
3891
3892
3893
3894
3895
3896
3897
3898
3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928
3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958
3959
3960
3961
3962
3963
3964
3965
3966
3967
3968
3969
3970
3971
3972
3973
3974
3975
3976
3977
3978
3979
3980
3981
3982
3983
3984
3985
3986
3987
3988
3989
3990
3991
3992
3993
3994
3995
3996
3997
3998
3999
4000
    
```

```

:*****
: 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.
:*****
    
```

```

3999
4000
    
```

```

3721 010710 022712 000130      CMP      #130,(R2)      ;SEQUENCE ERROR?
3722 010714 001011              BNE      TS131-10      ;BR TO ERROR HALT ON SEQ ERROR
3723 010716 012737 052652 000000  MOV      #52652,@#0    ;MOVE 1'S AND 0' PATTERN TO LOC. 0
3724 010724 005000              CLR      R0            ;R0=0
3725 010726 153700 000000      BISB     @#0,R0        ;TRY R0 252 W/ MODE 3 - EVEN BYTE
3726 010732 022700 000252      CMP      #252,R0      ;BISB W/ EVEN BYTE SUCCESSFUL?
3727 010736 001404              BEQ      TS131
3728
3729                          ; TO SCOPE: CLEAR 'F' RIGHT BYTE OF THIS <====
3730                          ;          CONDITIONAL BRANCH INST. AND <====
3731                          ;          REPLACE THE MOVE INSTRUCTION <====
3732 010740 012742 000237      MOV      #237,-(R2)    ;MOVE TO MAILBOX # ***** 237 *****
3733 010744 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
3734 010746 000000              HALT                ;BISB W/ MODE 3 - EVEN BYTE FAILED
3735                          ; OR SEQUENCE ERROR
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759 011002 012742 000240      MOV      #240,-(R2)    ;MOVE TO MAILBOX # ***** 240 *****
3760 011006 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
3761 011010 000000              HALT                ;BISB W/ MODE 3 - ODD BYTE FAILED
3762                          ; OR SEQUENCE ERROR
3763
3764
3765
3766
3767
3768
3769
3770
3771
3772
3773
3774
3775
3776

```

```

:*****
:
:      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.

```

```
3777 011042 100404      BMI      DNMB0B
3778
3779                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3780                      ;          CONDITIONAL BRANCH INST. AND <====
3781                      ;          REPLACE THE MOVE INSTRUCTION <====
3782                      ;          WHICH FOLLOWS W/ 766 <====
3782 011044
3783 011044 012742 000241 DNMB0A:  MOV      #241,-(R2) ;MOVE TO MAILBOX # ***** 241 *****
3784 011050 005242      INC      -(R2) ;SET MSGTYP TO FATAL ERROR
3785 011052 000000      HALT     ;CC'S INCORRECT
3786 011054 105100 DNMB0B: COMB   R0 ;CHECK DESTINATION DATA
3787 011056 001404      BEQ     TS133
3788
3789                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3790                      ;          CONDITIONAL BRANCH INST. AND <====
3791                      ;          REPLACE THE MOVE INSTRUCTION <====
3792                      ;          WHICH FOLLOWS W/ 760 <====
3792 011060 012742 000242      MOV      #242,-(R2) ;MOVE TO MAILBOX # ***** 242 *****
3793 011064 005242      INC      -(R2) ;SET MSGTYP TO FATAL ERROR
3794 011066 000000      HALT     ;DEST. DATA MODIFIED
3795
3796                      ; OR SEQUENCE ERROR
3797
3798 *****
3799 ;TEST 133 TEST DEST. MODE 1 W/ DOP NON-MODIFYING INST
3800 *****
3800 011070 005212 TS133:  INC      (R2) ;UPDATE TEST NUMBER
3801 011072 022712 000133      CMP      #133,(R2) ;SEQUENCE ERROR?
3802 011076 001017      BNE     TS134-10 ;BR TO ERROR HALT ON SEQ ERROR
3803 011100 005000      CLR     R0 ;R0=0
3804 011102 005010      CLR     (R0) ;LOC. 0=0
3805 011104 000241      CLC     ;CLEAR C BIT
3806 011106 032710 177777      BIT     #177777,(R0) ;TRY DOPNM DEST. MODE 1
3807 011112 100403      BMI     DNM1A ;BR TO ERROR IF N BIT SET
3808 011114 102402      BVS     DNM1A ;BR TO ERROR IF V BIT SET
3809 011116 103401      BCS     DNM1A ;BR TO ERROR IF C BIT SET
3810 011120 001404      BEQ     DNM1B
3811
3812                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <= --
3813                      ;          CONDITIONAL BRANCH INST. AND <= --
3814                      ;          REPLACE THE MOVE INSTRUCTION <= --
3815                      ;          WHICH FOLLOWS W/ 766 <= --
3815 011122
3816 011122 742 000243 DNMB1A:  MOV      #243,-(R2) ;MOVE TO MAILBOX # ***** 243 *****
3817 011126 005242      INC      -(R2) ;SET MSGTYP TO FATAL ERROR
3818 011130 000000      HALT     ;COND. CODES INCORRECT
3819 011132 005710 DNMB1B:  TST     (R0) ;CHECK TEST DATA
3820 011134 001404      BEQ     TS134
3821
3822                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- --
3823                      ;          CONDITIONAL BRANCH INST. AND <- --
3824                      ;          REPLACE THE MOVE INSTRUCTION <- --
3825                      ;          WHICH FOLLOWS W/ 760 <- --
3825 011136 012742 000244      MOV      #244,-(R2) ;MOVE TO MAILBOX # ***** 244 *****
3826 011142 005242      INC      -(R2) ;SET MSGTYP TO FATAL ERROR
3827 011144 000000      HALT     ;DESTINATION DATA MODIFIED
3828
3829                      ; OR SEQUENCE ERROR
3830 *****
3831 ;TEST 134 TEST DEST, MODE 2 W/ DOP NON-MODIFYING INST.
3832 *****
```



```
3833 011146 005212 TS134: INC (R2) ;UPDATE TEST NUMBER
3834 011150 022712 000134 CMP #134,(R2) ;SEQUENCE ERROR?
3835 011154 001027 BNE TS135-10 ;BR TO ERROR HALT ON SEQ ERROR
3836 011156 005000 CLR R0 ;R0=0
3837 011160 005010 CLR (R0) ;LOC. 0=0
3838 011162 052710 125252 BIS #125252,(R0) ;LOC. 0=125252
3839 011166 032720 077777 BIT #77777,(R0)+ ;TRY DOPNM INST W/ MODE 2
3840 011172 102402 BVS DNM2A ;BR TO ERROR IF V BIT SET
3841 011174 001401 BEQ DNM2A ;BR TO ERROR IF Z-BIT SET
3842 011176 100004 BPL DNM2B
3843 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3844 ; CONDITIONAL BRANCH INST. AND <====
3845 ; REPLACE THE MOVE INSTRUCTION <====
3846 ; WHICH FOLLOWS W/ 766 <====
3847 011200 DNM2A:
3848 011200 012742 000245 MOV #245,-(R2) ;MOVE TO MAILBOX # ***** 245 *****
3849 011204 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3850 011206 000000 HALT ;COND. CODES INCORRECT
3851 011210 005300 DNM2B: DEC R0 ;DECREMENT R0 TO CHECK IT.
3852 011212 005300 DEC R0
3853 011214 001404 BEQ DNM2D
3854 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3855 ; CONDITIONAL BRANCH INST. AND <====
3856 ; REPLACE THE MOVE INSTRUCTION <====
3857 ; WHICH FOLLOWS W/ 757 <====
3858 011216 DNM2C:
3859 011216 012742 000246 MOV #246,-(R2) ;MOVE TO MAILBOX # ***** 246 *****
3860 011222 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3861 011224 000000 HALT ;MODF 2 REGISTER NOT INCREMENTED BY 2
3862 011226 022710 125252 DNM2D: CMP #125252,(R0) ;CHECK DEST. DATA
3863 011232 001404 BEQ TS135
3864 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3865 ; CONDITIONAL BRANCH INST. AND <====
3866 ; REPLACE THE MOVE INSTRUCTION <====
3867 ; WHICH FOLLOWS W/ 750 <====
3868 011234 012742 000247 MOV #247,-(R2) ;MOVE TO MAILBOX # ***** 247 *****
3869 011240 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
3870 011242 000000 HALT ;DEST. DATA MODIFIED
3871 ; OR SEQUENCE ERROR
3872
3873 ;*****
3874 ;TEST 135 TEST DEST. MODE 2-BYTE, W/DOP NON-MODIFYING INST
3875 ;*****
3876 011244 005212 TS135: INC (R2) ;UPDATE TEST NUMBER
3877 011246 022712 000135 CMP #135,(R2) ;SEQUENCE ERROR?
3878 011252 001051 BNE TS136-10 ;BR TO ERROR HALT ON SEQ ERROR
3879 011254 005000 CLR R0 ;R0=0
3880 011256 005010 CLR (R0) ;LOC. 0=0
3881 011260 052710 052652 BIS #52652,(R0) ;LOC. 0=52652
3882 011264 000263 +SEC!SEV ;SET C AND V BITS
3883 011266 132720 000201 BITB #201,(R0)+ ;TRY DOPNM INST. W/ MODE 2 EVEN BYTE
3884 011272 001403 BEQ DNM2A ;BR TO ERROR IF Z-BIT SET
3885 011274 103002 BCC DNM2A ;BR TO ERROR IF C-BIT CLEAR
3886 011276 102401 BVS DNM2A ;BR TO ERROR IF V-BIT SET
3887 011300 100404 BMI DNM2B
3888 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < - -
```

```
3889
3890
3891
3892 011302
3893 011302 012742 000250
3894 011306 005242
3895 011310 000000
3896 011312 005300
3897 011314 001404
3898
3899
3900
3901
3902 011316 012742 000251
3903 011322 005242
3904 011324 000000
3905 011326 005200
3906 011330 132720 000201
3907 011334 001402
3908 011336 102401
3909 011340 100004
3910
3911
3912
3913
3914 011342
3915 011342 012742 000252
3916 011346 005242
3917 011350 000000
3918 011352 005300
3919 011354 005300
3920 011356 001404
3921
3922
3923
3924
3925 011360 012742 000253
3926 011364 005242
3927 011366 000000
3928 011370 022710 052652
3929 011374 001404
3930
3931
3932
3933
3934 011376 012742 000254
3935 011402 005242
3936 011404 000000
3937
3938
3939
3940
3941
3942
3943 011406 005212
3944 011410 022712 000136
```

DNMB2A: MOV #250,-(R2) ;MOVE TO MAILBOX # ***** 250 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;COND. CODES INCORRECT
DNMB2B: DEC R0 ;CHECK DEST. REGISTER.
BEQ DNMB2C
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====
; CONDITIONAL BRANCH INST. AND <=====
; REPLACE THE MOVE INSTRUCTION <=====
; WHICH FOLLOWS W/ 764 <=====
DNMB2C: MOV #251,-(R2) ;MOVE TO MAILBOX # ***** 251 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DEST. REGISTER NOT INCREMENTED BY 1
DNMB2C: INC R0 ;R0=1
BITB #201,(R0)+ ;TRY DOPNM INST. W/MODE 2-ODD BYTE
BEQ DNMB2D ;BR TO ERROR IF Z-BIT SET
BVS DNMB2D ;BR TO ERROR IF V-BIT SET
BPL DNMB2E
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====
; CONDITIONAL BRANCH INST. AND <=====
; REPLACE THE MOVE INSTRUCTION <=====
; WHICH FOLLOWS W/ 744 <=====
DNMB2D: MOV #252,-(R2) ;MOVE TO MAILBOX # ***** 252 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;COND. CODES INCORRECT
DNMB2E: DEC R0 ;DEC R0 TO CHECK IT.
DEC R0
BEQ DNMB2F
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====
; CONDITIONAL BRANCH INST. AND <=====
; REPLACE THE MOVE INSTRUCTION <=====
; WHICH FOLLOWS W/ 735 <=====
DNMB2F: 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?


```
4001
4002
4003
4004 011546 005212
4005 011550 022712 000137
4006 011554 001033
4007 011556 005000
4008 011560 005010
4009 011562 052710 125252
4010 011566 052700 000002
4011 011572 000277
4012 011574 032740 020000
4013 011600 100403
4014 011602 102402
4015 011604 103001
4016 011606 001004
4017
4018
4019
4020
4021 011610
4022 011610 012742 000261
4023 011614 005242
4024 011616 000000
4025 011620 005700
4026 011622 001404
4027
4028
4029
4030
4031 011624 012742 000262
4032 011630 005242
4033 011632 000000
4034 011634 022737 125252 000000
4035 011642 001404
4036
4037
4038
4039
4040 011644 012742 000263
4041 011650 005242
4042 011652 000000
4043
4044
4045
4046
4047
4048 011654 005212
4049 011656 022712 000140
4050 011662 001051
4051 011664 005000
4052 011666 005010
4053 011670 052710 052652
4054 011674 052700 000002
4055 011700 000257
4056 011702 132740 000201
```

```
*****
;TEST 137 TEST DEST. MODE 4 W/DOP NON-MODIFYING INST.
*****
TS137: INC (R2) ;UPDATE TEST NUMBER
CMP #137,(R2) ;SEQUENCE ERROR?
BNE TS140-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC. 0=0
BIS #125252,(R0) ;LOC. 0=125125
BIS #2,R0 ;R0=2
SCC ;SET ALL COND. CODE BITS
BIT #20000,-(R0) ;TRY DOPNM W/ MODE 4
BMI DNM4A ;BR TO ERROR IF N-BIT SET
BVS DNM4A ;BR TO ERROR IF V-BIT SET
BCC DNM4A ;BR TO ERROR IF C-BIT CHAR
BNE DNM4B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 762 <====
DNM4A: MOV #261,-(R2) ;MOVE TO MAILBOX # ***** 261 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;COND. CODES INCORRECT
DNM4B: TST R0 ;CHECK DEST. REGISTER
BEQ DNM4C
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 754 <====
MOV #262,-(R2) ;MOVE TO MAILBOX # ***** 262 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DEST. REGISTER NOT DECREMENTED BY 2
DNM4C: CMP #125252,#0 ;CHECK DEST. DATA
BEQ TS140
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 744 <====
MOV #263,-(R2) ;MOVE TO MAILBOX # ***** 263 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DEST. DATA MODIFIED
; OR SEQUENCE ERROR
*****
;TEST 140 TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
*****
TS140: INC (R2) ;UPDATE TEST NUMBER
CMP #140,(R2) ;SEQUENCE ERROR?
BNE TS141-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC. 0=0
BIS #52652,(R0) ;LOC. 0=52652
BIS #2,R0 ;R0=2
CCC ;COND. CODES=0
BITB #201,-(R0) ;TRY DOPNM INST W/MODE 4 ODD BYTE
```

```
4057 011706 102403          BVS      DNMB4A          ;BR TO ERROR IF V BIT SFT
4058 011710 001402          BEQ      DNMB4A          ;BR TO ERROR IF Z BIT SET
4059 011712 103401          BCS      DNMB4A          ;BR TO ERROR IF C BIT SET
4060 011714 001004          BNE      DNMB4B
4061
4062                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4063                          ;          CONDITIONAL BRANCH INST. AND <====
4064                          ;          REPLACE THE MOVE INSTRUCTION <====
4065                          ;          WHICH FOLLOWS W/ 762 <====
4065 011716          DNMB4A:
4066 011716 012742 000264      MOV      #264,-(R2)      ;MOVE TO MAILBOX # ***** 264 *****
4067 011722 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4068 011724 000000          HALT
4069 011726 022700 000001      DNMB4B: CMP      #1,R0      ;COND. CODES INCORRECT
4070 011732 001404          BEQ      DNMB4C          ;CHECK DEST. REGISTER
4071
4072                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4073                          ;          CONDITIONAL BRANCH INST. AND <====
4074                          ;          REPLACE THE MOVE INSTRUCTION <====
4075                          ;          WHICH FOLLOWS W/ 753 <====
4075 011734 012742 000265      MOV      #265,-(R2)      ;MOVE TO MAILBOX # ***** 265 *****
4076 011740 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4077 011742 000000          HALT
4078 011744 132740 000201      DNMB4C: BITB     #201,-(R0) ;DEST REG. NOT DECREMENTED BY 1
4079 011750 001401          BEQ      DNMB4D          ;TRY DOPNM INST. W/MODE 4 EVEN BYTE
4080 011752 100404          BMI      DNMB4E          ;BR TO ERROR IF Z-BIT SET
4081
4082                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4083                          ;          CONDITIONAL BRANCH INST. AND <====
4084                          ;          REPLACE THE MOVE INSTRUCTION <====
4085                          ;          WHICH FOLLOWS W/ 743 <====
4085 011754          DNMB4D:
4086 011754 012742 000266      MOV      #266,-(R2)      ;MOVE TO MAILBOX # ***** 266 *****
4087 011760 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4088 011762 000000          HALT
4089 011764 005700          DNMB4E: TST      R0      ;COND. CODES INCORRECT
4090 011766 001404          BEQ      DNMB4F          ;CHECK DEST. REGISTER
4091
4092                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4093                          ;          CONDITIONAL BRANCH INST. AND <====
4094                          ;          REPLACE THE MOVE INSTRUCTION <====
4095                          ;          WHICH FOLLOWS W/ 735 <====
4095 011770 012742 000267      MOV      #267,-(R2)      ;MOVE TO MAILBOX # ***** 267 *****
4096 011774 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4097 011776 000000          HALT
4098 012000 022710 052652      DNMB4F: CMP      #52652,(R0) ;DEST. REG. NOT DECREMENTED BY 1
4099 012004 001404          BEQ      TS141          ;CHECK DESTINATION DATA
4100
4101                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4102                          ;          CONDITIONAL BRANCH INST. AND <====
4103                          ;          REPLACE THE MOVE INSTRUCTION <====
4104                          ;          WHICH FOLLOWS W/ 726 <====
4104 012006 012742 000270      MOV      #270,-(R2)      ;MOVE TO MAILBOX # ***** 270 *****
4105 012012 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4106 012014 000000          HALT
4107                          ; DEST. DATA MODIFIED
4108                          ; OR SEQUENCE ERROR
4109
4110          ;*****
4110          ;TEST 141          TEST DEST MODE 5 W/DOP NON-MODIFYING INST.
4111          ;*****
4112 012016 005212          TS141: INC      (R2)          ;UPDATE TEST NUMBER
```

```
4113 012020 022712 000141      CMP      #141,(R2)      ;SEQUENCE ERROR?
4114 012024 001034      BNE      TS142-10      ;BR TO ERROR HALT ON SEQ ERROR
4115 012026 005000      CLR      R0           ;R0=0
4116 012030 005010      CLR      (R0)         ;LOC 0=0
4117 012032 052710 100000      BIS      #100000,(R0) ;LOC. 0=100000
4118 012036 052700 000402      BIS      #402,R0      ;R0=2
4119 012042 000277      SCC      ;SET ALL COND. CODE BITS
4120 012044 032750 100000      BIT      #100000,@-(R0);TRY DOPNM W/MODE 5
4121 012050 102403      BVS      DNM5A        ;BR TO ERROR IF V-BIT SET
4122 012052 103002      BCC      DNM5A        ;BR TO ERROR IF C-BIT CLEAR
4123 012054 001401      BEQ      DNM5A        ;BR TO ERROR IF Z-BIT SET
4124 012056 100404      BMI      DNM5B
4125
4126      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4127      ;          CONDITIONAL BRANCH INST. AND <====
4128      ;          REPLACE THE MOVE INSTRUCTION <====
4129      ;          WHICH FOLLOWS W/ 762 <====
4129 012060      DNM5A:
4130 012060 012742 000271      MOV      #271,-(R2)   ;MOVE TO MAILBOX # ***** 271 *****
4131 012064 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4132 012066 000000      HALT
4133 012070 022700 000400      DNM5B: CMP      #400,R0 ;COND. CODES INCORRECT
4134 012074 001404      BEQ      DNM5C        ;CHECK DEST. REGISTER
4135
4136      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4137      ;          CONDITIONAL BRANCH INST. AND <====
4138      ;          REPLACE THE MOVE INSTRUCTION <====
4139      ;          WHICH FOLLOWS W/ 753 <====
4139 012076 012742 000272      MOV      #272,-(R2)   ;MOVE TO MAILBOX # ***** 272 *****
4140 012102 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4141 012104 000000      HALT
4142 012106 022737 100000 000000 DNM5C: CMP      #100000,@#0 ;DEST. REGISTER NOT DECREMENTED BY 2
4143 012114 001404      BEQ      TS142        ;CHECK DESTINATION DATA
4144
4145      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4146      ;          CONDITIONAL BRANCH INST. AND <====
4147      ;          REPLACE THE MOVE INSTRUCTION <====
4148      ;          WHICH FOLLOWS W/ 743 <====
4148 012116 012742 000273      MOV      #273,-(R2)   ;MOVE TO MAILBOX # ***** 273 *****
4149 012122 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4150 012124 000000      HALT
4151      ;DEST. DATA INCORRECTLY MODIFIED
4152      ; OR SEQUENCE ERROR
4153
4154      ;*****
4155      ;TEST 142 TEST DEST. MODE 6 W/DOP NON-MODIFYING INST.
4156      ;*****
4156 012126 005212      TS142: INC      (R2)      ;UPDATE TEST NUMBER
4157 012130 022712 000142      CMP      #142,(R2)   ;SEQUENCE ERROR?
4158 012134 001033      BNE      TS143-10      ;BR TO ERROR HALT ON SEQ ERROR
4159 012136 005000      CLR      R0           ;R0=0
4160 012140 005010      CLR      (R0)         ;LOC> 0=0
4161 012142 052710 000001      BIS      #1,(R0)      ;LOC. 0=1
4162 012146 005100      COM      R0           ;R0=-1 C-BIT=1
4163 012150 032760 000001 000001      BIT      #1,1(R0)    ;TRY DOPNM W/MODE 6
4164 012156 001403      BEQ      DNM6A        ;BR TO ERROR IF Z-BIT SET
4165 012160 102402      BVS      DNM6A        ;BR TO ERROR IF V-BIT SET
4166 012162 103001      BCC      DNM6A        ;BR TO ERROR IF C-BIT CLEAR
4167 012164 100004      BPL      DNM6B
4168      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
```

```
4169                                     :                                     <====
4170                                     :                                     <====
4171                                     :                                     <====
4172 012166                               DNM6A: MOV #274,-(R2) ;MOVE TO MAILBOX # ***** 274 *****
4173 012166 012742 000274                 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4174 012172 005242                         HALT ;COND CODES INCORRECT
4175 012174 000000                         BEQ DNM6C ;CHECK DEST. REGISTER
4176 012176 022700 177777                 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4177 012202 001404                         :                                     <====
4178                                     :                                     <====
4179                                     :                                     <====
4180                                     :                                     <====
4181                                     :                                     <====
4182 012204 012742 000275                 MOV #275,-(R2) ;MOVE TO MAILBOX # ***** 275 *****
4183 012210 005242                         INC -(R2) ;SET MSGTYP TO FATAL ERROR
4184 012212 000000                         HALT ;DEST. REGISTER MODIFIED
4185 012214 022737 000001 000000 DNM6C: CMP #1,@#0 ;CHECK DEST. DATA
4186 012222 001404                         BEQ TS143 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4187                                     :                                     <====
4188                                     :                                     <====
4189                                     :                                     <====
4190                                     :                                     <====
4191 012224 012742 000276                 MOV #276,-(R2) ;MOVE TO MAILBOX # ***** 276 *****
4192 012230 005242                         INC -(R2) ;SET MSGTYP TO FATAL ERROR
4193 012232 000000                         HALT ;DEST. DATA MODIFIED
4194                                     : OR SEQUENCE ERROR
4195
4196
4197 :*****
4198 :TEST 143 TEST DEST MODE 7 W/DOP NON-MODIFYING INST.
4199 :*****
4200 012234 005212 TS143: INC (R2) ;UPDATE TEST NUMBER
4201 012236 022712 000143                 CMP #143,(R2) ;SEQUENCE ERROR?
4202 012242 001034                         BNE TS144-10 ;BR TO ERROR HALT ON SEQ ERROR
4203 012244 005000                         CLR R0 ;R0=0
4204 012246 005010                         CLR (R0) ;LOC. 0=0 C-BIT=0
4205 012250 052710 125125                 BIS #125125,(R0) ;LOC. 0=125125
4206 012254 052700 000001                 BIS #1,R0 ;R0=1
4207 012260 132770 000125 000403         BITB #125,@403(R0) ;TRY DOPNM W/MODE 7
4208 012266 102403                         BVS DNM7A ;BR TO ERROR IF V-BIT SET
4209 012270 100402                         BMI DNM7A ;BR TO ERROR IF N-BIT SET
4210 012272 103401                         BCS DNM7A ;BR TO ERROR IF C-BIT SET
4211 012274 001404                         BEQ DNM7B
4212                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <= =
4213                                     :                                     <---
4214                                     :                                     <---
4215                                     :                                     <---
4216 012276                               DNM7A: MOV #277,-(R2) ;MOVE TO MAILBOX # ***** 277 *****
4217 012276 012742 000277                 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4218 012302 005242                         HALT ;COND. CODES INCORRECT
4219 012304 000000                         BEQ DNM7B ;CHECK DEST. REGISTER
4220 012306 022700 000001                 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=-
4221 012312 001404                         :                                     <--
4222                                     :                                     <---
4223                                     :                                     <---
4224                                     :                                     <---
```

```

4225 012314 012742 000300      MOV    #300,-(R2)      ;MOVE TO MAILBOX # ***** 300 *****
4226 012320 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
4227 012322 000000              HALT                    ;DESTINATION REGISTER MODIFIED
4228 012324 022737 125125 000000 DNM7C: CMP    #125125,@#0    ;CHECK DEST. DATA
4229 012332 001404              BEQ    TS144
4230
4231
4232
4233
4234 012334 012742 000301      MOV    #301,-(R2)      ;MOVE TO MAILBOX # ***** 301 *****
4235 012340 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
4236 012342 000000              HALT                    ;DEST. DATA INCORRECT
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248 012344 005212 000144      TS144: INC    (R2)          ;UPDATE TEST NUMBER
4249 012346 022712              CMP    #144,(R2)      ;SEQUENCE ERROR?
4250 012352 001016              BNE    TS145-10       ;BR TO ERROR HALT ON SEQ ERROR
4251 012354 005000              CLR    R0             ;R0=0
4252 012356 005010              CLR    (R0)          ;LOC. 0=0
4253 012360 005100              COM    R0             ;R0=-1
4254 012362 005004              CLR    R4             ;R4 POINTS TO LOC. 0
4255 012364 010014              MOV    R0,(R4)        ;TRY MOVE MODE 0,1
4256 012366 102402              BVS    MDM1A          ;BR TO ERROR IF V SET
4257 012370 001401              BEQ    MDM1A          ;BR TO ERROR IF Z SET
4258 012372 100404              BMI    MDM1B
4259
4260
4261
4262
4263 012374
4264 012374 012742 000302      MDM1A: MOV    #302,-(R2)      ;MOVE TO MAILBOX # ***** 302 *****
4265 012400 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
4266 012402 000000              HALT                    ;CONDITION CODE NOT CORRECT
4267 012404 005704      MDM1B: TST    R4
4268 012406 001404              BEQ    TS145
4269
4270
4271
4272
4273 012410 012742 000303      MOV    #303,-(R2)      ;MOVE TO MAILBOX # ***** 303 *****
4274 012414 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
4275 012416 000000              HALT                    ;DESTINATION REGISTER INCORRECTLY ALTERED
4276
4277
4278
4279
4280
    
```

```

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

```

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

```

; 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/ 761 <-
    
```

```

:*****
: THIS TEST VERIFIES THE MOV DESTINATION MODE 2 INSTRUCTION.
:*****
    
```



```
4281 : DATA IS SET IN R0 USING SOP INSTRUCTIONS AND THEN MOVED
4282 : TO LOCATION 0 USING MOV SRC MODE 0, DEST. MODE 1.
4283 :
4284 :*****
4285 :TEST 145 TEST MOV DESTINATION MODE 2
4286 :*****
4287 012420 005212 TS145: INC (R2) ;UPDATE TEST NUMBER
4288 012422 022712 000145 CMP #145,(R2) ;SEQUENCE ERROR?
4289 012426 001026 BNE TS146-10 ;BR TO ERROR HALT ON SEQ ERROR
4290 012430 005000 CLR R0 ;R0=0
4291 012432 005001 CLR R1 ;R1=0
4292 012434 005010 CLR (R0) ;LOC.0=0
4293 012436 005110 COM (R0) ;LOC. 0- 1
4294 012440 010120 MOV R1,(R0)+ ;TRY MOVE MODE 0,2
4295 012442 100402 BMI MDM2A ;BR TO ERROR IF N SET
4296 012444 102401 BVS MDM2A ;BR TO ERROR IF V SET
4297 012446 001404 BEQ MDM2B
4298 :
4299 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--
4300 : CONDITIONAL BRANCH INST. AND <-
4301 : REPLACE THE MOVE INSTRUCTION <--
4302 : WHICH FOLLOWS W/ 767 <==
4302 012450 MDM2A: MOV #304,-(R2) ;MOVE TO MAILBOX # ***** 304 *****
4303 012450 012742 000304 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4304 012454 005242 HALT ;CC'S INCORRECT
4305 012456 000000
4306 012460 005300 MDM2B: DEC R0
4307 012462 005300 DEC R0
4308 012464 001404 BEQ MDM2D
4309 :
4310 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
4311 : CONDITIONAL BRANCH INST. AND <===
4312 : REPLACE THE MOVE INSTRUCTION <===
4313 : WHICH FOLLOWS W/ 760 <===
4313 012466 MDM2C: MOV #305,-(R2) ;MOVE TO MAILBOX # ***** 305 *****
4314 012466 012742 000305 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4315 012472 005242 HALT ;DESTINATION REGISTER NOT INCREMENTED PROPERLY
4316 012474 000000
4317 012476 005737 000000 MDM2D: TST #0
4318 012502 001404 BEQ TS146
4319 :
4320 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
4321 : CONDITIONAL BRANCH INST. AND <===
4322 : REPLACE THE MOVE INSTRUCTION <===
4323 : WHICH FOLLOWS W/ 751 <===
4323 012504 012742 000306 MOV #306,-(R2) ;MOVE TO MAILBOX # ***** 306 *****
4324 012510 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4325 012512 000000 HALT ;DESTINATION DATA INCORRECT
4326 : OR SEQUENCE ERROR
4327 :
4328 :*****
4329 :
4330 : THIS TEST VERIFIES DESTINATION MODE 2 W/MOVB INSTS. TWO DIFFERENT MOVB
4331 : INSTRUCTIONS ARE USED TO MOVE A TEST PATTERN FIRST TO BYTE 0 THEN TO BYTE 1.
4332 :
4333 :*****
4334 :TEST 146 TEST MOV-BYTE DESTINATION MODE 2
4335 :*****
4336 012514 005212 TS146: INC (R2) ;UPDATE TEST NUMBER
```

```
4337 012516 022712 000146      CMP      #146,(R2)      ;SEQUENCE ERROR?
4338 012522 001046      BNE      TS147-10     ;BR TO ERROR HALT ON SEQ ERROR
4339 012524 005000      CLR      R0          ;R0=0
4340 012526 005010      CLR      (R0)        ;LOC. 0=0
4341 012530 112720 000125      MOVB     #125,(R0)+   ;TRY DESTINATION MODE 2 W/EVEN BYTE
4342 012534 102402      BVS      MBDM2A      ;BR TO ERROR IF V SET
4343 012536 001401      BEQ      MBDM2A      ;BR TO ERROR IF Z SET
4344 012540 100004      BPL      MBDM2B
4345
4346
4347
4348
4349 012542      MBDM2A:
4350 012542 012742 000307      MOV      #307,-(R2)   ;MOVE TO MAILBOX # ***** 307 *****
4351 012546 005242      INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
4352 012550 000000      HALT
4353 012552 022700 000001      MBDM2B: CMP      #1,R0   ;CC'S INCORRECT
4354 012556 001404      BEQ      MBDM2C
4355
4356
4357
4358
4359 012560 012742 000310      MOV      #310,-(R2)   ;MOVE TO MAILBOX # ***** 310 *****
4360 012564 005242      INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
4361 012566 000000      HALT
4362 012570 112720 000252      MBDM2C: MOVB     #252,(R0)+ ;REGISTER NOT INCREMENTED BY ONE
4363 012574 102402      BVS      MBDM2D      ;TRY DESTINATION MODE 2 W/ODD BYTE
4364 012576 001401      BEQ      MBDM2D
4365 012600 100404      BMI      MBDM2E
4366
4367
4368
4369
4370 012602      MBDM2D:
4371 012602 012742 000311      MOV      #311,-(R2)   ;MOVE TO MAILBOX # ***** 311 *****
4372 012606 005242      INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
4373 012610 000000      HALT
4374 012612 022700 000002      MBDM2E: CMP      #2,R0   ;CC'S NOT SET CORRECT
4375 012616 001404      BEQ      MBDM2F
4376
4377
4378
4379
4380 012620 012742 000312      MOV      #312,-(R2)   ;MOVE TO MAILBOX # ***** 312 *****
4381 012624 005242      INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
4382 012626 000000      HALT
4383 012630 022737 125125 000000 MBDM2F: CMP      #125125,@#0 ;REGISTER NOT INCREMENTED BY ONE
4384 012636 001404      BEQ      TS147        ;CHECK DATA
4385
4386
4387
4388
4389 012640 012742 000313      MOV      #313,-(R2)   ;MOVE TO MAILBOX # ***** 313 *****
4390 012644 005242      INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
4391 012646 000000      HALT                ;DESTINATION DATA INCORRECT
4392
; OR SEQUENCE ERROR
```

4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448

012650 005212
012652 022712 000147
012656 001057
012660 012700 000400
012664 005010
012666 005037 000000
012672 012730 125252
012676 102402
012700 001401
012702 100404

012704
012704 012742 000314
012710 005242
012712 000000
012714 022700 000402
012720 001404

012722 012742 000315
012726 005242
012730 000000
012732 022737 125252 000000
012740 001404

012742 012742 000316
012746 005242
012750 000000
012752 112737 000125 000000
012760 022737 125125 000000
012766 001404

012770 012742 000317
012774 005242
012776 000000
013000 112737 000525 000001

```
*****
: 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, MOV B
: 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
BVS MDM3A ;BR TO ERROR IF V SET
BEQ MDM3A ;BR TO ERROR IF Z SET
BMI MDM3B

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

MDM3A: MOV #314,-(R2) ;MOVE TO MAILBOX # ***** 314 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S INCORRECT
MDM3B: CMP #402,R0 ;CHECK DEST. MODE REGISTER
BEQ MDM3C

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

MOV #315,-(R2) ;MOVE TO MAILBOX # ***** 315 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;REGISTER NOT INCREMENTED BY 2
MDM3C: CMP #125252,@#0 ;CHECK DESTINATION DATA
BEQ MDM3D

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

MOV #316,-(R2) ;MOVE TO MAILBOX # ***** 316 *****
INC --(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DESTINATION DATA INCORRECT
MDM3D: MOV B #125,@#0 ;TRY MOV B DESTINATION MODE 2 EVEN BYTE
CMP #125125,@#0 ;CHECK DATA
BEQ MDM3E

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
; CONDITIONAL BRANCH INST. AND <=
; REPLACE THE MOVE INSTRUCTION <
; WHICH FOLLOWS W/ 733 <

MOV #317,-(R2) ;MOVE TO MAILBOX # ***** 317 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DESTINATION DATA INCORRECT
MDM3E: MOV B #525,@#1 ;TRY MOV B DESTINATION MODE 2 ODD BYTE
```

```
4449 013006 022737 052525 000000      CMP      #52525,@#0      ;CHECK DATA
4450 013014 001404                BEQ      TS150
4451                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
4452                                ; CONDITIONAL BRANCH INST. AND <===
4453                                ; REPLACE THE MOVE INSTRUCTION <===
4454                                ; WHICH FOLLOWS W/ 720 <===
4455 013016 012742 000320      MOV      #320,-(R2)      ;MOVE TO MAILBOX # ***** 320 *****
4456 013022 005242                INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4457 013024 000000                HALT
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
```

```
*****
: THIS TEST VERIFIES THE MOV DESTINATION MODE 4 INSTRUCTION.
: SOP INSTRUCTIONS ON R0 ARE USED TO CLEAR TARGET LOCATION 0.
: R4 IS USED AS THE MODE 4 ADDRESSING REGISTER, AND
: CONDITIONAL BRANCHES ARE USED TO VERIFY THE DATA.
*****
```

```
4469 013026 005212                ;TEST 150      TEST MOV DESTINATION MODE 4
4470 013030 022712 000150      TS150: INC      (R2)          ;UPDATE TEST NUMBER
4471 013034 001026                CMP      #150,(R2)      ;SEQUENCE ERROR?
4472 013036 005000                BNE     TS151-10        ;BR TO ERROR HALT ON SEQ ERROR
4473 013040 005010                CLR     R0              ;R0=0
4474 013042 012704 000002      CLR     (R0)            ;LOC 0=0
4475 013046 012744 012345      MOV     #2,R4           ;R4=2
4476 013052 102402                MOV     #12345,-(R4)    ;TRY MOV DEST. MODE 4
4477 013054 001401                BVS     MDM4A           ;BR TO ERROR IF V-BIT SET
4478 013056 100004                BEQ     MDM4A           ;BR TO ERROR IF Z-BIT SET
4479
4480
4481
4482
```

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

```
4483 013060                MDM4A:
4484 013060 012742 000321      MOV     #321,-(R2)      ;MOVE TO MAILBOX # ***** 321 *****
4485 013064 005242                INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
4486 013066 000000                HALT   ;CC'S NOT CORRECT
4487 013070 005704                MDM4B: TST     R4         ;CHECK DECREMENTING OF MODE 4 REG.
4488 013072 001404                BEQ     MDM4C
4489
4490
4491
4492
```

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

```
4493 013074 012742 000322      MOV     #322,-(R2)      ;MOVE TO MAILBOX # ***** 322 *****
4494 013100 005242                INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
4495 013102 000000                HALT   ;DESTINATION MODE REGISTER NOT DECREMENTED BY 2
4496 013104 022710 012345      MDM4C: CMP     #12345,(R0) ;CHECK DESTINATION DATA
4497 013110 001404                BEQ     TS151
4498
4499
```

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

```
4502 013112 012742 000323      MOV     #323,-(R2)      ;MOVE TO MAILBOX # ***** 323 *****
4503 013116 005242                INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
4504 013120 000000                HALT   ;DESTINATION DATA INCORRECT
```

4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560

; OR SEQUENCE ERROR

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

:TEST 151 TEST MOV B DESTINATION MODE 4

TS151: INC (R2) ;UPDATE TEST NUMBER
CMP #151,(R2) ;SEQUENCE ERROR?
BNE TS152-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R4 ;R4=0
CLR (R4) ;LOC. 0=0
MOV #2,R0 ;R0 = 2
MOV B #125125,-(R0) ;TRY MOV B DEST. MODE 4--ODD BYTE
CMP R0,#1 ;CHECK THAT DEST. REG. WAS DECREMENTED
BEQ MBDM4A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
MOV #324,-(R2) ;MOVE TO MAILBOX # ***** 324 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DESTINATION REG. NOT DECREMENTED BY 1
MBDM4A: CMP (R4),#52400 ;CHECK DEST. DATA
BEQ MBDM4B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 757 <====
MOV #325,-(R2) ;MOVE TO MAILBOX # ***** 325 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DEST. DATA NOT CORRECT
MBDM4B: MOV B #125125,-(R0) ;TRY MOV B DEST. MODE 4--EVEN BYTE
BVS MBDM4C ;BR. TO ERROR IF V-BIT SET
BEQ MBDM4C ;BR TO ERROR IF Z-BIT SET
BPL MBDM4D
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 746 <====
MBDM4C: MOV #326,-(R2) ;MOVE TO MAILBOX # ***** 326 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;COND. CODES INCORRECT
MBDM4D: TST R0 ;CHECK MODE 4 DEST. REGISTER
BEQ MBDM4E
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 740 <====

```
4561 013230 012742 000327      MOV      #327,-(R2)      ;MOVE TO MAILBOX # ***** 327 *****
4562 013234 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4563 013236 000000              HALT                    ;DESTINATION REG NOT DECREMENTED BY 1
4564 013240 021427 052525      MBDM4E: CMP      (R4),#52525 ;CHECK DEST. DATA
4565 013244 001404              BEQ      TS152
4566                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
4567                                ;          CONDITIONAL BRANCH INST. AND <---
4568                                ;          REPLACE THE MOVE INSTRUCTION <---
4569                                ;          WHICH FOLLOWS W/ 731 <---
4570 013246 012742 000330      MOV      #330,-(R2)      ;MOVE TO MAILBOX # ***** 330 *****
4571 013252 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4572 013254 000000              HALT                    ;DESTINATION DATA INCORRECT
4573                                ; OR SEQUENCE ERROR
```

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

```
*****
: TEST 152 TEST MOV DESTINATION MODE 5
: *****
```

```
4587 013256 005212              TS152: INC      (R2)          ;UPDATE TEST NUMBER
4588 013260 022712 000152      CMP      #152,(R2)      ;SEQUENCE ERROR?
4589 013264 001051              BNE     TS153-10        ;BR TO ERROR HALT ON SEQ ERROR
4590 013266 005004              CLR     R4              ;R4=0
4591 013270 005014              CLR     (R4)            ;LOC. 0 = 0
4592 013272 012700 000400      MOV     #400,R0         ;R0=400
4593 013276 012750 004321      MOV     #4321,@-(R0)    ;TRY MOV DEST. MODE 5
4594 013302 102402              BVS     MDM5A           ;BR TO ERROR IF V-BIT SET
4595 013304 001401              BEQ     MDM5A           ;BR TO ERROR IF Z-BIT SET
4596 013306 100004              BPL     MDM5B
4597                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
4598                                ;          CONDITIONAL BRANCH INST. AND <
4599                                ;          REPLACE THE MOVE INSTRUCTION <
4600                                ;          WHICH FOLLOWS W/ 766 <
4601 013310              MDM5A:
4602 013310 012742 000331      MOV     #331,-(R2)      ;MOVE TO MAILBOX # ***** 331 *****
4603 013314 005242              INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
4604 013316 000000              HALT                    ;COND. CODES INCORRECT
4605 013320 022700 000376      MDM5B: CMP     #376,R0     ;CHECK MODE 5 REG. WAS DECREMENTED
4606 013324 001404              BEQ     MDM5C
4607                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
4608                                ;          CONDITIONAL BRANCH INST. AND <
4609                                ;          REPLACE THE MOVE INSTRUCTION <
4610                                ;          WHICH FOLLOWS W/ 757 <
4611 013326 012742 000332      MOV     #332,-(R2)      ;MOVE TO MAILBOX # ***** 332 *****
4612 013332 005242              INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
4613 013334 000000              HALT                    ;MODE 5 REGISTER NOT DECREMENTED BY 2
4614 013336 022714 004321      MDM5C: CMP     #4321,(R4) ;CHECK DEST. DATA
4615 013342 001404              BEQ     MDM5D
4616                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
```

```
4617 ;
4618 ;           CONDITIONAL BRANCH INST. AND <====
4619 ;           REPLACE THE MOVE INSTRUCTION <====
4620 013344 012742 000333          MOV #333,-(R2) ;MOVE TO MAILBOX # ***** 333 ***** <====
4621 013350 005242          INC -(R2) ;SET MSGTYP TO FATAL ERROR
4622 013352 000000          HALT ;DEST. DATA INCORRECT
4623 013354 012700 000406 MDM5D: MOV #406,R0 ;R0=406
4624 013360 112750 000377          MOVB #377,@-(R0) ;TRY MOV DEST. MODE 5 --EVEN BYTE
4625 013364 022700 000404          CMP #404,R0 ;CHECK MODE 5 REG.
4626 013370 001404          BEQ MDM5E
4627 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4628 ;           CONDITIONAL BRANCH INST. AND <====
4629 ;           REPLACE THE MOVE INSTRUCTION <====
4630 ;           WHICH FOLLOWS W/ 750 <====
4631 013372 012742 000334          MOV #334,-(R2) ;MOVE TO MAILBOX # ***** 334 *****
4632 013376 005242          INC -(R2) ;SET MSGTYP TO FATAL ERROR
4633 013400 000000          HALT ;MODE 5 REGISTER NOT DECREMENTED BY 2
4634 013402 022714 177721 MDM5E: CMP #177721,(R4) ;CHECK DEST. DATA
4635 013406 001404          BEQ TS153
4636 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4637 ;           CONDITIONAL BRANCH INST. AND <====
4638 ;           REPLACE THE MOVE INSTRUCTION <====
4639 ;           WHICH FOLLOWS W/ 726 <====
4640 013410 012742 000335          MOV #335,-(R2) ;MOVE TO MAILBOX # ***** 335 *****
4641 013414 005242          INC -(R2) ;SET MSGTYP TO FATAL ERROR
4642 013416 000000          HALT ;DEST. DATA INCORRECT
4643 ; OR SEQUENCE ERROR
4644 ;
4645 ;
4646 ;
4647 ;
4648 ;
4649 ;
4650 ;
4651 ;
4652 ;
4653 ;
4654 ;
4655 ;
```

```
*****
: 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.
*****
```

```
4656 013420 005212          INC (R2) ;UPDATE TEST NUMBER
4657 013422 022712 000153 MDM6A: CMP #153,(R2) ;SEQUENCE ERROR?
4658 013426 001054          BNE TS154-10 ;BR TO ERROR HALT ON SEQ ERROR
4659 013430 005000          CLR R0 ;R0=0
4660 013432 005010          CLR (R0) ;LOC. 0=0
4661 013434 005200          INC R0 ;R0=1
4662 013436 012760 052525 177777 MDM6A: MOV #052525,-1(R0) ;TRY MOV DEST. MODE 6
4663 013444 102402          BVS MDM6A ;BR TO ERROR IF V-BIT SET
4664 013446 001401          BEQ MDM6A ;BR TO ERROR IF Z-BIT SET
4665 013450 100004          BPL MDM6B
4666 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4667 ;           CONDITIONAL BRANCH INST. AND <====
4668 ;           REPLACE THE MOVE INSTRUCTION <====
4669 ;           WHICH FOLLOWS W/ 766 <====
4670 013452          MDM6A:
4671 013452 012742 000336          MOV #336,-(R2) ;MOVE TO MAILBOX # ***** 336 *****
4672 013456 005242          INC -(R2) ;SET MSGTYP TO FATAL ERROR
```

```

4673 013460 000000
4674 013462 022700 000001      MDM6B: HALT
4675 013466 001404              CMP      #1,R0
4676                                BEQ      MDM6C
4677                                ;COND. CODES INCORRECT
4678                                ;CHECK DEST. REGISTER UNALTERED
4679                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
4680                                ;          CONDITIONAL BRANCH INST. AND
4681                                ;          REPLACE THE MOVE INSTRUCTION
4682                                ;          WHICH FOLLOWS W/ 757
4683                                ;          <====
4684                                ;          <====
4685                                ;          <====
4686                                ;          <====
4687                                ;          <====
4688                                ;          <====
4689 013470 012742 000337      MOV      #337,-(R2)
4690 013474 005242              INC      -(R2)
4691 013476 000000              HALT
4692 013500 022737 052525 000000 MDM6C: CMP      #52525,@#0
4693 013506 001404              BEQ      MDM6D
4694                                ;COND. CODES INCORRECT
4695                                ;CHECK DEST. REGISTER UNALTERED
4696                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
4697                                ;          CONDITIONAL BRANCH INST. AND
4698                                ;          REPLACE THE MOVE INSTRUCTION
4699                                ;          WHICH FOLLOWS W/ 747
4700                                ;          <====
4701                                ;          <====
4702                                ;          <====
4703                                ;          <====
4704                                ;          <====
4705                                ;          <====
4706                                ;          <====
4707                                ;          <====
4708                                ;          <====
4709 013510 012742 000340      MOV      #340,-(R2)
4710 013514 005242              INC      -(R2)
4711 013516 000000              HALT
4712 013520 012700 000002      MDM6D: MOV      #2,R0
4713 013524 112760 000377 177777 MOVB    #377,-1(R0)
4714 013532 022700 000002      CMP      #2,R0
4715 013536 001404              BEQ      MDM6E
4716                                ;COND. CODES INCORRECT
4717                                ;CHECK DEST. REGISTER UNALTERED
4718                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
4719                                ;          CONDITIONAL BRANCH INST. AND
4720                                ;          REPLACE THE MOVE INSTRUCTION
4721                                ;          WHICH FOLLOWS W/ 733
4722                                ;          <
4723                                ;          <
4724                                ;          <
4725                                ;          <
4726                                ;          <
4727                                ;          <
4728                                ;          <
4729 013540 012742 000341      MOV      #341,-(R2)
4730 013544 005242              INC      -(R2)
4731 013546 000000              HALT
4732 013550 022737 177525 000000 MDM6E: CMP      #177525,@#0
4733 013556 001404              BEQ      TS154
4734                                ;COND. CODES INCORRECT
4735                                ;CHECK DEST. REGISTER UNALTERED
4736                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
4737                                ;          CONDITIONAL BRANCH INST. AND
4738                                ;          REPLACE THE MOVE INSTRUCTION
4739                                ;          WHICH FOLLOWS W/ 723
4740                                ;          <
4741                                ;          <
4742                                ;          <
4743                                ;          <
4744                                ;          <
4745                                ;          <
4746                                ;          <
4747                                ;          <
4748 013560 012742 000342      MOV      #342,-(R2)
4749 013564 005242              INC      -(R2)
4750 013566 000000              HALT
4751                                ;COND. CODES INCORRECT
4752                                ;CHECK DEST. REGISTER UNALTERED
4753                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
4754                                ;          CONDITIONAL BRANCH INST. AND
4755                                ;          REPLACE THE MOVE INSTRUCTION
4756                                ;          WHICH FOLLOWS W/ 723
4757                                ;          <
4758                                ;          <
4759                                ;          <
4760                                ;          <
4761                                ;          <
4762                                ;          <
4763                                ;          <
4764                                ;          <
4765                                ;          <
4766                                ;          <
4767                                ;          <
4768                                ;          <
4769                                ;          <
4770                                ;          <
4771                                ;          <
4772                                ;          <
4773                                ;          <
4774                                ;          <
4775                                ;          <
4776                                ;          <
4777                                ;          <
4778                                ;          <
4779                                ;          <
4780                                ;          <
4781                                ;          <
4782                                ;          <
4783                                ;          <
4784                                ;          <
4785                                ;          <
4786                                ;          <
4787                                ;          <
4788                                ;          <
4789                                ;          <
4790                                ;          <
4791                                ;          <
4792                                ;          <
4793                                ;          <
4794                                ;          <
4795                                ;          <
4796                                ;          <
4797                                ;          <
4798                                ;          <
4799                                ;          <
4800                                ;          <
4801                                ;          <
4802                                ;          <
4803                                ;          <
4804                                ;          <
4805                                ;          <
4806                                ;          <
4807                                ;          <
4808                                ;          <
4809                                ;          <
4810                                ;          <
4811                                ;          <
4812                                ;          <
4813                                ;          <
4814                                ;          <
4815                                ;          <
4816                                ;          <
4817                                ;          <
4818                                ;          <
4819                                ;          <
4820                                ;          <
4821                                ;          <
4822                                ;          <
4823                                ;          <
4824                                ;          <
4825                                ;          <
4826                                ;          <
4827                                ;          <
4828                                ;          <
4829                                ;          <
4830                                ;          <
4831                                ;          <
4832                                ;          <
4833                                ;          <
4834                                ;          <
4835                                ;          <
4836                                ;          <
4837                                ;          <
4838                                ;          <
4839                                ;          <
4840                                ;          <
4841                                ;          <
4842                                ;          <
4843                                ;          <
4844                                ;          <
4845                                ;          <
4846                                ;          <
4847                                ;          <
4848                                ;          <
4849                                ;          <
4850                                ;          <
4851                                ;          <
4852                                ;          <
4853                                ;          <
4854                                ;          <
4855                                ;          <
4856                                ;          <
4857                                ;          <
4858                                ;          <
4859                                ;          <
4860                                ;          <
4861                                ;          <
4862                                ;          <
4863                                ;          <
4864                                ;          <
4865                                ;          <
4866                                ;          <
4867                                ;          <
4868                                ;          <
4869                                ;          <
4870                                ;          <
4871                                ;          <
4872                                ;          <
4873                                ;          <
4874                                ;          <
4875                                ;          <
4876                                ;          <
4877                                ;          <
4878                                ;          <
4879                                ;          <
4880                                ;          <
4881                                ;          <
4882                                ;          <
4883                                ;          <
4884                                ;          <
4885                                ;          <
4886                                ;          <
4887                                ;          <
4888                                ;          <
4889                                ;          <
4890                                ;          <
4891                                ;          <
4892                                ;          <
4893                                ;          <
4894                                ;          <
4895                                ;          <
4896                                ;          <
4897                                ;          <
4898                                ;          <
4899                                ;          <
4900                                ;          <
4901                                ;          <
4902                                ;          <
4903                                ;          <
4904                                ;          <
4905                                ;          <
4906                                ;          <
4907                                ;          <
4908                                ;          <
4909                                ;          <
4910                                ;          <
4911                                ;          <
4912                                ;          <
4913                                ;          <
4914                                ;          <
4915                                ;          <
4916                                ;          <
4917                                ;          <
4918                                ;          <
4919                                ;          <
4920                                ;          <
4921                                ;          <
4922                                ;          <
4923                                ;          <
4924                                ;          <
4925                                ;          <
4926                                ;          <
4927                                ;          <
4928                                ;          <
4929                                ;          <
4930                                ;          <
4931                                ;          <
4932                                ;          <
4933                                ;          <
4934                                ;          <
4935                                ;          <
4936                                ;          <
4937                                ;          <
4938                                ;          <
4939                                ;          <
4940                                ;          <
4941                                ;          <
4942                                ;          <
4943                                ;          <
4944                                ;          <
4945                                ;          <
4946                                ;          <
4947                                ;          <
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                                ;          <

```

```

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

```

```

4722 :TEST 154      TEST MOV DESTINATION MODE 7
4723 :*****
4724 TS154: INC      (R2)      ;UPDATE TEST NUMBER
4725      CMP      #154,(R2) ;SEQUENCE ERROR?
4726      BNE     TS155-10   ;BR TO ERROR HALT ON SEQ ERROR
4727      CLR     R4         ;R4=0
4728      CLR     (R4)      ;LOC.0=0

```



```
4729 013604 012700 000403      MOV      #403,R0      ;R0=403
4730 013610 012770 070707 177777  MOV      #70707,@-1(R0) ;TRY MOV W/DEST MODE 7
4731 013616 102402      BVS      MDM7A        ;BR. TO ERROR IF V-BIT SET
4732 013620 001401      BEQ      MDM7A        ;BR TO ERROR IF Z-BIT SET
4733 013622 100004      BPL      MDM7B
4734                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4735                                     ;          CONDITIONAL BRANCH INST. AND <====
4736                                     ;          REPLACE THE MOVE INSTRUCTION <====
4737                                     ;          WHICH FOLLOWS W/ 765 <====
4738 013624                                     MDM7A:
4739 013624 012742 000343      MOV      #343,-(R2)    ;MOVE TO MAILBOX # ***** 343 *****
4740 013630 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4741 013632 000000      HALT
4742 013634 022700 000403      MDM7B: CMP      #403,R0    ;COND. CODES INCORRECT
4743 013640 001404      BEQ      MDM7C        ;CHECK DEST. REGISTER
4744                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4745                                     ;          CONDITIONAL BRANCH INST. AND <====
4746                                     ;          REPLACE THE MOVE INSTRUCTION <====
4747                                     ;          WHICH FOLLOWS W/ 756 <====
4748 013642 012742 000344      MOV      #344,-(R2)    ;MOVE TO MAILBOX # ***** 344 *****
4749 013646 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4750 013650 000000      HALT
4751 013652 022737 070707 000000  MDM7C: CMP      #70707,@#0  ;DEST. REGISTER INCORRECTLY ALTERED
4752 013660 001404      BEQ      MDM7D        ;CHECK DEST. DATA
4753                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4754                                     ;          CONDITIONAL BRANCH INST. AND <====
4755                                     ;          REPLACE THE MOVE INSTRUCTION <====
4756                                     ;          WHICH FOLLOWS W/ 746 <====
4757 013662 012742 000345      MOV      #345,-(R2)    ;MOVE TO MAILBOX # ***** 345 *****
4758 013666 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4759 013670 000000      HALT
4760 013672 112770 107070 000001  MDM7D: MOVB     #107070,@1(R0) ;DEST. DATA INCORRECT
4761 013700 022700 000403      CMP      #403,R0    ;TRY MOVW W/DEST MODE 7--ODD BYTE
4762 013704 001404      BEQ      MDM7E        ;CHECK MODE 7 DEST. REG.
4763                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4764                                     ;          CONDITIONAL BRANCH INST. AND <====
4765                                     ;          REPLACE THE MOVE INSTRUCTION <====
4766                                     ;          WHICH FOLLOWS W/ 734 <====
4767 013706 012742 000346      MOV      #346,-(R2)    ;MOVE TO MAILBOX # ***** 346 *****
4768 013712 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4769 013714 000000      HALT
4770 013716 022737 034307 000000  MDM7E: CMP      #34307,@#0 ;DEST. DATA INCORRECT
4771 013724 001404      BEQ      TS155        ;CHECK DEST. DATA
4772                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4773                                     ;          CONDITIONAL BRANCH INST. AND <====
4774                                     ;          REPLACE THE MOVE INSTRUCTION <====
4775                                     ;          WHICH FOLLOWS W/ 724 <====
4776 013726 012742 000347      MOV      #347,-(R2)    ;MOVE TO MAILBOX # ***** 347 *****
4777 013732 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4778 013734 000000      HALT
4779                                     ;DESTINATION DATA INCORRECT
4780                                     ; OR SEQUENCE ERROR
4781
4782
4783
4784
```

: THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.
: THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A

```

4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796 013736 005212
4797 013740 022712 000155
4798 013744 001015
4799 013746 012700 014020
4800 013752 014037 014020
4801 013756 064037 014020
4802 013762 144037 014020
4803 013766 154037 014021
4804 013772 024037 014020
4805 013776 001411
4806
4807
4808
4809
4810 014000
4811 014000 012742 000350
4812 014004 005242
4813 014006 000000
4814
4815
4816 014010 125252
4817 014012 052652
4818 014014 053125
4819 014016 125252
4820 014020 000000
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834 014022 005212
4835 014024 022712 000156
4836 014030 001015
4837 014032 012700 014106
4838 014036 015037 014020
4839 014042 065037 014020
4840 014046 145037 014020

```

```

:TABLE OF OPERANDS. THE TABLE OF OPERANDS AND THE WORK LOCATION IS
:STORED FOLLOWING THE TEST CODE. A SERIES OF 5 DOP INSTRUCTIONS UTILIZES
:THE DATA IN THE TABLE TO CYCLE THE WORK LOCATION THRU A SET OF
:VALUE. THE DATA HAS BEEN CHOSEN TO INSURE THAT NO SINGLE ERROR WILL
:GO UNDETECTED. WORD AND BYTE INSTRUCTION ACCESSING BOTH EVEN AND
:ODD ADDRESSES ARE USED IN THE TEST. THE LISTING SHOWS THE
:EXPECTED INTERMEDIATE RESULT AS EACH INSTRUCTION IS EXECUTED.

```

```

:*****
:TEST 155 TEST MODE 4 W/ DOP INSTS.
:*****

```

```

TS155: INC (R2) ;UPDATE TEST NUMBER
      CMP #155,(R2) ;SEQUENCE ERROR?
      BNE DOP4 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #TBL1,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 TS156
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 762 <====

```

```

DOP4: MOV #350,-(R2) ;MOVE TO MAILBOX # ***** 350 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;RESULT OF MODE 4 INSTS. INCORRECT
; OR SEQUENCE ERROR

```

```

125252
52652
53125
125252
TBL1: 0

```

```

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

```

```

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

```

```

TS156: INC (R2) ;UPDATE TEST NUMBER
      CMP #156,(R2) ;SEQUENCE ERROR?
      BNE DOP5 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #TBL2+2,R0 ;INITIALIZE R0
      MOV @-(R0),@#TBL1 ;TBL1=125252
      ADD @-(R0),@#TBL1 ;TBL1=000377
      BICB @-(R0),@#TBL1 ;TBL1=000252

```

4841	014052	155037	014021
4842	014056	025037	014020
4843	014062	001411	
4844			
4845			
4846			
4847			
4848	014064		
4849	014064	012742	000351
4850	014070	005242	
4851	014072	000000	
4852			
4853	014074	014010	
4854	014076	014012	
4855	014100	014013	
4856	014102	014014	
4857	014104	014016	

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

```

TBL2:

```

TBL1-10
TBL1-6
TBL1-5
TBL1-4
TBL1-2

```

```

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

```

4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869

```

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

```

4870
4871

:TEST 157 TEST MODE 6 W/ DOP INSTS.

4872	014106	005212	
4873	014110	022712	000157
4874	014114	001022	
4875	014116	012700	014014
4876	014122	016037	000000 014020
4877	014130	066037	000000 014020
4878	014136	146037	177777 014020
4879	014144	156037	177776 014021
4880	014152	026037	177774 014020
4881	014160	001404	

```

TS157: INC (R2) ;UPDATE TEST NUMBER
CMP #157,(R2) ;SEQUENCE ERROR?
BNE TS160-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #TBL1-4,R0 ;INITIALIZE R0
MOV 2(R0),@TBL1 ;TBL1=125252
ADD 0(R0),@TBL1 ;TBL1=000377
BICB -1(R0),@TBL1 ;TBL1=000252
BISB -2(R0),@TBL1+1 ;TBL1=125252
CMP -4(R0),@TBL1 ;CHECK RESULT
BEQ TS160

```

```

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

```

4882
4883
4884
4885
4886
4887
4888
4889

```

MOV #352,-(R2) ;MOVE TO MAILBOX # ***** 352 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULT OF MODE 6 INSTS. INCORRECT
; OR SEQUENCE ERROR

```

4890
4891
4892
4893
4894
4895
4896

```

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

```

4897
4898
4899
4900
4901
4902
4903 014172 005212
4904 014174 022712 000160
4905 014200 001022
4906 014202 012700 014100
4907 014206 017037 000004 014020
4908 014214 067037 000002 014020
4909 014222 147037 000000 014020
4910 014230 157037 177776 014021
4911 014236 027037 177774 01402C
4912 014244 001404
4913
4914
4915
4916
4917 014246 012742 000353
4918 014252 005242
4919 014254 000000
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932 014256 005212
4933 014260 022712 000161
4934 014264 001026
4935 014266 012700 125252
4936 014272 000261
4937 014274 006100
4938 014276 102004
4939 014300 103003
4940 014302 022700 052525
4941 014306 001404
4942
4943
4944
4945
4946 014310
4947 014310 012742 000354
4948 014314 005242
4949 014316 000000
4950 014320 012700 125252
4951 014324 000261
4952 014326 106100

: IN THE MODE 7 INSTRUCTIONS. THE DATA RESULTS ARE IDENTICAL TO
: THOSE EXPECTED IN THE MODE 5 TESTS.

: TEST 160 TEST MODE 7 W/ DOP INSTS.

TS160: INC (R2) ; UPDATE TEST NUMBER
CMP #160,(R2) ; SEQUENCE ERROR?
BNE TS161-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #TBL2-4,R0 ; INITIALIZE R0
MOV @4(R0),@#TBL1 ; TBL1=125252
ADD @2(R0),@#TBL1 ; TBL1=000377
BICB @0(R0),@#TBL1 ; TBL1=000252
BISB @-2(R0),@#TBL1+1 ; TBL1=125252
CMP @-4(R0),@#TBL1 ; CHECK RESULT
BEQ TS161

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 755 <====
MOV #353,-(R2) ; MOVE TO MAILBOX # ***** 353 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; RESULT OF MODE 7 INSTS INCORRECT
; OR SEQUENCE ERROR

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

: TEST 161 TEST ROTATE INSTRUCTIONS OF MODE 0

TS161: 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 R0 ; TRY ROL W/ MODE 0
BVC R0TOA ; CC=0011
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 <==--=
:

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 R0 ; TRY ROL W/ MODE 0 EVEN BYTE

```

4953 014330 102004          BVC     ROTOC          ;CC=0011
4954 014332 103003          BCC     ROTOC
4955 014334 022700 125125  CMP     #125125,R0    ;CHECK DATA
4956 014340 001404          BEQ     TS162
4957                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4958                                     ;                                     <====
4959                                     ;                                     <====
4960                                     ;                                     <====
4961 014342          ROTOC:
4962 014342 012742 000355  MOV     #355,-(R2)    ;MOVE TO MAILBOX # ***** 355 *****
4963 014346 005242          INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
4964 014350 000000          HALT                    ;ROLB MODE 0 FAILED
4965                                     ; OR SEQUENCE ERROR
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980 014352 005212          TS162: INC     (R2)        ;UPDATE TEST NUMBER
4981 014354 022712 000162  CMP     #162,(R2)    ;SEQUENCE ERROR?
4982 014360 001051          BNE     TS163-10    ;BR TO ERROR HALT ON SEQ ERROR
4983 014362 005000          CLR     R0          ;POINT TO LOC. 0
4984 014364 012710 052525  MOV     #52525,(R0) ;INITIALIZE DATA
4985 014370 000241          CLC                    ;CLEAR C-BIT
4986 014372 006110          ROL     (R0)         ;TRY ROL W/ MODE 1
4987 014374 102005          BVC     ROT1A        ;CC=1010
4988 014376 103404          BCS     ROT1A
4989 014400 023727 000000 125252  CMP     @#0,#125252 ;CHECK RESULT
4990 014406 001404          BEQ     ROT1B
4991                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4992                                     ;                                     <====
4993                                     ;                                     <====
4994                                     ;                                     <====
4995 014410          ROT1A:
4996 014410 012742 000356  MOV     #356,-(R2)    ;MOVE TO MAILBOX # ***** 356 *****
4997 014414 005242          INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
4998 014416 000000          HALT                    ;ROLB MODE 1 FAILED
4999 014420 000261          ROT1B: SEC
5000 014422 012710 125252  MOV     #125252,(R0) ;INITIALIZE DATA
5001 014426 106110          ROLB    (R0)         ;TRY ROLB W/ MODE 1 EVEN BYTE
5002 014430 102005          BVC     ROT1C        ;CC=1011
5003 014432 103004          BCC     ROT1C
5004 014434 022737 125125 000000  CMP     #125125,@#0 ;TEST RESULT
5005 014442 001404          BEQ     ROT1D
5006                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5007                                     ;                                     <====
5008                                     ;                                     <====

```

 : 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
 : *****

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

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

```
5009 ; WHICH FOLLOWS W/ 746 <====-
5010 014444 ROT1C: ;
5011 014444 012742 000357 MOV #357,-(R2) ;MOVE TO MAILBOX # ***** 357 *****
5012 014450 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5013 014452 000000 HALT ;ROL3 W/ MODE 1 EVEN BYTE FAILED
5014 014454 012710 125252 ROT1D: MOV #125252,(R0)
5015 014460 005000 CLR R0 ;POINT TO ODD BYTE
5016 014462 005200 INC R0
5017 014464 000261 SEC ;SET C-BIT
5018 014466 106110 ROLB (R0) ;TRY ROLB W/ MODE 1 ODD BYTE
5019 014470 102005 BVC ROT1E ;CC=0011
5020 014472 103004 BCC ROT1E
5021 014474 022737 052652 000000 CMP #052652,@#0 ;CHECK DATA
5022 014502 001404 BEQ TS163
5023 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-- -
5024 ; CONDITIONAL BRANCH INST. AND <====-
5025 ; REPLACE THE MOVE INSTRUCTION <====-
5026 ; WHICH FOLLOWS W/ 726 <-- -
5027 014504 ROT1E:
5028 014504 012742 000360 MOV #360,-(R2) ;MOVE TO MAILBOX # ***** 360 *****
5029 014510 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5030 014512 000000 HALT ;ROLB W/ MODE 1 ODD BYTE FAILED
5031 ; OR SEQUENCE ERROR
5032
5033 :*****
5034 :
5035 : THIS TEST VERIFIES MODE 2 ROTATE INSTRUCTIONS.
5036 : THE SAME PROCEDURE AS IN THE OTHER ROTATE TESTS ARE USED. R0
5037 : IS USED AS THE ADDRESSING REGISTER AND IS CHECKED FOR PROPER
5038 : INCREMENTING. BYTE INSTRUCTIONS ARE ALSO CHECKED.
5039 :
5040 :*****
5041 :TEST 163 TEST ROTATE INSTRUCTIONS W/ MODE 2
5042 :*****
5043 014514 005212 TS163: INC (R2) ;UPDATE TEST NUMBER
5044 014516 022712 000163 CMP #163,(R2) ;SEQUENCE ERROR?
5045 014522 001057 BNE TS164-10 ;BR TO ERROR HALT ON SEQ ERROR
5046 014524 005000 CLR R0 ;POINT TO LOC 0
5047 014526 012710 173737 MOV #173737,(R0) ;INITIALIZE DATA
5048 014532 000241 CLC ;CLEAR C-BIT
5049 014534 006120 ROL (R0)+ ;TRY ROL W/ MODE 2
5050 014536 103007 BCC ROT2A ;CHECK C-BIT
5051 014540 022737 167676 000000 CMP #167676,@#0 ;CHECK DATA
5052 014546 001003 BNE ROT2A ;BRANCH IF RESULT INCORRECT
5053 014550 005300 DEC R0 ;TEST R0
5054 014552 005300 DEC R0
5055 014554 001404 BEQ ROT2B
5056 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====-
5057 ; CONDITIONAL BRANCH INST. AND <= -
5058 ; REPLACE THE MOVE INSTRUCTION <====-
5059 ; WHICH FOLLOWS W/ 762 <= -
5060 014556 ROT2A:
5061 014556 012742 000361 MOV #361,-(R2) ;MOVE TO MAILBOX # ***** 361 *****
5062 014562 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5063 014564 000000 HALT ;ROL W/ MODE 2 FAILED
5064 014566 005000 ROT2B: CLR R0 ;POINT TO LOC 0
```

```

5065 014570 012710 004040      MOV      #4040,(R0)      ;INITIALIZE DATA
5066 014574 000741              CLC                    ;CLEAR C-BIT
5067 014576 106 20              ROLB     (R0)+         ;TRY ROLB W/ MODE 2 EVEN BYTE
5068 014600 103406              BCS     ROT2C          ;CHECK C-BIT
5069 014602 022737 004100 000000  CMP      #4100,@#0     ;CHECK DATA
5070 014610 001002              BNE     ROT2C          ;BRANCH IF DATA INCORRECT
5071 014612 005300              DEC     R0             ;CHECK R0
5072 014614 001404              BEQ     ROT2D
5073                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5074                                ;          CONDITIONAL BRANCH INST. AND <====
5075                                ;          REPLACE THE MOVE INSTRUCTION <====
5076                                ;          WHICH FOLLOWS W/ 742 <====

```

```

5077 014616              ROT2C:
5078 014616 012742 000362      MOV      #362,-(R2)     ;MOVE TO MAILBOX # ***** 362 *****
5079 014622 005242              INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
5080 014624 000000              HALT                    ;ROLB W/ MODE 2 EVEN BYTE FAILED
5081 014626 005000              ROT2D: CLR     R0       ;POINT TO LOC 0
5082 014630 012710 004040      MOV      #4040,(R0)     ;INITIALIZE DATA
5083 014634 005200              INC     R0             ;POINT TO ODD BYTE OF DATA
5084 014636 000261              SEC                    ;SET C-BIT
5085 014640 106120              ROLB     (R0)+         ;TRY ROL W/ MODE 2 ODD BYTE
5086 014642 103407              BCS     ROT2E          ;CHECK C-BIT
5087 014644 022737 010440 000000  CMP      #10440,@#0    ;CHECK DATA
5088 014652 001003              BNE     ROT2E          ;BRANCH IF DATA INCORRECT
5089 014654 005300              DEC     R0             ;CHECK R0
5090 014656 005300              DEC     R0
5091 014660 001404              BEQ     TS164

```

```

5092                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5093                                ;          CONDITIONAL BRANCH INST. AND <====
5094                                ;          REPLACE THE MOVE INSTRUCTION <====
5095                                ;          WHICH FOLLOWS W/ 720 <====

```

```

5096 014662              ROT2E:
5097 014662 012742 000363      MOV      #363,-(R2)     ;MOVE TO MAILBOX # ***** 363 *****
5098 014666 005242              INC     -(R2)          ;SET MSGTYP TO FATAL ERROR
5099 014670 000000              HALT                    ;ROLB W/ MODE 2 ODD BYTE FAILED
5100                                ; OR SEQUENCE ERROR

```

```

*****
: THIS TEST VERIFIES MODE 3 ROTATE INSTRUCTIONS.
: THIS TEST USES THE SAME PROCEDURES AS IN THE OTHER ROTATE
: TESTS. THE DATA IS STORED IN LOC. 0 AND IS ADDRESSED USING
: MODE 37. BYTE ADDRESSING IS ALSO CHECKED FOR EVEN AND ODD BYTES.
*****
: TEST 164 TEST ROTATE INSTRUCTIONS /W MODE 3
*****

```

```

5113 014672 005212              TS164: INC     (R2)       ;UPDATE TEST NUMBER
5114 014674 022712 000164      CMP      #164,(R2)     ;SEQUENCE ERROR?
5115 014700 001051              BNE     TS165-10      ;BR TO ERROR HALT ON SEQ ERROR
5116 014702 012737 052525 000000  MOV      #52525,@#0    ;INITIALIZE DATA IN LOC 0
5117 014710 000261              SEC                    ;SET C-BIT
5118 014712 006137 000000      ROL     @#0           ;TRO ROL W/ MODE 3
5119 014716 103404              BCS     ROT3A          ;CHECK C-BIT
5120 014720 022737 125253 000000  CMP      #125253,@#0   ;CHECK DATA

```

```

CJKDB-C DCF11-AA CPU DIAG. MACY11 30A(1052) 07-MAR-80 12:18 D 8 PAGE 94
CJKDBC.P11 07-MAR-80 12:17 T164 TEST ROTATE INSTRUCTIONS /W MODE 3 SEQ 0094

5121 014726 001404 BEQ ROT3B
5122 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5123 : CONDITIONAL BRANCH INST. AND <====
5124 : REPLACE THE MOVE INSTRUCTION <====
5125 : WHICH FOLLOWS W/ 764 <====
5126 014730 ROT3A:
5127 014730 012742 000364 MOV #364,-(R2) ;MOVE TO MAILBOX # ***** 364 *****
5128 014734 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5129 014736 000000 HALT ;ROL W/ MODE 3 FAILED
5130 014740 012737 125252 000000 ROT3B: MOV #125252,@#0 ;INITIALIZE DATA
5131 014746 000241 CLC ;CLEAR C-BIT
5132 014750 106137 000000 ROLB @#0 ;TRY ROL W/ MODE 3 EVEN BYTE
5133 014754 103004 BCC ROT3C ;CHECK C-BIT
5134 014756 023727 000000 125124 4$: CMP @#0,#125124 ;CHECK DATA
5135 014764 001404 BEQ ROT3D
5136 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5137 : CONDITIONAL BRANCH INST. AND <====
5138 : REPLACE THE MOVE INSTRUCTION <====
5139 : WHICH FOLLOWS W/ 745 <====
5140 014766 ROT3C:
5141 014766 012742 000365 MOV #365,-(R2) ;MOVE , MAILBOX # ***** 365 *****
5142 014772 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5143 014774 000000 HALT ;ROL W/ MODE 3 EVEN BYTE FAILED
5144 014776 012737 125252 000000 ROT3D: MOV #125252,@#0 ;INITIALIZE DATA IN LOC. 0
5145 015004 000261 SEC ;SET C-BIT
5146 015006 106137 000001 ROLB @#1 ;TRY ROL W/ MODE 3 ODD BYTE
5147 015012 103004 BCC ROT3E ;CHECK C-BIT
5148 015014 022737 052652 000000 CMP #052652,@#0 ;CHECK DATA
5149 015022 001404 BEQ TS165
5150 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5151 : CONDITIONAL BRANCH INST. AND <====
5152 : REPLACE THE MOVE INSTRUCTION <====
5153 : WHICH FOLLOWS W/ 726 <====
5154 015024 ROT3E:
5155 015024 012742 000366 MOV #366,-(R2) ;MOVE TO MAILBOX # ***** 366 *****
5156 015030 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5157 015032 000000 HALT ;ROL W/ MODE 3 ODD BYTE FAILED
5158 : OR SEQUENCE ERROR
5159
5160 :*****
5161 :
5162 : THIS TEST VERIFIES MODE 4 ROTATE INSTRUCTIONS. THE DATA IS
5163 : STORED IN LOC. 0. R0 IS SET TO 2 AND THE CARRY IS SET. AN ROL MODE 4
5164 : IS USED TO ROTATE LOCATION 0 USING R0. THE DATA IS CHECKED
5165 : AND THE C AND V BITS ARE TESTED. THE PROPER DECREMENTING OF
5166 : R0 IS VERIFIED.
5167 :
5168 :*****
5169 :TEST 165 TEST MODE 4 W/ ROTATE INSTRUCTIONS
5170 :*****
5171 015034 005212 TS165: INC (R2) ;UPDATE TEST NUMBER
5172 015036 022712 000165 CMP #165,(R2) ;SEQUENCE ERROR?
5173 015042 001016 BNE TS166-10 ;BR TO ERROR HALT ON SEQ ERROR
5174 015044 012737 070707 000000 MOV #070707,@#0 ;INITIALIZE DATA IN LOC. 0
5175 015052 012700 000002 MOV #2,R0 ;INITIALIZE R0 AS POINTER
5176 015056 000261 SEC ;SET C-BIT

```


5177 015060 006140
5178 015062 103406
5179 015064 022737 161617 000000
5180 015072 001002
5181 015074 005700
5182 015076 001404

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

5187 015100
5188 015100 012742 000367
5189 015104 005242
5190 015106 000000

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

5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203

: 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.

5204
5205

: TEST 166 TEST MODE 5 W/ ROTATE INSTRUCTIONS

5206 015110 005212
5207 015112 022712 000166
5208 015116 001021
5209 015120 012737 015172 000000
5210 015126 012700 000002
5211 015132 012767 107070 000032
5212 015140 000241
5213 015142 006150
5214 015144 103006
5215 015146 022737 016160 015172
5216 015154 001002
5217 015156 005700
5218 015160 001405

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
BCC ROT5 ;CHECK C-BIT
CMP #016160,@#ROTX ;CHECK DATA
BNE ROT5 ;BRANCH IF DATA INCORRECT
TST R0 ;CHECK MODE 5 REGISTER
BEQ TS167

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

5219
5220
5221
5222

5223 015162
5224 015162 012742 000370
5225 015166 005242
5226 015170 000000

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

5227
5228 015172 000000
5229
5230
5231
5232

ROTX: 0

: THIS TEST VERIFIES MODE 6 ROTATE INSTRUCTIONS.

5233
5234
5235
5236
5237
5238
5239
5240 015174 005212
5241 015176 022712 000167
5242 015202 001013
5243 015204 012737 125252 015172
5244 015212 000261
5245 015214 006167 177752
5246 015220 103004
5247 015222 022737 052525 015172
5248 015230 001404
5249
5250
5251
5252
5253 015232
5254 015232 012742 000371
5255 015236 005242
5256 015240 000000
5257
5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270 015242 005212
5271 015244 022712 000170
5272 015250 001016
5273 015252 012737 052525 015172
5274 015260 012737 015172 015316
5275 015266 000241
5276 015270 006177 000022
5277 015274 103404
5278 015276 023727 015172 125252
5279 015304 001405
5280
5281
5282
5283
5284 015306
5285 015306 012742 000372
5286 015312 005242
5287 015314 000000
5288

: 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

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

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

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

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

: TEST 170 TEST MODE 7 W/ ROTATE INSTRUCTIONS

TS170: IN (R2) ; UPDATE TEST NUMBER
CMP #170,(R2) ; SEQUENCE ERROR?
BNE ROT7 ; BR TO ERROR HALT ON SEQ ERROR
MOV #52525,@#ROTX ; INITIALIZE DATA
MOV #ROTX,@#ROTXAD ; INITIALIZE ADDRESS POINTER
CLC ; CLEAR C-BIT
ROL @ROTXAD ; TRY ROL W/ MODE 7
BCS ROT7 ; CHECK C-BIT
CMP @#ROTX,#125252 ; CHECK DATA
BEQ TS171

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

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

5289 015316 000000

POTXAD: 0

5290
5291
5292
5293
5294
5295
5296
5297
5298
5299
5300
5301
5302
5303
5304
5305
5306
5307
5308
5309
5310
5311
5312
5313
5314
5315
5316
5317
5318
5319
5320
5321
5322
5323
5324
5325
5326
5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344

015320 005212
015322 022712 000171
015326 001013
015330 012700 177400
015334 000300
015336 100404

015340 012742 000373
015344 005242
015346 000000
015350 022700 000377
015354 001404

015356 012742 000374
015362 005242
015364 000000

015366 005212
015370 022712 000172
015374 001011
015376 012737 125652 000000
015404 005000
015406 000310
015410 022737 125253 000000
015416 001404

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

: TEST 171 TEST MODE 0 W/ SWAB INST.

TS171: INC (R2) ; UPDATE TEST NUMBER
CMP #171,(R2) ; SEQUENCE ERROR?
BNE TS172-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #177400,R0 ; MOVE TEST PATTERN TO R0
SWAB R0 ; TRY SWAB MODE 0
BMI SBO ;

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 773 <====
MOV #373,-(R2) ; MOVE TO MAILBOX # ***** 373 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; SWAB DID NOT SET CC'S CORRECT
SBO: CMP #377,R0 ; CHECK RESULT
BEQ TS172 ;

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

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

: TEST 172 TEST MODE 1 W/ SWAB INST

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

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

```
5345  
5346  
5347  
5348 015420 012742 000375      MOV    #375,-(R2)      ; MOVE TO MAILBOX # ***** 375 *****  
5349 015424 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR  
5350 015426 000000              HALT                    ; RESULT OF SWAB MODE 1 FAILED  
5351  
5352  
5353  
5354  
5355  
5356  
5357  
5358  
5359  
5360  
5361  
5362  
5363  
5364  
5365 015430 005212              TS173: INC    (R2)           ; UPDATE TEST NUMBER  
5366 015432 022712 000173      CMP    #173,(R2)       ; SEQUENCE ERROR?  
5367 015436 001020              BNE    TS174-10        ; BR TO ERROR HALT ON SEQ ERROR  
5368 015440 012737 125152 000000  MOV    #125152,@#0     ; MOVE TEST PATTERN TO LOC. 0  
5369 015446 005000              CLR    R0              ; R0=0  
5370 015450 000320              SWAB   (R0)+           ; TRY SWAB MODE 2  
5371 015452 022737 065252 000000  CMP    #65252,@#0     ; CHECK RESULT  
5372 015460 001404              BEQ    SB2  
5373  
5374  
5375  
5376  
5377 015462 012742 000376      MOV    #376,-(R2)      ; MOVE TO MAILBOX # ***** 376 *****  
5378 015466 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR  
5379 015470 000000              HALT                    ; RESULT OF SWAB MODE 0 FAILED  
5380 015472 162700 000002      SB2:  SUB    #2,R0       ; CHECK EFFECT OF REG.  
5381 015476 001404              BEQ    TS174  
5382  
5383  
5384  
5385  
5386 015500 012742 000377      MOV    #377,-(R2)      ; MOVE TO MAILBOX # ***** 377 *****  
5387 015504 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR  
5388 015506 000000              HALT                    ; REGISTER VALUE INCORRECT  
5389  
5390  
5391  
5392  
5393  
5394  
5395  
5396  
5397  
5398  
5399  
5400
```

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

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

```
TS173: INC    (R2)           ; UPDATE TEST NUMBER  
CMP    #173,(R2)       ; SEQUENCE ERROR?  
BNE    TS174-10        ; BR TO ERROR HALT ON SEQ ERROR  
MOV    #125152,@#0     ; MOVE TEST PATTERN TO LOC. 0  
CLR    R0              ; R0=0  
SWAB   (R0)+           ; TRY SWAB MODE 2  
CMP    #65252,@#0     ; CHECK RESULT  
BEQ    SB2
```

```
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
: CONDITIONAL BRANCH INST. AND <=====  
: REPLACE THE MOVE INSTRUCTION <=====  
: WHICH FOLLOWS W/ 766 <=====  
: MOVE TO MAILBOX # ***** 376 *****
```

```
: SET MSGTYP TO FATAL ERROR  
: RESULT OF SWAB MODE 0 FAILED  
: CHECK EFFECT OF REG.
```

```
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
: CONDITIONAL BRANCH INST. AND <=====  
: REPLACE THE MOVE INSTRUCTION <=====  
: WHICH FOLLOWS W/ 757 <=====  
: MOVE TO MAILBOX # ***** 377 *****
```

```
: SET MSGTYP TO FATAL ERROR  
: REGISTER VALUE INCORRECT  
: OR SEQUENCE ERROR
```

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

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

```
5401
5402 015510 005212
5403 015512 022712 000174
5404 015516 001011
5405 015520 012737 000377 000000
5406 015526 000337 000000
5407 015532 022737 177400 000000
5408 015540 001404
5409
5410
5411
5412
5413 015542 012742 000400
5414 015546 005242
5415 015550 000000
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430 015552 005212
5431 015554 022712 000175
5432 015560 001020
5433 015562 012737 125652 000000
5434 015570 012700 000002
5435 015574 000340
5436 015576 022737 125253 000000
5437 015604 001404
5438
5439
5440
5441
5442 015606 012742 000401
5443 015612 005242
5444 015614 000000
5445 015616 005700
5446 015620 001404
5447
5448
5449
5450
5451 015622 012742 000402
5452 015626 005242
5453 015630 000000
5454
5455
5456

*****
TS174: IN (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
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
      MOV #400,-(R2) ;MOVE TO MAILBOX # ***** 400 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;RESULT OF SWAB INCORRECT
; OR SEQUENCE ERROR

*****
THIS TEST VERIFIES MODE 4 SWAB INSTRUCTIONS. THE DATA
IS MOVED TO LOC 0. R0 IS SET TO 2 AND USED AS THE MODE 4 ADDRESSING
REGISTER. THE DATA IS CHECKED WITH A COMPARE AND P0 IS CHECKED
FOR PROPER DECREMENTING.
*****
TEST 175 TEST MODE 4 W/ SWAB INST
*****
TS175: INC (R2) ;UPDATE TEST NUMBER
      CMP #175,(R2) ;SEQUENCE ERROR?
      BNE TS176-10 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #125652,@#0 ;MOVE TEST PATTERN TO LOC. 0
      MOV #2,R0 ;SET UP REGISTER POINTER
      SWAB -(R0) ;TRY SWAB MODE 4
      CMP #125253,@#0 ;CHECK RESULT
      BEQ SB4
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 765 <====
      MOV #401,-(R2) ;MOVE TO MAILBOX # ***** 401 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;RESULT OF SWAB INCORRECT
SB4: TST R0 ;CHECK EFFECT ON REG.
      BEQ TS176
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 757 <====
      MOV #402,-(R2) ;MOVE TO MAILBOX # ***** 402 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;REGISTER VALUE INCORRECT
; OR SEQUENCE ERROR
```

5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491
5492
5493
5494
5495
5496
5497
5498
5499
5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511
5512

: THIS TEST VERIFIES MODE 5 SWAB INSTRUCTION. THE TEST USES
: TWO LOCATIONS FOLLOWING THE TEST CODE. SB5X HOLDS THE DATA;
: SB5XAD IS A POINTER TO THE DATA LOCATION. THE DATA IS MOVED TO
: SB5X AND R0 IS SET TO TWO PLUS THE ADDRESS OF SB5XAD. FOLLOWING
: THE MODE 5 SWAB SB5X IS CHECKED FOR THE PROPER DATA. R0 IS
: CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.

TEST 176 TEST MODE 5 W/ SWAB INST.

```
TS176:  INC      (R2)           ;UPDATE TEST NUMBER
        CMP      #176,(R2)    ;SEQUENCE ERROR?
        BNE     SB5          ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #SB5XAD+2,R0  ;SET UP POINTER TO WORK LOCATION
        MOV     #125125,SB5X  ;MOVE PATTERN TO WORK LOCATION
        SWAB   @-(R0)         ;TRY SWAB MODE 5
        CMP     #52652,SB5X   ;CHECK RESULT
        BEQ    SB5A

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 765 <====
        MOV     #403,-(R2)    ;MOVE TO MAILBOX # ***** 403 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT                    ;RESULT OF SWAB INCORRECT
SB5A:   CMP     R0,#SB5XAD    ;CHECK RESULT OF REG.
        BEQ    TS177

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 756 <====
SB5:   MOV     #404,-(R2)    ;MOVE TO MAILBOX # ***** 404 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT                    ;REGISTER VALUE INCORRECT
;          OR SEQUENCE ERROR
SB5X:   0                    ;WORK LOCATION
SB5XAD: SB5X
```

: 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.

```

5513 015720 005212 TS177: INC (R2) ;UPDATE TEST NUMBER
5514 015722 022712 000177 CMP #177,(R2) ;SEQUENCE ERROR?
5515 015726 001013 BNE SB6 ;BR TO ERROR HALT ON SEQ ERROR
5516 015730 012767 125125 000030 MOV #125125,SB6X ;MOVE PATTERN TO WORK LOCATION
5517 015736 012700 015760 MOV #SB6X-6,R0 ;MOVE OFFSET POINTER TO R0
5518 015742 000360 000006 SWAB 6(R0) ;TRY SWAB W/ MODE 6
5519 015746 022760 052652 000006 CMP #52652,6(R0) ;CHECK RESULT
5520 015754 001405 BEQ TS200
5521 ;
5522 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5523 ; CONDITIONAL BRANCH INST. AND <====
5524 ; REPLACE THE MOVE INSTRUCTION <====
5525 ; WHICH FOLLOWS W/ 764 <====
5526 015756 012742 000405 SB6: MOV #405,-(R2) ;MOVE TO MAILBOX # ***** 405 *****
5527 015762 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5528 015764 000000 HALT ;RESULT OF SWAB INCORRECT
5529 ; OR SEQUENCE ERROR
5530 015766 000000 SB6X: 0 ;WORK LOCATION
5531
5532
5533
5534
5535
5536
5537
5538
5539
5540
5541
5542
5543
5544
5545
5546

```

```

:*****
:
: THIS TEST VERIFIES MODE 7 SWAB INSTRUCTION. THIS TEST
: USES TWO LOCATIONS FOLLOWING THE TEST CODE: A WORK LOCATION
: (SB7X) AND A POINTER TO THE WORK LOCATION (SB7XAD). DATA IS MOVED
: TO THE WORK LOCATION. R0 IS LOADED WITH 72 LESS THAN THE ADDRESS
: OF THE ADDRESS POINTER. THE DATA IS SWAB'ED USING A MODE 7
: INSTRUCTION WITH AN OFFSET OF +72. THE DATA IS VERIFIED WITH A
: COMPARE.
:
:*****

```

```

5547 015770 005212 TS200: INC (R2) ;UPDATE TEST NUMBER
5548 015772 022712 000200 CMP #200,(R2) ;SEQUENCE ERROR?
5549 015776 001013 BNE SB7 ;BR TO ERROR HALT ON SEQ ERROR
5550 016000 012767 177400 000030 MOV #177400,SB7X ;MOVE PATTERN TO WORK LOCATION
5551 016006 012700 015746 MOV #SB7XAD-72,R0 ;MOVE OFFSET POINTER TO R0
5552 016012 000370 000072 SWAB @72(R0) ;TRY SWAB MODE 7
5553 016016 027027 000072 000377 CMP @72(R0),#377 ;CHECK RESULTS
5554 016024 001406 BEQ TS201
5555 ;
5556 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5557 ; CONDITIONAL BRANCH INST. AND <====
5558 ; REPLACE THE MOVE INSTRUCTION <====
5559 ; WHICH FOLLOWS W/ 764 <====
5560 016026 012742 000406 SB7: MOV #406,-(R2) ;MOVE TO MAILBOX # ***** 406 *****
5561 016032 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5562 016034 000000 HALT ;RESULT OF SWAB INCORRECT
5563 ; OR SEQUENCE ERROR
5564 016036 000000 SB7X: 0 ;WORK LOCATION
5565 016040 016036 SB7XAD: SB7X ;POINTER TO WORK LOCATION
5566
5567
5568

```

5569
5570
5571
5572
5573
5574
5575
5576
5577
5578
5579
5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593
5594
5595
5596
5597
5598
5599
5600
5601
5602
5603
5604
5605
5606
5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624

016042 005212
016044 022712 000201
016050 001150
016052 005067 000326
016056 012700 016136
016062 000110
016064 022700 016066
016070 001404

016072 012742 000407
016076 005242
016100 000000
016102 026727 000276 000001
016110 001404

016112 012742 000410

```
*****
: THIS TEST VERIFIES ALL LEGAL MODES OF THE JMP INSTRUCTION.
: BECAUSE OF THE NATURE OF THE INSTRUCTION UNDER TEST, THIS TEST
: UTILIZES SEVERAL DIFFERENT TECHNIQUES. THE CODE IS NOT EXECUTED
: IN A LINEAR FASHION. THE DIFFERENT MODES ARE EXECUTED IN ORDER
: FROM 1-7; HOWEVER, THE CODE IS ARRANGED SO THAT CONTROL LEAP
: FROGS THRU THE TEST CODE. THE ORDER OF APPEARANCE OF THE CODE
: IS:
:     JMP MODE 1
:     JMP MODE 3
:     JMP MODE 2
:     JMP MODE 4
:     JMP MODE 6
:     JMP MODE 5
:     JMP MODE 7
: AN INTERNAL SEQUENCE TEST (JMPSEQ) IS USED TO INSURE THAT THE
: JUMPS ARE OCCURRING IN THE PROGRAMMED SEQUENCE.
: THE TEST IS MADE UP OF SEVERAL BLOCKS OF CODE. EACH CODE
: BEGINS WITH A LABEL WHICH INDICATES THE MODE BEING EXECUTED IN
: THAT BLOCK. A SIMPLE PROCEDURE IS FOLLOWED IN EACH BLOCK. FOR
: EXAMPLE THE CODE BEGINNING AT JMP3 WILL FIRST COMPARE THE RESULTS
: OF THE PREVIOUS MODE 2 JUMP. (ANY REGISTER CHANGES ARE VERIFIED
: AND THE SEQUENCE CHECK IS MADE). THEN THE REGISTERS ARE SETUP
: FOR A MODE 3 JUMP TO THE NEXT TEST BLOCK (HERE, JMP4), THE SEQUENCE
: CHECKER IS UPDATED AND THE JUMP IS EXECUTED.
: IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
: DETERMINING JUST WHICH MODE FAILED. IF THE SEQUENCE IS CORRECT
: THEN THE ERROR DETECTED WAS A MODE FAILURE (E.G. FAILURE OF THE
: REGISTER TO BE INCREMENTED IN MODE 2 JUMP.)
*****
: TEST 201 TEST THE JMP INSTRUCTION IN ALL MODES
*****
TS201: INC (R2) ;UPDATE TEST NUMBER
      CMP #201,(R2) ;SEQUENCE ERROR?
      BNE JMPCK+6 ;BR TO ERROR HALT ON SEQ ERROR
      CLR JMPSEQ ;ESTABLISH A SEQUENCE CHECKER
      MOV #JMP2,R0 ;SET R0=JUMP TARGET
      JMP (R0) ;TRY JMP MODE 1
JMP3: CMP #.+2,R0 ;CHECK RESULT OF MODE 2 JUMP
      BEQ JMP3A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
      MOV #407,-(R2) ;MOVE TO MAILBOX # ***** 407 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;REGISTER VALUE AFTER JMP MODE 2 INCORRECT
JMP3A: CMP JMPSEQ,#1 ;MAKE SURE JUMPS ARE IN SEQUENCE: JMPSEQ=1?
      BEQ JMP3B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 757 <====
      MOV #410,-(R2) ;MOVE TO MAILBOX # ***** 410 *****
```


5625	016116	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR
5626	016120	000000				HALT			:SHOULD BE HERE FROM JMP MODE 2 ONLY
5627	016122	012700	016134		JMP3B:	MOV	#IIMP4,R0		:POINT R0 TO INDIRECT JMP ADDR.
5628	016126	005267	000252			INC	JMPSEQ		:UPDATE SEQUENCE CHECKER
5629	016132	000130				JMP	@(R0)+		:TRY JMP MODE 3
5630	016134	016166			IIMP4:	JMP4			:ADDRESS INDIRECT JUMP
5631									
5632	016136	005767	000242		JMP2:	TST	JMPSEQ		:CHECK THAT JMPs ARE IN SEQUENCE: JMPSEQ=0?
5633	016142	001404				BEQ	JMP2A		
5634									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5635									: CONDITIONAL BRANCH INST. AND <====
5636									: REPLACE THE MOVE INSTRUCTION <====
5637									: WHICH FOLLOWS W/ 742 <====
5638	016144	012742	000411			MOV	#411,-(R2)		:MOVE TO MAILBOX # ***** 411 *****
5639	016150	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR
5640	016152	000000				HALT			:SHOULD BE HERE FROM JMP MODE 1 ONLY
5641	016154	005267	000224		JMP2A:	INC	JMPSEQ		:UPDATE SEQUENCE CHECKER
5642	016160	012700	016064			MOV	#JMP3,R0		:SET R0=JUMP TARGET
5643	016164	000120				JMP	(R0)+		:TRY A JUMP MODE 2 TO 'JMP3'
5644	016166	022700	016136		JMP4:	CMP	#IIMP4+2,R0		:CHECK RESULT OF REGISTER IN MODE 3 JUMP
5645	016172	001404				BEQ	JMP4A		
5646									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5647									: CONDITIONAL BRANCH INST. AND <====
5648									: REPLACE THE MOVE INSTRUCTION <====
5649									: WHICH FOLLOWS W/ 726 <====
5650	016174	012742	000412			MOV	#412,-(R2)		:MOVE TO MAILBOX # ***** 412 *****
5651	016200	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR
5652	016202	000000				HALT			:REGISTER VALUE AFTER MODE 3 JUMP INCORRECT
5653	016204	022767	000002	000172	JMP4A:	CMP	#2,JMPSEQ		:CHECK JUMP SEQUENCE: JMPSEQ=2?
5654	016212	001404				BEQ	JMP4B		
5655									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5656									: CONDITIONAL BRANCH INST. AND <====
5657									: REPLACE THE MOVE INSTRUCTION <====
5658									: WHICH FOLLOWS W/ 716 <====
5659	016214	012742	000413			MOV	#413,-(R2)		:MOVE TO MAILBOX # ***** 413 *****
5660	016220	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR
5661	016222	000000				HALT			:SHOULD BE ONLY FROM MODE 3 JUMP
5662	016224	012700	016274		JMP4B:	MOV	#JMP5+2,R0		:SET UP POINTER TO JUMP TARGET
5663	016230	005267	000150			INC	JMPSEQ		:UPDATE SEQUENCE CHECKER
5664	016234	000140				JMP	-(R0)		:TRY JUMP MODE 4 TO 'JMP4'
5665									
5666	016236	022767	000004	000140	JMP6:	CMP	#4,JMPSEQ		:CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?
5667	016244	001404				BEQ	JMP6A		
5668									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5669									: CONDITIONAL BRANCH INST. AND <====
5670									: REPLACE THE MOVE INSTRUCTION <====
5671									: WHICH FOLLOWS W/ 701 <====
5672	016246	012742	000414			MOV	#414,-(R2)		:MOVE TO MAILBOX # ***** 414 *****
5673	016252	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR
5674	016254	000000				HALT			:SHOULD BE HERE ONLY FROM MODE 5 JUMP
5675	016256	012700	016724		JMP6A:	MOV	#JMP7+376,R0		:SET UP OFFSET POINTER TO JUMP TARGET
5676	016262	005267	000116			INC	JMPSEQ		:UPDATE JUMP SEQUENCE
5677	016266	000160	177402			JMP	-376(R0)		:TRY MODE 6 JUMP
5678									
5679	016272	022767	000003	000104	JMP5:	CMP	#3,JMPSEQ		:CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?
5680	016300	001404				BEQ	JMP5A		

```
5681 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5682 ; CONDITIONAL BRANCH INST. AND <====
5683 ; REPLACE THE MOVE INSTRUCTION <====
5684 ; WHICH FOLLOWS W/ 663 <====
5685 016302 012742 000415 MOV #415,-(R2) ;MOVE TO MAILBOX # ***** 415 *****
5686 016306 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5687 016310 000000 HALT ;SHOULD ONLY BE HERE FROM MODE 4 JUMP
5688 016312 012700 016326 JMP5A: MOV #I JMP5+2,R0 ;SET UP POINTER TO INDIRECT JUMP ADDR.
5689 016316 005267 000062 INC JMPSEQ ;UPDATE JUMP SEQUENCE
5690 016322 000150 JMP @-(R0) ;TRY JUMP MODE 5 TO 'JMP6'
5691 016324 016236 IJMP5: JMP6 ;INDIRECT ADDRESS POINTER
5692
5693 016326 022767 000005 000050 JMP7: CMP #5,JMPSEQ ;CHECK JUMPS IN SEQUENCE: JMPSEQ=5?
5694 016334 001404 BEQ JMP7A
5695 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5696 ; CONDITIONAL BRANCH INST. AND <====
5697 ; REPLACE THE MOVE INSTRUCTION <====
5698 ; WHICH FOLLOWS W/ 645 <====
5699 016336 012742 000416 MOV #416,-(R2) ;MOVE TO MAILBOX # ***** 416 *****
5700 016342 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5701 016344 000000 HALT ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5702 016346 012700 016372 JMP7A: MOV #I JMP+10,R0 ;SET UP OFFSET POINTER TO INDIRECT ADDR.
5703 016352 005267 000026 INC JMPSEQ ;UPDATE JUMP SEQUENCE
5704 016356 000170 177770 JMP @-10(R0) ;TRY MODE 7 JUMP
5705 016362 016364 IJMP: JMPCK ;INDIRECT ADDRESS
5706
5707 016364 026727 000014 000006 JMPCK: CMP JMPSEQ,#6 ;CHECK JUMPS IN SEQUENCE: JMPSEQ
5708 016372 001405 BEQ TS202
5709 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5710 ; CONDITIONAL BRANCH INST. AND <====
5711 ; REPLACE THE MOVE INSTRUCTION <====
5712 ; WHICH FOLLOWS W/ 626 <====
5713 016374 012742 000417 MOV #417,-(R2) ;MOVE TO MAILBOX # ***** 417 *****
5714 016400 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5715 016402 000000 HALT ;SHOULD ONLY BE HERE FROM MODE 6 JUMP
5716 ; OR SEQUENCE ERROR
5717 016404 000000 JMPSEQ: 0
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
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758
5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775

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
016524 000000

```
*****  
: 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 MSGT TO FATAL ERROR  
HALT ;JSR MODE 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  
HALT ;JSR MODE 3 MALFUNCTIONED
```

```
5776 016526 005237 017034 JSR3B: INC @#JSRSEQ ;UPDATE SEQUENCE CHECKER
5777 016532 004137 016620 JSR R1,@#JSR4 ;TRY JSR MODE 4
5778
5779 016536 005737 017034 JSR2: TST @#JSRSEQ ;CHECK SEQUENCE: JSRSEQ=0?
5780 016542 001011 BNE JSR2A ;BRANCH IF OUT OF SEQUENCE
5781 016544 020127 016446 CMP R1,#JSR1A ;PROPER PC SAVED?
5782 016550 001006 BNE JSR2A ;BRANCH IF PC WRONG
5783 016552 022706 000776 CMP #STBOT-2,R6 ;R6 DECREMENT?
5784 016556 001003 BNE JSR2A ;BRANCH IF R6 IS INCORRECT
5785 016560 021627 177777 CMP (R6), #-1 ;REGISTER SAVED?
5786 016564 001404 BEQ JSR2B
5787
5788 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5789 ; CONDITIONAL BRANCH INST. AND <====
5790 ; REPLACE THE MOVE INSTRUCTION <====
5791 ; WHICH FOLLOWS W/ 713 <====
5791 016566 JSR2A:
5792 016566 012742 000422 MOV #422,-(R2) ;MOVE TO MAILBOX # ***** 422 *****
5793 016572 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5794 016574 000000 HALT ;JSR MODE 1 MALFUNCTIONED
5795 016576 012706 001000 JSR2B: MOV #STBOT,R6 ;INITIALIZE R6
5796 016602 012701 125252 MOV #125252,R1 ;INITIALIZE R1
5797 016606 005237 017034 INC @#JSRSEQ ;UPDATE SEQUENCE CHECKER
5798 016612 012700 016456 MOV #JSR3,R0 ;SET TARGET ADDRESS
5799 016616 004120 JSR R1,(R0)+ ;TRY JSR MODE 2
5800
5801 016620 022737 000002 017034 JSR4: CMP #2,@#JSRSEQ ;CHECK SEQUENCE: JSRSEQ=2?
5802 016626 001003 BNE JSR4A ;BRANCH IF OUT OF SEQUENCE
5803 016630 022701 016536 CMP #JSR2,R1 ;PROPER PC SAVED?
5804 016634 001404 BEQ JSR4B
5805
5806 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5807 ; CONDITIONAL BRANCH INST. AND <====
5808 ; REPLACE THE MOVE INSTRUCTION <====
5809 ; WHICH FOLLOWS W/ 667 <====
5809 016636 JSR4A:
5810 016636 012742 000423 MOV #423,-(R2) ;MOVE TO MAILBOX # ***** 423 *****
5811 016642 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5812 016644 000000 HALT ;JSR MODE 3 MALFUNCTIONED
5813 016646 000000 JSR4B: INC @#JSRSEQ ;UPDATE SEQUENCE CHECKER
5814 016652 000000 MOV #JSR5+2,R0 ;SET TARGET ADDRESS
5815 016656 004140 JSR R1,-(R0) ;TRY JSR MODE 4
5816
5817 016660 022767 000004 000146 JSR6: CMP #4,JSRSEQ ;CHECK SEQUENCE: JSRSEQ=4?
5818 016666 001006 BNE JSR6A ;BRANCH IF OUT OF SEQUENCE
5819 016670 022701 016772 CMP #JSR7,R1 ;PROPER PC SAVED?
5820 016674 001003 BNE JSR6A ;BRANCH IF PC WRONG
5821 016676 022700 017030 CMP #JSR6AD,R0 ;MODE 5 REGISTER CORRECT?
5822 016702 001404 BEQ JSR6B
5823
5824 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5825 ; CONDITIONAL BRANCH INST. AND <====
5826 ; REPLACE THE MOVE INSTRUCTION <====
5827 ; WHICH FOLLOWS W/ 644 <====
5827 016704 JSR6A:
5828 016704 012742 000424 MOV #424,-(R2) ;MOVE TO MAILBOX # ***** 424 *****
5829 016710 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5830 016712 000000 HALT ;JSR MODE 5 FAILED
5831 016714 005237 017034 JSR6B: INC @#JSRSEQ ;UPDATE SEQUENCE CHECKER
```

```

CJKDB-C DCF11-AA CPU DIAG. MACY11 30A(1052) 07-MAR-80 12:18 D 9 PAGE 107
CJKDBC.P11 07-MAR-80 12:17 T202 TEST JSR INSTRUCTION W/ ALL MODES SEQ 0107

5832 016720 004167 000046 JSR R1,JSR7 ;TRY JSR MODE 6
5833 016724 022767 000003 000102 JSR5: CMP #3,JSRSEQ ;CHECK SEQUENCE: JSRSEQ=3?
5834 016732 001006 BNE JSR5A ;BRANCH IF OUT OF SEQUENCE
5835 016734 022701 016660 CMP #JSR6,R1 ;PROPER PC SAVED?
5836 016740 001003 BNE JSR5A ;BRANCH IF PC WRONG
5837 016742 022700 016724 CMP #JSR5,R0 ;CHECK MODE 4 REGISTER
5838 016746 001404 BEQ JSR5B
5839 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5840 ; CONDITIONAL BRANCH INST. AND <====
5841 ; REPLACE THE MOVE INSTRUCTION <====
5842 ; WHICH FOLLOWS W/ 622 <====
5843 016750 JSR5A:
5844 016750 012742 000425 MOV #425,-(R2) ;MOVE TO MAILBOX # ***** 425 *****
5845 016754 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5846 016756 000000 HALT ;JSR MODE 4 MALFUNCTIONED
5847 016760 005237 017034 JSR5B: INC @#JSRSEQ ;UPDATE SEQUENCE CHECKER
5848 016764 012700 017032 MOV #JSR6AD+2,R0 ;POINT R0 TO TARGET ADDRESS
5849 016770 004150 JSR R1,@-(R0) ;TRY JSR MODE 5
5850
5851 016772 022737 000005 017034 JSR7: CMP #5,@#JSRSEQ ;CHECK SEQUENCE: JSRSEQ=5?
5852 017000 001003 BNE JSR7A ;BRANCH IF OUT OF SEQUENCE
5853 017002 022701 016724 CMP #JSR5,R1 ;PROPER PC SAVED?
5854 017006 001404 BEQ JSR7B
5855 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5856 ; CONDITIONAL BRANCH INST. AND <====
5857 ; REPLACE THE MOVE INSTRUCTION <====
5858 ; WHICH FOLLOWS W/ 602 <====
5859 017010 JSR7A:
5860 017010 012742 000426 MOV #426,-(R2) ;MOVE TO MAILBOX # ***** 426 *****
5861 017014 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5862 017016 000000 HALT ;JSR MODE 6 FAILED
5863 017020 005237 017034 JSR7B: INC @#JSRSEQ ;UPDATE SEQUENCE CHECKER
5864 017024 004177 000002 JSR R1,@JSRCKAD ;TRY JSR MODE 7
5865
5866 017030 016660 JSR6AD: JSR6 ;MODE 5 TARGET ADDRESS
5867 017032 017036 JSRCKAD: JSRCK ;MODE 7 TARGET ADDRESS
5868 017034 000000 JSRSEQ: 0 ;SEQUENCE CHECKER
5869
5870 017036 022767 000006 177770 JSRCK: CMP #6,JSRSEQ ;CHECK SEQUENCE: JSRSEQ=6?
5871 017044 001003 BNE JSRCK1 ;BRANCH IF OUT OF SEQUENCE
5872 017046 022701 017030 CMP #JSR6AD,R1 ;PROPER PC SAVED?
5873 017052 001404 BEQ TS203
5874 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5875 ; CONDITIONAL BRANCH INST. AND <====
5876 ; REPLACE THE MOVE INSTRUCTION <====
5877 ; WHICH FOLLOWS W/ 560 <====
5878 017054 JSRCK1:
5879 017054 012742 000427 MOV #427,-(R2) ;MOVE TO MAILBOX # ***** 427 *****
5880 017060 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5881 017062 000000 HALT ;JSR MODE 7 MALFUNCTIONED
5882 ; OR SEQUENCE ERROR
5883
5884
5885
5886 ;*****
5887 ;

```

```

5888
5889
5890
5891
5892
5893
5894
5895
5896 017064 005212
5897 017066 022712 000203
5898 017072 001016
5899 017074 012706 001000
5900 017100 012746 052525
5901 017104 012700 017122
5902 017110 000200
5903
5904
5905 017112 012742 000430
5906 017116 005242
5907 017120 000000
5908 017122 022700 052525
5909 017126 001404
5910
5911
5912
5913
5914 017130 012742 000431
5915 017134 005242
5916 017136 000000
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936 017140 005212
5937 017142 022712 000204
5938 017146 001022
5939 017150 000277
5940 017152 000251
5941 017154 012700 100000
5942 017160 101402
5943 017162 102401

```

```

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

```

```

:*****
:
: THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF A GROUP
: OF FOUR INSTRUCTIONS. THE GROUP CONSISTS OF THE INSTRUCTIONS:
: MOV, BIC, BIT, AND BIS. THESE INSTRUCTIONS ARE SIMILAR IN THE
: WAY THEY EFFECT THE C AND V BITS. THEY ALL LEAVE THE V-BIT
: CLEAR AND THE C-BIT UNAFFECTED.
: THE TEST PROCEDURE IS AS FOLLOWS: THE N, Z, AND V BITS
: ARE LOADED WITH THE COMPLEMENT OF THE EXPECTED RESULTS, THE C-BIT
: IS LOADED WITH THE DESIRED RESULT. THE INSTRUCTION IS EXECUTED
: WITH DIFFERENT DATA PATTERNS AND THE RESULTS ARE VERIFIED WITH
: A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS. THE DATA IS CHOSEN
: TO PRODUCT ALL POSSIBLE COMBINATIONS OF THE C AND V BITS.
:*****

```

```

:*****
:TEST 204 TEST MOV INSTRUCTION
:*****
TS204: INC (R2) ;UPDATE TEST NUMBER
CMP #204,(R2) ;SEQUENCE ERROR?
BNE TS205-10 ;BR TO ERROR HALT ON SEQ ERROR
SCC ;CC-0110
+CLN!CLC
MOV #100000,R0 ;CC-1000
BLOS MOV1
BVS MOV1

```

```
5944 017164 100404      BMI      MOV2
5945
5946                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5947                    ;          CONDITIONAL BRANCH INST. AND <====
5948                    ;          REPLACE THE MOVE INSTRUCTION <====
5949                    ;          WHICH FOLLOWS W/ 770 <====
5949 017166      MOV1:
5950 017166 012742 000432      MOV      #432,-(R2)      ;MOVE TO MAILBOX # ***** 432 *****
5951 017172 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5952 017174 000000      HALT                    ;MOV DID NOT SET CC'S CORRECTLY
5953
5954 017176 000277      MOV2:      SCC                    ;CC=1011
5955 017200 000244      CLZ
5956 017202 012700 000000      MOV      #0,R0          ;CC=0101
5957 017206 101002      BHI      MOV3          ;C OR Z = 0?
5958 017210 102401      BVS      MOV3          ;V=1?
5959 017212 100004      BPL      TS205
5960
5961                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5962                    ;          CONDITIONAL BRANCH INST. AND <====
5963                    ;          REPLACE THE MOVE INSTRUCTION <====
5964                    ;          WHICH FOLLOWS W/ 755 <====
5964 017214      MOV3:
5965 017214 012742 000433      MOV      #433,-(R2)      ;MOVE TO MAILBOX # ***** 433 *****
5966 017220 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5967 017222 000000      HALT                    ;MOV DID NOT SET CC'S CORRECTLY
5968                    ; OR SEQUENCE ERROR
5969
5970                    ;*****
5971                    ;TEST 205 TEST BIT INSTRUCTION
5972                    ;*****
5972 017224 005212      TS205:      INC      (R2)          ;UPDATE TEST NUMBER
5973 017226 022712 000205      CMP      #205,(R2)      ;SEQUENCE ERROR?
5974 017232 001024      BNE      TS206-10      ;BR TO ERROR HA T ON SEQ ERROR
5975 017234 012700 100001      MOV      #100001,R0
5976 017240 000277      SCC                    ;CC=0110
5977 017242 000251      +CLN!CLC
5978 017244 032700 100000      BIT      #100000,R0      ;CC=1000
5979 017250 101402      BLOS     BIT1
5980 017252 102401      BVS      BIT1
5981 017254 100404      BMI      BIT2
5982
5983                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5984                    ;          CONDITIONAL BRANCH INST. AND <====
5985                    ;          REPLACE THE MOVE INSTRUCTION <====
5986                    ;          WHICH FOLLOWS W/ 766 <====
5986 017256      BIT1:
5987 017256 012742 000434      MOV      #434,-(R2)      ;MOVE TO MAILBOX # ***** 434 *****
5988 017262 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
5989 017264 000000      HALT                    ;BIT DID NOT SET CC'S CORRECTLY
5990
5991 017266 000277      BIT2:      SCC                    ;CC=1011
5992 017270 000244      CLZ
5993 017272 032700 077776      BIT      #77776,R0      ;CC=0101
5994 017276 101002      BHI      BIT3
5995 017300 102401      BVS      BIT3
5996 017302 100004      BPL      TS206
5997
5998                    ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5999                    ;          CONDITIONAL BRANCH INST. AND <====
                    ;          REPLACE THE MOVE INSTRUCTION <====
```

```
6000
6001 017304 BIT3:                                : WHICH FOLLOWS W/ 753 <===-
6002 017304 012742 000435 MOV #435,-(R2)      ;MOVE TO MAILBOX # ***** 435 *****
6003 017310 005242 INC -(R2)                   ;SET MSGTYP TO FATAL ERROR
6004 017312 000000 HALT                          ;BIT DID NOT SET CC'S CORRECTLY
6005                                          ; OR SEQUENCE ERROR
6006
6007 ;*****
6008 ;TEST 206 TEST BIC INSTRUCTION
6009 ;*****
6009 017314 005212 TS206: INC (R2)                ;UPDATE TEST NUMBER
6010 017316 022712 000206 CMP #206,(R2)        ;SEQUENCE ERROR?
6011 017322 001024 BNE TS207-10                ;BR TO ERROR HALT ON SEQ ERROR
6012 017324 012700 177777 MOV #177777,R0
6013 017330 000277 SCC                          ;CC=0110
6014 017332 000251 +CLN:CLC
6015 017334 042700 077777 BIC #77777,R0        ;CC=1000
6016 017340 101402 BLOS BIC1
6017 017342 102401 BVS BIC1
6018 017344 100404 BMI BIC2
6019
6020 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
6021 ; CONDITIONAL BRANCH INST. AND <---
6022 ; REPLACE THE MOVE INSTRUCTION <= -
6023 ; WHICH FOLLOWS W/ 766 <====
6023 017346 BIC1: MOV #436,-(R2)                ;MOVE TO MAILBOX # ***** 436 *****
6024 017346 012742 000436 INC -(R2)             ;SET MSGTYP TO FATAL ERROR
6025 017352 005242 HALT                          ;BIC DID NOT SET CC'S CORRECTLY
6026 017354 000000 BIC2: SCC                      ;CC=1011
6027 017356 000277 CLZ
6028 017360 000244 BIC #100000,R0                ;CC=0101
6029 017362 042700 100000 BHI BIC3
6030 017366 101002 BVS BIC3
6031 017370 102401 BPL TS207
6032 017372 100004
6033
6034 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6035 ; CONDITIONAL BRANCH INST. AND <====
6036 ; REPLACE THE MOVE INSTRUCTION <====
6037 ; WHICH FOLLOWS W/ 753 <====
6037 017374 BIC3: MOV #437,-(R2)                ;MOVE TO MAILBOX # ***** 437 *****
6038 017374 012742 000437 INC -(R2)             ;SET MSGTYP TO FATAL ERROR
6039 017400 005242 HALT                          ;BIC DID NOT SET CC'S CORRECTLY
6040 017402 000000 ; OR SEQUENCE ERROR
6041
6042 ;*****
6043 ;TEST 207 TEST BIS INSTRUCTION
6044 ;*****
6045 017404 005212 TS207: INC (R2)                ;UPDATE TEST NUMBER
6046 017406 022712 000207 CMP #207,(R2)        ;SEQUENCE ERROR?
6047 017412 001025 BNE TS210-10                ;BR TO ERROR HALT ON SEQ ERROR
6048 017414 005000 CLR R0
6049 017416 000277 SCC                          ;R0=0
6050 017420 000251 +CLN:CLC                      ;CC=1010
6051 017422 052700 000000 BIS #0,R0            ;CC=0100 R0=0
6052 017426 103403 BCS BIS1
6053 017430 102402 BVS BIS1
6054 017432 100401 BMI BIS1
6055 017434 001404 BEQ BIS2
```



```

6056                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6057                                     ;                               <====
6058                                     ;                               <====
6059                                     ;                               <====
6060 017436                               BIS1: MOV #440,-(R2) ;MOVE TO MAILBOX # ***** 440 *****
6061 017436 012742 000440                 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6062 017442 005242                         HALT ;BIS DID NOT SET CC'S CORRECTLY
6063 017444 000000                         BIS2: SCC ;CC=0111
6064 017446 000277                         CLN
6065 017450 000250                         BIS #177777,R0 ;CC=1001
6066 017452 052700 177777                 BCC BIS3
6067 017456 103003                         BVS BIS3
6068 017460 102402                         BEQ BIS3
6069 017462 001401                         BMI TS210
6070 017464 100404
6071                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6072                                     ;                               <====
6073                                     ;                               <====
6074                                     ;                               <====
6075 017466                               BIS3: MOV #441,-(R2) ;MOVE TO MAILBOX # ***** 441 *****
6076 017466 012742 000441                 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6077 017472 005242                         HALT ;BIS DID NOT SET CC'S CORRECTLY
6078 017474 000000                         ; OR SEQUENCE ERROR
6079
6080
6081
6082
6083
6084
6085
6086
6087
6088
6089
6090
6091
6092
6093
6094
6095

```

```

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

```

```

6096 017476 005212                               :TEST 210 TEST INC INSTRUCTION
6097 017500 022712 000210                 TS210: INC (R2) ;UPDATE TEST NUMBER
6098 017504 001037                               CMP #210,(R2) ;SEQUENCE ERROR?
6099 017506 012700 077777                 BNE TS211-10 ;BR TO ERROR HALT ON SEQ ERROR
6100 017512 000257                               MOV #077777,R0 ;R0=077777
6101 017514 000264                               CCC ;CC=0100
6102 017516 005200                               SEZ
6103 017520 101402                               INC R0 ;CC=1010 R0 10000
6104 017522 100001                               BLOS INC1
6105 017524 102404                               BPL INC1
6106                               BVS INC2
6107                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6108                                     ;                               <====
6109                                     ;                               <====
6110 017526                               INC1: MOV #442,-(R2) ;MOVE TO MAILBOX # ***** 442 *****
6111 017526 012742 000442

```



```

6168 017654 000261      DEC2:  SEC                ;CC=1011
6169 017656 000244      CLZ
6170 017660 005300      DEC      R0                ;CC=0101  R0=0
6171 017662 101002      BHI      DEC3
6172 017664 100401      BMI      DEC3
6173 017666 102004      BVC      DEC4
6174
6175      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6176      ;          CONDITIONAL BRANCH INST. AND <----
6177      ;          REPLACE THE MOVE INSTRUCTION <====
6178      ;          WHICH FOLLOWS W/ 755 <----
6178 017670      DEC3:
6179 017670 012742 000446  MOV      #446,-(R2)      ;MOVE TO MAILBOX # ***** 446 *****
6180 017674 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6181 017676 000000      HALT
6182 017700 000277      DEC4:  SCC                ;DEC DID NOT SET CC'S CORRECTLY
6183 017702 000251      +CLN!CLC              ;CC=0110
6184 017704 005300      DEC      R0                ;CC=1000  R0=177777
6185 017706 101402      BLOS     DEC5
6186 017710 102401      BVS      DEC5
6187 017712 100404      BMI      DEC6
6188
6189      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6190      ;          CONDITIONAL BRANCH INST. AND <----
6191      ;          REPLACE THE MOVE INSTRUCTION <----
6192      ;          WHICH FOLLOWS W/ 743 <----
6192 017714      DEC5:
6193 017714 012742 000447  MOV      #447,-(R2)      ;MOVE TO MAILBOX # ***** 447 *****
6194 017720 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6195 017722 000000      HALT
6196 017724 042700 077777  DEC6:  BIC      #77777,R0    ;DEC DID NOT SET CC'S CORRECTLY
6197 017730 000277      SCC                ;R0=100000
6198 017732 000252      +CLN.CLV              ;CC=0101
6199 017734 005300      DEC      R0                ;CC=1011  R0=77777
6200 017736 100403      BMI      DEC7
6201 017740 001402      BEQ      DEC7
6202 017742 102001      BVC      DEC7
6203 017744 103404      BCS      TS212
6204
6205      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6206      ;          CONDITIONAL BRANCH INST. AND <====
6207      ;          REPLACE THE MOVE INSTRUCTION <----
6208      ;          WHICH FOLLOWS W/ 726 <====
6208 017746      DEC7:
6209 017746 012742 000450  MOV      #450,-(R2)      ;MOVE TO MAILBOX # ***** 450 *****
6210 017752 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6211 017754 000000      HALT                ;DEC DID NOT SET CC'S CORRECTLY
6212
6213
6214
6215
6216
6217
6218
6219
6220
6221
6222
6223
;*****
;
; THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE CLR,
; TST, AND SWAB INSTRUCTIONS. THESE THREE INSTRUCTIONS ALL LEAVE
; THE C AND V BITS CLEARED. AGAIN, THE CONDITION CODES ARE PRESET,
; THE INSTRUCTION EXECUTED AND THE RESULTS CHECKED WITH CONDITIONAL
; BRANCH INSTRUCTIONS. THE PROCEDURE IS REPEATED TO PRODUCE OTHER
; COMBINATIONS OF CONDITION CODES.
    
```

6224
6225
6226
6227
6228 017756 005212
6229 017760 022712 000212
6230 017764 001007
6231 017766 000277
6232 017770 000244
6233 017772 005000
6234 017774 100403
6235 017776 102402
6236 020000 103401
6237 020002 007404
6238
6239
6240
6241
6242 020004
6243 020004 012742 000451
6244 020010 005242
6245 020012 000000
6246
6247
6248
6249
6250
6251 020014 005212
6252 020016 022712 000213
6253 020022 001022
6254 020024 000277
6255 020026 000244
6256 020030 005700
6257 020032 100403
6258 020034 102402
6259 020036 103401
6260 020040 001404
6261
6262
6263
6264
6265 020042
6266 020042 012742 000452
6267 020046 005242
6268 020050 000000
6269 020052 005300
6270 020054 000277
6271 020056 000250
6272 020060 005700
6273 020062 101402
6274 020064 102401
6275 020066 100404
6276
6277
6278
6279

```
*****
: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
        CLZ
        CLR      R0          ;CC=0100  R0=0
        BMI     CLR1
        BVS     CLR1
        BCS     CLR1
        BEQ     TS213

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

CLR1:   MOV      #451,-(R2)    ;MOVE TO MAILBOX # ***** 451 *****
        INC      -(R2)
        HALT
        ;SET MSGTYP TO FATAL ERROR
        ;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
        CLZ
        TST     R0          ;CC=0100
        BMI     TEST1
        BVS     TEST1
        BCS     TEST1
        BEQ     TEST2

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

TEST1:  MOV      #452,-(R2)    ;MOVE TO MAILBOX # ***** 452 *****
        INC      -(R2)
        HALT
        ;SET MSGTYP TO FATAL ERROR
        ;TEST DID NOT SET CC'S CORRECTLY

TEST2:  DEC      R0          ;MAKE R0 NEGATIVE
        SCC
        CLN
        TST     R0          ;CC=1000
        BLOS   TEST3
        BVS     TEST3
        BMI     TS214

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

6280 020070
6281 020070 012742 000453
6282 020074 005242
6283 020076 000000

TEST3:
MOV #453,-(R2) ;MOVE TO MAILBOX # ***** 453 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR

6284
6285
6286
6287
6288 020100 005212
6289 020102 022712 000214
6290 020106 001023
6291 020110 012700 170000
6292 020114 000277
6293 020116 000250
6294 020120 000300
6295 020122 101402
6296 020124 102401
6297 020126 100404

:TEST 214 TEST SWAB INSTRUCTION

TS214: INC (R2) ;UPDATE TEST NUMBER
CMP #214,(R2) ;SEQUENCE ERROR?
BNE TS215-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #170000,R0 ;R0=170000
SCC ;CC=0111
CLN
SWAB R0 ;CC=1000 R0=360
BLOS SWB1
BVS SWB1
BMI SWB2

6298
6299
6300
6301
6302 020130
6303 020130 012742 000454
6304 020134 005242
6305 020136 000000
6306 020140 000277
6307 020142 000244
6308 020144 000300
6309 020146 102403
6310 020150 103402
6311 020152 100401
6312 020154 001404

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
SWB1: MOV #454,-(R2) ;MOVE TO MAILBOX # ***** 454 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;SWAB DID NOT SET CC'S CORRECTLY
SWB2: SCC ;CC=1011
CLZ
SWAB R0 ;CC=0100 R0=170000
BVS SWB3
BCS SWB3
BMI SWB3
BEQ TS215

6313
6314
6315
6316
6317 020156
6318 020156 012742 000455
6319 020162 005242
6320 020164 000000

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 754 <====
SWB3: MOV #455,-(R2) ;MOVE TO MAILBOX # ***** 455 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;

6321
6322
6323
6324
6325
6326
6327
6328
6329
6330
6331
6332
6333
6334
6335

: 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

```
6336 020166 005212 TS215: INC (R2) ;UPDATE TEST NUMBER
6337 020170 022712 000215 CMP #215,(R2) ;SEQUENCE ERROR?
6338 020174 001062 BNE TS216-10 ;BR TO ERROR HALT ON SEQ ERROR
6339 020176 012700 040000 MOV #40000,R0 ;RO=40000
6340 020202 000277 SCC ;CC=1111
6341 020204 062700 030000 ADD #30000,R0 ;CC=0000 RO=70000
6342 020210 101402 BLOS ADD1
6343 020212 102401 BVS ADD1
6344 020214 100004 BPL ADD2
6345 ;
6346 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6347 ; CONDITIONAL BRANCH INST. AND <====
6348 ; REPLACE THE MOVE INSTRUCTION <====
6349 ; WHICH FOLLOWS W/ 767 <====
6349 020216 ADD1:
6350 020216 012742 000456 MOV #456,-(R2) ;MOVE TO MAILBOX # ***** 456 *****
6351 020222 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6352 020224 000000 HALT ;ADD DID NOT SET CC'S CORRECTLY
6353 020226 000264 ADD2: SEZ ;CC=0100
6354 ;
6355 020230 062700 010000 ADD #10000,R0 ;CC=1010 40-100000
6356 020234 101402 BLOS ADD3
6357 020236 102001 BVC ADD3
6358 020240 100404 BMI ADD4
6359 ;
6360 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6361 ; CONDITIONAL BRANCH INST. AND <====
6362 ; REPLACE THE MOVE INSTRUCTION <====
6363 ; WHICH FOLLOWS W/ 755 <====
6363 020242 ADD3:
6364 020242 012742 000457 MOV #457,-(R2) ;MOVE TO MAILBOX # ***** 457 *****
6365 020246 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6366 020250 000000 HALT ;ADD DID NOT SET CC'S CORRECTLY
6367 020252 000257 ADD4: CCC ;CC=1000
6368 020254 000270 SEN
6369 020256 062700 100000 ADD #100000,R0 ;CC=0111 RO=0
6370 020262 101002 BHI ADD5
6371 020264 102001 BVC ADD5
6372 020266 100004 BPL ADD6
6373 ;
6374 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6375 ; CONDITIONAL BRANCH INST. AND <====
6376 ; REPLACE THE MOVE INSTRUCTION <====
6377 ; WHICH FOLLOWS W/ 742 <--
6377 020270 ADD5:
6378 020270 012742 000460 MOV #460,-(R2) ;MOVE TO MAILBOX # ***** 460 *****
6379 020274 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6380 020276 000000 HALT ;ADD DID NOT SET CC'S CORRECTLY
6381 020300 062700 177777 ADD6: ADD #177777,R0 ;CC 1000 RO=177777
6382 020304 101402 BLOS ADD7
6383 020306 102401 BVS ADD7
6384 020310 100404 BMI ADD8
6385 ;
6386 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
6387 ; CONDITIONAL BRANCH INST. AND < -
6388 ; REPLACE THE MOVE INSTRUCTION < --
6389 ; WHICH FOLLOWS W/ 731 <
6389 020312 ADD7:
6390 020312 012742 000461 MOV #461,-(R2) ;MOVE TO MAILBOX # ***** 461 *****
6391 020316 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
```

```
6392 020320 000000
6393 020322 000277
6394 020324 000245
6395 020326 062700 000001
6396 020332 102403
6397 020334 103002
6398 020336 100401
6399 020340 001404
6400
6401
6402
6403
6404 020342
6405 020342 012742 000462
6406 020346 005242
6407 020350 000000
6408
6409
6410
6411
6412
6413 020352 005212
6414 020354 022712 000216
6415 020360 001037
6416 020362 012700 077777
6417 020366 000277
6418 020370 000252
6419 020372 005500
6420 020374 101402
6421 020376 102001
6422 020400 100404
6423
6424
6425
6426
6427 020402
6428 020402 012742 000463
6429 020406 005242
6430 020410 000000
6431 020412 052700 077777
6432 020416 000277
6433 020420 000244
6434 020422 005500
6435 020424 101002
6436 020426 102401
6437 020430 100004
6438
6439
6440
6441
6442 020432
6443 020432 012742 000464
6444 020436 005242
6445 020440 000000
6446 020442 000277
6447 020444 000245

ADD8:  HALT          ;ADD DID NOT SET CC'S CORRECTLY
        SCC          ;CC=1010
        +CLC!CLZ
        ADD #1,R0    ;CC=0101 R=0
        BVS ADD9
        BCC ADD9
        BMI ADD9
        BEQ TS216

        ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
        ; CONDITIONAL BRANCH INST. AND <====
        ; REPLACE THE MOVE INSTRUCTION <====
        ; WHICH FOLLOWS W/ 715 <====

ADD9:  MOV #462,-(R2) ;MOVE TO MAILBOX # ***** 462 *****
        INC -(R2)    ;SET MSGTYP TO FATAL ERROR
        HALT        ;ADD DID NOT SET CC'S CORRECTLY
                ; OR SEQUENCE ERROR

;*****
;TEST 216 TEST ADC INSTRUCTION
;*****
TS216:  INC (R2)      ;UPDATE TEST NUMBER
        CMP #216,(R2) ;SEQUENCE ERROR?
        BNE TS217-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #077777,R0
        SCC          ;CC=0101
        +CLN!CLV
        ADC R0       ;CC=1010
        BLOS ADC1
        BVC ADC1
        BMI ADC2

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

ADC1:  MOV #463,-(R2) ;MOVE TO MAILBOX # ***** 463 *****
        INC -(R2)    ;SET MSGTYP TO FATAL ERROR
        HALT        ;ADC DID NOT SET CC'S CORRECTLY

ADC2:  BIS #77777,R0
        SCC          ;CC=1011
        CLZ
        ADC R0       ;CC=0101 R0=0
        BHI ADC3
        BVS ADC3
        BPL ADC4

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

ADC3:  MOV #464,-(R2) ;MOVE TO MAILBOX # ***** 464 *****
        INC -(R2)    ;SET MSGTYP TO FATAL ERROR
        HALT        ;ADC DID NOT SET CC'S CORRECTLY

ADC4:  SCC
        +CLZ!CLL    ;CC=1010
```

```

6448 020446 005500          ADC      R0          ;CC=0100
6449 020450 102403          BVS      ADC5
6450 020452 103402          BCS      ADC5
6451 020454 100401          BMI      ADC5
6452 020456 001404          BEQ      TS217
6453
6454                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
6455                      ;          CONDITIONAL BRANCH INST. AND  <====
6456                      ;          REPLACE THE MOVE INSTRUCTION  <====
6457                      ;          WHICH FOLLOWS W/ 740  <====
6457 020460          ADC5:
6458 020460 012742 000465      MOV      #465,-(R2)   ;MOVE TO MAILBOX # ***** 465 *****
6459 020464 005242          INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
6460 020466 000000          HALT             ;ADC DID NOT SET CC'S CORRECTLY
6461                      ; OR SEQUENCE ERROR
6462
6463
6464
6465
6466
6467
6468
6469
6470
6471
6472
6473
6474
6475
6476

```

```

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

```

```

6476
6477 020470 005212          TS217: INC      (R2)          ;UPDATE TEST NUMBER
6478 020472 022712 000217      CMP      #217,(R2)   ;SEQUENCE ERROR?
6479 020476 001042          BNE      TS220-10    ;BR TO ERROR HALT ON SEQ ERROR
6480 020500 012700 000001      MOV      #1,R0
6481 020504 000277          SCC
6482 020506 000251          +CLN!CLC          ;CC=0110
6483 020510 005400          NEG      R0          ;CC=1001 R0-177777
6484 020512 103003          BCC      NEG1
6485 020514 102402          BVS      NEG1
6486 020516 001401          BEQ      NEG1
6487 020520 100404          BMI      NEG2
6488
6489                      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
6490                      ;          CONDITIONAL BRANCH INST. AND  <====
6491                      ;          REPLACE THE MOVE INSTRUCTION  <====
6492                      ;          WHICH FOLLOWS W/ 766  <====
6492 020522          NEG1:
6493 020522 012742 000466      MOV      #466,-(R2)   ;MOVE TO MAILBOX # ***** 466 *****
6494 020526 005242          INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
6495 020530 000000          HALT             ;NEG DID NOT SET CC'S CORRECTLY
6496 020532 042700 077777      NEG2: BIC      #77777,R0
6497 020536 000257          CCC
6498 020540 000264          SEZ
6499 020542 005400          NEG      R0          ;CC=0100
6500 020544 102003          BVC      NEG3
6501 020546 103002          BCC      NEG3
6502 020550 001401          BEQ      NEG3
6503 020552 100404          BMI      NEG4

```



```
6504 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6505 ; CONDITIONAL BRANCH INST. AND <====
6506 ; REPLACE THE MOVE INSTRUCTION <====
6507 ; WHICH FOLLOWS W/ 751 <====
6508 020554 NEG3:
6509 020554 012742 000467 MOV #467,-(R2) ;MOVE TO MAILBOX # ***** 467 *****
6510 020560 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6511 020562 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
6512 020564 005000 NEG4: CLR R0 ;CC-1011
6513 020566 000277 SCC ;CC=0100 R0=0
6514 020570 000244 CLZ
6515 020572 005400 NEG R0
6516 020574 102403 BVS NEG5
6517 020576 103402 BCS NEG5
6518 020600 001001 BNE NEG5
6519 020602 100004 BPL TS220
6520 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <= -
6521 ; CONDITIONAL BRANCH INST. AND <----
6522 ; REPLACE THE MOVE INSTRUCTION <----
6523 ; WHICH FOLLOWS W/ 735 <====
6524 020604 NEG5:
6525 020604 012742 000470 MOV #470,-(R2) ;MOVE TO MAILBOX # ***** 470 *****
6526 020610 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6527 020612 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
6528 ; OR SEQUENCE ERROR
6529
6530 ;*****
6531 ;TEST 220 TEST CMP INSTRUCTION
6532 ;*****
6533 020614 005212 TS220: INC (R2) ;UPDATE TEST NUMBER
6534 020616 022712 000220 CMP #220,(R2) ;SEQUENCE ERROR?
6535 020622 001060 BNE TS221-10 ;BR TO ERROR HALT ON SEQ ERROR
6536 020624 012700 000005 MOV #5,R0
6537 020630 000257 CCC ;CC=1010
6538 020632 000271 +SEN.SEC
6539 020634 022700 000005 CMP #5,R0 ;CC=0101
6540 020640 101002 BHI CMP1
6541 020642 102401 BVS CMP1
6542 020644 100004 BPL CMP2
6543 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6544 ; CONDITIONAL BRANCH INST. AND <====
6545 ; REPLACE THE MOVE INSTRUCTION <====
6546 ; WHICH FOLLOWS W/ 766 <====
6547 020646 CMP1:
6548 020646 012742 000471 MOV #471,-(R2) ;MOVE TO MAILBOX # ***** 471 *****
6549 020652 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6550 020654 000000 HALT ;CMP DID NOT SET CC'S CORRECTLY
6551 020656 012700 100000 CMP2: MOV #100000,R0
6552 020662 000277 SCC ;CC=1101
6553 020664 000242 CLV
6554 020666 020027 077777 CMP R0,#77777 ;CC=0010
6555 020672 101402 BLOS CMP3
6556 020674 102001 BVC CMP3
6557 020676 100004 BPL CMP4
6558 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6559 ; CONDITIONAL BRANCH INST. AND <====
```

```
6560                                     :                               <====  
6561                                     :                               <====  
6562 020700                             :                               <====  
6563 020700 012742 000472               :                               <====  
6564 020704 005242                       :                               <====  
6565 020706 000000                       :                               <====  
6566 020710 052700 040000               :                               <====  
6567 020714 000257                       :                               <====  
6568 020716 000264                       :                               <====  
6569 020720 022700 040000               :                               <====  
6570 020724 102003                       :                               <====  
6571 020726 103002                       :                               <====  
6572 020730 001401                       :                               <====  
6573 020732 100404                       :                               <====  
6574                                     :                               <====  
6575                                     :                               <====  
6576                                     :                               <====  
6577                                     :                               <====  
6578 020734                             :                               <====  
6579 020734 012742 000473               :                               <====  
6580 020740 005242                       :                               <====  
6581 020742 000000                       :                               <====  
6582 020744 042700 040000               :                               <====  
6583 020750 000277                       :                               <====  
6584 020752 022700 177777               :                               <====  
6585 020756 101402                       :                               <====  
6586 020760 102401                       :                               <====  
6587 020762 100004                       :                               <====  
6588                                     :                               <====  
6589                                     :                               <====  
6590                                     :                               <====  
6591                                     :                               <====  
6592 020764                             :                               <====  
6593 020764 012742 000474               :                               <====  
6594 020770 005242                       :                               <====  
6595 020772 000000                       :                               <====  
6596                                     :                               <====  
6597                                     :                               <====  
6598                                     :                               <====  
6599                                     :                               <====  
6600                                     :                               <====  
6601 020774 005212                       :                               <====  
6602 020776 022712 000221               :                               <====  
6603 021002 001010                       :                               <====  
6604 021004 012700 177777               :                               <====  
6605 021010 000257                       :                               <====  
6606 021012 000265                       :                               <====  
6607 021014 005100                       :                               <====  
6608 021016 101002                       :                               <====  
6609 021020 102401                       :                               <====  
6610 021022 100004                       :                               <====  
6611                                     :                               <====  
6612                                     :                               <====  
6613                                     :                               <====  
6614                                     :                               <====  
6615 021024                             :                               <====
```

REPLACE THE MOVE INSTRUCTION WHICH FOLLOWS W/ 751

MOV #472, -(R2) ; MOVE TO MAILBOX # ***** 472 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; CMP DID NOT SET CC'S CORRECTLY
BIS #40000, R0 ; R0=140000
CCC ; CC=0100
SEZ ;
CMP #40000, R0 ; CC=1011
BVC CMP5
BCC CMP5
BEQ CMP5
BMI CMP6

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS CONDITIONAL BRANCH INST. AND REPLACE THE MOVE INSTRUCTION WHICH FOLLOWS W/ 733

MOV #473, -(R2) ; MOVE TO MAILBOX # ***** 473 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; CMP DID NOT SET CC'S CORRECTLY
BIC #40000, R0 ; CC=1111
SCC ; CC=0000
CMP #-1, R0
BLOS CMP7
BVS CMP7
BPL TS221

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS CONDITIONAL BRANCH INST. AND REPLACE THE MOVE INSTRUCTION WHICH FOLLOWS W/ 717

MOV #474, -(R2) ; MOVE TO MAILBOX # ***** 474 *****
INC -(R2) ; SET MSGTYP TO FATAL ERROR
HALT ; CMP DID NOT SET CC'S CORRECTLY
OR SEQUENCE ERROR

: TEST 221 TEST COM INSTRUCTION

TS221: INC (R2) ; UPDATE TEST NUMBER
CMP #221, (R2) ; SEQUENCE ERROR?
BNE TS222-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #-1, R0
CCC ; CC=1010
+SEC!SEZ
COM R0 ; CC=0101
BHI COM1
BVS COM1
BPL TS222

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

```
6616 021024 012742 000475      MOV    #475,-(R2)      ;MOVE TO MAILBOX # ***** 475 *****
6617 021030 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6618 021032 000000              HALT                  ;COM DID NOT SET CC'S CORRECTLY
6619                                     ; OR SEQUENCE ERROR
```

6620
6621
6622
6623
6624
6625
6626
6627
6628
6629
6630
6631
6632
6633

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

6634
6635
6636
6637
6638
6639
6640
6641
6642
6643
6644
6645
6646
6647
6648
6649

```
.....
: 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 <== -
```

6650
6651
6652
6653
6654
6655
6656
6657
6658
6659
6660
6661
6662
6663
6664

```
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
```

6665
6666
6667
6668
6669
6670
6671

```
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 751 <====
SUB3:  MOV    #477,-(R2) ;MOVE TO MAILBOX # ***** 477 *****
        INC    -(R2)    ;SET MSGTYP TO FATAL ERROR
        HALT
SUB4:  COM    R0
        SCC   ;R0=177777
        ;CC=11111
```

```
6672 021134 162700 100000      SUB      #100000,R0      ;CC=0000  R0=77777
6673 021140 101402      BLOS     SUB5
6674 021142 102401      BVS      SUB5
6675 021144 100004      BPL      SUB6
6676
6677      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6678      ;          CONDITIONAL BRANCH INST. AND <====
6679      ;          REPLACE THE MOVE INSTRUCTION <====
6680      ;          WHICH FOLLOWS W/ 736 <====
6680 021146      SUB5:
6681 021146 012742 000500      MOV      #500,-(R2)      ;MOVE TO MAILBOX # ***** 500 *****
6682 021152 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6683 021154 000000      HALT
6684 021156 000257      SUB6: CCC              ;SUB DID NOT SET CC'S CORRECTLY
6685 021160 000264      SEZ
6686 021162 162700 140000      SUB      #140000,R0      ;CC=1011
6687 021166 102003      BVC      SUB7
6688 021170 103002      BCC      SUB7
6689 021172 001401      BEQ      SUB7
6690 021174 100404      BMI      TS223
6691
6692      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6693      ;          CONDITIONAL BRANCH INST. AND <====
6694      ;          REPLACE THE MOVE INSTRUCTION <====
6695      ;          WHICH FOLLOWS W/ 722 <====
6695 021176      SUB7:
6696 021176 012742 000501      MOV      #501,-(R2)      ;MOVE TO MAILBOX # ***** 501 *****
6697 021202 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6698 021204 000000      HALT
6699
6700      ;*****
6701      ;TEST 223      TEST SBC INSTRUCTION
6702      ;*****
6703 021206 005212      TS223: INC      (R2)          ;UPDATE TEST NUMBER
6704 021210 022712 000223      CMP      #223,(R2)      ;SEQUENCE ERROR?
6705 021214 001053      BNE      TS224-10      ;BR TO ERROR HALT ON SEQ ERROR
6706 021216 012700 000001      MOV      #1,R0
6707 021222 000277      SCC
6708 021224 000244      CLZ
6709 021226 005600      SBC      R0              ;CC=0100  R=0
6710 021230 103403      BCS      SBC1
6711 021232 102402      BVS      SBC1
6712 021234 100401      BMI      SBC1
6713 021236 001404      BEQ      SBC2
6714
6715      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6716      ;          CONDITIONAL BRANCH INST. AND <====
6717      ;          REPLACE THE MOVE INSTRUCTION <====
6718      ;          WHICH FOLLOWS W/ 766 <====
6718 021240      SBC1:
6719 021240 012742 000502      MOV      #502,-(R2)      ;MOVE TO MAILBOX # ***** 502 *****
6720 021244 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6721 021246 000000      HALT
6722 021250 000277      SBC2: SCC              ;SBC DID NOT SET CC'S CORRECTLY
6723 021252 000245      +CLZ!CLC              ;CC=1010
6724 021254 005600      SBC      R0              ;CC=0100  R=0
6725 021256 103403      BCS      SBC3
6726 021260 102402      BVS      SBC3
6727 021262 100401      BMI      SBC3
```

```

6728 021264 001404          BEQ      SBC4          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6729                                     ;          CONDITIONAL BRANCH INST. AND <====
6730                                     ;          REPLACE THE MOVE INSTRUCTION <====
6731                                     ;          WHICH FOLLOWS W/ 753 <====
6732
6733 021266          SBC3:
6734 021266 012742 000503    MOV      #503,-(R2)    ;MOVE TO MAILBOX # ***** 503 *****
6735 021272 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
6736 021274 000000          HALT
6737 021276 000277          SBC4:    SCC
6738 021300 000250          CLN
6739 021302 005600          SBC      R0          ;CC=1001  R0=177777
6740 021304 103003          BCC      SBC5
6741 021306 102402          BVS      SBC5
6742 021310 001401          BEQ      SBC5
6743 021312 100404          BMI      SBC6
6744                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6745                                     ;          CONDITIONAL BRANCH INST. AND <====
6746                                     ;          REPLACE THE MOVE INSTRUCTION <====
6747                                     ;          WHICH FOLLOWS W/ 740 <====
6748
6749 021314          SBC5:
6750 021314 012742 000504    MOV      #504,-(R2)    ;MOVE TO MAILBOX # ***** 504 *****
6751 021320 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
6752 021322 000000          HALT
6753 021324 042700 077777    SBC6:    BIC      #77777,R0    ;SBC DID NOT SET CC'S CORRECTLY
6754 021330 000277          SCC          ;R0=100000
6755 021332 000242          CLV          ;CC=1101
6756 021334 005600          SBC      R0          ;CC=0010
6757 021336 101402          BLOS     SBC7
6758 021340 102001          BVC      SBC7
6759 021342 100004          BPL      TS224
6760                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6761                                     ;          CONDITIONAL BRANCH INST. AND <====
6762                                     ;          REPLACE THE MOVE INSTRUCTION <====
6763                                     ;          WHICH FOLLOWS W/ 724 <====
6764
6765 021344          SBC7:
6766 021344 012742 000505    MOV      #505,-(R2)    ;MOVE TO MAILBOX # ***** 505 *****
6767 021350 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
6768 021352 000000          HALT          ;SBC DID NOT SET CC'S CORRECTLY
6769                                     ; OR SEQUENCE ERROR

```

```

6770
6771
6772
6773
6774
6775
6776
6777
6778
6779
6780
6781
6782
6783 021354 005212          TS224:  INC      (R2)          ;UPDATE TEST NUMBER

```

: THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF THE ROL,
: ROR, ASL AND ASR INSTRUCTIONS. SPECIAL DATA PATTERNS ARE LOADED
: AND ROTATED SEVERAL TIMES FOR EACH TEST. THE CONDITION CODES
: ARE PRESET BEFORE EACH ROTATION AND THE CONDITION CODES ARE
: CHECKED AFTER EACH ROTATION. THE FINAL CHECK IN EACH TEST IS
: TO VERIFY THE CUMULATIVE DATA RESULT. THE DATA PATTERNS HAVE
: BEEN SELECTED TO PRODUCE ALL COMBINATIONS OF THE C AND V BITS.

6784	021356	022712	000224		CMP	#224,(R2)		:SEQUENCE ERROR?	
6785	021362	001053			BNE	TS225-10		:BR TO ERROR HALT ON SEQ ERROR	
6786	021364	012700	144000		MOV	#144000,R0		:R0=144000	
6787	021370	000257			CCC			:CC=0110	
6788	021372	000266			+SEZ:SEV				
6789	021374	006100			ROL	R0		:CC=1001 R0=110000	
6790	021376	103003			BCC	ROL1			
6791	021400	102402			BVS	ROL1			
6792	021402	001401			BEQ	ROL1			
6793	021404	100404			BMI	ROL2			
6794								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
6795								: CONDITIONAL BRANCH INST. AND	<====
6796								: REPLACE THE MOVE INSTRUCTION	<====
6797								: WHICH FOLLOWS W/ 766	<====
6798	021406			ROL1:					
6799	021406	012742	000506		MOV	#506,-(R2)		:MOVE TO MAILBOX # ***** 506 *****	
6800	021412	005242			INC	-(R2)		:SET MSGTYP TO FATAL ERROR	
6801	021414	000000			HALT				
6802	021416	000277		ROL2:	SCC			:CC=1100	
6803	021420	000243			+CLV:CLC				
6804	021422	006100			ROL	R0		:CC=0011 R0=020000	
6805	021424	103003			BCC	ROL3			
6806	021426	102002			BVC	ROL3			
6807	021430	001401			BEG	ROL3			
6808	021432	100004			BPL	ROL4			
6809								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
6810								: CONDITIONAL BRANCH INST. AND	<====
6811								: REPLACE THE MOVE INSTRUCTION	<====
6812								: WHICH FOLLOWS W/ 753	<====
6813	021434			ROL3:					
6814	021434	012742	000507		MOV	#507,-(R2)		:MOVE TO MAILBOX # ***** 507 *****	
6815	021440	005242			INC	-(R2)		:SET MSGTYP TO FATAL ERROR	
6816	021442	000000			HALT			:ROL DID NOT SET CC'S CORRECTLY	
6817	021444	000277		ROL4:	SCC			:CC=0111	
6818	021446	000250			CLN				
6819	021450	006100			ROL	R0		:CC=0000 R0=040001	
6820	021452	101402			BLOS	ROL5			
6821	021454	102401			BVS	ROL5			
6822	021456	100004			BPL	ROL6			
6823								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
6824								: CONDITIONAL BRANCH INST. AND	<====
6825								: REPLACE THE MOVE INSTRUCTION	<====
6826								: WHICH FOLLOWS W/ 741	<====
6827	021460			ROL5:					
6828	021460	012742	000510		MOV	#510,-(R2)		:MOVE TO MAILBOX # ***** 510 *****	
6829	021464	005242			INC	-(R2)		:SET MSGTYP TO FATAL ERROR	
6830	021466	000000			HALT			:ROL DID NOT SET CC'S CORRECTLY	
6831	021470	000257		ROL6:	CCC			:CC=0101	
6832	021472	000265			+SEZ:SEC				
6833	021474	006100			ROL	R0		:CC=1010 R0=100003	
6834	021476	101405			BLOS	ROL7			
6835	021500	102004			BVC	ROL7			
6836	021502	100003			BPL	ROL7			
6837	021504	022700	100003		CMP	#100003,R0			
6838	021510	001404			BEQ	TS225			
6839								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====

```
6840  
6841  
6842  
6843 021512  
6844 021512 012742 000511 ROL7:  
6845 021516 005242 MOV #511,-(R2) ;MOVE TO MAILBOX # ***** 511 *****  
6846 021520 000000 INC -(R2) ;SET MSGTYP TO FATAL ERROR  
6847  
6848  
6849  
6850  
6851 021522 005212  
6852 021524 022712 000225 TS225:  
6853 021530 001051  
6854 021532 012700 000023  
6855 021536 000277  
6856 021540 000250  
6857 021542 006000  
6858 021544 102403  
6859 021546 103002  
6860 021550 001401  
6861 021552 100404  
6862  
6863  
6864  
6865  
6866 021554  
6867 021554 012742 000512 ROR1:  
6868 021560 005242  
6869 021562 000000  
6870 021564 000257 ROR2:  
6871 021566 000274  
6872 021570 006000  
6873 021572 102003  
6874 021574 103002  
6875 021576 001401  
6876 021600 100004 BPL  
6877  
6878  
6879  
6880  
6881 021602  
6882 021602 012742 000513 ROR3:  
6883 021606 005242  
6884 021610 000000  
6885 021612 000277 ROR4:  
6886 021614 000241  
6887 021616 006000  
6888 021620 101403  
6889 021622 102402  
6890 021624 001401  
6891 021626 100004 BPL  
6892  
6893  
6894  
6895
```

CONDITIONAL BRANCH INST. AND <====
REPLACE THE MOVE INSTRUCTION <====
WHICH FOLLOWS W/ 724 <====

;TEST 225 TEST ROR 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/ 753 <--

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

```
6896 021630
6897 021630 012742 000514
6898 021634 005242
6899 021636 000000
6900 021640 000257
6901 021642 000265
6902 021644 006000
6903 021646 101402
6904 021650 102001
6905 021652 100404
6906
6907
6908
6909
6910 021654
6911 021654 012742 000515
6912 021660 005242
6913 021662 000000
6914
6915
6916
6917
6918 021664 005212
6919 021666 022712 000226
6920 021672 001054
6921 021674 012700 144000
6922 021700 000257
6923 021702 000271
6924 021704 006300
6925 021706 103003
6926 021710 102402
6927 021712 001401
6928 021714 100404
6929
6930
6931
6932
6933 021716
6934 021716 012742 000516
6935 021722 005242
6936 021724 000000
6937 021726 000277
6938 021730 000243
6939 021732 006300
6940 021734 103003
6941 021736 102002
6942 021740 001401
6943 021742 100004
6944
6945
6946
6947
6948 021744
6949 021744 012742 000517
6950 021750 005242
6951 021752 000000

ROR5:
MOV #514,-(R2) ;MOVE TO MAILBOX # ***** 514 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ROR DID NOT SET CC'S CORRECTLY
ROR6: CCC ;CC=0101
+SEC!SEZ ;CC=1010 R0=110001
ROR R0
BLOS ROR7
BVC ROR7
BMI TS226

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

ROR7:
MOV #515,-(R2) ;MOVE TO MAILBOX # ***** 515 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ROR DID NOT PRODUCE CORRECT RESULTS
; OR SEQUENCE ERROR

;*****
;TEST 226 TEST ASL INSTRUCTION
;*****
TS226: INC (R2) ;UPDATE TEST NUMBER
CMP #226,(R2) ;SEQUENCE ERROR?
BNE TS227-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #144000,R0 ;R0=14000
CCC ;CC=0110
+SEN.SEC
ASL R0 ;CC=1001 R0=110000
BCC ASL1
BVS ASL1
BEQ ASL1
BMI ASL2

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

ASL1:
MOV #516,-(R2) ;MOVE TO MAILBOX # ***** 516 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT
ASL2: SCC ;CC=1100
+CLV!CLC ;CC=0011 R0=020000
ASL R0
BCC ASL3
BVC ASL3
BEQ ASL3
BPL ASL4

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

ASL3:
MOV #517,-(R2) ;MOVE TO MAILBOX # ***** 517 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;ASL DID NOT SET CC'S CORRECTLY
```



```
6952 021754 000277 ASL4: SCC ;CC=0111
6953 021756 000250 CLN
6954 021760 006300 ASL R0 ;CC=0000 R0=040000
6955 021762 101402 BLOS ASL5
6956 021764 102401 BVS ASL5
6957 021766 100004 BPL ASL6
6958 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6959 ; CONDITIONAL BRANCH INST. AND <====
6960 ; REPLACE THE MOVE INSTRUCTION <====
6961 ; WHICH FOLLOWS W/ 741 <====
6962 021770 ASL5:
6963 021770 012742 000520 MOV #520,-(R2) ;MOVE TO MAILBOX # ***** 520 *****
6964 021774 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6965 021776 000000 HALT ;ASL DID NOT SET CC'S CORRECTLY
6966 022000 000257 ASL6: CCC ;CC=0101
6967 022002 000265 +SEZ!SEC
6968 022004 006300 ASL R0 ;CC=1010 R0=100000
6969 022006 103406 BCS ASL7
6970 022010 001405 BEQ ASL7
6971 022012 102004 BVC ASL7
6972 022014 100003 BPL ASL7
6973 022016 022700 100000 CMP #100000,R0
6974 022022 001404 BEQ TS227
6975 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6976 ; CONDITIONAL BRANCH INST. AND <====
6977 ; REPLACE THE MOVE INSTRUCTION <====
6978 ; WHICH FOLLOWS W/ 723 <====
6979 022024 ASL7:
6980 022024 012742 000521 MOV #521,-(R2) ;MOVE TO MAILBOX # ***** 521 *****
6981 022030 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6982 022032 000000 HALT ;ASL MALFUNCTIONED
6983 ; OR SEQUENCE ERROR
6984
6985 ;*****
6986 ;TEST 227 TEST ASR INSTRUCTION
6987 ;*****
6988 022034 005212 TS227: INC (R2) ;UPDATE TEST NUMBER
6989 022036 022712 000227 CMP #227,(R2) ;SEQUENCE ERROR?
6990 022042 001060 BNE TS230-10 ;BR TO ERROR HALT ON SEQ ERROR
6991 022044 012700 100023 MOV #100023,R0 ;R0=100023
6992 022050 000277 SCC ;CC=0110
6993 022052 000250 CLN
6994 022054 006200 ASR R0 ;CC=1001 RP=140011
6995 022056 102403 BVS ASR1
6996 022060 103002 BCC ASR1
6997 022062 001401 BEQ ASR1
6998 022064 100404 BMI ASR2
6999 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7000 ; CONDITIONAL BRANCH INST. AND <====
7001 ; REPLACE THE MOVE INSTRUCTION <====
7002 ; WHICH FOLLOWS W/ 766 <====
7003 022066 ASR1:
7004 022066 012742 000522 MOV #522,-(R2) ;MOVE TO MAILBOX # ***** 522 *****
7005 022072 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7006 022074 000000 HALT ;ASR DID NOT SET CC'S CORRECTLY
7007 022076 042700 100000 ASR2: BIC #100000,R0 ;R0=40011
```

```

7008 022102 000277          SCC          ;CC=1100
7009 022104 000243          +CLV!CLC
7010 022106 006200          ASR      R0          ;CC=0011  R0=020004
7011 022110 102003          BVC      ASR3
7012 022112 103002          BCC      ASR3
7013 022114 001401          BEQ      ASR3
7014 022116 100004          BPL      ASR4
7015
7016          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7017          ;          CONDITIONAL BRANCH INST. AND <====
7018          ;          REPLACE THE MOVE INSTRUCTION <====
7019          ;          WHICH FOLLOWS W/ 751 <====
7019 022120          ASR3:
7020 022120 012742 000523    MOV      #523,-(R2)    ;MOVE TO MAILBOX # ***** 523 *****
7021 022124 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7022 022126 000000          HALT
7023 022130 000277          ASR4: SCC          ;ASR DID NOT SET CC'S CORRECTLY
7024          ;CC=1111
7025 022132 006200          ASR      R0          ;CC=0000  R0=010002
7026 022134 101403          BLOS     ASR5
7027 022136 102402          BVS     ASR5
7028 022140 001401          BEQ     ASR5
7029 022142 100004          BPL     ASR6
7030
7031          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7032          ;          CONDITIONAL BRANCH INST. AND <====
7033          ;          REPLACE THE MOVE INSTRUCTION <====
7034          ;          WHICH FOLLOWS W/ 737 <====
7034 022144          ASR5:
7035 022144 012742 000524    MOV      #524,-(R2)    ;MOVE TO MAILBOX # ***** 524 *****
7036 022150 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7037 022152 000000          HALT
7038 022154 052700 100000    ASR6: BIS      #100000,R0 ;ASR DID NOT SET CC'S CORRECTLY
7039 022160 000257          CCC
7040 022162 000265          +SEZ!SEC ;R0=110002
7041 022164 006200          ASR      R0          ;CC=0101
7042 022166 101406          BLOS     ASR7          ;C=1010  R0=144001
7043 022170 102005          BVC     ASR7
7044 022172 100004          BPL     ASR7
7045 022174 001403          BEQ     ASR7
7046 022176 022700 144001    CMP      #144001,R0    ;CHECK RESULT OF ASR'S
7047 022202 001404          BEQ     TS230
7048
7049          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7050          ;          CONDITIONAL BRANCH INST. AND <====
7051          ;          REPLACE THE MOVE INSTRUCTION <====
7052          ;          WHICH FOLLOWS W/ 717 <====
7052 022204          ASR7:
7053 022204 012742 000525    MOV      #525,-(R2)    ;MOVE TO MAILBOX # ***** 525 *****
7054 022210 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7055 022212 000000          HALT          ;ASR DID NOT FUNCTION CORRECTLY
7056          ; OR SEQUENCE ERROR
7057
7058
7059
7060
7061
7062
7063
:*****
:
: THIS TEST VERIFIES THE SXT INSTRUCTION. CONDITION CODES
: ARE PRESET IN EACH OF THE TWO POSSIBLE CASES. WITH THE N-BIT SET.

```

7064
7065
7066
7067
7068
7069
7070
7071 022214 005212
7072 022216 022712 000230
7073 022222 001033
7074 022224 005000
7075 022226 000277
7076 022230 000244
7077 022232 006700
7078 022234 100006
7079 022236 001405
7080 022240 102404
7081 022242 103003
7082 022244 022700 177777
7083 022250 001404
7084
7085
7086
7087
7088 022252
7089 022252 012742 000526
7090 022256 005242
7091 022260 000000
7092 022262 005000
7093 022264 005010
7094 022266 005110
7095 022270 000257
7096 022272 000266
7097 022274 006710
7098 022276 001005
7099 022300 103404
7100 022302 102403
7101 022304 100402
7102 022306 005710
7103 022310 001404
7104
7105
7106
7107
7108 022312
7109 022312 012742 000527
7110 022316 005242
7111 022320 000000
7112
7113
7114
7115
7116
7117
7118
7119

:THE TEST CHECKS FOR ALL ONES IN THE DESTINATION. WITH THE N-BIT
:CLEAR, THE DESTINATION SHOULD CONTAIN ALL ZEROES. THE DATA
:IS VERIFIED BY CONDITIONAL BRANCHES.
:*****
:TEST 230 TEST THE SXT INSTRUCTION
:*****
TS230: INC (R2) ;UPDATE TEST NUMBER
CMP #230,(R2) ;SEQUENCE ERROR?
BNE TS231-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0
SCC ;SET CC=1011
CLZ
SXT R0 ;TRY SXT
BPL SXT0 ;TEST CC=1001
BEQ SXT0
BVS SXT0
BCC SXT0
CMP #-1,R0 ;CHECK DATA RESULT
BEQ SXT1
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 764 <====
SXT0: MOV #526,-(R2) ;MOVE TO MAILBOX # ***** 526 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULTS OF SXT INCORRECT
SXT1: CLR R0 ;R0=0
CLR (R0) ;LOC. 0=0
COM (R0) ;LOC. 0=177777
CCC ;SET CC=0110
+SEZ!SEV
SXT (R0)
BNE SXT2 ;TEST CC=0100
BCS SXT2
BVS SXT2
BMI SXT2
TST (R0)
BEQ TS231
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 744 <====
SXT2: MOV #527,-(R2) ;MOVE TO MAILBOX # ***** 527 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULTS OF SXT INCORRECT
: OR SEQUENCE ERROR
:*****
: 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

```
7120 ;REPRODUCE THE ORIGINAL VALUE IF R0=31525.
7121 ;
7122 ;*****
7123 ;TEST 231 TEST THE XOR INSTRUCTION
7124 ;*****
7125 022322 005212 TS231: INC (R2) ;UPDATE TEST NUMBER
7126 022324 022712 000231 CMP #231,(R2) ;SEQUENCE ERROR?
7127 022330 001035 BNE TS232-10 ;BR TO ERROR HALT ON SEQ ERROR
7128 022332 012700 007463 MOV #7463,R0 ;SET UP R0
7129 022336 012701 031525 MOV #31525,R1 ;SET UP R1
7130 022342 000277 SCC ;SET CC=1110
7131 022344 000241 CLC
7132 022346 074100 XOR R1,R0 ;TRY XOR
7133 022350 101406 BLOS XOR1 ;CC=0000?
7134 022352 102405 BVS XOR1
7135 022354 001404 BEQ XOR1
7136 022356 100403 BMI XOR1
7137 022360 022700 036146 CMP #36146,R0 ;DATA RESULT CORRECT?
7138 022364 001404 BEQ XOR2
7139 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7140 ; CONDITIONAL BRANCH INST. AND <====
7141 ; REPLACE THE MOVE INSTRUCTION <====
7142 ; WHICH FOLLOWS W/ 761 <====
7143 022366 XOR1:
7144 022366 012742 000530 MOV #530,-(R2) ;MOVE TO MAILBOX # ***** 530 *****
7145 022372 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7146 022374 000000 HALT ;
7147 022376 010104 XOR2: MOV R1,R4
7148 022400 000261 SEC ;CC=1110
7149 022402 000241 CLC
7150 022404 074400 XOR R4,R0 ;TRY XOR MODE 0,0
7151 022406 101406 BLOS XOR3 ;CC=0000?
7152 022410 102405 BVS XOR3
7153 022412 001404 BEQ XOR3
7154 022414 100403 BMI XOR3
7155 022416 022700 007463 CMP #7463,R0
7156 022422 001404 BEQ TS232
7157 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7158 ; CONDITIONAL BRANCH INST. AND <====
7159 ; REPLACE THE MOVE INSTRUCTION <====
7160 ; WHICH FOLLOWS W/ 742 <====
7161 022424 XOR3:
7162 022424 012742 000531 MOV #531,-(R2) ;MOVE TO MAILBOX # ***** 531 *****
7163 022430 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7164 022432 000000 HALT ;RESULT OF XOR INCORRECT
7165 ; OR SEQUENCE ERROR
7166 ;
7167 ;*****
7168 ; THIS TEST VERIFIES THE SOB INSTRUCTION. R4 IS USED AS A
7169 ; COUNTER WHILE R0 IS THE ADDRESS REGISTER. CONDITIONAL
7170 ; BRANCHES ARE USED TO VERIFY PROPER TRANSFER OF CONTROL
7171 ; WHILE R4 IS CHECKED TO INSURE PROPER DECREMENTING OF R0.
7172 ;
7173 ;*****
7174 ;TEST 232 TEST SOB INSTRUCTION
7175 ;
```

```
7176
7177 022434 005212
7178 022436 022712 000232
7179 022442 001023
7180 022444 012700 000525
7181 022450 010004
7182 022452 000277
7183 022454 101002
7184 022456 100001
7185 022460 102404
7186
7187
7188
7189
7190 022462
7191 022462 012742 000532
7192 022466 005242
7193 022470 000000
7194 022472 005304
7195 022474 000277
7196 022476 077012
7197 022500 101004
7198 022502 100003
7199 022504 102002
7200 022506 005704
7201 022510 001404
7202
7203
7204
7205
7206 022512
7207 022512 012742 000533
7208 022516 005242
7209 022520 000000
7210
7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222 022522 005212
7223 022524 022712 000233
7224 022530 001061
7225 022532 012706 001000
7226 022536 012746 125252
7227 022542 162706 000074
7228 022546 012705 022572
7229 022552 012746 006436
7230 022556 000277
7231 022560 000116

:*****
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
        SCC
        SOB1:  BHI     SOB2           ;SET CC=1111
        BPL     SOB2           ;CC=1111?
        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
        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

:*****
: THIS TEST VERIFIES THE MARK INSTRUCTION. THE EFFECTS
: OF THE MARK INSTRUCTION ARE SIMULATED BY THE PROGRAM INSTRUCTIONS.
: THE CONTENTS OF R5 AND THE STACK POINTER ARE CHECKED AFTER EACH
: OF THE TWO ROUTINES IN THE TEST.
:*****
:TEST 233 TEST MARK INSTRUCTION
:*****
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
        JMP     (SP)           ;XFER CONTL TO MARK 36 INST. ON STACK
```

```

7232 022562 012742 000534      MOV      #534,-(R2)      ;MOVE TO MAILBOX # ***** 534 *****
7233 022566 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
7234 022570 000000              HALT                    ;MARK INST. SHOULD HAVE JUMPED TO MRK1
7235 022572 101010      MRK1:  BHI      MRK2      ;TEST CC UNAFFECTED
7236 022574 100007              BPL      MRK2          ;IE. CC=1111
7237 022576 102006              BVC      MRK2
7238 022600 020527 125252      CMP      R5,#125252    ;CHECK R5 RESTORED FROM STACK
7239 022604 001003              BNE      MRK2
7240 022606 027706 001000      CMP      #STBOT,R6    ;CHECK STACK POINTER READJUSTED CORRECTLY.
7241 022612 001404              BEQ      MRK3
7242                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
7243                                ;          CONDITIONAL BRANCH INST. AND
7244                                ;          REPLACE THE MOVE INSTRUCTION
7245                                ;          WHICH FOLLOWS W/ 746
7246 022614              MRK2:
7247 022614 012742 000535      MOV      #535,-(R2)    ;MOVE TO MAILBOX # ***** 535 *****
7248 022620 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
7249 022622 000000              HALT                    ;RESULTS OF MARK INCORRECT
7250 022624 012746 052525      MRK3:  MOV      #52525,-(SP) ;PUT MARK 0 INST. ON STACK
7251 022630 012746 006400      MOV      #6400,-(SP)  ;SET ADDR. OF MARK INST. IN R5
7252 022634 010605              MOV      SP,R5         ;DO JSR
7253 022636 004737 022646      JSR      PC,@MRK4
7254 022642 000137 022660      JMP      @MRK5
7255 022646 000205      MRK4:  RTS      R5         ;DO RTS WITH R5 TO MARK INST ON STACK
7256 022650 012742 000536      MOV      #536,-(R2)    ;MOVE TO MAILBOX # ***** 536 *****
7257 022654 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
7258 022656 000000              HALT                    ;RTS,MARK SEQUENCE FAILED
7259 022660 022706 001000      MRK5:  CMP      #STBOT,R6    ;STACK ADJUSTED CORRECTLY
7260 022664 001003              BNE      MRK6
7261 022666 022705 052525      CMP      #52525,R5    ;IF NOT: BR
7262 022672 001404              BEQ      TS234         ;CHECK IF R5 RESTORED FROM STACK
7263                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
7264                                ;          CONDITIONAL BRANCH INST. AND
7265                                ;          REPLACE THE MOVE INSTRUCTION
7266                                ;          WHICH FOLLOWS W/ 716
7267 022674              MRK6:
7268 022674 012742 000537      MOV      #537,-(R2)    ;MOVE TO MAILBOX # ***** 537 *****
7269 022700 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
7270 022702 000000              HALT                    ;RESULTS OF MARK INCORRECT
7271                                ; OR SEQUENCE ERROR
7272
7273              PS 177776
7274
7275      ;*****
7276      ;
7277      ; THESE NEXT SEVEN TESTS VERIFY THE MTPS INSTRUCTION IN ALL
7278      ; MODES. THE PSW IS DEFINED BY AN EQUATE STATEMENT BEFORE THE
7279      ; FIRST MTPS TEST. IN EACH TEST A PATTERN OF ONES AND
7280      ; ZEROES IS SET IN A DATA REGISTER AND MOVED TO THE PSW.
7281      ; THE DATA IN THE PSW, AND THE DATA REGISTER ADDRESS,
7282      ; ARE CHECKED TO VERIFY PROPER EXECUTION OF THE INSTRUCTION.
7283      ;*****
7284      ;TEST 234      TEST MTPS INSTRUCTION
7285      ;*****
7286 022704 005212      TS234:  INC      (R2)      ;UPDATE TEST NUMBER
7287 022706 022712 000234      CMP      #234,(R2)    ;SEQUENCE ERROR?

```



```
7344 023046 012742 000543      MOV    #543,-(R2)      ;MOVE TO MAILBOX # ***** 543 *****
7345 023052 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
7346 023054 000000              HALT                  ;DEST REGISTER NOT INCREMENTED BY 1
7347                                     ; OR SEQUENCE ERROR
```

```
*****
:TEST 236      TEST MTPS MODE 3
*****
```

```
7351
7352 023056 005212              TS236: INC    (R2)          ;UPDATE TEST NUMBER
7353 023060 022712 000236      CMP    #236,(R2)      ;SEQUENCE ERROR?
7354 023064 001024              BNE    TS237-10       ;BR TO ERROR HALT ON SEQ ERROR
7355 023066 012700 000402      MOV    #402,R0        ;R0=402
7356 023072 005010              CLR    (R0)           ;LOC. 402=0
7357 023074 012737 052652 000000  MOV    #52652,@#0     ;LOC. 0=52652
7358 023102 005037 177776      CLR    @#PS           ;PS=0
7359 023106 106430              MTPS   @ (R0)+        ;TRY MTPS W/MODE 3
7360 023110 022737 000252 177776  CMP    #252,@#PS      ;CHECK DEST. DATA
7361 023116 001404              BEQ    MTPS3
```

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

```
7365
7366 023120 012742 000544      MOV    #544,-(R2)      ;MOVE TO MAILBOX # ***** 544 *****
7367 023124 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
7368 023126 000000              HALT                  ;DEST. DATA INCORRECT
7369 023130 022700 000404      MTPS3: CMP    #404,R0   ;CHECK MODE 3 REGISTER.
7370 023134 001404              BEQ    TS237
```

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

```
7375 023136 012742 000545      MOV    #545,-(R2)      ;MOVE TO MAILBOX # ***** 545 *****
7376 023142 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
7377 023144 000000              HALT                  ;MODE 3 REGISTER INCORRECT
7378                                     ; OR SEQUENCE ERROR
```

```
*****
:TEST 237      TEST MTPS MODE 4
*****
```

```
7383 023146 005212              TS237: INC    (R2)          ;UPDATE TEST NUMBER
7384 023150 022712 000237      CMP    #237,(R2)      ;SEQUENCE ERROR?
7385 023154 001022              BNE    TS240-10       ;BR TO ERROR HALT ON SEQ ERROR
7386 023156 012700 000001      MOV    #1,R0          ;R0=1
7387 023162 012737 125125 000000  MOV    #125125,@#0    ;LOC. 0 = 125125
7388 023170 005037 177776      CLR    @#PS           ;PS=0
7389 023174 106440              MTPS   -(R0)         ;TRY MTPS W/MODE 4
7390 023176 022737 000105 177776  CMP    #105,@#PS      ;CHECK DEST. DATA
7391 023204 001404              BEQ    MTPS4
```

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

```
7396 023206 012742 000546      MOV    #546,-(R2)      ;MOVE TO MAILBOX # ***** 546 *****
7397 023212 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
7398 023214 000000              HALT                  ;DEST. DATA INCORRECT
7399 023216 005700      MTPS4: TST    R0      ;CHECK MODE 4 REGISTER
```



```
7400 023220 001404          BEQ      TS240
7401
7402
7403
7404
7405 023222 012742 000547    MOV      #547,-(R2)
7406 023226 005242          INC      -(R2)
7407 023230 000000          HALT
7408
7409
7410
7411
7412
7413
7414 023232 005212          TS240:  INC      (R2)
7415 023234 022712 000240    CMP      #240,(R2)
7416 023240 001021          BNE     TS241-10
7417 023242 012700 000404    MOV      #404,R0
7418 023246 012737 177400 000000  MOV      #177400,@#0
7419 023254 000277          SCC
7420 023256 106450          MTPS   @-(R0)
7421 023260 005737 177776    TST     @#PS
7422 023264 001404          BEQ     MTPS5
7423
7424
7425
7426 023266 012742 000550    MOV      #550,-(R2)
7427 023272 005242          INC      -(R2)
7428 023274 000000          HALT
7429 023276 022700 000402    MTPS5:  CMP      #402,R0
7430 023302 001404          BEQ     TS241
7431
7432
7433
7434
7435 023304 012742 000551    MOV      #551,-(R2)
7436 023310 005242          INC      -(R2)
7437 023312 000000          HALT
7438
7439
7440
7441
7442
```

```
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 755 <====
: MOVE TO MAILBOX # ***** 547 *****
: SET MSGTYP TO FATAL ERROR
: MODE 4 REGISTER NOT DECREMENTED BY 1
: OR SEQUENCE ERROR
```

```
*****
:TEST 240          TEST MTPS MODE 5
*****
```

```
TS240:  INC      (R2)
        CMP      #240,(R2)
        BNE     TS241-10
        MOV      #404,R0
        MOV      #177400,@#0
        SCC
        MTPS   @-(R0)
        TST     @#PS
        BEQ     MTPS5
        :UPDATE TEST NUMBER
        :SEQUENCE ERROR?
        :BR TO ERROR HALT ON SEQ ERROR
        :R0=404
        :LOC. 0=177400
        :SET ALL COND. CODES
        :TRY MTPS W/MODE 5
        :CHECK DEST. DATA.
```

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

```
MTPS5:  CMP      #402,R0
        BEQ     TS241
```

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

```
: MOVE TO MAILBOX # ***** 551 *****
: SET MSGTYP TO FATAL ERROR
: MODE 5 REGISTER NOT DECREMENTED BY 2
: OR SEQUENCE ERROR
```

```
*****
:TEST 241          TEST MTPS MODE 6
*****
```

```
TS241:  INC      (R2)
        CMP      #241,(R2)
        BNE     TS242-10
        MOV      #52652,@#0
        MOV      #406,R0
        CLR     @#PS
        MTPS   -406(R0)
        CMP     #252,@#PS
        BEQ     MTPS6
        :UPDATE TEST NUMBER
        :SEQUENCE ERROR?
        :BR TO ERROR HALT ON SEQ ERROR
        :LOC. 0=52652
        :R0=406
        :PS=0
        :TRY MTPS W/MODE 6
        :CHECK DEST. DATA
```

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

```
7452
7453
7454
7455
```

```
7456 023356 012742 000552      MOV    #552,-(R2)      ;MOVE TO MAILBOX # ***** 552 *****
7457 023362 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
7458 023364 000000              HALT                    ;DEST. DATA INCORRECT
7459 023366 022700 000406      MTPS6: CMP    #406,R0   ;CHECK MODE 6 REGISTER
7460 023372 001404              BEQ    TS242
7461                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7462                                ;          CONDITIONAL BRANCH INST. AND <====
7463                                ;          REPLACE THE MOVE INSTRUCTION <====
7464                                ;          WHICH FOLLOWS W/ 753 <====
7465 023374 012742 000553      MOV    #553,-(R2)      ;MOVE TO MAILBOX # ***** 553 *****
7466 023400 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
7467 023402 000000              HALT                    ;MODE 6 REGISTER MODIFIED
7468                                ; OR SEQUENCE ERROR
```

:TEST 242 TEST MTPS MODE 7

```
7472 TS242: INC    (R2)          ;UPDATE TEST NUMBER
7473 023404 005212              CMP    #242,(R2)      ;SEQUENCE ERROR?
7474 023406 022712 000242      BNE    TS243-10        ;BR TO ERROR HALT ON SEQ ERROR
7475 023412 001024              MOV    #52652,@#0     ;LOC. 0=52652
7476 023414 012737 052652 000000  MOV    #410,R0        ;R0=410
7477 023422 012700 000410      CLR    @#PS           ;PS=0
7478 023426 005037 177776      MTPS   @-2(R0)        ;TRY MTPS W/MODE 7
7479 023432 106470 177776      CMP    #105,@#PS     ;CHECK DEST. DATA
7480 023436 022737 000105 177776  BEQ    MTPS7
7481 023444 001404
```

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

```
7486 023446 012742 000554      MOV    #554,-(R2)      ;MOVE TO MAILBOX # ***** 554 *****
7487 023452 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
7488 023454 000000              HALT                    ;DESTINATION DATA INCORRECT
7489 023456 022700 000410      MTPS7: CMP    #410,R0   ;CHECK MODE 7 REGISTER
7490 023462 001404              BEQ    TS243
```

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

```
7495 023464 012742 000555      MOV    #555,-(R2)      ;MOVE TO MAILBOX # ***** 555 *****
7496 023470 005242              INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
7497 023472 000000              HALT                    ;MODE 7 REGISTER MODIFIED
7498                                ; OR SEQUENCE ERROR
```

: THESE NEXT SEVEN TESTS VERIFY THE MFPS INSTRUCTION IN ALL
: MODES. IN EACH TEST, A PATTERN OF ONES AND ZEROES 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

7500
7501
7502
7503
7504
7505
7506
7507
7508
7509
7510
7511

```

7512 023474 005212          TS243:  INC      (R2)          ;UPDATE TEST NUMBER
7513 023476 022712 000243    CMP      #243,(R2)        ;SEQUENCE ERROR?
7514 023502 001025          BNE     TS244-10         ;BR TO ERROR HALT ON SEQ ERROR
7515 023504 012737 000377 177776  MOV     #377,@#PS
7516 023512 106700          MFPS    R0
7517 023514 022700 177757    CMP     #177757,R0
7518 023520 001404          BEQ     MFPS1
7519
7520
7521
7522
7523 023522 012742 000556    MOV     #556,-(R2)      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7524 023526 005242          INC     -(R2)           ;          CONDITIONAL BRANCH INST. AND <====
7525 023530 000000          HALT                    ;          REPLACE THE MOVE INSTRUCTION <====
7526
7527 023532 005000          MFPS1: CLR     R0        ;          WHICH FOLLOWS W/ 770 <===
7528 023534 012737 177777 000000  MOV     #-1,@#0        ;MOVE TO MAILBOX # ***** 556 *****
7529 023542 005037 177776    CLR     @#PS           ;SET MSGTYP TO FATAL ERROR
7530 023546 106710          MFPS    (R0)          ;MFPS FAILED
7531 023550 105737 000000    TSTB   @#0
7532 023554 001404          BEQ     TS244
7533
7534
7535
7536
7537 023556 012742 000557    MOV     #557,-(R2)      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-- -
7538 023562 005242          INC     -(R2)           ;          CONDITIONAL BRANCH INST. AND <= -
7539 023564 000000          HALT                    ;          REPLACE THE MOVE INSTRUCTION <-- -
7540
7541
7542
7543
7544
7545 023566 005212          *****
7546 023570 022712 000244    TS244:  INC      (R2)          ;*****
7547 023574 001031          CMP     #244,(R2)        ;TEST 244 TEST MFPS MODE 2
7548 023576 005000          BNE     TS245-10         ;*****
7549 023600 005010          CLR     R0              ;UPDATE TEST NUMBER
7550 023602 012737 000377 177776  CLR     (R0)            ;SEQUENCE ERROR?
7551 023610 106720          MOV     #377,@#PS        ;BR TO ERROR HALT ON SEQ ERROR
7552 023612 103003          MFPS    (R0)+           ;R0=0
7553 023614 102402          BCC    MFPS2A           ;LOC. 0=0
7554 023616 001401          BVS    MFPS2A           ;SET PS=357
7555 023620 100404          BEQ    MFPS2A           ;TRY MFPS W/MODE 2
7556
7557
7558
7559
7560 023622          MFPS2A:
7561 023622 012742 000560    MOV     #560,-(R2)      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-- =
7562 023626 005242          INC     -(R2)           ;          CONDITIONAL BRANCH INST. AND <= =
7563 023630 000000          HALT                    ;          REPLACE THE MOVE INSTRUCTION <= =
7564 023632 022737 000357 000000  MFPS2B: CMP     #357,@#0    ;          WHICH FOLLOWS W/ 765 <==
7565 023640 001404          BEQ    MFPS2C
7566
7567
    
```

```
7568                                     :          REPLACE THE MOVE INSTRUCTION <====
7569                                     :          WHICH FOLLOWS W/ 755          <====
7570 023642 012742 000561                :          :MOVE TO MAILBOX # ***** 561 *****
7571 023646 005242                      :          :SET MSGTYP TO FATAL ERROR
7572 023650 000000                      :          :DEST. DATA INCORRECT
7573 023652 022700 000001                :          :CHECK MODE 2 REGISTER
7574 023656 001404                      :          :
7575                                     :          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7576                                     :          :          CONDITIONAL BRANCH INST. AND <====
7577                                     :          :          REPLACE THE MOVE INSTRUCTION <====
7578                                     :          :          WHICH FOLLOWS W/ 746          <====
7579 023660 012742 000562                :          :MOVE TO MAILBOX # ***** 562 *****
7580 023664 005242                      :          :SET MSGTYP TO FATAL ERROR
7581 023666 000000                      :          :MODE 2 REGISTER NOT INCREMENTED 1
7582                                     :          : OR SEQUENCE ERROR
7583
7584                                     :*****
7585 :TEST 245          TEST MFPS MODE 3
7586 :*****
7587 023670 005212                TS245: INC      (R2)          :UPDATE TEST NUMBER
7588 023672 022712 000245          :          :CMP      #245,(R2)      :SEQUENCE ERROR?
7589 023676 001033                :          :BNE     TS246-10        :BR TO ERROR HALT ON SEQ ERROR
7590 023700 012700 000406          :          :MOV     #406,R0         :R0=406
7591 023704 005037 000000          :          :CLR     @#0             :LOC. 0=0
7592 023710 012737 000252 177776 :          :MOV     #252,@#PS      :PS=252
7593 023716 106730                :          :MFPS   @(R0)+          :TRY MFPS WITH MODE 3
7594 023720 103403                :          :BCS    MFPS3A          :BR TO ERROR IF C-BIT SET
7595 023722 102402                :          :BVS    MFPS3A          :BR TO ERROR IF V-BIT SET
7596 023724 001401                :          :BEQ    MFPS3A          :BR TO ERROR IF Z-BIT SET
7597 023726 100404                :          :BMI    MFPS3B
7598                                     :          :
7599                                     :          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7600                                     :          :          CONDITIONAL BRANCH INST. AND <====
7601                                     :          :          REPLACE THE MOVE INSTRUCTION <====
7602                                     :          :          WHICH FOLLOWS W/ 763          <====
7603 023730                MFPS3A: MOV     #563,-(R2)      :MOVE TO MAILBOX # ***** 563 *****
7604 023734 005242                      :          :INC     -(R2)          :SET MSGTYP TO FATAL ERROR
7605 023736 000000                      :          :HALT
7606 023740 022737 125000 000000 :          :MFPS3B: CMP     #125000,@#0 :CONDITION CODES INCORRECT
7607 023746 001404                :          :BEQ    MFPS3C          :CHECK DEST. DATA
7608                                     :          :
7609                                     :          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=
7610                                     :          :          CONDITIONAL BRANCH INST. AND <---
7611                                     :          :          REPLACE THE MOVE INSTRUCTION <---
7612                                     :          :          WHICH FOLLOWS W/ 753          <
7613 023750 012742 000564                :          :MOVE TO MAILBOX # ***** 564 *****
7614 023754 005242                      :          :INC     -(R2)          :SET MSGTYP TO FATAL ERROR
7615 023756 000000                      :          :HALT
7616 023760 020027 000410                :          :MFPS3C: CMP     R0,#410   :DEST DATA INCORRECT
7617 023764 001404                :          :BEQ    TS246          :CHECK MODE 3 REGISTER.
7618                                     :          :
7619                                     :          : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
7620                                     :          :          CONDITIONAL BRANCH INST. AND <
7621                                     :          :          REPLACE THE MOVE INSTRUCTION <--
7622                                     :          :          WHICH FOLLOWS W/ 744          <
7623 023766 012742 000565                :          :MOVE TO MAILBOX # ***** 565 *****
7624 023772 005242                      :          :INC     -(R2)          :SET MSGTYP TO FATAL ERROR
7625 023774 000000                      :          :HALT
7626                                     :          :MODE 3 REGISTER NOT INCREMENTED BY 2
```

```
7624 ; OR SEQUENCE ERROR
7625
7626
7627 :*****
7628 :TEST 246 TEST MFPS MODE 4
7629 :*****
7629 023776 005212 TS246: INC (R2) ;UPDATE TEST NUMBER
7630 024000 022712 000246 CMP #246,(R2) ;SEQUENCE ERROR?
7631 024004 001033 BNE TS247-10 ;BR TO ERROR HALT ON SEQ ERROR
7632 024006 012700 000002 MOV #2,R0 ;R0=2
7633 024012 005037 000000 CLR @#0 ;LOC. 0=0
7634 024016 012737 000125 177776 MOV #125,@#PS ;PS=125
7635 024024 106740 MFPS -(R0) ;TRY MFPS W/MODE 4
7636 024026 103003 BCC MFPS4A ;BR TO ERROR IF C-BIT CLEAR
7637 024030 102402 BVS MFPS4A ;BR TO ERROR IF V-BIT SET
7638 024032 001401 BEQ MFPS4A ;BR TO ERROR IF Z-BIT SET
7639 024034 100004 BPL MFPS4B
7640 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7641 ; CONDITIONAL BRANCH INST. AND <====
7642 ; REPLACE THE MOVE INSTRUCTION <====
7643 ; WHICH FOLLOWS W/ 763 <====
7644 024036 MFPS4A:
7645 024036 012742 000566 MOV #566,-(R2) ;MOVE TO MAILBOX # ***** 566 *****
7646 024042 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7647 024044 000000 HALT ;COND. CODES INCORRECT
7648 024046 022737 042400 000000 MFPS4B: CMP #42400,@#0 ;CHECK DEST. DATA
7649 024054 001404 BEQ MFPS4C
7650 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7651 ; CONDITIONAL BRANCH INST. AND <-==
7652 ; REPLACE THE MOVE INSTRUCTION <-
7653 ; WHICH FOLLOWS W/ 753 <-
7654 024056 012742 000567 MOV #567,-(R2) ;MOVE TO MAILBOX # ***** 567 *****
7655 024062 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7656 024064 000000 HALT ;DEST. DATA INCORRECT
7657 024066 020027 000001 MFPS4C: CMP R0,#1 ;CHECK MODE 4 REGISTER
7658 024072 001404 BEQ TS247
7659 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
7660 ; CONDITIONAL BRANCH INST. AND <
7661 ; REPLACE THE MOVE INSTRUCTION <
7662 ; WHICH FOLLOWS W/ 744 <
7663 024074 012742 000570 MOV #570,-(R2) ;MOVE TO MAILBOX # ***** 570 *****
7664 024100 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7665 024102 000000 HALT ;MODE 4 REGISTER NOT DECREMENTED BY 1
7666 ; OR SEQUENCE ERROR
7667
7668 :*****
7669 :TEST 247 TEST MFPS MODE 5
7670 :*****
7671 024104 005212 TS247: INC (R2) ;UPDATE TEST NUMBER
7672 024106 022712 000247 CMP #247,(R2) ;SEQUENCE ERROR?
7673 024112 001033 BNE TS250-10 ;BR TO ERROR HALT ON SEQ ERROR
7674 024114 012700 000410 MOV #410,R0 ;R0=410
7675 024120 012737 177777 000000 MOV #-1,@#0 ;LOC. 0=-1
7676 024126 005037 177776 CLR @#PS ;PS=0
7677 024132 106750 MFPS @-(R0) ;TRY MFPS W/MODE 5
7678 024134 103403 BCS MFPS5A ;BR TO ERROR IF C-BIT SET
7679 024136 102402 BVS MFPS5A ;BR TO ERROR IF V-BIT SET
```

```
7680 024140 100401      BMI      MFPS5A      ;BR TO ERROR IF N-BIT SET
7681 024142 001404      BEQ      MFPS5B
7682                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7683                                     ;          CONDITIONAL BRANCH INST. AND <====
7684                                     ;          REPLACE THE MOVE INSTRUCTION <====
7685                                     ;          WHICH FOLLOWS W/ 763 <====
7686 024144      MFPS5A:
7687 024144 012742 000571      MOV      #571,-(R2)    ;MOVE TO MAILBOX # ***** 571 *****
7688 024150 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7689 024152 000000      HALT
7690 024154 022737 000377 000000 MFPS5B: CMP      #377,@#0    ;COND. CODES INCORRECT
7691 024162 001404      BEQ      MFPS5C      ;CHECK DEST. DATA
7692                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7693                                     ;          CONDITIONAL BRANCH INST. AND <====
7694                                     ;          REPLACE THE MOVE INSTRUCTION <====
7695                                     ;          WHICH FOLLOWS W/ 753 <====
7696 024164 012742 000572      MOV      #572,-(R2)    ;MOVE TO MAILBOX # ***** 572 *****
7697 024170 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7698 024172 000000      HALT
7699 024174 020027 000406 MFPS5C: CMP      R0,#406    ;DEST DATA INCORRECT
7700 024200 001404      BEQ      TS250      ;CHECK MODE 5 REGISTER
7701                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7702                                     ;          CONDITIONAL BRANCH INST. AND <====
7703                                     ;          REPLACE THE MOVE INSTRUCTION <====
7704                                     ;          WHICH FOLLOWS W/ 744 <====
7705 024202 012742 000573      MOV      #573,-(R2)    ;MOVE TO MAILBOX # ***** 573 *****
7706 024206 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7707 024210 000000      HALT
7708                                     ;MODE 5 REGISTER NOT DECREMENTED BY 2
7709                                     ; OR SEQUENCE ERROR
7710
7711 ;*****
7712 ;TEST 250 TEST MFPS MODE 6
7713 ;*****
7713 024212 005212      TS250: INC      (R2)        ;UPDATE TEST NUMBER
7714 024214 022712 000250      CMP      #250,(R2)    ;SEQUENCE ERROR?
7715 024220 001034      BNE      TS251-10    ;BR TO ERROR HALT ON SEQ ERROR
7716 024222 012700 000401      MOV      #401,R0      ;R0=410
7717 024226 005037 000000      CLR      @#0          ;LOC. 0=0
7718 024232 012737 000252 177776 MFPS      #252,@#PS    ;PS=252
7719 024240 106760 177377      MFPS      -401(R0)    ;TRY MFPS W/MODE 6
7720 024244 102403      BVS      MFPS6A      ;BR TO ERROR IF V-BIT SET
7721 024246 103402      BCS      MFPS6A      ;BR TO ERROR IF C-BIT SET
7722 024250 001401      BEQ      MFPS6A      ;BR TO ERROR IF Z-BIT SET
7723 024252 100404      BMI      MFPS6B
7724                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7725                                     ;          CONDITIONAL BRANCH INST. AND <====
7726                                     ;          REPLACE THE MOVE INSTRUCTION <====
7727                                     ;          WHICH FOLLOWS W/ 762 <====
7728 024254      MFPS6A:
7729 024254 012742 000574      MOV      #574,-(R2)    ;MOVE TO MAILBOX # ***** 574 *****
7730 024260 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7731 024262 000000      HALT
7732 024264 022737 000252 000000 MFPS6B: CMP      #252,@#0    ;COND. CODES INCORRECT
7733 024272 001404      BEQ      MFPS6C      ;CHECK DEST. DATA
7734                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7735                                     ;          CONDITIONAL BRANCH INST. AND <====
```

```
7736
7737
7738 024274 012742 000575      MOV    #575,-(R2)      ;
7739 024300 005242              INC    -(R2)          ; REPLACE THE MOVE INSTRUCTION <====
7740 024302 000000              HALT                               ; WHICH FOLLOWS W/ 752 <====
7741 024304 022700 000401      MFPS6C: CMP    #401,R0 ; MOVE TO MAILBOX # ***** 575 *****
7742 024310 001404              BEQ    TS251          ; SET MSGTYP TO FATAL ERROR
7743
7744
7745
7746
7747 024312 012742 000576      MOV    #576,-(R2)      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7748 024316 005242              INC    -(R2)          ; CONDITIONAL BRANCH INST. AND <====
7749 024320 000000              HALT                               ; REPLACE THE MOVE INSTRUCTION <====
7750
7751
7752
7753
7754
7755 024322 005212              TS251: INC    (R2)      ; UPDATE TEST NUMBER
7756 024324 022712 000251      CMP    #251,(R2)      ; SEQUENCE ERROR?
7757 024330 001034              BNE    TS252-10       ; BR TO ERROR HALT ON SEQ ERROR
7758 024332 012700 000777      MOV    #777,R0        ; R0=777
7759 024336 005037 000000      CLR    @#0            ; LOC. 0=0
7760 024342 012737 000125 177776  MOV    #125,@#PS      ; PS=125
7761 024350 106770 177407      MFPS   @-371(R0)      ; TRY MFPS W/MODE 7
7762 024354 102403              BVS    MFPS7A         ; BR TO ERROR IF V-BIT SET
7763 024356 103002              BCC    MFPS7A         ; BR TO ERROR IF C-BIT SET
7764 024360 001401              BEQ    MFPS7A         ; BR TO ERROR IF Z-BIT SET
7765 024362 100004              BPL    MFPS7B
7766
7767
7768
7769
7770 024364              MFPS7A:
7771 024364 012742 000577      MOV    #577,-(R2)      ;
7772 024370 005242              INC    -(R2)          ; MOVE TO MAILBOX # ***** 577 *****
7773 024372 000000              HALT                               ; SET MSGTYP TO FATAL ERROR
7774 024374 022737 042400 000000  MFPS7B: CMP    #42400,@#0 ; CONDITION CODE INCORRECT
7775 024402 001404              BEQ    MFPS7C         ; CHECK DESTINATION DATA
7776
7777
7778
7779
7780 024404 012742 000600      MOV    #600,-(R2)      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7781 024410 005242              INC    -(R2)          ; CONDITIONAL BRANCH INST. AND <====
7782 024412 000000              HALT                               ; REPLACE THE MOVE INSTRUCTION <====
7783 024414 022700 000777      MFPS7C: CMP    #777,R0 ; WHICH FOLLOWS W/ 752 <====
7784 024420 001404              BEQ    TS252          ; MOVE TO MAILBOX # ***** 600 *****
7785
7786
7787
7788
7789 024422 012742 000601      MOV    #601,-(R2)      ;
7790 024426 005242              INC    -(R2)          ; MOVE TO MAILBOX # * ***** 601 *****
7791 024430 000000              HALT                               ; SET MSGTYP TO FATAL ERROR
; MODE 7 REGISTER MODIFIED
```

7792
7793
7794
7795
7796
7797
7798
7799
7800
7801
7802
7803
7804
7805
7806
7807
7808
7809
7810
7811
7812
7813
7814
7815
7816
7817
7818
7819
7820
7821
7822
7823
7824

024432 005212
024434 022712 000252
024440 001010
024442 012737 000357 177776
024450 000005
024452 022737 000357 177776
024460 001404

024462 012742 000602
024466 005242
024470 000000

024472

; OR SEQUENCE ERROR

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

TEST 252 TEST THAT RESET DOES NOT CLEAR PSW

TS252: INC (R2) ;UPDATE TEST NUMBER
CMP #252,(R2) ;SEQUENCE ERROR?
BNE TS253-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #357,@#PS ;MOV ONES TO PSW
RESET ;
CMP #357,@#PS ;PSW CORRECT?
BEQ TS253

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
; MOVE TO MAILBOX # ***** 602 *****
; SET MSGTYP TO FATAL ERROR
; RESET ALTERED PSW
; OR SEQUENCE ERROR

REST:

THE FOLLOWING TEST CHECKS THE INDEPENDENT FUNCTIONING OF BASIC

7825
7826
7827
7828
7829
7830 024472 005212
7831 024474 022712 000253
7832 024500 001022
7833 024502 012767 000340 153266
7834 024510 052767 140000 153260
7835 024516 012706 000001
7836 024522 000241
7837 024524 006106
7838 024526 103376
7839 024530 001407
7840 024532 042767 140000 153236
7841 024540 012742 000603
7842 024544 005242
7843 024546 000000
7844 024550 042767 140000 153220
7845
7846
7847
7848
7849
7850
7851
7852
7853
7854
7855
7856
7857 024556 005212
7858 024560 022712 000254
7859 024564 001046
7860 024566 012767 000340 153202
7861 024574 052767 140000 153174
7862 024602 012706 177777
7863 024606 022706 177777
7864 024612 001407
7865 024614 042767 140000 153154
7866 024622 012742 000604
7867 024626 005242
7868 024630 000000
7869 024632 042767 140000 153136
7870 024640 022706 177777
7871 024644 001004
7872
7873
7874
7875
7876 024646 012742 000605
7877 024652 005242
7878 024654 000000
7879 024656 005006
7880 024660 052767 140000 153110

```
;DATA PATH COMPONENTS WITH USER MODE SET.  
:*****  
:TEST 253 TEST USER MODE R6 CAN HOLD A ONE IN EVERY POSITION  
:*****  
TS253: INC (R2) ;UPDATE TEST NUMBER  
CMP #253,(R2) ;SEQUENCE ERROR?  
BNE TS254-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #340,PS ;LOCK OUT INTERRUPTS WHILE PLAYING WITH R6  
BIS #USRM,PS ;SET USER MODE  
MOV #1,R6 ;SET BIT0  
CLC ;CLEAR C-BIT  
USP1: ROL R6 ;ROTATE 1 POSITION  
BCC USP1 ;BR IF NOT ALL DONE  
BEQ USP1A ;BR IF NO BITS PICKED  
BIC #USRM,PS ;CLEAR USER MODE  
MOV #603,-(R2) ;MOVE TO MAILBOX # ***** 603 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;USER MODE R6 PICKED A BIT  
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  
:*****  
TS254: INC (R2) ;UPDATE TEST NUMBER  
CMP #254,(R2) ;SEQUENCE ERROR?  
BNE USP4-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #340,PS ;LOCK OUT INTERRUPTS WHILE PLAYING WITH R6  
BIS #USRM,PS ;SET USER MODE  
MOV #-1,R6 ;SET USER R6 TO ALL ONES  
CMP #-1,R6 ;READ AND CHECK USER R6  
BEQ USP2 ;BR IF NO ERROR  
BIC #USRM,PS ;CLEAR USER MODE  
MOV #604,-(R2) ;MOVE TO MAILBOX # ***** 604 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;USER R6 WILL NOT HOLD ALL ONES  
USP2: BIC #USRM,PS ;SET KERNEL MODE  
CMP #-1,R6 ;KERNEL MODE R6 ADDR. FROM USER MODE?>>  
BNE USP3  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 747 <====  
MOV #605,-(R2) ;MOVE TO MAILBOX # ***** 605 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;DUAL ADDRESSING ERROR USER/KERNEL R6  
USP3: CLR R6 ;CLEAR KERNEL MODE SP  
BIS #USRM,PS ;SET USER MODE
```

```

7881 024666 022706 177777          CMP      #-1,R6          ;CHECK USEF R6 NOT ADDR. FROM KERNEL MODE
7882 024672 042767 140000 153076  BIC      #USRM,PS      ;CLEAR USER MODE
7883 024700 001404          BEQ      USP4          ;BR IF NO ERROR
7884 024702 012742 000606          MOV      #606,-(R2)    ;MOVE TO MAILBOX # ***** 606 *****
7885 024706 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7886 024710 000000          HALT                    ;DUAL ADDRESSING ERROR OR SEQUENCE ERROR
7887 024712 012706 001000  USP4:  MOV      #STBOT,R6    ;RESTORE SP USER
7888 024716 042767 140000 153052  BIC      #USRM,PS      ;SET KERNEL MODE
7889 024724 012706 001000          MOV      #STBOT,R6    ;RESTORE SP KERNEL

```

```

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

```

```

:TEST 255      TEST MFPI WITH R6 IN MODE 0
:*****

```

```

7899 024730 005212          TS255:  INC      (R2)          ;UPDATE TEST NUMBER
7900 024732 022712 000255          CMP      #255,(R2)    ;SEQUENCE ERROR?
7901 024736 001032          BNE      TS256-10     ;BR TO ERROR HALT ON SEQ ERROR
7902 024740 012706 001000          MOV      #STBOT,R6    ;INITIALIZE KERNEL STACK POINTER
7903 024744 012767 140000 153024  MOV      #USRM,PS      ;SET USER MODE/PREVIOUS KERNEL
7904 024752 012706 027324          MOV      #USTBOT,R6   ;INITIALIZE USER STACK POINTER
7905 024756 006506          MFPI    R6            ;TRY MFPI WITH MODE 0
7906 024760 022767 140000 153010  CMP      #140000,PS    ;CHECK PSW
7907 024766 001407          BEQ      MFPI0        ;BR IF NO ERROR
7908 024770 042767 140000 153000  BIC      #USRM,PS      ;CLEAR USER MODE
7909 024776 012742 000607          MOV      #607,-(R2)    ;MOVE TO MAILBOX # ***** 607 *****
7910 025002 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7911 025004 000000          HALT                    ;INCORRECT PSW FROM MFPI
7912 025006 042767 140000 152762  MFPI0:  BIC      #USRM,PS      ;CLEAR USER MODE
7913 025014 022767 001000 002300  CMP      #STBOT,USTBOT-2 ;CHECK DATA ON STACK
7914 025022 001404          BEQ      MFPI0A       ;BR IF NO ERROR
7915 025024 012742 000610          MOV      #610,-(R2)    ;MOVE TO MAILBOX # ***** 610 *****
7916 025030 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7917 025032 000000          HALT                    ;INCORRECT DATA FROM MFPI
7918 025034          MFPI0A:

```

```

:*****
:TEST 256      TEST MTPI WITH R6 IN MODE 0
:*****

```

```

7923 025034 005212          TS256:  INC      (R2)          ;UPDATE TEST NUMBER
7924 025036 022712 000256          CMP      #256,(R2)    ;SEQUENCE ERROR?
7925 025042 001035          BNE      TS257-10     ;BR TO ERROR HALT ON SEQ ERROR
7926 025044 012767 000340 152724  MOV      #340,PS      ;SET KERNEL MODE AND LOCK OUT INTERRUPTS
7927 025052 005006          CLR      R6            ;INITIALIZE KERNEL R6
7928 025054 052767 140000 152714  BIS      #USRM,PS      ;SET USER MODE/PREVIOUS KERNEL
7929 025062 012706 027324          MOV      #USTBOT,R6   ;INITIALIZE USER STACK POINTER
7930 025066 012746 001000          MOV      #STBOT,-(R6) ;SET UP TARGET DATA
7931 025072 006606          MTPI    R6            ;TRY MODE 0 MTPI
7932 025074 022767 140340 152674  CMP      #140340,PS    ;CHECK PSW
7933 025102 001407          BEQ      MTPI0        ;BR IF NO ERROR
7934 025104 042767 140000 152664  BIC      #USRM,PS      ;CLEAR USER MODE
7935 025112 012742 000611          MOV      #611,-(R2)    ;MOVE TO MAILBOX # ***** 611 *****
7936 025116 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR

```

```

7937 025120 000000
7938 025122 042767 140000 152646 MTP10: HALT ;PS INCORRECT FOLLOWING MTPI
7939 025130 020627 001000 BIC #USRM,PS ;SET KERNEL MODE
7940 025134 001404 CMP R6,#STBOT ;CHECK TARGET DATA
7941 BEQ TS257
7942 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7943 ; CONDITIONAL BRANCH INST. AND <= --
7944 ; REPLACE THE MOVE INSTRUCTION <-- --
7945 ; WHICH FOLLOWS W/ 742 <== --
7945 025136 012742 000612 MOV #612,-(R2) ;MOVE TO MAILBOX # ***** 612 *****
7946 025142 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7947 025144 000000 HALT ;DATA INCORRECT FOLLOWING MTPI
7948 ; OR SEQUENCE ERROR
7949
7950
7951
7952
7953

```

```

: 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

```

7977 025146 005212
7978 025150 022712 000257 TS257: INC (R2) ;UPDATE TEST NUMBER
7979 025154 001062 BNE ER ;SEQUENCE ERROR?
7980 025156 012700 027214 SETUP: MOV #BRTAB,R0 ;BR TO ERROR HALT ON SEQ ERROR
7981 025162 012704 027252 MOV #YNTAB,R4 ;INITIALIZE BRANCH TABLE POINTER
7982 025166 012767 000017 000142 MOV #15.,BRCT ;INITIALIZE YES/NO BRANCH MAP POINTER
7983 025174 012067 000110 SETBR: MOV (R0)+,BRH ;INITIALIZE BRANCH TABLE COUNT
7984 025200 012401 MOV (R4)+,R1 ;GET NEXT BRANCH INST.
7985 025202 012767 177777 000074 MOV #-1,CC1 ;GET NEXT BRANCH MAP
7986 025210 012703 000020 MOV #16.,R3 ;INITIALIZE CONDITION CODE VALUE
7987 025214 005267 000064 SETCC: INC CC1 ;INITIALIZE CONDITION CODE COUNT
7988 025220 032701 100000 BIT #100000,R1 ;SET FOR NEXT CC VALUE
7989 025224 013705 177776 MOV @#177776,R5 ;SEE IF SHOULD BR W/ THESE CC'S
7990 025230 042705 177773 BIC #177773,R5 ;SIMULATE A JNE
7991 025234 000165 025240 JMP +4(R5) ; (JUMP NOT EQUAL)
7992 025240 000167 000020 JMP SET2BR ; TO SET2BR

```

```

7993 025244 012767 025340 000042      MOV      #CONT,NBR      ;SET TO CONTINUE IF NO BRANCH
7994 025252 012767 025322 000040      MOV      #ER,YBR       ;SET TO REPORT ERROR IF BRANCH
7995 025260 000167 000014      JMP      AROUND        ;GO AROUND OPPOSITE CONDITION
7996 025264 012767 025322 000022  SET2BR: MOV      #ER,NBR      ;SET TO REPORT ERROR IF NO BRANCH
7997 025272 012767 025340 000020      MOV      #CONT,YBR     ;SET TO CONTINUE IF BRANCH
7998 025300 006101      AROUND: ROL      R1      ;UPDATE BIT MAP
7999
8000 025302 012737      MOV      (PC)+,@(PC)+  ;SET CONDITION CODE
8001 025304 000000      CC1:    0              ;NEW CC VALUE GOES HERE
8002 025306 177776      177776
8003 025310 000000      BRH:    0              ;BRANCH INST. GOES HERE
8004 025312 000137      JMP      @(PC)+        ;THIS JUMP IF NO BRANCH
8005 025314 000000      NBR:    0              ;WHERE TO GO IF NO BRANCH OCCURS
8006 025316 000137      JMP      @(PC)+        ;THIS JUMP IF BRANCH OCCURS
8007 025320 000000      YBR:    0              ;WHERE TO GO IF BRANCH OCCURS
8008 025322 012702 000304      ER:     MOV      #STESTN,R2 ;RESTORE POINTER
8009 025326 012742 000613      MOV      #613,-(R2)    ;MOVE TO MAILBOX # ***** 613 *****
8010 025332 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
8011 025334 000000      HALT
8012 025336 000000      BRCT:   0
8013 025340 005303      CONT:   DEC      R3      ;CC'S DONE?
8014 025342 013705 177776      MOV      @#177776,R5   ;SIMULATE A JNE
8015 025346 042705 177773      BIC      #177773,R5    ;
8016 025352 000165 025356      JMP      .+4(R5)       ; (JUMP NOT EQUAL)
8017 025356 000167 177632      JMP      SETCC        ; TO SETCC
8018 025362 005367 177750      DEC      BRCT         ;BR'S DONE?
8019 025366 013705 177776      MOV      @#177776,R5   ;SIMULATE A JNE
8020 025372 042705 177773      BIC      #177773,R5    ;
8021 025376 000165 025402      JMP      .+4(R5)       ; (JUMP NOT EQUAL)
8022 025402 000167 177566      JMP      SETBR        ; TO SETBR
8023
8024
8025
8026
8027
8028
8029
8030
8031
8032
8033
8034
8035 025406 005212 000260      TS260: INC      (R2)      ;UPDATE TEST NUMBER
8036 025410 022712 000260      CMP      #260,(R2)    ;SEQUENCE ERROR?
8037 025414 001051      BNE      DAERR        ;BR TO ERROR HALT ON SEQ ERROR
8038 025416 005000      BITCLR: CLR      R0      ;INITIALIZE ALL REGISTERS
8039 025420 005001      CLR      R1
8040 025422 005002      CLR      R2
8041 025424 005003      CLR      R3
8042 025426 005004      CLR      R4
8043 025430 005005      CLR      R5
8044 025432 052700 000001      BITSET: BIS      #1,R0   ;SET R0=1
8045 025436 052701 000002      BIS      #2,R1         ;R1=2
8046 025442 052702 000004      BIS      #4,R2         ;R2 4
8047 025446 052703 000010      BIS      #10,R3        ;R3 10
8048 025452 052704 000020      BIS      #20,R4        ;R4 20

```

```

;*****
;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

```

8049	025456	052705	000040	BIS	#40,R5	:R5=40
8050	025462	052706	001000	BIS	#1000,R6	:R6=1000
8051	025466	022706	001000	BITCHK: CMP	#1000,R6	:TEST THAT NO DUAL ADDRESSING OCCURRED
8052	025472	001022		BNE	DAERR	:BR TO ERROR HALT IF ANY OTHER BITS ARE SET
8053	025474	022705	000040	CMP	#40,R5	
8054	025500	001017		BNE	DAERR	
8055	025502	022704	000020	CMP	#20,R4	
8056	025506	001014		BNE	DAERR	
8057	025510	022703	000010	CMP	#10,R3	
8058	025514	001011		BNE	DAERR	
8059	025516	022702	000004	CMP	#4,R2	
8060	025522	001006		BNE	DAERR	
8061	025524	022701	000002	CMP	#2,R1	
8062	025530	001003		BNE	DAERR	
8063	025532	022700	000001	CMP	#1,R0	
8064	025536	001404		BEQ	BITCON	

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

8069	025540			DAERR:		
8070	025540	012742	000614	MOV	#614,-(R2)	:MOVE TO MAILBOX # ***** 514 *****
8071	025544	005242		INC	-(R2)	:SET MSGTYP TO FATAL ERROR
8072	025546	000000		HALT		:DUAL ADDRESSING ERROR
8073	025550	012702	000304	BITCON: MOV	#\$TESTN,R2	:RESTORE POINTER

8074
8075
8076
8077
8078
8079
8080
8081
8082
8083
8084
:*****
: 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.
:*****

8085
8086
8087
8088
8089
8090
8091
8092
8093
8094
8095
8096
8097
8098
8099
8100
8101
8102
8103
8104

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

TS261:	INC	(R2)				:UPDATE TEST NUMBER
8086	025556	022712	000261	CMP	#261,(R2)	:SEQUENCE ERROR?
8087	025562	001012		BNE	BTERR	:BR TO ERROR HALT ON SEQ ERROR
8088	025564	052737	170357	BIS	#170357,@#PS	:SET ALL POSSIBLE BITS IN PSW
8089	025572	105037	177776	CLRB	@#PS	:CLR PR LEVEL AND CC'S
8090	025576	013700	177776	MOV	@#PS,R0	:COPY CONTENTS OF PSW
8091	025602	022700	170004	CMP	#170004,R0	:TEST THAT CLRB AFFECTED ONLY LOW BYTE
8092	025606	001006		BNE	BTCON	:CONTINUE IF OK
8093	025610	005037	177776	BTERR: CLR	@#PS	:RETURN TO KERNEL MODE
8094	025614	012742	000615	MOV	#615,-(R2)	:MOVE TO MAILBOX # ***** 615 *****
8095	025620	005242		INC	-(R2)	:SET MSGTYP TO FATAL ERROR
8096	025622	000000		HALT		:BYTE INSTRUCTION ALTERED PSW
8097	025624	005037	177776	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.
:*****

8105
8106
8107
8108 025630 005212
8109 025632 022712 000262
8110 025636 001010
8111 025640 000277
8112 025642 000252
8113 025644 000167 000000
8114 025650 100403
8115 025652 001002
8116 025654 102401
8117 025656 103404
8118
8119
8120
8121
8122 025660
8123 025660 012742 000616
8124 025664 005242
8125 025666 000000
8126
8127
8128
8129
8130
8131
8132
8133
8134
8135
8136
8137
8138
8139
8140
8141
8142
8143
8144
8145 025670 005212
8146 025672 022712 000263
8147 025676 001062
8148 025700 012767 000240 000024
8149 025706 012767 000017 000032
8150 025714 012767 000261 000102
8151 025722 012767 000001 000110
8152 025730 000277
8153 025732 000000
8154 025734 013704 177776
8155 025740 042704 177760
8156 025744 022704
8157 025746 000000
8158 025750 001404
8159
8160

```
*****  
:TEST 262 TEST THAT JMP INSTRUCTION DOES NOT AFFECT CONDITION CODES  
*****  
TS262: INC (R2) ;UPDATE TEST NUMBER  
CMP #262,(R2) ;SEQUENCE ERROR?  
BNE TS263-10 ;BR TO ERROR HALT ON SEQ ERROR  
SCC  
+CLN!CLV ;CC=0101  
JMP JMP ;JUMP TO TEST PSW  
JMP: BMI JMPERR ;BR TO ERROR HALT IF N-BIT IS SET  
BNE JMPERR ;BR TO ERROR HALT IF Z-BIT IS CLEAR  
BVS JMPERR ;BR TO ERROR HALT IF V-BIT IS SET  
BCS TS263  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===  
; CONDITIONAL BRANCH INST. AND <===  
; REPLACE THE MOVE INSTRUCTION <===  
; WHICH FOLLOWS W/ 767 <===  
JMPERR: MOV #616,-(R2) ;MOVE TO MAILBOX # ***** 616 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;JMP INSTRUCTION AFFECTED CC'S  
; OR SEQUENCE ERROR
```

```
*****  
: THIS TEST VERIFIES THE SET AND CLEAR CONDITION CODE INSTRUCTIONS.  
: THE TEST CONSISTS OF TWO ROUTINES, ONE TO TEST ALL CLEAR CC  
: INSTRUCTIONS, AND THE SECOND TO TEST ALL SET CC INSTRUCTIONS. ALL  
: POSSIBLE COMBINATIONS OF CONDITION CODES ARE TESTED, INCLUDING NOP'S.  
: TO TEST THE CLEAR CC INSTRUCTIONS, ALL CONDITION CODES ARE  
: INITIALLY SET. THE INSTRUCTION IS EXECUTED, AND THE PSW IS CHECKED  
: TO VERIFY THE PROPER COMBINATION OF CONDITION CODES.  
: TO TEST THE SET CC INSTRUCTIONS, THE CONDITION CODES ARE  
: INITIALLY CLEARED, AND ONLY THE REQUIRED BITS ARE SET BY THE SET CC  
: INSTRUCTION. THE CONTENTS OF THE PSW ARE CHECKED TO VERIFY THAT  
: ONLY THE REQUIRED BITS WERE SET.  
*****
```

```
*****  
:TEST 263 TEST SET CC AND CLEAR CC INSTRUCTIONS  
*****  
TS263: INC (R2) ;UPDATE TEST NUMBER  
CMP #263,(R2) ;SEQUENCE ERROR?  
BNE CCERR ;BR TO ERROR HALT ON SEQ ERROR  
MOV #240,CC3 ;INITIALIZE CLR CC INSTRUCTION CODES  
MOV #17,CC2 ;INITIALIZE OCTAL MAP  
MOV #261,SC3 ;INITIALIZE SET CC INSTRUCTION CODES  
MOV #1,SC4 ;INITIALIZE OCTAL MAP  
CLRCD: SCC ;SET ALL CONDITION CODES  
CC3: 0 ;CONDITION CODE INSTRUCTION  
MOV @PS,R4 ;COPY THE PSW  
BIC #177760,R4 ;ISOLATE CONDITION CODES  
CMP (PC)+,R4 ;CHECK THAT PROPER CC'S WERE CLEARED  
CC2: 0 ;OCTAL REPRESENTATION OF CC'S  
BEQ CON1  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===  
; CONDITIONAL BRANCH INST. AND <===
```

8161								:	REPLACE THE MOVE INSTRUCTION	<===
8162								:	WHICH FOLLOWS W/ 752	<===
8163	025752	012742	000617					:	MOVE TO MAILBOX # ***** 617 *****	
8164	025756	005242						:	SET MSGTYP TO FATAL ERROR	
8165	025760	000000						:	CLEAR CC INSTRUCTION FAILED	
8166	025762	005367	177760		CON1:	DEC	CC2	:	SET NEXT OCTAL MAP OF CC'S	
8167	025766	005267	177740			INC	CC3	:	GET NEXT CLEAR CC INSTRUCTION	
8168	025772	026727	177734	000257		CMP	CC3,#257	:	TEST FOR CCC INSTRUCTION	
8169	026000	003753				BLE	CLRCO	:	GO TEST NEXT INSTRUCTION IF NOT FOUND	
8170	026002	026727	177724	000260		CMP	CC3,#2:0	:	CHECK FOR NOP=260	
8171	026010	001004				BNE	SETCD	:	GO TEST SET CC INSTRUCTIONS	
8172	026012	012767	000017	177726		MOV	#17,CC2	:	SET OCTAL MAP TO TEST NOP	
8173	026020	000743				BR	CLRCO	:	GO TEST NOP	
8174	026022	000257			SETCD:	CCC		:	CLEAR ALL CONDITION CODES	
8175	026024	000000			SC3:	0		:	CONDITION CODE INSTRUCTION	
8176	026026	013704	177776			MOV	@#PS,R4	:	COY PSW	
8177	026032	042704	177760			BIC	#177760,R4	:	CLEAR AWAY UNWANTED BITS	
8178	026036	022704				CMP	(PC)+,R4	:	CHECK THAT PROPER CC'S WERE SET	
8179	026040	000000			SC4:	0		:	OCTAL REPRESENTATION OF CC'S	
8180	026042	001404				BEQ	CON2	:	TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<===
8181								:		
8182								:	CONDITIONAL BRANCH INST. AND	<===
8183								:	REPLACE THE MOVE INSTRUCTION	<===
8184								:	WHICH FOLLOWS W/ 715	<===
8185	026044				CCERR:			:		
8186	026044	012742	000620			MOV	#620,-(R2)	:	MOVE TO MAILBOX # ***** 620 *****	
8187	026050	005242				INC	-(R2)	:	SET MSGTYP TO FATAL ERROR	
8188	026052	000000				HALT		:	SET CC FAILED OR SEQUENCE ERROR	
8189	026054	005267	177760		CON2:	INC	SC4	:	SET NEXT OCTAL MAP	
8190	026060	005267	177740			INC	SC3	:	PREPARE NEXT SET CC INSTRUCTION	
8191	026064	026727	177734	000277		CMP	SC3,#277	:	FINISHED?	
8192	026072	003753				BLE	SETCD	:	BR IF NO	
8193	026074	000167	000006			JMP	MORO	:	JUMP TO NEXT TESTS	

8194
8195
8196
8197
8198
8199
8200
8201
8202
8203
8204
8205
8206
8207
8208
8209
8210
8211
8212
8213
8214
8215
8216
8217
8218
8219
8220
8221
8222
8223
8224
8225
8226
8227
8228
8229
8230
8231
8232
8233
8234
8235
8236
8237
8238
8239
8240
8241
8242
8243
8244
8245
8246
8247
8248
8249

026100 000000 000000 000000
026106
026106 005212
026110 022712 000264
026114 001020
026116 005037 026100
026122 012700 026100
026126 060020
026130 022700 026102
026134 001404
026136 012742 000621
026142 005242
026144 000000
026146 022737 026102 026100
026154 001404
026156 012742 000622
026162 005242
026164 000000
026166 005212
026170 022712 000265
026174 001020
026176 005037 026100
026202 012700 026102
026206 060040
026210 022700 026100
026214 001404

```
*****  
:SBTTL TEST INSTRUCTIONS USING SAME REGISTER FOR SOURCE & DESTINATION  
:IN AUTO INCREMENT (DECREMENT) MODES AND  
:AUTO INCREMENT (DECREMENT) DEFERRED MODES,  
:CONTENTS OF THE REGISTER IN USED ARE  
:INCREMENTED (DECREMENTED) BY 2  
:BEFORE USED AS THE SOURCE OPERAND.  
:A: .WORD 0,0,0  
:MOR0:  
:*****  
:TEST 264 TEST AUTO-INCREMENT MODE, USING R0  
:*****  
TS264: INC (R2) ;UPDATE TEST NUMBER  
CMP #264,(R2) ;SEQUENCE ERROR?  
BNE TS265-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR @A ;CLEAR LOC A  
MOV #A,R0 ;R0 STORES ADDR OF A  
ADD R0,(R0)+ ;CHECK THAT R0 IS INCR BY 2 BEFORE  
;BEING USED AS THE SOURCE OPERAND  
;R0 INCR BY 2?  
CMP #A+2,R0  
BEQ MOR1  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 767 <====  
MOV #621,-(R2) ;MOVE TO MAILBOX # ***** 621 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;R0 WAS NOT INCREMENTED BY 2  
MOR1: CMP #A+2,@A ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE  
;BEING USED IN THE 'ADD' INSTR  
;LOC A CONTAINS (A+2)?  
BEQ TS265  
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
; CONDITIONAL BRANCH INST. AND <====  
; REPLACE THE MOVE INSTRUCTION <====  
; WHICH FOLLOWS W/ 757 <====  
MOV #622,-(R2) ;MOVE TO MAILBOX # ***** 622 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;WRONG SUM IN LOC A  
; OR SEQUENCE ERROR  
:*****  
:TEST 265 AUTO-DECREMENT MODE, USING R0  
:*****  
TS265: INC (R2) ;UPDATE TEST NUMBER  
CMP #265,(R2) ;SEQUENCE ERROR?  
BNE TS266-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR @A ;CLEAR LOC A  
MOV #A+2,R0 ;R0 STORES ADDR OF A+2  
ADD R0,-(R0) ;CHECK THAT R0 IS DECR BY 2 BEFORE  
;BEING USED AS THE SOURCE OPERAND  
;R0 DECR BY 2?  
CMP #A,R0  
BEQ MOR2
```



```

8250 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8251 ; CONDITIONAL BRANCH INST. AND <====
8252 ; REPLACE THE MOVE INSTRUCTION <====
8253 ; WHICH FOLLOWS W/ 767 <====
8254 026216 012742 000623 MOV #623,-(R2) ;MOVE TO MAILBOX # ***** 623 *****
8255 026222 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8256 026224 000000 HALT ;R0 WAS NOT DECREMENTED BY 2
8257
8258 026226 022737 026100 026100 MOR2: CMP #A,@#A ;CONTENT OF R0 WAS DECR BY 2 BEFORE
8259 ;BEING USED IN THE 'ADD' INSTR
8260 ;LOC A CONTAINS (R0)
8261 026234 001404 BEQ TS266
8262
8263 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8264 ; CONDITIONAL BRANCH INST. AND <====
8265 ; REPLACE THE MOVE INSTRUCTION <====
8266 ; WHICH FOLLOWS W/ 757 <====
8266 026236 012742 000624 MOV #624,-(R2) ;MOVE TO MAILBOX # ***** 624 *****
8267 026242 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8268 026244 000000 HALT ;WRONG SUM IN LOC A
8269 ; OR SEQUENCE ERROR
8270
8271
8272 *****
8273 ;TEST 266 TEST AUTO-INCREMENT DEFERRED MODE, USING R0
8274 026246 005212 TS266: INC (R2) ;UPDATE TEST NUMBER
8275 026250 022712 000266 CMP #266,(R2) ;SEQUENCE ERROR?
8276 026254 001044 BNE TS267-10 ;BR TO ERROR HALT ON SEQ ERROR
8277 026256 005037 026100 CLR @#A ;CLEAR LOC A
8278 026262 005037 026104 CLR @#A+4 ;CLEAR LOC A+4
8279 026266 012737 026100 026102 MOV #A,@#A+2 ;STORE ADDR A IN LOC A+2
8280 026274 012700 026102 MOV #A+2,R0 ;R0 STORES ADDR A+2
8281 026300 060030 ADD R0,@(R0)+ ;CHECK THAT R0 IS INCR BY 2 BEFORE
8282 ;BEING USED AS THE SOURCE OPERAND
8283 026302 022700 026104 CMP #A+4,R0 ;R0 INCR BY 2?
8284 026306 001404 BEQ MOR3
8285
8286 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8287 ; CONDITIONAL BRANCH INST. AND <====
8288 ; REPLACE THE MOVE INSTRUCTION <====
8289 ; WHICH FOLLOWS W/ 762 <====
8289 026310 012742 000625 MOV #625,-(R2) ;MOVE TO MAILBOX # ***** 625 *****
8290 026314 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8291 026316 000000 HALT ;R0 WAS NOT INCREMENTED BY 2
8292
8293 026320 022737 026100 026102 MOR3: CMP #A,@#A+2 ;LOC A+2 STILL STORES ADDR A?
8294 026326 001404 BEQ MOR4
8295
8296 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8297 ; CONDITIONAL BRANCH INST. AND <====
8298 ; REPLACE THE MOVE INSTRUCTION <====
8299 ; WHICH FOLLOWS W/ 752 <====
8299 026330 012742 000626 MOV #626,-(R2) ;MOVE TO MAILBOX # ***** 626 *****
8300 026334 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8301 026336 000000 HALT ;LOC A+2 STORES WRONG DATA
8302
8303 026340 022737 026104 026100 MOR4: CMP #A+4,@#A ;CHECK CONTENT OF R0 WAS INCR BY 2 BEFORE
8304 ;BEING USED IN THE 'ADD' INSTR
8305 026346 001404 BEQ MOR5

```

```

8306 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8307 ; CONDITIONAL BRANCH INST. AND <====
8308 ; REPLACE THE MOVE INSTRUCTION <====
8309 ; WHICH FOLLOWS W/ 742 <====
8310 026350 012742 000627 MOV #627,-(R2) ;MOVE TO MAILBOX # ***** 627 *****
8311 026354 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8312 026356 000000 HALT ;LOC A STORES WRONG DATA
8313
8314 026360 005737 026104 MOR5: TST @#A+4 ;LOC A+4 STILL STORES 0?
8315 026364 001404 BEQ TS267
8316
8317 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8318 ; CONDITIONAL BRANCH INST. AND <====
8319 ; REPLACE THE MOVE INSTRUCTION <====
8320 ; WHICH FOLLOWS W/ 733 <====
8320 026366 012742 000630 MOV #630,-(R2) ;MOVE TO MAILBOX # ***** 630 *****
8321 026372 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8322 026374 000000 HALT ;LOC A+4 DID NOT STAY CLFAR
8323 ; OR SEQUENCE ERROR
8324
8325 ;*****
8326 ;TEST 267 TEST AUTO-DECREMENT DEFERRED, USING R0
8327 ;*****
8328 026376 005212 TS267: INC (R2) ;UPDATE TEST NUMBER
8329 026400 022712 000267 CMP #267,(R2) ;SEQUENCE ERROR?
8330 026404 001044 BNE TS270-10 ;BR TO ERROR HALT ON SEQ ERROR
8331 026406 005037 026100 CLR @#A ;CLEAR LOC A
8332 026412 005037 026104 CLR @#A+4 ;CLEAR LOC A+4
8333 026416 012700 026104 MOV #A+4,R0 ;R0 STORES ADDR A+4
8334 026422 012737 026100 026102 MOV #A,@#A+2 ;STORE ADDR A IN LOC A+2
8335 026430 060050 ADD R0,@-(R0) ;CHECK THAT R0 IS DECR BY 2 BEFORE
8336 ;BEING USED AS THE SOURCE OPERAND
8337 026432 022700 026102 CMP #A+2,R0 ;R0 DECREMENTED BY 2?
8338 026436 001404 BEQ MOR6
8339
8340 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8341 ; CONDITIONAL BRANCH INST. AND <====
8342 ; REPLACE THE MOVE INSTRUCTION <====
8343 ; WHICH FOLLOWS W/ 762 <====
8343 026440 012742 000631 MOV #631,-(R2) ;MOVE TO MAILBOX # ***** 631 *****
8344 026444 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8345 026446 000000 HALT ;R0 WAS NOT DECREMENTED BY 2
8346
8347 026450 022737 026102 026100 MOR6: CMP #A+2,@#A ;CHECK CONTENT OF R0 WAS DECR BY 2 BEFORE
8348 ;BEING USED IN THE 'ADD' INSTR
8349 026456 001404 BEQ MOR7
8350
8351 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8352 ; CONDITIONAL BRANCH INST. AND <====
8353 ; REPLACE THE MOVE INSTRUCTION <====
8354 ; WHICH FOLLOWS W/ 752 <====
8354 026460 012742 000632 MOV #632,-(R2) ;MOVE TO MAILBOX # ***** 632 *****
8355 026464 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8356 026466 000000 HALT ;LOC A STORES WRONG DATA
8357
8358
8359 026470 022737 026100 026102 MOR7: CMP #A,@#A+2 ;LOC A+2 STILL STORES A?
8360 026476 001404 BEQ MOR8
8361 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====

```

8362						:	CONDITIONAL BRANCH INST. AND	<====
8363						:	REPLACE THE MOVE INSTRUCTION	<====
8364						:	WHICH FOLLOWS W/ 742	<====
8365	026500	012742	000633			:	MOVE TO MAILBOX # ***** 633 *****	
8366	026504	005242				:	SET MSGTYP TO FATAL ERROR	
8367	026506	000000				:	LOC A+2 STORES WRONG DATA	
8368						:		
8369	026510	005737	026104	MOR8:	TST	@A+4	LOC A+4 STILL STORES 0?	
8370	026514	001404			BEQ	TS270		
8371						:	TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
8372						:	CONDITIONAL BRANCH INST. AND	<====
8373						:	REPLACE THE MOVE INSTRUCTION	<====
8374						:	WHICH FOLLOWS W/ 733	<====
8375	026516	012742	000634			:	MOVE TO MAILBOX # ***** 634 *****	
8376	026522	005242				:	SET MSGTYP TO FATAL ERROR	
8377	026524	000000				:	LOC A+4 DID NOT STAY CLEAR	
8378						:	OR SEQUENCE ERROR	
8379						:		

```
8380 ;*****
8381 ;SBTTL INSTRUCTION USING PC AS SOURCE REGISTER
8382 ;
8383 ;IN INDEX, INDEX DEFERRED, RELATIVE, AND
8384 ;RELATIVE DEFERRED MODES, DESTINATION WILL CONTAIN
8385 ;THE PC COUNT OF THE CURRENT INSTRUCTION +4.
8386 ;
8387 ;*****
8388 ;TEST 270 TEST PC AS SOURCE IN MODE 0, USING R0
8389 ;*****
8390 026526 005212 TS270: INC (R2) ;UPDATE TEST NUMBER
8391 026530 022712 000270 CMP #270,(R2) ;SEQUENCE ERROR?
8392 026534 001006 BNE TS271-10 ;BR TO ERROR HALT ON SEQ ERROR
8393 026536 012700 177777 MOV #-1,R0 ;SET ALL 1 IN R0
8394 026542 010700 PCN01: MOV PC,R0 ;STORES PC IN R0
8395 026544 022700 026544 CMP #PCN01+2,R0 ;R0 STORES PC+2?
8396 026550 001404 BEQ TS271
8397 ;
8398 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8399 ; CONDITIONAL BRANCH INST. AND <====
8400 ; REPLACE THE MOVE INSTRUCTION <====
8401 ; WHICH FOLLOWS W/ 771 <====
8401 026552 012742 000635 MOV #635,-(R2) ;MOVE TO MAILBOX # ***** 635 *****
8402 026556 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8403 026560 000000 HALT ;R0 STORED WRONG VALUE
8404 ; OR SEQUENCE ERROR
8405 ;
8406 ;*****
8407 ;TEST 271 TEST PC AS SOURCE IN MODE 6, USING R0
8408 ;*****
8409 026562 005212 TS271: INC (R2) ;UPDATE TEST NUMBER
8410 026564 022712 000271 CMP #271,(R2) ;SEQUENCE ERROR?
8411 026570 001010 BNE TS272-10 ;BR TO ERROR HALT ON SEQ ERROR
8412 026572 012700 026100 MOV #A,R0 ;R0 STORES ADDR A
8413 026576 010760 000004 PCN2: MOV PC,4(R0) ;EFFECTIVE ADDR IS A+4
8414 026602 022737 026602 026104 CMP #PCN2+4,@#A+4 ;LOC A+4 STORES PC+4?
8415 026610 001404 BEQ TS272
8416 ;
8417 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8418 ; CONDITIONAL BRANCH INST. AND <====
8419 ; REPLACE THE MOVE INSTRUCTION <====
8420 ; WHICH FOLLOWS W/ 767 <====
8420 026612 012742 000636 MOV #636,-(R2) ;MOVE TO MAILBOX # ***** 636 *****
8421 026616 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8422 026620 000000 HALT ;LOC A+4 STORED WRONG VALUE
8423 ; OR SEQUENCE ERROR
8424 ;
8425 ;*****
8426 ;TEST 272 TEST PC AS SOURCE IN MODE 7, USING R0
8427 ;*****
8428 026622 005212 TS272: INC (R2) ;UPDATE TEST NUMBER
8429 026624 022712 000272 CMP #272,(R2) ;SEQUENCE ERROR?
8430 026630 001013 BNE TS273-10 ;BR TO ERROR HALT ON SEQ ERROR
8431 026632 012737 026100 026104 MOV #A,@#A+4 ;LOC A+4 STORES ADDR A
8432 026640 012700 026100 MOV #A,R0 ;R0 STORES ADDR A
8433 026644 010770 000004 PCN3: MOV PC,@4(R0) ;EFFECTIVE ADDR IS A
8434 026650 022737 026650 026100 CMP #PCN3+4,@#A ;LOC A STORES PC+4?
8435 026656 001404 BEQ TS273
```

```
8436  
8437  
8438  
8439  
8440 026660 012742 000637      MOV    #637,-(R2)      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
8441 026664 005242              INC    -(R2)           ; CONDITIONAL BRANCH INST. AND <====  
8442 026666 000000              HALT                    ; REPLACE THE MOVE INSTRUCTION <====  
                        ; WHICH FOLLOWS W/ 764 <====  
                        ; MOVE TO MAILBOX # ***** 637 *****  
                        ; SET MSGTYP TO FATAL ERROR  
                        ; LOC A STORED WRONG VALUE  
                        ; OR SEQUENCE ERROR
```

```
8445  
8446  
8447  
8448 026670 005212  
8449 026672 022712 000273      ;*****  
                        ;TEST 273      TEST PC AS SOURCE IN RELATIVE DEFERRED MODE ,USING R0  
                        ;*****
```

```
8448 TS273: INC    (R2)           ;UPDATE TEST NUMBER  
8449      CMP    #273,(R2)      ;SEQUENCE ERROR?  
8450      BNE    TS274-10       ;BR TO ERROR HALT ON SEQ ERROR  
8451 026700 012737 026102 026100  MOV    #A+2,@#A      ;LOC A STORES ADDR A+2  
8452 026706 010777 177166      PCN4:  MOV    PC,@A    ;EFFECTIVE ADDR IS A+2  
8453 026712 022737 026712 026102  CMP    #PCN4+4,@#A+2 ;LOC A+2 STORES PC+4?  
8454 026720 001404      BEQ    TS274
```

```
8455  
8456  
8457  
8458  
8459 026722 012742 000640      MOV    #640,-(R2)      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
8460 026726 005242              INC    -(R2)           ; CONDITIONAL BRANCH INST. AND <====  
8461 026730 000000              HALT                    ; REPLACE THE MOVE INSTRUCTION <====  
                        ; WHICH FOLLOWS W/ 766 <====  
                        ; MOVE TO MAILBOX # ***** 640 *****  
                        ; SET MSGTYP TO FATAL ERROR  
                        ; LOC A+2 STORED WRONG VALUE  
                        ; OR SEQUENCE ERROR
```

```
8464  
8465  
8466  
8467 026732 005212  
8468 026734 022712 000274      ;*****  
                        ;TEST 274      TEST PC AS SOURCE IN RELATIVE MODE ,USING R0  
                        ;*****
```

```
8467 TS274: INC    (R2)           ;UPDATE TEST NUMBER  
8468      CMP    #274,(R2)      ;SEQUENCE ERROR?  
8469      BNE    TS275-10       ;BR TO ERROR HALT ON SEQ ERROR  
8470 026742 005037 026100      CLR    @#A            ;CLEAR A  
8471 026746 010767 177126      PCN5:  MOV    PC,A     ;EFFECTIVE ADDR IS A  
8472 026752 022737 026752 026100  CMP    #PCN5+4,@#A   ;LOC A STORES PC+4?  
8473 026760 001404      BEQ    TS275
```

```
8474  
8475  
8476  
8477  
8478 026762 012742 000641      MOV    #641,-(R2)      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
8479 026766 005242              INC    -(R2)           ; CONDITIONAL BRANCH INST. AND <====  
8480 026770 000000              HALT                    ; REPLACE THE MOVE INSTRUCTION <====  
                        ; WHICH FOLLOWS W/ 767 <====  
                        ; MOVE TO MAILBOX # ***** 641 *****  
                        ; SET MSGTYP TO FATAL ERROR  
                        ; LOCATION A STORED WRONG VALUE  
                        ; OR SEQUENCE ERROR
```

```
8483  
8484  
8485  
8486  
8487  
8488 026772 005212  
8489 026774 022712 000275      ;*****  
                        ;SBTTL THE NEXT THREE TESTS EXERCISE MASKING ACTION OF MICROCODES.  
                        ;*****  
                        ;TEST 275      TEST SUB INSTRUCTION, SM=0, DM=2  
                        ;*****
```

```
8488 TS275: INC    (R2)           ;UPDATE TEST NUMBER  
8489      CMP    #275,(R2)      ;SEQUENCE ERROR?  
8490      BNE    TS276-10       ;BR TO ERROR HALT ON SEQ ERROR  
8491 027002 012737 052525 000000  MOV    #052525,@#0   ;SET UP LOC 0
```

```
8492 027010 012701 050505      MOV      #050505,R1      ;SET UP R1
8493 027014 005000              CLR      R0              ;CLEAR R0
8494 027016 160120              SUB      R1,(R0)+       ;SUBTRACTION, SM=0,DM=2
8495 027020 022737 002020 000000  CMP      #2020,@#0      ;CHECK DIFFERENCE AT LOC 0
8496 027026 001404              BEQ      TS276
8497
8498
8499
8500
8501 027030 012742 000642      MOV      #642,-(R2)     ;MOVE TO MAILBOX # ***** 642 *****
8502 027034 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
8503 027036 000000              HALT                    ;WRONG RESULT FROM SUBTRACTION
8504
8505
8506
8507
8508
8509
8510
8511
8512
8513
8514
8515
8516
8517
8518
8519
8520
8521
8522
8523
8524
8525
8526
8527
8528
8529
8530
8531
8532
8533
8534
8535
8536
8537
8538
8539
8540
8541
8542
8543
8544
8545
8546
8547
```

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

```
*****
;TEST 276 TEST MFPD WITH R0, IN MODE 2
*****
```

```
TS276: INC      (R2)          ;UPDATE TEST NUMBER
      CMP      #276,(R2)    ;SEQUENCE ERROR?
      BNE      TS277-10    ;BR TO ERROR HALT ON SEQ ERROR
      MOV      #052525,@#0 ;SET UP LOC 0
      CLR      R0          ;CLEAR R0
      MOV      #170000,PS   ;SET USER MODE ON, CURRENT & PREVIOUS
      MOV      #USTBOT,R6  ;SET USER STACK POINTER
      MFPD     (R0)+       ;MODE 2, MFPD
      CLR      PS          ;SET KERNEL MODE
      CMP      #052525,USTBOT-2 ;CHECK DATA ON STACK
      BEQ      BRMFPD      ;BR IF NO ERROR
      MOV      #643,-(R2)  ;MOVE TO MAILBOX # ***** 643 *****
      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
      HALT                    ;INCORRECT DATA FROM MFPD
```

BRMFPD:

```
*****
;TEST 277 TEST MTPD WITH R0, IN MODE 2
*****
```

```
TS277: INC      (R2)          ;UPDATE TEST NUMBER
      CMP      #277,(R2)    ;SEQUENCE ERROR?
      BNE      END1        ;BR TO ERROR HALT ON SEQ ERROR
      MOV      #170000,PS   ;SET USER MODE ON, CURRENT & PREVIOUS
      MOV      #USTBOT,R6  ;SET USER STACK POINTER
      MOV      #125252,-(R6) ;PUSH DATA IN USER STACK
      MOV      #0,@#0      ;CLEAR LOC 0
      CLR      R0          ;CLEAR R0
      MTPD     (R0)+       ;MODE 2, MTPD
      CLR      PS          ;SET KERNEL MODE
      CMP      #125252,@#0  ;CHECK DATA ON LOC 0
      BEQ      SECPRT      ;BR TO TRAP TEST IF NO ERROR
      MOV      #644,-(R2)  ;MOVE TO MAILBOX # ***** 644 *****
      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
      HALT                    ;INCORRECT DATA FROM MTPD
```

END1:

```
      MOV      #645,-(R2)  ;MOVE TO MAILBOX # ***** 645 *****
      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
      HALT                    ;SEQUENCE ERROR
```

```

8548
8549 027214 000402      BRTAB: BR      .+6
8550 027216 001002      BNE      .+6
8551 027220 001402      BEQ      .+6
8552 027222 002002      BGE      .+6
8553 027224 002402      BLT      .+6
8554 027226 003002      BGT      .+6
8555 027230 003402      BLE      .+6
8556 027232 100002      BPL      .+6
8557 027234 100402      BMI      .+6
8558 027236 101002      BHI      .+6
8559 027240 101402      BLOS     .+6
8560 027242 102002      BVC      .+6
8561 027244 102402      BVS      .+6
8562 027246 103002      BCC      .+6      ;SAME AS BHIS
8563 027250 103402      BCS      .+6      ;SAME AS BLO
8564
8565          000002      .RADIX 2
8566 027252 177777      YNTAB: 1111111111111111      ;BR
8567 027254 170360      1111000011110000      ;BNE: Z=0
8568 027256 007417      0000111100001111      ;BEQ: Z=1
8569 027260 146063      1100110000110011      ;BGE: N XOR V =0
8570 027262 031714      0011001111001100      ;BLT: N XOR V =1
8571 027264 140060      1100000000110000      ;BGT: Z+(N XOR V) =0
8572 027266 037717      0011111111001111      ;BLE: Z+(N XOR V) =1
8573
8574 027270 177400      1111111100000000      ;BPL: N=0
8575 027272 000377      0000000011111111      ;BMI: N=1
8576 027274 120240      1010000010100000      ;BHI: C+Z=0
8577 027276 057537      0101111101011111      ;BLOS: C+Z=1
8578 027300 146314      1100110011001100      ;BVC: V=0
8579 027302 031463      0011001100110011      ;BVS: V=1
8580 027304 125252      1010101010101010      ;BCC: C=0
8581 027306 052525      0101010101010101      ;BCS: C=1
8582          000010      .RADIX 8
8583
8584          .EVEN
8585 027310 000006      .BLKW 6
8586 027324      USTBOT:
8587      ;*****
8588      ; THE FOLLOWING ARE SPECIAL CPU TRAP
8589      ; HANDLERS TO TRAP AND REPORT SPECIAL TRAPS.
8590      ;
8591      ;*****
8592
8593 027324      T04:
8594 027324 012742 000646      MOV      #646,-(R2)      ;MOVE TO MAILBOX # ***** 646 *****
8595 027330 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
8596 027332 000000      HALT                    ;TRAPPED THRU LOC. 4
8597 027334
8598 027334 012742 000647      T010:
8599 027340 005242      MOV      #647,-(R2)      ;MOVE TO MAILBOX # ***** 647 *****
8600 027342 000000      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
8601 027344      HALT                    ;TRAPPED THRU LOC. 10
8602 027344 012742 000650      T014:
8603 027350 005242      MOV      #650,-(R2)      ;MOVE TO MAILBOX # ***** 650 *****
      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR

```

8604	027352	000000		HALT		;TRAPPED THRU LOC. 14		
8605	027354			T030:				
8606	027354	012742	000651	MOV	#651,-(R2)	;MOVE TO MAILBOX # *****	651	*****
8607	027360	005242		INC	-(R2)	;SET MSGTYP TO FATAL ERROR		
8608	027362	000000		HALT		;TRAPPED THRU LOC. 30		
8609	027364			T034:				
8610	027364	012742	000652	MOV	#652,-(R2)	;MOVE TO MAILBOX # *****	652	*****
8611	027370	005242		INC	-(R2)	;SET MSGTYP TO FATAL ERROR		
8612	027372	000000		HALT		;TRAPPED THRU LOC. 34		
8613	027374			T0114:				
8614	027374	012742	000653	MOV	#653,-(R2)	;MOVE TO MAILBOX # *****	653	*****
8615	027400	005242		INC	-(R2)	;SET MSGTYP TO FATAL ERROR		
8616	027402	000000		HALT		;TRAPPED THRU LOC. 114		
8617	027404			T0244:				
8618	027404	012742	000654	MOV	#654,-(R2)	;MOVE TO MAILBOX # *****	654	*****
8619	027410	005242		INC	-(R2)	;SET MSGTYP TO FATAL ERROR		
8620	027412	000000		HALT		;TRAPPED THRU LOC. 244		
8621	027414			T0250:				
8622	027414	012742	000655	MOV	#655,-(R2)	;MOVE TO MAILBOX # *****	655	*****
8623	027420	005242		INC	-(R2)	;SET MSGTYP TO FATAL ERROR		
8624	027422	000000		HALT		;TRAPPED THRU LOC. 250		
8625				.SBTTL	** STARTING OF TRAP TEST **			
8626	027424			SECPRT:				


```
8627
8628      000000      .REPT 0
8629
8630
8631      PART TWO:      F11 TRAP TEST, THIS IS THE SECOND
8632                        PART OF THE MAIN PROGRAM.
8633
8634
8635      ABSTRACT
8636
8637      THIS IS A TEST OF ALL OPERATIONS AND INSTRUCTIONS THAT CAUSE
8638      TRAPS. ALSO TESTED ARE TRAP OVERFLOW CONDITIONS,
8639      ODDITIES OF REGISTER 6, INTERRUPTS , THE RESET AND WAIT INSTRUCTIONS.
8640
8641      .ENDR
8642
8643      .LIST ME
8644      .NLIST MC,MD,CND
8645      .ABS
8646      SP=%6
8647      R6=%6
8648      TAB=%3
8649      LAST=%1
8650      FIRST=%5
8651      R2=%2
8652      HLT-HALT
8653      TRT=3
8654      ITRAP5=4
8655      RTRAP5=4
8656      RTRAP4=14
8657      RTRAP3=30
8658      RTRAP2=20
8659      RTRAP1=34
8660      TTCR=177564
8661      TRCSR=177560
8662      TPS=177564
8663      TPB=177566
8664      BELL=240
8665      NOP=240
8666      STATUS=177776
8667      TRAPA=77
8668      RTRAP=10
8669      ILLA=004700
8670      ILLB=100
8671      CC=177776
8672      BUFF=STBOT
8673
8674
8675
```

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

```

8676
8677 ;SPECIAL CASE OF ODD;.EVEN .BYTE AND REGISTER 6
8678 000000 HERE=0
8679
8680 027424 000167 000026 JMP TESTN1
8681 027430 000000 K1: 0
8682 027432 000000 K2: 0
8683 027434 000000 K3: 0
8684 027436 000000 K4: 0
8685 027440 000000 K5: 0
8686 027442 000000 K6: 0
8687 027444 052525 K7: 052525
8688 027446 052400 K10: 052400
8689 027450 000000 K11: 0
8690 027452 000000 K12: 0
8691 027454 000176 SWR: 176
8692
8693 027456 032737 000001 000320 TESTN1: BIT #1,@SWENV ;ARE WE RUNNING IN APT MODE
8694 027464 001403 BEQ 1$ ;IF NO USE SOFTWARE SWITCH REGISTER
8695 027466 012767 000322 177760 MOV #SWREG,SWR ;IF YES USE APT SWITCH REGISTER
8696 027474 032777 000001 177752 1$: BIT #1,@SWR ;IF BIT IS NON-ZERO SHIP ALL TRAPS TESTS
8697 027502 001402 BEQ 2$
8698 027504 000167 012722 JMP THRPRT
8699 027510 2$:
8700 ;*****
8701 ;TEST 300 TEST AUTO INCREMENT AND DECREMENT OF R6 FOR WORD AND BYTES
8702 ;*****
8703 027510 005212 TS300: INC (R2) ;UPDATE TEST NUMBER
8704 027512 022712 000300 CMP #300,(R2) ;SEQUENCE ERROR?
8705 027516 001127 BNE TS301-10 ;BR TO ERROR HALT ON SEQ ERROR
8706 027520 005006 CLR %6
8707 027522 112667 150252 MOVB (6)+,HERE ;SIX SHOULD INCREMENT BY TWO
8708 027526 020627 000002 CMP %6,#2
8709 027532 001404 BEQ BR1
8710
8711 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
8712 ; CONDITIONAL BRANCH INST. AND <---
8713 ; REPLACE THE MOVE INSTRUCTION <---
8714 ; WHICH FOLLOWS W/ 771 <---
8715 027534 012742 000656 MOV #656,-(R2) ;MOVE TO MAILBOX # ***** 656 *****
8716 027540 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8717 027542 000000 HALT ;R6 DID NOT AUTO INCREMENT BY TWO
8718 027544 012706 001000 BR1: MOV #1000,%6
8719 027550 114627 000000 MOVB -(6),#HERE ;SHOULD DECREMENT BY TWO
8720 027554 020627 000776 CMP %6,#776
8721 027560 001404 BEQ BR2
8722
8723 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -
8724 ; CONDITIONAL BRANCH INST. AND <== -
8725 ; REPLACE THE MOVE INSTRUCTION <--
8726 ; WHICH FOLLOWS W/ 756 <--
8727 027562 012742 000657 MOV #657,-(R2) ;MOVE TO MAILBOX # ***** 657 *****
8728 027566 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8729 027570 000000 HALT ;R6 DID NOT AUTO DECREMENT BY 2
8730 027572 005006 BR2: CLR %6
8731 027574 112626 MOVB (6)+,(6)+ ;DOUBLES AUTO INCREMENT OF R6

```

8732	027576	020627	000004		CMP	%6,#4			
8733	027602	001404			BEQ	BR3			
8734								; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
8735								CONDITIONAL BRANCH INST. AND	<====
8736								REPLACE THE MOVE INSTRUCTION	<====
8737								WHICH FOLLOWS W/ 745	<====
8738	027604	012742	000660		MOV	#660,-(R2)		;MOVE TO MAILBOX # ***** 660 *****	
8739	027610	005242			INC	-(R2)		;SET MSGTYP TO FATAL ERROR	
8740	027612	000000			HALT			;WRONG AUTO INCREMENT OF R6	
8741									
8742	027614	005006		BR3:	CLR	%6			
8743	027616	005004			CLR	%4			
8744	027620	122624			CMPB	(6)+,(4)+		;TEST INCREMENT OF R6	
8745	027622	020627	000002		CMP	%6,#2			
8746	027626	001404			BEQ	BR4			
8747								; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
8748								CONDITIONAL BRANCH INST. AND	<====
8749								REPLACE THE MOVE INSTRUCTION	<====
8750								WHICH FOLLOWS W/ 733	<====
8751	027630	012742	000661		MOV	#661,-(R2)		;MOVE TO MAILBOX # ***** 661 *****	
8752	027634	005242			INC	-(R2)		;SET MSGTYP TO FATAL ERROR	
8753	027636	000000			HALT			;WRONG INCREMENT OF R6	
8754									
8755	027640	005006		BR4:	CLR	%6			
8756	027642	005004			CLR	%4			
8757	027644	122426			CMPB	(4)+,(6)+		;TEST INCREMENT OF R6	
8758	027646	020627	000002		CMP	%6,#2			
8759	027652	001404			BEQ	BR5			
8760								; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
8761								CONDITIONAL BRANCH INST. AND	<====
8762								REPLACE THE MOVE INSTRUCTION	<====
8763								WHICH FOLLOWS W/ 721	<====
8764	027654	012742	000662		MOV	#662,-(R2)		;MOVE TO MAILBOX # ***** 662 *****	
8765	027660	005242			INC	-(R2)		;SET MSGTYP TO FATAL ERROR	
8766	027662	000000			HALT			;WRONG INCREMENT OF R6	
8767									
8768	027664	005006		BR5:	CLR	%6			
8769	027666	005004			CLR	%4			
8770	027670	122624			CMPB	(6)+,(4)+		;TEST INCREMENT OF R4	
8771	027672	020427	000001		CMP	%4,#1			
8772	027676	001404			BEQ	BR6			
8773								; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
8774								CONDITIONAL BRANCH INST. AND	<====
8775								REPLACE THE MOVE INSTRUCTION	<====
8776								WHICH FOLLOWS W/ 707	<====
8777	027700	012742	000663		MOV	#663,-(R2)		;MOVE TO MAILBOX # ***** 663 *****	
8778	027704	005242			INC	-(R2)		;SET MSGTYP TO FATAL ERROR	
8779	027706	000000			HALT			;WRONG INCREMENT OF R4	
8780	027710	005006		BR6:	CLR	%6			
8781	027712	005004			CLR	%4			
8782	027714	122426			CMPB	(4)+,(6)+		;TEST INCREMENT OF R6	
8783	027716	020627	000002		CMP	%6,#2			
8784	027722	001404			BEQ	BR7			
8785								; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
8786								CONDITIONAL BRANCH INST. AND	<====
8787								REPLACE THE MOVE INSTRUCTION	<====


```

8844 030112 026727 177312 050456      CMP      K1,#050456
8845 030120 001404                      BEQ      BR12
8846                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8847                                     ;          CONDITIONAL BRANCH INST. AND <====
8848                                     ;          REPLACE THE MOVE INSTRUCTION <====
8849                                     ;          WHICH FOLLOWS W/ 735 <====
8850 030122 012742 000670      MOV      #670,-(R2)
8851 030126 005242      INC      -(R2)
8852 030130 000000      HALT
8853                                     ;MOVE TO MAILBOX # ***** 670 *****
8854 030132 012767 123456 177270 BR12:  MOV      #123456,K1
8855 030140 012767 050505 177272      MOV      #050505,K5
8856 030146 012705 027430      MOV      #K1,%5
8857 030152 012706 027440      MOV      #K5,%6
8858 030156 112526      MOVB    (5)+,(6)+
8859 030160 022767 050456 177252      MOVB    (5)+,(6)+
8860 030166 001404      CMP      #050456,K5
8861                                     BEQ      BR13
8862                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8863                                     ;          CONDITIONAL BRANCH INST. AND <====
8864                                     ;          REPLACE THE MOVE INSTRUCTION <====
8865                                     ;          WHICH FOLLOWS W/ 712 <====
8865 030170 012742 000671      MOV      #671,-(R2)
8866 030174 005242      INC      -(R2)
8867 030176 000000      HALT
8868                                     ;MOVE TO MAILBOX # ***** 671 *****
8869 030200 012767 123456 177222 BR13:  MOV      #123456,K1
8870 030206 012767 050505 177224      MOV      #050505,K5
8871 030214 012705 027431      MOV      #K1+1,%5
8872 030220 012706 027440      MOV      #K5,%6
8873 030224 112526      MOVB    (5)+,(6)+
8874 030226 026727 177206 050647      MOVB    (5)+,(6)+
8875 030234 001404      CMP      K5,#050647
8876                                     BEQ      BR14
8877                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -
8878                                     ;          CONDITIONAL BRANCH INST. AND <= =
8879                                     ;          REPLACE THE MOVE INSTRUCTION <====
8880                                     ;          WHICH FOLLOWS W/ 667 <====
8880 030236 012742 000672      MOV      #672,-(R2)
8881 030242 005242      INC      -(R2)
8882 030244 000000      HALT
8883                                     ;MOVE TO MAILBOX # ***** 672 *****
8884 030246 012767 123456 177154 BR14:  MOV      #123456,K1
8885 030254 012767 050505 177156      MOV      #050505,K5
8886 030262 012705 027431      MOV      #K1+1,%5
8887 030266 012706 027440      MOV      #K5,%6
8888 030272 112625      MOVB    (6)+,(5)+
8889 030274 022767 042456 177126      MOVB    (6)+,(5)+
8890 030302 001404      CMP      #042456,K1
8891                                     BEQ      TS302
8892                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8893                                     ;          CONDITIONAL BRANCH INST. AND <====
8894                                     ;          REPLACE THE MOVE INSTRUCTION <====
8895                                     ;          WHICH FOLLOWS W/ 644 <== =
8895 030304 012742 000673      MOV      #673,-(R2)
8896 030310 005242      INC      -(R2)
8897 030312 000000      HALT
8898                                     ;MOVE TO MAILBOX # ***** 673 *****
8899                                     ;SET MSGTYP TO FATAL ERROR
8899                                     ;FAILED LOW OF 6 TO HIGH OF 5,OR WRONG $STNM
8899                                     ; OR SEQUENCE ERROR

```

```
8900 ;TEST 302 TEST BYTE OPERATION WITH SEQUENTIAL ODD-EVEN ADDRESS
8901 :*****
8902 030314 005212 TS302: INC (R2) ;UPDATE TEST NUMBER
8903 030316 022712 000302 CMP #302,(R2) ;SEQUENCE ERROR?
8904 030322 001074 BNE TS303-10 ;BR TO ERROR HALT ON SEQ ERROR
8905 030324 126767 177114 177113 CMPB K7,K7+1 ;SAME .WORD LOW TO HIGH
8906 030332 001404 BEQ BR15
8907 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8908 ; CONDITIONAL BRANCH INST. AND <====
8909 ; REPLACE THE MOVE INSTRUCTION <====
8910 ; WHICH FOLLOWS W/ 773 <====
8911 030334 012742 000674 MOV #674,-(R2) ;MOVE TO MAILBOX # ***** 674 *****
8912 030340 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8913 030342 000000 HALT ;SHOULD COMPARE LOW TO HIGH
8914
8915 030344 126767 177075 177072 BR15: CMPB K7+1,K7 ;COMPARE ODD TO .EVEN SAME .WORD
8916 030352 001404 BEQ BR16
8917 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8918 ; CONDITIONAL BRANCH INST. AND <====
8919 ; REPLACE THE MOVE INSTRUCTION <====
8920 ; WHICH FOLLOWS W/ 763 <====
8921 030354 012742 000675 MOV #675,-(R2) ;MOVE TO MAILBOX # ***** 675 *****
8922 030360 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8923 030362 000000 HALT ;ODD TO .EVEN .BYTE FAILURE
8924
8925 030364 126767 177057 177052 BR16: CMPB K10+1,K7 ;SEQUENTIAL .BYTES
8926 030372 001404 BEQ BR17
8927 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8928 ; CONDITIONAL BRANCH INST. AND <====
8929 ; REPLACE THE MOVE INSTRUCTION <====
8930 ; WHICH FOLLOWS W/ 753 <====
8931 030374 012742 000676 MOV #676,-(R2) ;MOVE TO MAILBOX # ***** 676 *****
8932 030400 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8933 030402 000000 HALT ;ODD TO .EVEN FAILED
8934
8935 030404 126767 177036 177030 BR17: CMPB K10,K6
8936 030412 001404 BEQ BR20
8937 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8938 ; CONDITIONAL BRANCH INST. AND <====
8939 ; REPLACE THE MOVE INSTRUCTION <====
8940 ; WHICH FOLLOWS W/ 743 <====
8941 030414 012742 000677 MOV #677,-(R2) ;MOVE TO MAILBOX # ***** 677 *****
8942 030420 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8943 030422 000000 HALT ;.EVEN TO EVEN FAILED
8944 030424 126767 177015 177015 BR20: CMPB K7+1,K10+1
8945 030432 001404 BEQ BR21
8946 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
8947 ; CONDITIONAL BRANCH INST. AND <====
8948 ; REPLACE THE MOVE INSTRUCTION <====
8949 ; WHICH FOLLOWS W/ 733 <====
8950 030434 012742 000700 MOV #700,-(R2) ;MOVE TO MAILBOX # ***** 700 *****
8951 030440 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
8952 030442 000000 HALT ;ODD TO ODD FAILED
8953
8954 030444 126767 176776 176775 BR21: CMPB K10,K10+1
8955 030452 001004 BNE BR22
```

```

8956
8957
8958
8959
8960 030454 012742 000701      MOV    #701,-(R2)
8961 030460 005242              INC    -(R2)
8962 030462 000000              HALT
8963
8964 030464 126767 176757 176755 BR22:  CMPB   K10+1,K10+1
8965 030472 001404              BEQ   BR23
8966
8967
8968
8969
8970 030474 012742 000702      MOV    #702,-(R2)
8971 030500 005242              INC    -(R2)
8972 030502 000000              HALT
8973
8974 030504 126767 176736 176733 BR23:  CMPB   K10,K7+1
8975 030512 001004              BNE   TS303
8976
8977
8978
8979
8980 030514 012742 000703      MOV    #703,-(R2)
8981 030520 005242              INC    -(R2)
8982 030522 000000              HALT
8983
8984
8985
8986
8987
8988
8989 030524 005212
8990 030526 022712 000303      TS303: INC    (R2)
8991 030532 001053              CMP    #303,(R2)
8992 030534 000277              BNE   TS304-10
8993 030536 005067 147234      SCC
8994 030542 103004              CLR   STATUS
8995
8996
8997
8998
8999 030544 012742 000704      MOV    #704,-(R2)
9000 030550 005242              INC    -(R2)
9001 030552 000000              HALT
9002 030554
9003 030554 102004      BR33:  BVC   BR34
9004
9005
9006
9007
9008 030556 012742 000705      MOV    #705,-(R2)
9009 030562 005242              INC    -(R2)
9010 030564 000000              HALT
9011 030566
    
```

```

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 723 <====
: MOVE TO MAILBOX # ***** 701 *****
: SET MSGTYP TO FATAL ERROR
: LOW TO HIGH IN SAME .WORD FAILED

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 713 <====
: MOVE TO MAILBOX # ***** 702 *****
: SET MSGTYP TO FATAL ERROR
: HIGH TO LOW IN SAME .WORD FAILED

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 703 <====
: MOVE TO MAILBOX # ***** 703 *****
: SET MSGTYP TO FATAL ERROR
: EVEN TO ODD FAILED,OR WRONG $STNM
: OR SEQUENCE ERROR
    
```

```

:*****
:TEST 303 TEST THE CC BITS
:*****
    
```

```

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 773 <====
: MOVE TO MAILBOX # ***** 704 *****
: SET MSGTYP TO FATAL ERROR
: C NCT CLEAR

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 766 <====
: MOVE TO MAILBOX # ***** 705 *****
: SET MSGTYP TO FATAL ERROR
: V NOT CLEAR
    
```

9012	030566	001004		BNE	BR35						
9013											
9014											
9015											
9016											
9017	030570	012742	000706	MOV	#706,-(R2)						
9018	030574	005242		INC	-(R2)						
9019	030576	000000		HALT							
9020	030600										
9021	030600	100004		BR35:	BPL	BR36					
9022											
9023											
9024											
9025											
9026	030602	012742	000707	MOV	#707,-(R2)						
9027	030606	005242		INC	-(R2)						
9028	030610	000000		HALT							
9029	030612	000257		BR36:	CCC						
9030	030614	052767	000017 147154	BIS	#17,STATUS						
9031											
9032	030622	103404		BCS	BR37						
9033											
9034											
9035											
9036											
9037	030624	012742	000710	MOV	#710,-(R2)						
9038	030630	005242		INC	-(R2)						
9039	030632	000000		HALT							
9040	030634			BR37:	BVS	BR40					
9041	030634	102404									
9042											
9043											
9044											
9045											
9046	030636	012742	000711	MOV	#711,-(R2)						
9047	030642	005242		INC	-(R2)						
9048	030644	000000		HALT							
9049	030646			BR40:	BEQ	BR41					
9050	030646	001404									
9051											
9052											
9053											
9054											
9055	030650	012742	000712	MOV	#712,-(R2)						
9056	030654	005242		INC	-(R2)						
9057	030656	000000		HALT							
9058	030660			BR41:	BMI	TS304					
9059	030660	100404									
9060											
9061											
9062											
9063											
9064	030662	012742	000713	MOV	#713,-(R2)						
9065	030666	005242		INC	-(R2)						
9066	030670	000000		HALT							
9067											

```
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====>
; CONDITIONAL BRANCH INST. AND <====>
; REPLACE THE MOVE INSTRUCTION <====>
; WHICH FOLLOWS W/ 761 <====>
; MOVE TO MAILBOX # ***** 706 *****
; SET MSGTYP TO FATAL ERROR
; Z NOT CLEAR

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====>
; CONDITIONAL BRANCH INST. AND <====>
; REPLACE THE MOVE INSTRUCTION <====>
; WHICH FOLLOWS W/ 754 <====>
; MOVE TO MAILBOX # ***** 707 *****
; SET MSGTYP TO FATAL ERROR
; N NOT CLEAR
; CLEAR CONDITION CODES
; SET STATUS TO ONES

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====>
; CONDITIONAL BRANCH INST. AND <====>
; REPLACE THE MOVE INSTRUCTION <====>
; WHICH FOLLOWS W/ 743 <====>
; MOVE TO MAILBOX # ***** 710 *****
; SET MSGTYP TO FATAL ERROR
; C NOT SET

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====>
; CONDITIONAL BRANCH INST. AND <====>
; REPLACE THE MOVE INSTRUCTION <====>
; WHICH FOLLOWS W/ 736 <====>
; MOVE TO MAILBOX # ***** 711 *****
; SET MSGTYP TO FATAL ERROR
; V NOT SET

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====>
; CONDITIONAL BRANCH INST. AND <====>
; REPLACE THE MOVE INSTRUCTION <====>
; WHICH FOLLOWS W/ 731 <====>
; MOVE TO MAILBOX # ***** 712 *****
; SET MSGTYP TO FATAL ERROR
; Z NOT SET

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====>
; CONDITIONAL BRANCH INST. AND <====>
; REPLACE THE MOVE INSTRUCTION <====>
; WHICH FOLLOWS W/ 724 <====>
; MOVE TO MAILBOX # ***** 713 *****
; SET MSGTYP TO FATAL ERROR
; N NOT SET, OR WRONG $STNM
; OR SEQUENCE ERROR
```



```
9068 :*****
9069 :TEST 304 TEST THAT A TRAP OCCURS ON A RESERVED INSTRUCTION
9070 :*****
9071 030672 005212 TS304: INC (R2) ;UPDATE TEST NUMBER
9072 030674 022712 000304 CMP #304,(R2) ;SEQUENCE ERROR?
9073 030700 001006 BNE RETA ;BR TO ERROR HALT ON SEQ ERROR
9074 030702 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9075 030706 012767 030726 147074 MOV #RETAH,RTRAP ;RETURN LOCATION
9076 030714 000077 TRAPA ;RESERVED INSTRUCTION, SHOULD TRAP
9077 030716 RETA:
9078 030716 012742 000714 MOV #714,-(R2) ;MOVE TO MAILBOX # ***** 714 *****
9079 030722 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9080 030724 000000 HALT ;RESERVE INSTRUCTION DIDN'T TRAP,OR WRONG $STNM
9081 030726 RETAH:
9082 :*****
9083 :TEST 305 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9084 :*****
9085 030726 005212 TS305: INC (R2) ;UPDATE TEST NUMBER
9086 030730 022712 000305 CMP #305,(R2) ;SEQUENCE ERROR?
9087 030734 001011 BNE TS306-10 ;BR TO ERROR HALT ON SEQ ERROR
9088 030736 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9089 030742 012767 030752 147040 MOV #RETB,RTRAP ;RETURN POINTER
9090 030750 000077 TRAPA ;RESERVED INSTRUCTION
9091 030752 020627 000774 RETB: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
9092 030756 001404 BEQ TS306
9093 :
9094 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9095 : CONDITIONAL BRANCH INST. AND <====
9096 : REPLACE THE MOVE INSTRUCTION <====
9097 : WHICH FOLLOWS W/ 766 <====
9098 030760 012742 000715 MOV #715,-(R2) ;MOVE TO MAILBOX # ***** 715 *****
9099 030764 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9100 030766 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $STNM
9101 : OR SEQUENCE ERROR
9102 :*****
9103 :TEST 306 TEST THAT PROPER P.C. IS SAVED
9104 :*****
9105 030770 005212 TS306: INC (R2) ;UPDATE TEST NUMBER
9106 030772 022712 000306 CMP #306,(R2) ;SEQUENCE ERROR?
9107 030776 001012 BNE TS307-10 ;BR TO ERROR HALT ON SEQ ERROR
9108 031000 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9109 031004 012767 031014 146776 MOV #RETC,RTRAP ;RETURN FROM TRAP POINTER
9110 031012 000077 INSTC: TRAPA ;TRAP ON THIS INSTRUCTION
9111 031014 022767 031014 147752 RETC: CMP #,BUFF-4 ;CHECK FOR INCREMENTED P.C.
9112 031022 001404 BEQ TS307
9113 :
9114 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9115 : CONDITIONAL BRANCH INST. AND <====
9116 : REPLACE THE MOVE INSTRUCTION <====
9117 : WHICH FOLLOWS W/ 765 <====
9118 031024 012742 000716 MOV #716,-(R2) ;MOVE TO MAILBOX # ***** 716 *****
9119 031030 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9120 031032 000000 HALT ;INCORRECT P.C.,OR WRONG $STNM
9121 : OR SEQUENCE ERROR
9122 :*****
9123 031034 005212 :TEST 307 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
TS307: INC (R2) ;UPDATE TEST NUMBER
```

```

9124 031036 022712 000307      CMP      #307,(R2)      ;SEQUENCE ERROR?
9125 031042 001037      BNE      TS310-10     ;BR TO ERROR HALT ON SEQ ERROR
9126 031044 012706 001000      MOV      #BUFF,SP     ;SET UP
9127 031050 012767 031066 146732      MOV      #RETD,RTRAP  ;SET UP
9128 031056 005067 146714      CLR      CC           ;CLEAR CC AND PRIORITY
9129 031062 000257      CCC
9130 031064 000077      TRAPA
9131 031066 026727 147704 000000 RETD:  CMP      BUFF-2,#0     ;TEST THAT OLD STATUS WENT TO STACK
9132 031074 001404      BEQ      1$
9133      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9134      ;          CONDITIONAL BRANCH INST. AND <====
9135      ;          REPLACE THE MOVE INSTRUCTION <====
9136      ;          WHICH FOLLOWS W/ 762 <====
9137 031076 012742 000717      MOV      #717,-(R2)   ;MOVE TO MAILBOX # ***** 717 *****
9138 031102 005242      INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
9139 031104 000000      HALT
9140 031106 012706 001000 1$:      MOV      #BUFF,SP     ;SET UP
9141 031112 012767 031132 146670      MOV      #RETE,RTRAP  ;SET UP
9142 031120 012767 000357 146650      MOV      #357,CC     ;SET PRIORITY
9143 031126 000277      SCC
9144 031130 000077      TRAPA
9145 031132 026727 147640 000357 RETE:  CMP      BUFF-2,#357   ;COMPARES STATUS ON STACK
9146 031140 001404      BEQ      TS310
9147      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9148      ;          CONDITIONAL BRANCH INST. AND <====
9149      ;          REPLACE THE MOVE INSTRUCTION <====
9150      ;          WHICH FOLLOWS W/ 740 <====
9151 031142 012742 000720      MOV      #720,-(R2)   ;MOVE TO MAILBOX # ***** 720 *****
9152 031146 005242      INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
9153 031150 000000      HALT
9154      ; INCORRECT STATUS ON STACK,OR WRONG $STNM
9155      ; OR SEQUENCE ERROR
9156      ;*****
9157      ;TEST 310 TEST THAT 'NEW' STATUS IS CORRECT
9158      ;*****
9159 031152 005212      TS310: INC      (R2)      ;UPDATE TEST NUMBER
9160 031154 022712 000310      CMP      #310,(R2)   ;SEQUENCE ERROR?
9161 031160 001110      BNE      STPP
9162 031162 012706 001000      MOV      #BUFF,SP     ;BR TO ERROR HALT ON SEQ ERROR
9163 031166 012767 031202 146614      MOV      #RETF,RTRAP
9164 031174 005067 146612      CLR      RTRAP+2
9165 031200 000077      TRAPA
9166 031202 100004      RETF:  ;CLEAR FUTURE PRIORITY AND CC
9167      ;TEST FOR 'C' CLEARED
9168      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9169      ;          CONDITIONAL BRANCH INST. AND <====
9170      ;          REPLACE THE MOVE INSTRUCTION <====
9171      ;          WHICH FOLLOWS W/ 766 <====
9172 031204 012742 000721      MOV      #721,-(R2)   ;MOVE TO MAILBOX # ***** 721 *****
9173 031210 005242      INC      -(R2)       ;SET MSGTYP TO FATAL ERROR
9174 031212 000000      HALT
9175 031214 001004 1$:      BNE      2$
9176      ; N NOT CLEARED
9177      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9178      ;          CONDITIONAL BRANCH INST. AND <====
9179      ;          REPLACE THE MOVE INSTRUCTION <====
9179      ;          WHICH FOLLOWS W/ 761 <====
    
```

9180	031216	012742	000722			MOV	#722,-(R2)	:MOVE TO MAILBOX # ***** 722 *****	
9181	031222	005242				INC	-(R2)	:SET MSGTYP TO FATAL ERROR	
9182	031224	000000				HALT		:Z NOT CLEARED	
9183	031226			2\$:					
9184	031226	102004				BVC	3\$		
9185								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
9186								: CONDITIONAL BRANCH INST. AND	<====
9187								: REPLACE THE MOVE INSTRUCTION	<====
9188								: WHICH FOLLOWS W/ 754	<====
9189	031230	012742	000723			MOV	#723,-(R2)	:MOVE TO MAILBOX # ***** 723 *****	
9190	031234	005242				INC	-(R2)	:SET MSGTYP TO FATAL ERROR	
9191	031236	000000				HALT		:V NOT CLEARED	
9192	031240			3\$:					
9193	031240	103000				BCC	4\$		
9194								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
9195								: CONDITIONAL BRANCH INST. AND	<====
9196								: REPLACE THE MOVE INSTRUCTION	<====
9197								: WHICH FOLLOWS W/ 747	<====
9198	031242	012742	000724			MOV	#724,-(R2)	:MOVE TO MAILBOX # ***** 724 *****	
9199	031246	005242				INC	-(R2)	:SET MSGTYP TO FATAL ERROR	
9200	031250	000000				HALT		:C NOT CLEARED	
9201	031252	032767	000340	146516	4\$:	BIT	#340,CC	:TEST PRIORITY	
9202	031260	001404				BEQ	5\$		
9203								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
9204								: CONDITIONAL BRANCH INST. AND	<====
9205								: REPLACE THE MOVE INSTRUCTION	<====
9206								: WHICH FOLLOWS W/ 737	<====
9207	031262	012742	000725			MOV	#725,-(R2)	:MOVE TO MAILBOX # ***** 725 *****	
9208	031266	005242				INC	-(R2)	:SET MSGTYP TO FATAL ERROR	
9209	031270	000000				HALT		:PRIORITY NOT ZERO	
9210	031272	012706	001000		5\$:	MOV	#BUFF,SP		
9211	031276	012767	031314	146504		MOV	#RETG,RTRAP		
9212	031304	012767	000357	146500		MOV	#357,RTRAP+2	:SET NEW 'CC' AND PRIORITY	
9213	031312	000077				TRAPA		:TRAP HERE	
9214	031314				RETG:				
9215	031314	100404				BMI	1\$		
9216								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
9217								: CONDITIONAL BRANCH INST. AND	<====
9218								: REPLACE THE MOVE INSTRUCTION	<====
9219								: WHICH FOLLOWS W/ 721	<====
9220	031316	012742	000726			MOV	#726,-(R2)	:MOVE TO MAILBOX # ***** 726 *****	
9221	031322	005242				INC	-(R2)	:SET MSGTYP TO FATAL ERROR	
9222	031324	000000				HALT		:N NOT SET	
9223	031326				1\$:				
9224	031326	001404				BEQ	2\$		
9225								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
9226								: CONDITIONAL BRANCH INST. AND	<====
9227								: REPLACE THE MOVE INSTRUCTION	<====
9228								: WHICH FOLLOWS W/ 714	<====
9229	031330	012742	000727			MOV	#727,-(R2)	:MOVE TO MAILBOX # ***** 727 *****	
9230	031334	005242				INC	-(R2)	:SET MSGTYP TO FATAL ERROR	
9231	031336	000000				HALT		:Z NOT SET	
9232	031340				2\$:				
9233	031340	102404				BVS	3\$		
9234								: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
9235								: CONDITIONAL BRANCH INST. AND	<====

```
9236                                     :                                     <====
9237                                     :                                     WHICH FOLLOWS W/ 707 <====
9238 031342 012742 000730                MOV #730,-(R2)                ;MOVE TO MAILBOX # ***** 730 *****
9239 031346 005242                        INC -(R2)                   ;SET MSGTYP TO FATAL ERROR
9240 031350 000000                        HALT                       ;V NOT SET
9241 031352                                3$:
9242 031352 103404                        BCS 4$
9243                                     :
9244                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9245                                     : CONDITIONAL BRANCH INST. AND <====
9246                                     : REPLACE THE MOVE INSTRUCTION <--
9247 031354 012742 000731                MOV #731,-(R2)            ;MOVE TO MAILBOX # ***** 731 *****
9248 031360 005242                        INC -(R2)                   ;SET MSGTYP TO FATAL ERROR
9249 031362 000000                        HALT                       ;C NOT SET
9250 031364 016706 146406                4$: MOV CC,SP
9251 031370 042706 000017                BIC #17,SP
9252 031374 022706 000340                CMP #340,SP
9253 031400 001404                        BEQ STPPA
9254                                     :
9255                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9256                                     : CONDITIONAL BRANCH INST. AND <-
9257                                     : REPLACE THE MOVE INSTRUCTION <
9258                                     : WHICH FOLLOWS W/ 667 <
9258 031402                                STPP:
9259 031402 012742 000732                MOV #732,-(R2)            ;MOVE TO MAILBOX # ***** 732 *****
9260 031406 005242                        INC -(R2)                   ;SET MSGTYP TO FATAL ERROR
9261 031410 000000                        HALT                       ;PRIORITY WAS CHANGED,OR WRONG $STNM
9262 031412 012767 000012 146370        STPPA: MOV #12,10
9263 031420 005067 146366                CLR 12
9264                                     ;*****
9265                                     ;TEST 311 TEST THAT A TRAP OCCURS FOR A 'TRAP' INSTRUCTION
9266                                     ;*****
9267 031424 005212                                TS311: INC (R2)                ;UPDATE TEST NUMBER
9268 031426 022712 000311                CMP #311,(R2)             ;SEQUENCE ERROR?
9269 031432 001013                        BNE TS312-10              ;BR TO ERROR HALT ON SEQ ERROR
9270 031434 012767 000012 146346                MOV #12,10
9271 031442 005067 146344                CLR 12
9272 031446 012706 001000                MOV #BUFF,SP              ;STACK POINTER SETUP
9273 031452 012767 031472 146354                MOV #RETA1,RTRAP1        ;RETURN LOCATION
9274 031460 104400                        TRAP                       ;RESERVED INSTRUCTION, SHOULD TRAP
9275 031462 012742 000733                MOV #733,-(R2)            ;MOVE TO MAILBOX # ***** 733 *****
9276 031466 005242                        INC -(R2)                   ;SET MSGTYP TO FATAL ERROR
9277 031470 000000                        HALT                       ;TRAP DIDN'T TRAP,OR WRONG $STNM
9278 031472                                RETA1:
9279                                     ;*****
9280                                     ;TEST 312 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9281                                     ;*****
9282 031472 005212                                TS312: INC (R2)                ;UPDATE TEST NUMBER
9283 031474 022712 000312                CMP #312,(R2)             ;SEQUENCE ERROR?
9284 031500 001011                        BNE TS313-10              ;BR TO ERROR HALT ON SEQ ERROR
9285 031502 012706 001000                MOV #BUFF,SP              ;STACK POINTER SETUP
9286 031506 012767 031516 146320                MOV #RETB1,RTRAP1        ;RETURN POINTER
9287 031514 104400                        TRAP                       ;RESERVED INSTRUCTION
9288 031516 020627 000774                RETB1: CMP SP,#BUFF-4     ;TEST DECREMENT OF SP
9289 031522 001404                        BEQ TS313
9290                                     :
9291                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
                                     : CONDITIONAL BRANCH INST. AND <=
```

```
9292                                     :           REPLACE THE MOVE INSTRUCTION <====
9293                                     :           WHICH FOLLOWS W/ 766           <====
9294 031524 012742 000734                MOV    #734,-(R2)      :MOVE TO MAILBOX # ***** 734 *****
9295 031530 005242                        INC    -(R2)          :SET MSGTYP TO FATAL ERROR
9296 031532 000000                        HALT                  :NOT DECREMENTED TWO WORDS,OR WRONG $STNM
9297                                     :           OR SEQUENCE ERROR
9298                                     :*****
9299 :TEST 313 TEST THAT PROPER P.C. IS SAVED
9300 :*****
9301 031534 005212                        TS313: INC    (R2)      :UPDATE TEST NUMBER
9302 031536 022712 000313                CMP    #313,(R2)     :SEQUENCE ERROR?
9303 031542 001012                        BNE    TS314-10      :BR TO ERROR HALT ON SEQ ERROR
9304 031544 012706 001000                MOV    #BUFF,SP      :STACK POINTER SETUP
9305 031550 012767 031560 146256        MOV    #RETC1,RTRAP1 :RETURN FROM TRAP POINTER
9306 031556 104400                        TRAP                                :TRAP ON THIS INSTRUCTION
9307 031560 022767 031560 147206 RETC1: CMP    #.BUFF-4      :CHECK INCREMENTED P.C.
9308 031566 001404                        BEQ    TS314
9309                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9310                                     :           CONDITIONAL BRANCH INST. AND <====
9311                                     :           REPLACE THE MOVE INSTRUCTION <====
9312                                     :           WHICH FOLLOWS W/ 765           <====
9313 031570 012742 000735                MOV    #735,-(R2)     :MOVE TO MAILBOX # ***** 735 *****
9314 031574 005242                        INC    -(R2)          :SET MSGTYP TO FATAL ERROR
9315 031576 000000                        HALT                  :INCORRECT P.C.,OR WRONG $STNM
9316                                     :           OR SEQUENCE ERROR
9317                                     :*****
9318 :TEST 314 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
9319 :*****
9320 031600 005212                        TS314: INC    (R2)      :UPDATE TEST NUMBER
9321 031602 022712 000314                CMP    #314,(R2)     :SEQUENCE ERROR?
9322 031606 001036                        BNE    TS315-10      :BR TO ERROR HALT ON SEQ ERROR
9323 031610 012706 001000                MOV    #BUFF,SP      :SET UP
9324 031614 012767 031632 146212        MOV    #RETD1,RTRAP1 :SET UP
9325 031622 005067 146150                CLR    CC             :CLEAR CC AND PRIORITY
9326 031626 000257                        CCC
9327 031630 104400                        TRAP                                :TRAP
9328 031632 026727 147140 000000 RETD1: CMP    BUFF-2,#0   :TEST THAT OLD STATUS WENT TO STACK
9329 031640 001404                        BEQ    1$
9330                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9331                                     :           CONDITIONAL BRANCH INST. AND <====
9332                                     :           REPLACE THE MOVE INSTRUCTION <====
9333                                     :           WHICH FOLLOWS W/ 762           <====
9334 031642 012742 000736                MOV    #736,-(R2)     :MOVE TO MAILBOX # ***** 736 *****
9335 031646 005242                        INC    -(R2)          :SET MSGTYP TO FATAL ERROR
9336 031650 000000                        HALT                  :INCORRECT STATUS
9337 031652 012706 001000 1$: MOV    #BUFF,SP      :SET UP
9338 031656 012767 031674 146150        MOV    #RETE1,RTRAP1 :SET UP
9339 031664 012767 000357 146104        MOV    #357,CC       :SET PRIORITY
9340 031672 104400                        TRAP                                :SET CC
9341 031674 026727 147076 000357 RETE1: CMP    BUFF-2,#357 :COMPARES STATUS ON STACK
9342 031702 001404                        BEQ    TS315
9343                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9344                                     :           CONDITIONAL BRANCH INST. AND <====
9345                                     :           REPLACE THE MOVE INSTRUCTION <====
9346                                     :           WHICH FOLLOWS W/ 741           <====
9347 031704 012742 000737                MOV    #737,-(R2)     :MOVE TO MAILBOX # ***** 737 *****
```

```
9348 031710 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
9349 031712 000000          HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
9350                                     ; OR SEQUENCE ERROR
9351                                     ;*****
9352                                     ;TEST 315      TEST THAT 'NEW' STATUS IS CORRECT
9353                                     ;*****
9354 031714 005212          TS315: INC    (R2)          ;UPDATE TEST NUMBER
9355 031716 022712 000315    CMP    #315,(R2)        ;SEQUENCE ERROR?
9356 031722 001110          BNE    TS316-10        ;BR TO ERROR HALT ON SEQ ERROR
9357 031724 012706 001000    MOV    #BUFF,SP
9358 031730 012767 031744 146076  MOV    #RETF1,RTRAP1
9359 031736 005067 146074    CLR    RTRAP1+2        ;CLEAR FUTURE PRIORITY AND CC
9360 031742 104400          TRAP
9361 031744          RETF1:
9362 031744 100004          BPL    1$              ;TEST FOR 'C' CLEARED
9363                                     ;
9364                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < - -
9365                                     ; CONDITIONAL BRANCH INST. AND < - -
9366                                     ; REPLACE THE MOVE INSTRUCTION <
9367                                     ; WHICH FOLLOWS W/ 766 <-
9367 031746 012742 000740    MOV    #740,-(R2)      ;MOVE TO MAILBOX # ***** 740 *****
9368 031752 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
9369 031754 000000          HALT                    ;C NOT CLEARED
9370 031756          1$:
9371 031756 001004          BNE    2$
9372                                     ;
9373                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9374                                     ; CONDITIONAL BRANCH INST. AND <
9375                                     ; REPLACE THE MOVE INSTRUCTION <
9376                                     ; WHICH FOLLOWS W/ 761 <
9376 031760 012742 000741    MOV    #741,-(R2)      ;MOVE TO MAILBOX # ***** 741 *****
9377 031764 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
9378 031766 000000          HALT                    ;Z NOT CLEARED
9379 031770          2$:
9380 031770 102004          BVC    3$
9381                                     ;
9382                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <-
9383                                     ; CONDITIONAL BRANCH INST. AND <
9384                                     ; REPLACE THE MOVE INSTRUCTION <
9385                                     ; WHICH FOLLOWS W/ 754 <
9385 031772 012742 000742    MOV    #742,-(R2)      ;MOVE TO MAILBOX # ***** 742 *****
9386 031776 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
9387 032000 000000          HALT                    ;V NOT CLEARED
9388 032002          3$:
9389 032002 103004          BCC    4$
9390                                     ;
9391                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
9392                                     ; CONDITIONAL BRANCH INST. AND <
9393                                     ; REPLACE THE MOVE INSTRUCTION <- -
9394                                     ; WHICH FOLLOWS W/ 747 <
9394 032004 012742 000743    MOV    #743,-(R2)      ;MOVE TO MAILBOX # ***** 743 *****
9395 032010 005242          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
9396 032012 000000          HALT                    ;C NOT CLEARED
9397 032014 032767 000340 145754 4$: BIT    #340,CC
9398 032022 001404          BEQ    5$              ;TEST PRIORITY
9399                                     ;
9400                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9401                                     ; CONDITIONAL BRANCH INST. AND <
9402                                     ; REPLACE THE MOVE INSTRUCTION <
9403                                     ; WHICH FOLLOWS W/ 737 <
9403 032024 012742 000744    MOV    #744,-(R2)      ;MOVE TO MAILBOX # ***** 744 *****
```

```
9404 032030 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9405 032032 000000          HALT                    ;PRIORITY NOT ZERO
9406 032034 012706 001000    5$:      MOV      #BUFF,SP
9407 032040 012767 032056 145766    MOV      #RETG1,RTRAP1
9408 032046 012767 000357 145762    MOV      #357,RTRAP1+2 ;SET NEW 'CC' AND PRIORITY
9409 032054 104400          TRAP                    ;TRAP HERE
9410 032056          RETG1:
9411 032056 100404          BMI      1$
9412          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9413          ;          CONDITIONAL BRANCH INST. AND <====
9414          ;          REPLACE THE MOVE INSTRUCTION <====
9415          ;          WHICH FOLLOWS W/ 721 <====
9416 032060 012742 000745          MOV      #745,-(R2)    ;MOVE TO MAILBOX # ***** 745 *****
9417 032064 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9418 032066 000000          HALT                    ;N NOT SET
9419 032070          1$:
9420 032070 001404          BEQ      2$
9421          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9422          ;          CONDITIONAL BRANCH INST. AND <====
9423          ;          REPLACE THE MOVE INSTRUCTION <====
9424          ;          WHICH FOLLOWS W/ 714 <====
9425 032072 012742 000746          MOV      #746,-(R2)    ;MOVE TO MAILBOX # ***** 746 *****
9426 032076 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9427 032100 000000          HALT                    ;Z NOT SET
9428 032102          2$:
9429 032102 102404          BVS      3$
9430          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9431          ;          CONDITIONAL BRANCH INST. AND <====
9432          ;          REPLACE THE MOVE INSTRUCTION <====
9433          ;          WHICH FOLLOWS W/ 707 <====
9434 032104 012742 000747          MOV      #747,-(R2)    ;MOVE TO MAILBOX # ***** 747 *****
9435 032110 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9436 032112 000000          HALT                    ;V NOT SET
9437 032114          3$:
9438 032114 103404          BCS      4$
9439          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9440          ;          CONDITIONAL BRANCH INST. AND <====
9441          ;          REPLACE THE MOVE INSTRUCTION <====
9442          ;          WHICH FOLLOWS W/ 702 <====
9443 032116 012742 000750          MOV      #750,-(R2)    ;MOVE TO MAILBOX # ***** 750 *****
9444 032122 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9445 032124 000000          HALT                    ;C NOT SET
9446 032126 016706 145644          4$:      MOV      CC,SP
9447 032132 042706 000017    BIC      #17,SP
9448 032136 022706 000340    CMP      #340,SP
9449 032142 001404          BEQ      TS316
9450          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9451          ;          CONDITIONAL BRANCH INST. AND <
9452          ;          REPLACE THE MOVE INSTRUCTION <
9453          ;          WHICH FOLLOWS W/ 667 <
9454 032144 012742 000751          MOV      #751,-(R2)    ;MOVE TO MAILBOX # ***** 751 *****
9455 032150 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9456 032152 000000          HALT                    ;PRIORITY WAS CHANGED,OR WRONG $STNM
9457          ; OR SEQUENCE ERROR
9458          ;*****
9459          ;TEST 316          TEST THAT ALL COMBINATION OF 'TRAP' WILL CAUSE A TRAP
```

```
*****
9460                                     ;*****
9461 032154 005212 TS316: INC (R2) ;UPDATE TEST NUMBER
9462 032156 022712 000316 CMP #316,(R2) ;SEQUENCE ERROR?
9463 032162 001011 BNE BR45 ;BR TO ERROR HALT ON SEQ ERROR
9464                                     ;***** F11 **** ADD +376 TO SHORTEN TEST
9465 032164 012767 104776 000012 MOV #TRAP+376,RB1 ;INITIALIZE BASE TRAP INSTRUCTION
9466 032172 012767 032216 145634 MOV #RA1,34 ;RETURN FROM TRAP TO RA1
9467 032200 012706 001000 RC1: MOV #BUFF,SP ;SET UP STACK POINTER
9468 032204 104400 RB1: TRAP ;TRAP INST WILL BE MODIFIED TO TRAP+377
9469 032206 BR45:
9470 032206 012742 000752 MOV #752,-(R2) ;MOVE TO MAILBOX # ***** 752 *****
9471 032212 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9472 032214 000000 HALT ;PREVIOUS INST FAILED TO TRAP,OR WRONG $STNM
9473 032216 005267 177762 RA1: INC RB1 ;INCREMENT TRAP INSTRUCTION
9474 032222 022767 104777 177754 CMP #104777,RB1 ;TRAP+377 TO UPPER LIMIT
9475 032230 103363 BHIS RC1 ;HAVE WE TESTED ALL
9476 032232 012767 000036 145574 MOV #36,34
9477 032240 005067 145572 CLR 36
*****
9478                                     ;*****
9479 ;TEST 317 TEST THAT A TRAP OCCURES ON AN 'IOT' INSTRUCTION
9480                                     ;*****
9481 032244 005212 TS317: INC (R2) ;UPDATE TEST NUMBER
9482 032246 022712 000317 CMP #317,(R2) ;SEQUENCE ERROR?
9483 032252 001006 BNE TS320-10 ;BR TO ERROR HALT ON SEQ ERROR
9484 032254 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9485 032260 012767 032300 145532 MOV #RETA2,RTRAP2 ;RETURN LOCATION
9486 032266 000004 IOT ;RESERVE INSTRUCTION, SHOULD TRAP
9487 032270 012742 000753 MOV #753,-(R2) ;MOVE TO MAILBOX # ***** 753 *****
9488 032274 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9489 032276 000000 HALT ;IOT DIDN'T TRAP,OR WRONG $STNM
9490 032300 RETA2:
*****
9491                                     ;*****
9492 ;TEST 320 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9493                                     ;*****
9494 032300 005212 TS320: INC (R2) ;UPDATE TEST NUMBER
9495 032302 022712 000320 CMP #320,(R2) ;SEQUENCE ERROR?
9496 032306 001011 BNE TS321-10 ;BR TO ERROR HALT ON SEQ ERROR
9497 032310 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9498 032314 012767 032324 145476 MOV #RETB2,RTRAP2 ;RETURN POINTER
9499 032322 000004 IOT ;RESERVED INSTRUCTION
9500 032324 020627 000774 RETB2: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
9501 032330 001404 BEQ TS321
9502                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9503                                     ; CONDITIONAL BRANCH INST. AND <====
9504                                     ; REPLACE THE MOVE INSTRUCTION <====
9505                                     ; WHICH FOLLOWS W/ 766 <====
9506 032332 012742 000754 MOV #754,-(R2) ;MOVE TO MAILBOX # ***** 754 *****
9507 032336 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9508 032340 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $STNM
9509                                     ; OR SEQUENCE ERROR
*****
9510                                     ;*****
9511 ;TEST 321 TEST THAT PROPER P.C. IS SAVED
9512                                     ;*****
9513 032342 005212 TS321: INC (R2) ;UPDATE TEST NUMBER
9514 032344 022712 000321 CMP #321,(R2) ;SEQUENCE ERROR?
9515 032350 001012 BNE TS322-10 ;BR TO ERROR HALT ON SEQ ERROR
```



```

9516 032352 012706 001000      MOV      #BUFF,SP      ;STACK POINTER SETUP
9517 032356 012767 032366 145434  MOV      #RETC2,RTRAP2 ;RETURN FROM TRAP POINTER
9518 032364 000004      IOT                      ;TRAP ON THIS INSTRUCTION
9519 032366 022767 032366 146400  RETC2:  CMP      #.BUFF-4  ;CHECK FOR INCREMENTED P.C.
9520 032374 001404      BEQ      TS322
9521      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
9522      ;         CONDITIONAL BRANCH INST. AND <---
9523      ;         REPLACE THE MOVE INSTRUCTION <---
9524      ;         WHICH FOLLOWS W/ 765 <---
9525 032376 012742 000755      MOV      #755,-(R2)    ;MOVE TO MAILBOX # ***** 755 *****
9526 032402 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9527 032404 000000      HALT                    ;INCORRECT P.C.,OR WRONG $STNM
9528      ; OR SEQUENCE ERROR
9529
9530      ;*****
9531      ;TEST 322      TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
9532      ;*****
9532 032406 005212      TS322:  INC      (R2)    ;UPDATE TEST NUMBER
9533 032410 022712 000322      CMP      #322,(R2)    ;SEQUENCE ERROR?
9534 032414 001037      BNE      TS323-10    ;BR TO ERROR HALT ON SEQ ERROR
9535 032416 012706 001000      MOV      #BUFF,SP      ;SET UP
9536 032422 012767 032440 145370  MOV      #RETD2,RTRAP2 ;SET UP
9537 032430 005067 145342      CLR      CC          ;CLEAR CC AND PRIORITY
9538 032434 000257      CCC
9539 032436 000004      IOT
9540 032440 026727 146332 000000  RETD2:  CMP      BUFF-2,#0 ;TRAP
9541 032446 001404      BEQ      1$          ;TEST THAT OLD STATUS WENT TO STACK
9542      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9543      ;         CONDITIONAL BRANCH INST. AND <====
9544      ;         REPLACE THE MOVE INSTRUCTION <====
9545      ;         WHICH FOLLOWS W/ 762 <====
9546 032450 012742 000756      MOV      #756,-(R2)    ;MOVE TO MAILBOX # ***** 756 *****
9547 032454 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9548 032456 000000      HALT                    ;INCORRECT STATUS
9549 032460 012706 001000 1$:      MOV      #BUFF,SP      ;SET UP
9550 032464 012767 032504 145326  MOV      #RETE2,RTRAP2 ;SET UP
9551 032472 012767 000357 145276  MOV      #357,CC      ;SET PRIORITY
9552 032500 000277      SCC
9553 032502 000004      IOT
9554 032504 026727 146266 000357  RETE2:  CMP      BUFF-2,#357 ;TRAP
9555 032512 001404      BEQ      TS323      ;COMPARES STATUS ON STACK
9556      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9557      ;         CONDITIONAL BRANCH INST. AND <====
9558      ;         REPLACE THE MOVE INSTRUCTION <====
9559      ;         WHICH FOLLOWS W/ 740 <====
9560 032514 012742 000757      MOV      #757,-(R2)    ;MOVE TO MAILBOX # ***** 757 *****
9561 032520 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9562 032522 000000      HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $STNM
9563      ; OR SEQUENCE ERROR
9564      ;*****
9565      ;TEST 323      TEST THAT 'NEW' STATUS IS CORRECT
9566      ;*****
9567 032524 005212      TS323:  INC      (R2)    ;UPDATE TEST NUMBER
9568 032526 022712 000323      CMP      #323,(R2)    ;SEQUENCE ERROR?
9569 032532 001110      BNE      BR46        ;BR TO ERROR HALT ON SEQ ERROR
9570 032534 012706 001000      MOV      #BUFF,SP
9571 032540 012767 032554 145252  MOV      #RETF2,RTRAP2
    
```

```
9572 032546 005067 145250      CLR      RTRAP2+2      ;CLEAR FUTURE PRIORITY AND CC
9573 032552 000004              IOT
9574 032554              RETF2:              ;TEST FOR 'C' CLEARED
9575 032554 100004              BPL      1$
9576              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9577              ;          CONDITIONAL BRANCH INST. AND <====
9578              ;          REPLACE THE MOVE INSTRUCTION <====
9579              ;          WHICH FOLLOWS W/ 766 <====
9580 032556 012742 000760      MOV      #760,-(R2)    ;MOVE TO MAILBOX # ***** 760 *****
9581 032562 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9582 032564 000000              HALT     ;N NOT CLEARED
9583 032566              1$:
9584 032566 001004              BNE      2$
9585              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9586              ;          CONDITIONAL BRANCH INST. AND <====
9587              ;          REPLACE THE MOVE INSTRUCTION <====
9588              ;          WHICH FOLLOWS W/ 761 <====
9589 032570 012742 000761      MOV      #761,-(R2)    ;MOVE TO MAILBOX # ***** 761 *****
9590 032574 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9591 032576 000000              HALT     ;Z NOT CLEARED
9592 032600              2$:
9593 032600 102004              BVC      3$
9594              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9595              ;          CONDITIONAL BRANCH INST. AND <====
9596              ;          REPLACE THE MOVE INSTRUCTION <====
9597              ;          WHICH FOLLOWS W/ 754 <====
9598 032602 012742 000762      MOV      #762,-(R2)    ;MOVE TO MAILBOX # ***** 762 *****
9599 032606 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9600 032610 000000              HALT     ;V NOT CLEARED
9601 032612              3$:
9602 032612 103004              BCC      4$
9603              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
9604              ;          CONDITIONAL BRANCH INST. AND <
9605              ;          REPLACE THE MOVE INSTRUCTION <
9606              ;          WHICH FOLLOWS W/ 747 <
9607 032614 012742 000763      MOV      #763,-(R2)    ;MOVE TO MAILBOX # ***** 763 *****
9608 032620 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9609 032622 000000              HALT     ;C NOT CLEARED
9610 032624 032767 000340 145144 4$:  BIT      #340,CC
9611 032632 001404              BEQ      5$
9612              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -
9613              ;          CONDITIONAL BRANCH INST. AND <====
9614              ;          REPLACE THE MOVE INSTRUCTION <====
9615              ;          WHICH FOLLOWS W/ 737 <====
9616 032634 012742 000764      MOV      #764,-(R2)    ;MOVE TO MAILBOX # ***** 764 *****
9617 032640 005242              INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
9618 032642 000000              HALT     ;PRIORITY NOT ZERO
9619 032644 012706 001000 5$:  MOV      #BUFF,SP
9620 032650 012767 032666 145142  MOV      #RETF2,RTRAP2
9621 032656 012767 000357 145136  MOV      #357,RTRAP2+2
9622 032664 000004              IOT
9623 032666              RETG2:
9624 032666 100404              BMI      1$
9625              ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -
9626              ;          CONDITIONAL BRANCH INST. AND <
9627              ;          REPLACE THE MOVE INSTRUCTION <
```

```
9628
9629 032670 012742 000765      MOV    #765,-(R2)      ; WHICH FOLLOWS W/ 721 <====
9630 032674 005242              INC    -(R2)          ; MOVE TO MAILBOX # ***** 765 *****
9631 032676 000000              HALT                    ; SET MSGTYP TO FATAL ERROR
9632 032700                      1$: BEQ    2$          ; N NOT SET
9633 032700 001404
9634
9635                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9636                          ; CONDITIONAL BRANCH INST. AND <====
9637                          ; REPLACE THE MOVE INSTRUCTION <====
9638 032702 012742 000766      MOV    #766,-(R2)      ; WHICH FOLLOWS W/ 714 <====
9639 032706 005242              INC    -(R2)          ; MOVE TO MAILBOX # ***** 766 *****
9640 032710 000000              HALT                    ; SET MSGTYP TO FATAL ERROR
9641 032712                      2$: BVS    3$          ; Z NOT SET
9642 032712 102404
9643
9644                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9645                          ; CONDITIONAL BRANCH INST. AND <====
9646                          ; REPLACE THE MOVE INSTRUCTION <====
9647 032714 012742 000767      MOV    #767,-(R2)      ; WHICH FOLLOWS W/ 707 <====
9648 032720 005242              INC    -(R2)          ; MOVE TO MAILBOX # ***** 767 *****
9649 032722 000000              HALT                    ; SET MSGTYP TO FATAL ERROR
9650 032724                      3$: BCS    4$          ; V NOT SET
9651 032724 103404
9652
9653                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9654                          ; CONDITIONAL BRANCH INST. AND <====
9655                          ; REPLACE THE MOVE INSTRUCTION <====
9656 032726 012742 000770      MOV    #770,-(R2)      ; WHICH FOLLOWS W/ 702 <====
9657 032732 005242              INC    -(R2)          ; MOVE TO MAILBOX # ***** 770 *****
9658 032734 000000              HALT                    ; SET MSGTYP TO FATAL ERROR
9659 032736 016706 145034      4$: MOV    CC,SP        ; C NOT SET
9660 032742 042706 000017      BIC    #17,SP
9661 032746 022706 000340      CMP    #340,SP
9662 032752 001404      BEQ    BR46A
9663
9664                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9665                          ; CONDITIONAL BRANCH INST. AND <====
9666                          ; REPLACE THE MOVE INSTRUCTION <====
9667 032754                      BR46:
9668 032754 012742 000771      MOV    #771,-(R2)      ; WHICH FOLLOWS W/ 667 <====
9669 032760 005242              INC    -(R2)          ; MOVE TO MAILBOX # ***** 771 *****
9670 032762 000000              HALT                    ; SET MSGTYP TO FATAL ERROR
9671 032764 012767 000022 145026 BR46A: MOV    #22,20
9672 032772 005067 145024      CLR    22              ; PRIORITY WAS CHANGED, OR WRONG $TSTM
9673                          ; .+2
9674                          ; HALT
9675
9676                          ;*****
9676 032776 005212              ;TEST 324 TEST THAT A TRAP OCCURS ON AN EMT INSTRUCTION
9677 033000 022712 000324      TS324: INC    (R2)          ;*****
9678 033004 001006              ;UPDATE TEST NUMBER
9679 033006 012706 001000      CMP    #324,(R2)      ;SEQUENCE ERROR?
9680 033012 012767 033032 145010 MOV    TS325-10        ;BR TO ERROR HALT ON SEQ ERROR
9681 033020 104000              BNE
9682 033022 012742 000772      MOV    #BUFF,SP        ;STACK POINTER SETUP
9683 033026 005242              MOV    #RETA3,RTRAP3  ;RETURN LOCATION
9683                          EMT
9683                          ;RESERVE INSTRUCTION, SHOULD TRAP
9683                          MOV    #772,-(R2)      ;MOVE TO MAILBOX # ***** 772 *****
9683                          INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
```

```
9684 033030 000000          HALT          ;EMT DIDN'T TRAP,OR WRONG $STNM
9685 033032
9686
9687
9688
9689 033032 005212          RETA3:
9690 033034 022712 000325      ;*****
9691 033040 001011          ;TEST 325      TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9692 033042 012706 001000      ;*****
9693 033046 012767 033056 144754  TS325:  INC      (R2)          ;UPDATE TEST NUMBER
9694 033054 104000          ;*****
9695 033056 020627 000774      ;*****
9696 033062 001404          ;*****
9697
9698
9699
9700
9701 033064 012742 000773      ;*****
9702 033070 005242          ;*****
9703 033072 000000          ;*****
9704
9705
9706
9707
9708 033074 005212          ;*****
9709 033076 022712 000326      ;*****
9710 033102 001012          ;*****
9711 033104 012706 001000      ;*****
9712 033110 012767 033120 144712  TS326:  INC      (R2)          ;UPDATE TEST NUMBER
9713 033116 104000          ;*****
9714 033120 022767 033120 145646  RETC3:  CMP      #,BUFF-4      ;SEQUENCE ERROR?
9715 033126 001404          ;*****
9716
9717
9718
9719
9720 033130 012742 000774      ;*****
9721 033134 005242          ;*****
9722 033136 000000          ;*****
9723
9724
9725
9726
9727 033140 005212          ;*****
9728 033142 022712 000327      ;*****
9729 033146 001037          ;*****
9730 033150 012706 001000      ;*****
9731 033154 012767 033172 144646  RETD3:  INC      (R2)          ;UPDATE TEST NUMBER
9732 033162 005067 144610      ;*****
9733 033166 000257          ;*****
9734 033170 104000          ;*****
9735 033172 026727 145600 000000  RETD3:  CMP      BUFF-2,#0      ;SEQUENCE ERROR?
9736 033200 001404          ;*****
9737
9738
9739
```

```
9740
9741 033202 012742 000775      MOV    #775,-(R2)      ; MOVE TO MAILBOX # ***** 775 ***** <====
9742 033206 005242              INC    -(R2)          ; SET MSGTYP TO FATAL ERROR
9743 033210 000000              HALT                    ; INCORRECT STATUS
9744 033212 012706 001000      1$:  MOV    #BUFF,SP      ; SET UP
9745 033216 012767 033236 144604  MOV    #RETE3,RTRAP3   ; SET UP
9746 033224 012767 000357 144544  MOV    #357,CC         ; SET PRIORITY
9747 033232 000277              SCC                    ; SET CC
9748 033234 104000              EMT                    ; TRAP
9749 033236 026727 145534 000357 RETE3: CMP    BUFF-2,#357   ; COMPARES STATUS ON STACK
9750 033244 001404              BEQ    TS330
9751
9752
9753
9754
9755 033246 012742 000776      MOV    #776,-(R2)      ; MOVE TO MAILBOX # ***** 776 ***** <====
9756 033252 005242              INC    -(R2)          ; SET MSGTYP TO FATAL ERROR
9757 033254 000000              HALT                    ; INCORRECT STATUS ON STACK,OR WRONG $STNM
9758
9759
9760
9761
9762 033256 005212              TS330: INC    (R2)      ; UPDATE TEST NUMBER
9763 033260 022712 000330      CMP    #330,(R2)      ; SEQUENCE ERROR?
9764 033264 001106              BNE    TS331-10       ; BR TO ERROR HALT ON SEQ ERROR
9765 033266 012706 001000      MOV    #BUFF,SP
9766 033272 012767 033306 144530  MOV    #RETF3,RTRAP3
9767 033300 005067 144526      CLR    RTRAP3+2      ; CLEAR FUTURE PRIORITY AND CC
9768 033304 104000              EMT
9769 033306              RETF3:
9770 033306 100004              BPL    1$
9771
9772
9773
9774
9775 033310 012742 000777      MOV    #777,-(R2)      ; MOVE TO MAILBOX # ***** 777 ***** <====
9776 033314 005242              INC    -(R2)          ; SET MSGTYP TO FATAL ERROR
9777 033316 000000              HALT                    ; C NOT CLEARED
9778 033320              1$:
9779 033320 001004              BNE    2$
9780
9781
9782
9783
9784 033322 012742 001000      MOV    #1000,-(R2)     ; MOVE TO MAILBOX # ***** 1000 ***** <====
9785 033326 005242              INC    -(R2)          ; SET MSGTYP TO FATAL ERROR
9786 033330 000000              HALT                    ; Z NOT CLEARED
9787 033332              2$:
9788 033332 102004              BVC    3$
9789
9790
9791
9792
9793 033334 012742 001001      MOV    #1001,-(R2)     ; MOVE TO MAILBOX # ***** 1001 ***** <====
9794 033340 005242              INC    -(R2)          ; SET MSGTYP TO FATAL ERROR
9795 033342 000000              HALT                    ; V NOT CLEARED

;*****
;TEST 330 TEST THAT 'NEW' STATUS IS CORRECT
;*****
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 762 <====
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 740 <====
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 761 <====
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 754 <====
```

```
9796 033344          3$:
9797 033344 103004          BCC      4$
9798
9799
9800
9801
9802 033346 012742 001002          MOV      #1002,-(R2)
9803 033352 005242          INC      -(R2)
9804 033354 000000          HALT
9805 033356 032767 000340 144412 4$:    BIT      #340,CC
9806 033364 001404          BEQ      5$
9807
9808
9809
9810
9811 033366 012742 001003          MOV      #1003,-(R2)
9812 033372 005242          INC      -(R2)
9813 033374 000000          HALT
9814 033376 012706 001000          5$:    MOV      #BUFF,SP
9815 033402 012767 033420 144420    MOV      #RETG3,RTRAP3
9816 033410 012767 000357 144414    MOV      #357,RTRAP3+2
9817 033416 104000          EMT
9818 033420          RETG3:
9819 033420 100404          BMI      1$
9820
9821
9822
9823
9824 033422 012742 001004          MOV      #1004,-(R2)
9825 033426 005242          INC      -(R2)
9826 033430 000000          HALT
9827 033432          1$:
9828 033432 001404          BEQ      2$
9829
9830
9831
9832
9833 033434 012742 001005          MOV      #1005,-(R2)
9834 033440 005242          INC      -(R2)
9835 033442 000000          HALT
9836 033444          2$:
9837 033444 102404          BVS      3$
9838
9839
9840
9841
9842 033446 012742 001006          MOV      #1006,-(R2)
9843 033452 005242          INC      -(R2)
9844 033454 000000          HALT
9845 033456          3$:
9846 033456 103404          BCS      4$
9847
9848
9849
9850
9851 033460 012742 001007          MOV      #1007,-(R2)
```

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 747
MOVE TO MAILBOX # ***** 1002 *****
SET MSGTYP TO FATAL ERROR
C NOT CLEARED
TEST PRIORITY

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 737
MOVE TO MAILBOX # ***** 1003 *****
SET MSGTYP TO FATAL ERROR
PRIORITY NOT ZERO

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 721
MOVE TO MAILBOX # ***** 1004 *****
SET MSGTYP TO FATAL ERROR
N NOT SET

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 714
MOVE TO MAILBOX # ***** 1005 *****
SET MSGTYP TO FATAL ERROR
Z NOT SET

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 707
MOVE TO MAILBOX # ***** 1006 *****
SET MSGTYP TO FATAL ERROR
V NOT SET

TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
CONDITIONAL BRANCH INST. AND
REPLACE THE MOVE INSTRUCTION
WHICH FOLLOWS W/ 702
MOVE TO MAILBOX # ***** 1007 *****

```

9852 033464 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9853 033466 000000          HALT                    ;C NOT SET
9854 033470 000257          CCC
9855 033472 022767 000340 144276 4$:  CMP      #340,CC
9856 033500 001404          BEQ      TS331
9857                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9858                                     ;          CONDITIONAL BRANCH INST. AND <====
9859                                     ;          REPLACE THE MOVE INSTRUCTION <====
9860                                     ;          WHICH FOLLOWS W/ 671 <====
9861 033502 012742 001010          MOV      #1010,-(R2)    ;MOVE TO MAILBOX # ***** 1010 *****
9862 033506 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9863 033510 000000          HALT                    ;PRIORITY WAS CHANGED,OR WRONG $STNM
9864                                     ; OR SEQUENCE ERROR
9865 ;*****
9866 ;TEST 331 TEST THAT ALL COMBINATION OF EMT WILL CAUSE A TRAP
9867 ;*****
9868 033512 005212          TS331: INC      (R2)          ;UPDATE TEST NUMBER
9869 033514 022712 000331          CMP      #331,(R2)     ;SEQUENCE ERROR?
9870 033520 001011          BNE      3R47          ;BR TO ERROR HALT ON SEQ ERROR
9871                                     ;**** F11 **** ADD +376 TO SHORTEN TEST
9872 033522 012767 104376 000012          MOV      #EMT+376,RB   ;INITIALIZE BASE EMT INSTRUCTION
9873 033530 012767 033554 144272          MOV      #RA,30
9874 033536 012706 001000          RC:    MOV      #BUFF,SP ;RETURN FROM TRAP TO RA
9875 033542 104000          RB:    EMT          ;SET UP STACK POINTER
9876 033544          BR47:          ;TRAP INST. WILL BE MODIFIED TO EMT+377
9877 033544 012742 001011          MOV      #1011,-(R2)   ;MOVE TO MAILBOX # ***** 1011 *****
9878 033550 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9879 033552 000000          HALT                    ;PREVIOUS INST FAILED TO TRAP,OR WRONG $STNM
9880 033554 005267 177762          RA:    INC      RB          ;INCREMENT TRAP INSTRUCTION
9881 033560 022767 104377 177754          CMP      #104377,RB   ;EMT+377 TO EMT?
9882 033566 103363          BHIS   RC          ;HAVE WE TESTED ALL
9883                                     ;YES
9884 033570 012767 000032 144232          MOV      #32,30
9885 033576 005067 144230          CLR      32          ;/.+
9886                                     ;HALT
9887 ;*****
9888 ;TEST 332 TEST THAT A TRAP OCCURES ON AN 'TRACE-TRT' INSTRUCTION
9889 ;*****
9889 033602 005212          TS332: INC      (R2)          ;UPDATE TEST NUMBER
9890 033604 022712 000332          CMP      #332,(R2)     ;SEQUENCE ERROR?
9891 033610 001006          BNE      TS333-10      ;BR TO ERROR HALT ON SEQ ERROR
9892 033612 012706 001000          MOV      #BUFF,SP     ;STACK POINTER SETUP
9893 033616 012767 033636 144170          MOV      #RETA4,RTRAP4 ;RETURN LOCATION
9894 033624 000003          TRT                    ;RESERVED INSTRUCTION, SHOULD TRAP
9895 033626 012742 001012          MOV      #1012,-(R2)   ;MOVE TO MAILBOX # ***** 1012 *****
9896 033632 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
9897 033634 000000          HALT                    ;TRT DIDN'T TRAP,OR WRONG $STNM
9898          RETA4:
9899 ;*****
9900 ;TEST 333 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
9901 ;*****
9902 033636 005212          TS333: INC      (R2)          ;UPDATE TEST NUMBER
9903 033640 022712 000333          CMP      #333,(R2)     ;SEQUENCE ERROR?
9904 033644 001011          BNE      TS334-10      ;BR TO ERROR HALT ON SEQ ERROR
9905 033646 012706 001000          MOV      #BUFF,SP     ;STACK POINTER SETUP
9906 033652 012767 033662 144134          MOV      #RETB4,RTRAP4 ;RETURN POINTER
9907 033660 000003          TRT                    ;RESERVED INSTRUCTION
    
```

```

CJKDB-C DCF11-AA CPU DIAG. MACY11 30A(1052) 07-MAR-80 12:18 N 14 PAGE 182
CJKDBC.P11 07-MAR-80 12:17 T333 TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION SEQ 0182

9908 033662 020627 000774 RETB4: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
9909 033666 001404 BEQ TS334 ;
9910 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9911 ; CONDITIONAL BRANCH INST. AND <====
9912 ; REPLACE THE MOVE INSTRUCTION <====
9913 ; WHICH FOLLOWS W/ 766 <====
9914 033670 012742 001013 MOV #1013,-(R2) ;MOVE TO MAILBOX # ***** 1013 *****
9915 033674 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9916 033676 000000 HALT ;NOT DECREMENTED TWO WORDS,OR WRONG $STNM
9917 ; OR SEQUENCE ERROR
9918 ;*****
9919 ;TEST 334 TEST THAT PROPER P.C. IS SAVED
9920 ;*****
9921 033700 005212 TS334: INC (R2) ;UPDATE TEST NUMBER
9922 033702 022712 000334 CMP #334,(R2) ;SEQUENCE ERROR?
9923 033706 001012 BNE TS335-10 ;BR TO ERROR HALT ON SEQ ERROR
9924 033710 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP
9925 033714 012767 033724 144072 MOV #RETC4,RTRAP4 ;RETURN FROM TRAP POINTER
9926 033722 000003 TRT ;TRAP ON THIS INSTRUCTION
9927 033724 022767 033724 145042 RETC4: CMP #.BUFF-4 ;CHECK FOR INCREMENTED P.C.
9928 033732 001404 BEQ TS335 ;
9929 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9930 ; CONDITIONAL BRANCH INST. AND <====
9931 ; REPLACE THE MOVE INSTRUCTION <====
9932 ; WHICH FOLLOWS W/ 765 <====
9933 033734 012742 001014 MOV #1014,-(R2) ;MOVE TO MAILBOX # ***** 1014 *****
9934 033740 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9935 033742 000000 HALT ;INCORRECT P.C.,OR WRONG $STNM
9936 ; OR SEQUENCE ERROR
9937 ;*****
9938 ;TEST 335 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
9939 ;*****
9940 033744 005212 TS335: INC (R2) ;UPDATE TEST NUMBER
9941 033746 022712 000335 CMP #335,(R2) ;SEQUENCE ERROR?
9942 033752 001037 BNE TS336-10 ;BR TO ERROR HALT ON SEQ ERROR
9943 033754 012706 001000 MOV #BUFF,SP ;SET UP
9944 033760 012767 033776 144026 MOV #RETD4,RTRAP4 ;SET UP
9945 033766 005067 144004 CLR CC ;CLEAR CC AND PRIORITY
9946 033772 000257 CCC ;
9947 033774 000003 TRT ;TRAP
9948 033776 026727 144774 000000 RETD4: CMP BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK
9949 ;TEST FOR ALL ZEROS
9950 034004 001404 BEQ 1$ ;
9951 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
9952 ; CONDITIONAL BRANCH INST. AND <====
9953 ; REPLACE THE MOVE INSTRUCTION <====
9954 ; WHICH FOLLOWS W/ 762 <====
9955 034006 012742 001015 MOV #1015,-(R2) ;MOVE TO MAILBOX # ***** 1015 *****
9956 034012 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
9957 034014 000000 HALT ;INCORRECT STATUS
9958 034016 012706 001000 1$: MOV #BUFF,SP ;SET UP
9959 034022 012767 034042 143764 MOV #RETE4,RTRAP4 ;SET UP
9960 034030 012767 000357 143740 MOV #357,CC ;SET PRIORITY
9961 034036 000277 SCC ;SET-SET CC
9962 034040 000003 TRT ;TRAP
9963 034042 026727 144730 000357 RETE4: CMP BUFF-2,#357 ;COMPARES STATUS ON STACK

```



```
9964 034050 001174 BEQ TS336
9965
9966
9967
9968
9969 034052 012742 001016 MOV #1016,-(R2)
9970 034056 005242 INC -(R2)
9971 034060 000000 HALT
9972
9973
9974
9975
9976 034062 005212 TS336: INC (R2)
9977 034064 022712 000336 CMP #336,(R2)
9978 034070 001110 BNE BR51
9979 034072 012706 001000 MOV #BUFF,SP
9980 034076 012767 034112 143710 MOV #RETF4,RTRAP4
9981 034104 005067 143706 CLR RTRAP4+2
9982 034110 000003 TRT
9983 034112 RETF4:
9984 034112 100004 BPL 1$
9985
9986
9987
9988
9989 034114 012742 001017 MOV #1017,-(R2)
9990 034120 005242 INC -(R2)
9991 034122 000000 HALT
9992 034124 1$:
9993 034124 001004 BNE 2$
9994
9995
9996
9997
9998 034126 012742 001020 MOV #1020,-(R2)
9999 034132 005242 INC -(R2)
10000 034134 000000 HALT
10001 034136 2$:
10002 034136 102004 BVC 3$
10003
10004
10005
10006
10007 034140 012742 001021 MOV #1021,-(R2)
10008 034144 005242 INC -(R2)
10009 034146 000000 HALT
10010 034150 3$:
10011 034150 103004 BCC 4$
10012
10013
10014
10015
10016 034152 012742 001022 MOV #1022,-(R2)
10017 034156 005242 INC -(R2)
10018 034160 000000 HALT
10019 034162 032767 000340 143606 4$: BIT #340,CC
```

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

MOVE TO MAILBOX # ***** 1016 *****
SET MSGTYP TO FATAL ERROR
INCORRECT STATUS ON STACK,OR WRONG \$TSTNM
OR SEQUENCE ERROR

TEST 336 TEST THAT 'NEW' STATUS IS CORRECT

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

CLEAR FUTURE PRIORITY AND CC

TEST FOR 'C' CLEARED

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

MOVE TO MAILBOX # ***** 1017 *****
SET MSGTYP TO FATAL ERROR
C NOT CLEARED

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

MOVE TO MAILBOX # ***** 1020 *****
SET MSGTYP TO FATAL ERROR
Z NOT CLEARED

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

MOVE TO MAILBOX # ***** 1021 *****
SET MSGTYP TO FATAL ERROR
V NOT CLEARED

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

MOVE TO MAILBOX # ***** 1022 *****
SET MSGTYP TO FATAL ERROR
C NOT CLEARED
TEST PRIORITY


```
10076 034312  
10077 034312 012742 001030  
10078 034316 005242  
10079 034320 000000  
10080 034322 012767 000016 143464 BR51A: MOV #16,14  
10081 034330 005067 143462 CLR 16  
10082  
10083 ;PDP-11 ILLEGAL AND ADDRESS INSTRUCTION TEST  
10084 ;ALL INSTRUCTIONS THAT ARE RESERVED  
10085 ;SHOULD TRAP TO LOCATION 4, AND THE  
10086 ;PC THAT POINTS TO THE TRAPPING INSTRUCTION  
10087 ;SHOULD BE PLACED ON THE STACK  
10088  
10089 ;*****  
10090 ;TEST 337 TEST THAT A TRAP OCCURS ON AN ILLEGAL INSTRUCTION  
10091 ;*****  
10092 034334 005212 TS337: INC (R2) ;UPDATE TEST NUMBER  
10093 034336 022712 000337 CMP #337,(R2) ;SEQUENCE ERROR?  
10094 034342 001006 BNE TS340-10 ;BR TO ERROR HALT ON SEQ ERROR  
10095 034344 012706 001000 MOV #BUFF,SP ;STACK POINTER SETUP  
10096 034350 012767 034370 143426 MOV #RETA5,RTRAP5 ;RETURN LOCATION  
10097 034356 000100 JMP %0 ;ILLEGAL INSTRUCTION, SHOULD TRAP
```

```
10098 034360 012742 001031      MOV    #1031,-(R2)      ;MOVE TO MAILBOX # ***** 1031 *****
10099 034364 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10100 034366 000000              HALT                    ;ILLEGAL INSTRUCTION DIDN'T TRAP,OR WRONG $STSTM
10101 034370
10102
10103      RETA5:
;*****
;TEST 340      TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
;*****
10104
10105 034370 005212      TS340: INC    (R2)           ;UPDATE TEST NUMBER
10106 034372 022712 000340      CMP    #340,(R2)       ;SEQUENCE ERROR?
10107 034376 001011      BNE    TS341-10        ;BR TO ERROR HALT ON SEQ ERROR
10108 034400 012706 001000      MOV    #BUFF,SP        ;STACK POINTER SETUP
10109 034404 012767 034414 143372      MOV    #RETB5,RTRAP5   ;RETURN POINTER
10110 034412 000100      JMP    %0              ;RESERVED INSTRUCTION
10111 034414 020627 000774      RETB5: CMP    SP,#BUFF-4 ;TEST DECREMENT OF SP
10112 034420 001404      BEQ    TS341
10113
10114      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
10115      ;          CONDITIONAL BRANCH INST. AND <- -
10116      ;          REPLACE THE MOVE INSTRUCTION <- -
10117      ;          WHICH FOLLOWS W/ 766 <- -
10117 034422 012742 001032      MOV    #1032,-(R2)     ;MOVE TO MAILBOX # ***** 1032 *****
10118 034426 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10119 034430 000000              HALT                    ;NOT DECREMENTED TWO WORDS,OR WRONG $STSTM
10120
10121      ; OR SEQUENCE ERROR
10122      ;*****
;TEST 341      TEST THAT PROPER P.C. IS SAVED
;*****
10123
10124 034432 005212      TS341: INC    (R2)           ;UPDATE TEST NUMBER
10125 034434 022712 000341      CMP    #341,(R2)       ;SEQUENCE ERROR?
10126 034440 001012      BNE    TS342-10        ;BR TO ERROR HALT ON SEQ ERROR
10127 034442 012706 001000      MCV   #BUFF,SP        ;STACK POINTER SETUP
10128 034446 012767 034456 143330      MOV    #RETC5,RTRAP5   ;RETURN FROM TRAP POINTER
10129 034454 000100      JMP    %0              ;TRAP ON THIS INSTRUCTION
10130 034456 022767 034456 144310      RETC5: CMP    #.BUFF-4   ;CHECK FOR INCREMENTED P.C.
10131 034464 001404      BEQ    TS342
10132
10133      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10134      ;          CONDITIONAL BRANCH INST. AND <====
10135      ;          REPLACE THE MOVE INSTRUCTION <====
10136      ;          WHICH FOLLOWS W/ 765 <====
10136 034466 012742 001033      MOV    #1033,-(R2)     ;MOVE TO MAILBOX # ***** 1033 *****
10137 034472 005242              INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
10138 034474 000000              HALT                    ;INCORRECT P.C.,OR WRONG $STSTM
10139
10140      ; OR SEQUENCE ERROR
10141      ;*****
;TEST 342      TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
;*****
10142
10143 034476 005212      TS342: INC    (R2)           ;UPDATE TEST NUMBER
10144 034500 022712 000342      CMP    #342,(R2)       ;SEQUENCE ERROR?
10145 034504 001037      BNE    TS343-10        ;BR TO ERROR HALT ON SEQ ERROR
10146 034506 012706 001000      MOV    #BUFF,SP        ;SET UP
10147 034512 012767 034530 143264      MOV    #RETD5,RTRAP5   ;SET UP
10148 034520 005067 143252      CLR    CC              ;CLEAR CC AND PRIORITY
10149 034524 000257      CCC
10150 034526 000100      JMP    %0              ;TRAP
10151 034530 026727 144242 000000      RETD5: CMP    BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK
10152 034536 001404      BEQ    1$
10153
10153      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <==
```

```

10154      :                               :      CONDITIONAL BRANCH INST. AND <====
10155      :                               :      REPLACE THE MOVE INSTRUCTION <====
10156      :                               :      WHICH FOLLOWS W/ 762         <====
10157 034540 012742 001034      MOV      #1034,-(R2)      ;MOVE TO MAILBOX # ***** 1034 *****
10158 034544 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10159 034546 000000      HALT                    ;INCORRECT STATUS
10160 034550 012706 001000      1$:  MOV      #BUFF,SP      ;SET UP
10161 034554 012767 034574 143222  MOV      #RETE5,RTRAPS   ;SET UP
10162 034562 012767 000357 143206  MOV      #357,CC        ;SET PRIORITY
10163 034570 000277      SCC                    ;SET CC
10164 034572 000100      JMP      %0             ;TRAP
10165 034574 026727 144176 000357 RETE5: CMP      BUFF-2,#357     ;COMPARES STATUS ON STACK
10166 034602 001404      BEQ      TS343         ;
10167      :                               :      TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
10168      :                               :      CONDITIONAL BRANCH INST. AND <- -
10169      :                               :      REPLACE THE MOVE INSTRUCTION <- -
10170      :                               :      WHICH FOLLOWS W/ 740         <-
10171 034604 012742 001035      MOV      #1035,-(R2)   ;MOVE TO MAILBOX # ***** 1035 *****
10172 034610 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10173 034612 000000      HALT                    ;INCORRECT STATUS ON STACK,OR WRONG $TSTNM
10174      :                               :      OR SEQUENCE ERROR
10175      :*****
10176      :TEST 343      TEST THAT 'NEW' STATUS IS CORRECT
10177      :*****
10178 034614 005212      TS343: INC      (R2)      ;UPDATE TEST NUMBER
10179 034616 022712 000343      CMP      #343,(R2)     ;SEQUENCE ERROR?
10180 034622 001106      BNE      TS344-10      ;BR TO ERROR HALT ON SEQ ERROR
10181 034624 012706 001000      MOV      #BUFF,SP      ;
10182 034630 012767 034644 143146  MOV      #RETF5,RTRAPS  ;
10183 034636 005067 143144      CLR      RTRAPS+2      ;CLEAR FUTURE PRIORITY AND CC
10184 034642 000100      JMP      %0             ;
10185 034644      RETF5:              ;TEST FOR 'C' CLEARED
10186 034644 100004      BPL      1$            ;
10187      :                               :      TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
10188      :                               :      CONDITIONAL BRANCH INST. AND <
10189      :                               :      REPLACE THE MOVE INSTRUCTION <
10190      :                               :      WHICH FOLLOWS W/ 766         <
10191 034646 012742 001036      MOV      #1036,-(R2)   ;MOVE TO MAILBOX # ***** 1036 *****
10192 034652 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10193 034654 000000      HALT                    ;C NOT CLEARED
10194 034656      1$:              ;
10195 034656 001004      BNE      2$            ;
10196      :                               :      TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10197      :                               :      CONDITIONAL BRANCH INST. AND <====
10198      :                               :      REPLACE THE MOVE INSTRUCTION <====
10199      :                               :      WHICH FOLLOWS W/ 761         <====
10200 034660 012742 001037      MOV      #1037,-(R2)   ;MOVE TO MAILBOX # ***** 1037 *****
10201 034664 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10202 034666 000000      HALT                    ;Z NOT CLEARED
10203 034670      2$:              ;
10204 034670 102004      BVC      3$            ;
10205      :                               :      TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10206      :                               :      CONDITIONAL BRANCH INST. AND <====
10207      :                               :      REPLACE THE MOVE INSTRUCTION <====
10208      :                               :      WHICH FOLLOWS W/ 754         <====
10209 034672 012742 001040      MOV      #1040,-(R2)   ;MOVE TO MAILBOX # ***** 1040 *****

```

10210	034676	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR	
10211	034700	000000				HALT			:V NOT CLEARED	
10212	034702				3\$:					
10213	034702	103004				BCC	4\$			
10214									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
10215									: CONDITIONAL BRANCH INST. AND	<====
10216									: REPLACE THE MOVE INSTRUCTION	<====
10217									: WHICH FOLLOWS W/ 747	<====
10218	034704	012742	001041			MOV	#1041,-(R2)		:MOVE TO MAILBOX # ***** 1041 *****	
10219	034710	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR	
10220	034712	000000				HALT			:C NOT CLEARED	
10221	034714	032767	000357	143054	4\$:	BIT	#357,CC		:TEST PRIORITY	
10222	034722	001404				BEG	5\$			
10223									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
10224									: CONDITIONAL BRANCH INST. AND	<====
10225									: REPLACE THE MOVE INSTRUCTION	<====
10226									: WHICH FOLLOWS W/ 737	<====
10227	034724	012742	001042			MOV	#1042,-(R2)		:MOVE TO MAILBOX # ***** 1042 *****	
10228	034730	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR	
10229	034732	000000				HALT			:PRIORITY NOT ZERO	
10230	034734	012706	001000		5\$:	MOV	#BUFF,SP			
10231	034740	012767	034756	143036		MOV	#RETG5,RTRAP5			
10232	034746	012767	000357	143032		MOV	#357,RTRAP5+2		:SET NEW 'CC' AND PRIORITY	
10233	034754	000100				JMP	%0		:TRAP HERE	
10234	034756				RETG5:					
10235	034756	100404				BMI	1\$			
10236									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
10237									: CONDITIONAL BRANCH INST. AND	<====
10238									: REPLACE THE MOVE INSTRUCTION	<====
10239									: WHICH FOLLOWS W/ 721	<====
10240	034760	012742	001043			MOV	#1043,-(R2)		:MOVE TO MAILBOX # ***** 1043 *****	
10241	034764	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR	
10242	034766	000000				HALT			:N NOT SET	
10243	034770				1\$:					
10244	034770	001404				BEG	2\$			
10245									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
10246									: CONDITIONAL BRANCH INST. AND	<====
10247									: REPLACE THE MOVE INSTRUCTION	<====
10248									: WHICH FOLLOWS W/ 714	<====
10249	034772	012742	001044			MOV	#1044,-(R2)		:MOVE TO MAILBOX # ***** 1044 *****	
10250	034776	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR	
10251	035000	000000				HALT			:Z NOT SET	
10252	035002				2\$:					
10253	035002	102404				BVS	3\$			
10254									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
10255									: CONDITIONAL BRANCH INST. AND	<====
10256									: REPLACE THE MOVE INSTRUCTION	<====
10257									: WHICH FOLLOWS W/ 707	<====
10258	035004	012742	001045			MOV	#1045,-(R2)		:MOVE TO MAILBOX # ***** 1045 *****	
10259	035010	005242				INC	-(R2)		:SET MSGTYP TO FATAL ERROR	
10260	035012	000000				HALT			:V NOT SET	
10261	035014				3\$:					
10262	035014	103404				BCS	4\$			
10263									: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
10264									: CONDITIONAL BRANCH INST. AND	<====
10265									: REPLACE THE MOVE INSTRUCTION	<====

```
10266
10267 035016 012742 001046      MOV    #1046,-(R2)      ; MOVE TO MAILBOX # ***** 1046 *****
10268 035022 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
10269 035024 000000              HALT                    ; C NOT SET
10270 035026 016706 142744      4$:  MOV    CC,SP
10271 035032 022706 000357      CMP    #357,SP
10272 035036 001404              BEQ    TS344
10273
10274
10275
10276
10277 035040 012742 001047      MOV    #1047,-(R2)     ; MOVE TO MAILBOX # ***** 1047 *****
10278 035044 005242              INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
10279 035046 000000              HALT                    ; PRIORITY WAS CHANGED,OR WRONG $STNM
10280
10281
10282
10283
10284 035050 005212              *****
10285 035052 022712 000344      :TEST 344      TEST THAT A TRAP OCCURES ON ALL ILLEGAL INSTRUCTION
10286 035056 001006              *****
10287 035060 012706 001000      TS344:  INC    (R2)           ; UPDATE TEST NUMBER
10288 035064 012767 035104 142712  CMP    #344,(R2)       ; SEQUENCE ERROR?
10289 035072 004000              BNE    TS345-10        ; BR TO ERROR HALT ON SEQ ERROR
10290 035074 012742 001050      MOV    #BUFF,SP        ; STACK POINTER SETUP
10291 035100 005242              MOV    #RETH5,RTRAPS   ; RETURN LOCATION
10292 035102 000000              JSR    %0,%0           ; RESERVED INSTRUCTION, SHOULD TRAP
10293 035104              MOV    #1050,-(R2)     ; MOVE TO MAILBOX # ***** 1050 *****
10294
10295
10296
10297 035104 005212              *****
10298 035106 022712 000345      :TEST 345      TEST DECREMENT OF STACK POINTER ON A TRAP OPERATION
10299 035112 001011              *****
10300 035114 012706 001000      TS345:  INC    (R2)           ; UPDATE TEST NUMBER
10301 035120 012767 035130 142656  CMP    #345,(R2)       ; SEQUENCE ERROR?
10302 035126 004000              BNE    TS346-10        ; BR TO ERROR HALT ON SEQ ERROR
10303 035130 020627 000774      MOV    #BUFF,SP        ; STACK POINTER SETUP
10304 035134 001404              MOV    #RETJ,RTRAPS    ; RETURN POINTER
10305
10306
10307
10308
10309 035136 012742 001051      RETJ:  JSR    %0,%0     ; RESERVED INSTRUCTION
10310 035142 005242              CMP    SP,#BUFF-4      ; TEST DECREMENT OF SP
10311 035144 000000              BEQ    TS346
10312
10313
10314
10315
10316 035146 005212              *****
10317 035150 022712 000346      :TEST 346      TEST THAT PROPER P.C. IS SAVED
10318 035154 001012              *****
10319 035156 012706 001000      TS346:  INC    (R2)           ; UPDATE TEST NUMBER
10320 035162 012767 035172 142614  CMP    #346,(R2)       ; SEQUENCE ERROR?
10321 035170 004000              BNE    TS347-10        ; BR TO ERROR HALT ON SEQ ERROR
INSTK:  MOV    #BUFF,SP        ; STACK POINTER SETUP
        MOV    #RETK,RTRAPS   ; RETURN FROM TRAP POINTER
        JSR    %0,%0           ; TRAP ON THIS INSTRUCTION
```

```
10322 035172 022767 035172 143574 RETK:  CMP      #INSTK+2,BUFF-4 ;CHECK FOR INCREMENED P.C.
10323 035200 001404                BEQ      TS347
10324                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10325                                ;                CONDITIONAL BRANCH INST. AND <----
10326                                ;                REPLACE THE MOVE INSTRUCTION <----
10327                                ;                WHICH FOLLOWS W/ 765 <====
10328 035202 012742 001052                MOV      #1052,-(R2) ;MOVE TO MAILBOX # ***** 1052 *****
10329 035206 005242                INC      -(R2) ;SET MSGTYP TO FATAL ERROR
10330 035210 000000                HALT     ;INCORRECT P.C.,OR WRONG $STNM
10331                                ; OR SEQUENCE ERROR
10332
10333
10334
```

```
*****
:TEST 347 TEST THAT 'OLD' CC AND PRIORITY ARE PLACED ON STACK
*****
```

```
TS347:  INC      (R2) ;UPDATE TEST NUMBER
10337 035214 022712 000347                CMP      #347,(R2) ;SEQUENCE ERROR?
10338 035220 001037                BNE     TS350-10 ;BR TO ERROR HALT ON SEQ ERROR
10339 035222 012706 001000                MOV     #BUFF,SP ;SET UP
10340 035226 012767 035244 142550                MOV     #RETL,RTRAPS ;SET UP
10341 035234 005067 142536                CLR     CC ;CLEAR CC AND PRIORITY
10342 035240 000257                CCC
10343 035242 004000                JSR     %0,%0 ;TRAP
10344 035244 026727 143526 000000 RETL:  CMP     BUFF-2,#0 ;TEST THAT OLD STATUS WENT TO STACK
10345 035252 001404                BEQ     1$
```

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

```
10350 035254 012742 001053                MOV     #1053,-(R2) ;MOVE TO MAILBOX # ***** 1053 *****
10351 035260 005242                INC     -(R2) ;SET MSGTYP TO FATAL ERROR
10352 035262 000000                HALT    ;INCORRCT STATUS
10353 035264 012706 001000 1$:  MOV     #BUFF,SP ;SET UP
10354 035270 012767 035310 142506                MOV     #RETM,RTRAPS ;SET UP
10355 035276 012767 000357 142472                MOV     #357,CC ;SET PRIORITY
10356 035304 000277                SCC     ;SET CC
10357 035306 004000                JSR     %0,%0 ;TRAP
10358 035310 026727 143462 000357 RETM:  CMP     BUFF-2,#357 ;COMPARES STATUS ON STACK
10359 035316 001404                BEQ     TS350
```

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

```
10364 035320 012742 001054                MOV     #1054,-(R2) ;MOVE TO MAILBOX # ***** 1054 *****
10365 035324 005242                INC     -(R2) ;SET MSGTYP TO FATAL ERROR
10366 035326 000000                HALT    ;INCORRECT STATUS ON STACK,OR WRONG $STNM
10367                                ; OR SEQUENCE ERROR
10368
```

```
*****
:TEST 350 TEST THAT 'NEW' STATUS IS CORRECT
*****
```

```
TS350:  INC      (R2) ;UPDATE TEST NUMBER
10372 035332 022712 000350                CMP     #350,(R2) ;SEQUENCE ERROR?
10373 035336 001105                BNE     TS351-10 ;BR TO ERROR HALT ON SEQ ERROR
10374 035340 012706 001000                MOV     #BUFF,SP
10375 035344 012767 035360 142432                MOV     #RETN,RTRAPS
10376 035352 005067 142430                CLR     RTRAPS+2 ;CLEAR FUTURE PRIORITY AND CC
10377 035356 004000                JSR     %0,%0
```



```

10378 035360          RETN:          ;TEST FOR 'C' CLEARED
10379 035360 100004    BPL          1$
10380                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10381                ;          CONDITIONAL BRANCH INST. AND <====
10382                ;          REPLACE THE MOVE INSTRUCTION <====
10383                ;          WHICH FOLLOWS W/ 766 <====
10384 035362 012742 001055    MOV      #1055,-(R2)    ;MOVE TO MAILBOX # ***** 1055 *****
10385 035366 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10386 035370 000000          HALT          ;C NOT CLEARED
10387 035372          1$:
10388 035372 001004    BNE          2$
10389                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
10390                ;          CONDITIONAL BRANCH INST. AND <
10391                ;          REPLACE THE MOVE INSTRUCTION < --
10392                ;          WHICH FOLLOWS W/ 761 < = -
10393 035374 012742 001056    MOV      #1056,-(R2)    ;MOVE TO MAILBOX # ***** 1056 *****
10394 035400 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10395 035402 000000          HALT          ;Z NOT CLEARED
10396 035404          2$:
10397 035404 102004    BVC          3$
10398                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
10399                ;          CONDITIONAL BRANCH INST. AND <---
10400                ;          REPLACE THE MOVE INSTRUCTION <---
10401                ;          WHICH FOLLOWS W/ 754 <---
10402 035406 012742 001057    MOV      #1057,-(R2)    ;MOVE TO MAILBOX # ***** 1057 *****
10403 035412 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10404 035414 000000          HALT          ;V NOT CLEARED
10405 035416          3$:
10406 035416 103004    BCC          4$
10407                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10408                ;          CONDITIONAL BRANCH INST. AND <====
10409                ;          REPLACE THE MOVE INSTRUCTION <====
10410                ;          WHICH FOLLOWS W/ 747 <====
10411 035420 012742 001060    MOV      #1060,-(R2)    ;MOVE TO MAILBOX # ***** 1060 *****
10412 035424 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10413 035426 000000          HALT          ;C NOT CLEARED
10414 035430 016700 142342    MOV      CC,%0         ;TEMP STORAGE
10415 035434 001404          BEQ      5$
10416                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10417                ;          CONDITIONAL BRANCH INST. AND <====
10418                ;          REPLACE THE MOVE INSTRUCTION <====
10419                ;          WHICH FOLLOWS W/ 740 <====
10420 035436 012742 001061    MOV      #1061,-(R2)    ;MOVE TO MAILBOX # ***** 1061 *****
10421 035442 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
10422 035444 000000          HALT          ;PRIORITY NOT ZERO
10423 035446 012706 001000    MOV      #BUFF,SP
10424 035452 012767 035470 142324    MOV      #RETO,RTRAP5
10425 035460 012767 000357 142320    MOV      #357,RTRAP5+2
10426 035466 004000          JSR      %0,%0         ;SET NEW 'CC' AND PRIORITY
10427 035470          RETO:
10428 035470 100404    BMI          1$
10429                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10430                ;          CONDITIONAL BRANCH INST. AND <====
10431                ;          REPLACE THE MOVE INSTRUCTION <====
10432                ;          WHICH FOLLOWS W/ 722 <====
10433 035472 012742 001062    MOV      #1062,-(R2)    ;MOVE TO MAILBOX # ***** 1062 *****

```

```
10434 035476 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10435 035500 000000      HALT                    ;N NOT SET
10436 035502      1$:      BEQ      2$
10437 035502 001404
10438
10439
10440
10441
10442 035504 012742 001063      MOV      #1063,-(R2)   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <= --
10443 035510 005242      INC      -(R2)      ;          CONDITIONAL BRANCH INST. AND <- ==
10444 035512 000000      HALT                    ;          REPLACE THE MOVE INSTRUCTION < = -
10445 035514      2$:      BVS      3$          ;          WHICH FOLLOWS W/ 715 <-==
10446 035514 102404
10447
10448
10449
10450
10451 035516 012742 001064      MOV      #1064,-(R2)   ; MOVE TO MAILBOX # ***** 1063 *****
10452 035522 005242      INC      -(R2)      ; SET MSGTYP TO FATAL ERROR
10453 035524 000000      HALT                    ; Z NOT SET
10454 035526      3$:      BCS      4$
10455 035526 103404
10456
10457
10458
10459
10460 035530 012742 001065      MOV      #1065,-(R2)   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS < -
10461 035534 005242      INC      -(R2)      ;          CONDITIONAL BRANCH INST. AND <
10462 035536 000000      HALT                    ;          REPLACE THE MOVE INSTRUCTION <-
10463 035540 016700 142232      MOV      CC,%0        ;          WHICH FOLLOWS W/ 710 <
10464 035544 022700 000357      CMP      #357,%0
10465 035550 001404      BEQ      TS351
10466
10467
10468
10469
10470 035552 012742 001066      MOV      #1066,-(R2)   ; MOVE TO MAILBOX # ***** 1064 *****
10471 035556 005242      INC      -(R2)      ; SET MSGTYP TO FATAL ERROR
10472 035560 000000      HALT                    ; PRIORITY WAS CHANGED,OR WRONG $STNM
10473
10474
10475
10476
10477
10478 035562 005212      TS351: INC      (R2)      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10479 035564 022712 000351      CMP      #351,(R2)     ;          CONDITIONAL BRANCH INST. AND <====
10480 035570 001006      BNE      TS352-10      ;          REPLACE THE MOVE INSTRUCTION <====
10481 035572 012706 000150      MOV      #150,%6       ;          WHICH FOLLOWS W/ 672 <====
10482 035576 012767 035616 142200      MOV      #TDEC1,4      ; MOVE TO MAILBOX # ***** 1065 *****
10483 035604 005746      TST      -(6)          ; SET MSGTYP TO FATAL ERROR
10484 035606 012742 001067      MOV      #1067,-(R2)   ; PRIORITY WAS CHANGED,OR WRONG $STNM
10485 035612 005242      INC      -(R2)      ; OR SEQUENCE ERROR
10486 035614 000000      HALT
10487 035616      TDEC1:
10488
10489
```

```
10490 ;TEST 352 TEST FOR DECREMENT OF R6 ON OVERFLOW TRAP
10491 :*****
10492 035616 005212 TS352: INC (R2) ;UPDATE TEST NUMBER
10493 035620 022712 000352 CMP #352,(R2) ;SEQUENCE ERROR?
10494 035624 001011 BNE TS353-10 ;BR TO ERROR HALT ON SEQ ERROR
10495 035626 012706 000150 MOV #150,%6 ;R6 = 150
10496 035632 012767 035642 142144 MOV #TDEC2,4 ;TRAP POINTER
10497 035640 005746 TST -(6) ;WITH R6 = 150 SHOULD TRAP
10498 035642 020627 000142 TDEC2: CMP %6,#142 ;DID R6 DECREMENT
10499 035646 001404 BEQ TS353
10500 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10501 ; CONDITIONAL BRANCH INST. AND <====
10502 ; REPLACE THE MOVE INSTRUCTION <====
10503 ; WHICH FOLLOWS w/ 766 <====
10504 035650 012742 001070 MOV #1070,-(R2) ;MOVE TO MAILBOX # ***** 1070 *****
10505 035654 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10506 035656 000000 HALT ;R6 NOT = 142,OR WRONG $STNM
10507 ; OR SEQUENCE ERROR
10508
10509 :*****
10510 ;TEST 353 TEST DIFFERENT TYPES OF OVERFLOW
10511 :*****
10512 035660 005212 TS353: INC (R2) ;UPDATE TEST NUMBER
10513 035662 022712 000353 CMP #353,(R2) ;SEQUENCE ERROR?
10514 035666 001041 BNE TS354-10 ;BR TO ERROR HALT ON SEQ ERROR
10515 035670 012706 000150 MOV #150,%6
10516 035674 005067 142246 CLR 146 ;STATUS WORD OF LOC 10
10517 035700 012767 035710 142076 MOV #TDEC3,4 ;RETURN TO LOC 4
10518 035706 005246 INC -(6)
10519 035710 005767 142232 TDEC3: TST 146
10520 035714 001004 BNE 1$
10521 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
10522 ; CONDITIONAL BRANCH INST. AND <- -
10523 ; REPLACE THE MOVE INSTRUCTION <
10524 ; WHICH FOLLOWS w/ 764 < -
10525 035716 012742 001071 MOV #1071,-(R2) ;MOVE TO MAILBOX # ***** 1071 *****
10526 035722 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10527 035724 000000 HALT ;INCREMENT OPERATION NOT INHIBITED
10528 035726 012705 001000 1$: MOV #1000,%5
10529 035732 012706 000400 MOV #400,%6
10530 035736 012767 035756 142040 MOV #TDEC4,4
10531 035744 124645 CMPB -(6),-(5)
10532 035746 012742 001072 MOV #1072,-(R2) ;MOVE TO MAILBOX # ***** 1072 *****
10533 035752 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10534 035754 000000 HALT ;STACK - 400 AND DECREMENTED, SHOULD TRAP
10535 035756 012706 000400 TDEC4: MOV #400,%6
10536 035762 012767 036002 142014 MOV #TDEC7,4
10537 035770 134546 BITB -(5),-(6)
10538 035772 TDEC6:
10539 035772 012742 001073 MOV #1073,-(R2) ;MOVE TO MAILBOX # ***** 1073 *****
10540 035776 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10541 036000 000000 HALT ;NO STACK OVERFLOW,OR WRONG $STNM
10542 036002 TDEC7:
10543
10544 :*****
10545 ;TEST 354 TEST THAT AN 77 CAUSES AN OVERFLOW TRAP
```

```
10546 :*****
10547 036002 005212 TS354: INC (R2) ;UPDATE TEST NUMBER
10548 036004 022712 000354 CMP #354,(R2) ;SEQUENCE ERROR?
10549 036010 001011 BNE VDEC2 ;BR TO ERROR HALT ON SEQ ERROR
10550 036012 012706 000400 MOV #400,%6 ;SET UP STACK TO OVERFLOW
10551 036016 012767 036034 141764 MOV #VDEC2,10 ;SET UP 77 VECTOR
10552 036024 012767 036044 141752 MOV #VDEC1,4 ;SET UP OVERFLOW VECTOR
10553 036032 000077 77 ;THIS TRAP SHOULD CAUSE OVERFLOW
10554 036034 VDEC2:
10555 036034 012742 001074 MOV #1074,-(R2) ;MOVE TO MAILBOX # ***** 1074 *****
10556 036040 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10557 036042 000000 HALT ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $STNM
10558 036044 012767 000012 141736 VDEC1: MOV #10+2,10
10559 :*****
10560 :TEST 355 TEST THAT AN IOT CAUSES AN OVERFLOW TRAP
10561 :*****
10562 036052 005212 TS355: INC (R2) ;UPDATE TEST NUMBER
10563 036054 022712 000355 CMP #355,(R2) ;SEQUENCE ERROR?
10564 036060 001011 BNE VDEC4 ;BR TO ERROR HALT ON SEQ ERROR
10565 036062 012706 000400 MOV #400,%6 ;SET UP STACK TO OVERFLOW
10566 036066 012767 036104 141724 MOV #VDEC4,20 ;SET UP IOT VECTOR
10567 036074 012767 036114 141702 MOV #VDEC3,4 ;SET UP OVERFLOW VECTOR
10568 036102 000004 IOT ;THIS TRAP SHOULD CAUSE OVERFLOW
10569 036104 VDEC4:
10570 036104 012742 001075 MOV #1075,-(R2) ;MOVE TO MAILBOX # ***** 1075 *****
10571 036110 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10572 036112 000000 HALT ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $STNM
10573 036114 012767 000022 141676 VDEC3: MOV #20+2,20
10574 :*****
10575 :TEST 356 TEST THAT AN EMT CAUSES AN OVERFLOW TRAP
10576 :*****
10577 036122 005212 TS356: INC (R2) ;UPDATE TEST NUMBER
10578 036124 022712 000356 CMP #356,(R2) ;SEQUENCE ERROR?
10579 036130 001011 BNE VDEC6 ;BR TO ERROR HALT ON SEQ ERROR
10580 036132 012706 000400 MOV #400,%6 ;SET UP STACK TO OVERFLOW
10581 036136 012767 036154 141664 MOV #VDEC6,30 ;SET UP EMT VECTOR
10582 036144 012767 036164 141632 MOV #VDEC5,4 ;SET UP OVERFLOW VECTOR
10583 036152 104000 EMT ;THIS TRAP SHOULD CAUSE OVERFLOW
10584 036154 VDEC6:
10585 036154 012742 001076 MOV #1076,-(R2) ;MOVE TO MAILBOX # ***** 1076 *****
10586 036160 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10587 036162 000000 HALT ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $STNM
10588 036164 012767 000032 141636 VDEC5: MOV #30+2,30
10589 :*****
10590 :TEST 357 TEST THAT AN TRAP CAUSES AN OVERFLOW TRAP
10591 :*****
10592 036172 005212 TS357: INC (R2) ;UPDATE TEST NUMBER
10593 036174 022712 000357 CMP #357,(R2) ;SEQUENCE ERROR?
10594 036200 001011 BNE VDEC8 ;BR TO ERROR HALT ON SEQ ERROR
10595 036202 012706 000400 MOV #400,%6 ;SET UP STACK TO OVERFLOW
10596 036206 012767 036224 141620 MOV #VDEC8,34 ;SET UP TRAP VECTOR
10597 036214 012767 036234 141562 MOV #VDEC7,4 ;SET UP OVERFLOW VECTOR
10598 036222 104400 TRAP ;THIS TRAP SHOULD CAUSE OVERFLOW
10599 036224 VDEC8:
10600 036224 012742 001077 MOV #1077,-(R2) ;MOVE TO MAILBOX # ***** 1077 *****
10601 036230 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
```

```
10602 036232 000000          HALT                ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10603 036234 012767 000036 141572 VDEC7: MOV      #34+2,34
10604                                     ;*****
10605                                     ;TEST 360          TEST THAT AN TRT CAUSES AN OVERFLOW TRAP
10606                                     ;*****
10607 036242 005212          TS360: INC      (R2)                ;UPDATE TEST NUMBER
10608 036244 022712 000360          CMP      #360,(R2)                ;SEQUENCE ERROR?
10609 036250 001011          BNE     VDEC10                    ;BR TO ERROR HALT ON SEQ ERROR
10610 036252 012706 000400          MOV     #400,%6                    ;SET UP STACK TO OVERFLOW
10611 036256 012767 036274 141530          MOV     #VDEC10,14                ;SET UP TRT VECTOR
10612 036264 012767 036304 141512          MOV     #VDEC9,4                   ;SET UP OVERFLOW VECTOR
10613 036272 000003          TRT                                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10614 036274
10615 036274 012742 001100          VDEC10: MOV     #1100,-(R2)         ;MOVE TO MAILBOX # ***** 1100 *****
10616 036300 005242          INC     -(R2)                      ;SET MSGTYP TO FATAL ERROR
10617 036302 000000          HALT                                     ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10618 036304 012767 000016 141502 VDEC9: MOV     #14+2,14
10619                                     ;*****
10620                                     ;TEST 361          TEST THAT AN ILLA CAUSES AN OVERFLOW TRAP
10621                                     ;*****
10622 036312 005212          TS361: INC      (R2)                ;UPDATE TEST NUMBER
10623 036314 022712 000361          CMP     #361,(R2)                ;SEQUENCE ERROR?
10624 036320 001011          BNE     VDEC11                    ;BR TO ERROR HALT ON SEQ ERROR
10625 036322 012706 000400          MOV     #400,%6                    ;SET UP STACK TO OVERFLOW
10626 036326 012767 036344 141454          MOV     #VDEC11,10                ;SET UP ILLA VECTOR
10627 036334 012767 036354 141442          MOV     #VDEC12,4                 ;SET UP OVERFLOW VECTOR
10628 036342 004700          ILLA                                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10629 036344
10630 036344 012742 001101          VDEC11: MOV     #1101,-(R2)         ;MOVE TO MAILBOX # ***** 1101 *****
10631 036350 005242          INC     -(R2)                      ;SET MSGTYP TO FATAL ERROR
10632 036352 000000          HALT                                     ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
10633 036354 012767 000012 141426 VDEC12: MOV     #10+2,10
10634 036362 020627 000370          CMP     %6,#370                    ;STACK PUSHED FOUR WORDS?
10635 036366 001404          BEQ     TS362
10636                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <== -
10637                                     ;          CONDITIONAL BRANCH INST. AND <==--
10638                                     ;          REPLACE THE MOVE INSTRUCTION < = -
10639                                     ;          WHICH FOLLOWS W/ 754 < ==-
10640 036370 012742 001102          MOV     #1102,-(R2)                ;MOVE TO MAILBOX # ***** 1102 *****
10641 036374 005242          INC     -(R2)                      ;SET MSGTYP TO FATAL ERROR
10642 036376 000000          HALT                                     ;TRAP OVERFLOW DID NOT OCCUR
10643                                     ; OR SEQUENCE ERROR
10644                                     ;*****
10645                                     ;TEST 362          TEST THAT AN ILLB CAUSES AN OVERFLOW TRAP
10646                                     ;*****
10647 036400 005212          TS362: INC      (R2)                ;UPDATE TEST NUMBER
10648 036402 022712 000362          CMP     #362,(R2)                ;SEQUENCE ERROR?
10649 036406 001011          BNE     VDEC13                    ;BR TO ERROR HALT ON SEQ ERROR
10650 036410 012706 000400          MOV     #400,%6                    ;SET UP STACK TO OVERFLOW
10651 036414 012767 036432 141366          MOV     #VDEC13,10                ;SET UP ILLB VECTOR
10652 036422 012767 036442 141354          MOV     #VDEC14,4                 ;SET UP OVERFLOW VECTOR
10653 036430 000100          ILLB                                     ;THIS TRAP SHOULD CAUSE OVERFLOW
10654 036432
10655 036432 012742 001103          VDEC13: MOV     #1103,-(R2)         ;MOVE TO MAILBOX # ***** 1103 *****
10656 036436 005242          INC     -(R2)                      ;SET MSGTYP TO FATAL ERROR
10657 036440 000000          HALT                                     ;TRAP FLAG OVERFLOW DID NOT OCCUR,OR WRONG $TSTNM
```

```

10658 036442 012767 000012 141340 VDEC14: MOV #10+2,10
10659
10660
10661 ;*****
10662 ;TEST 363 TEST FOR FALSE OVERFLOW TRAP
10663 ;*****
10663 036450 005212 TS363: INC (R2) ;UPDATE TEST NUMBER
10664 036452 022712 000363 CMP #363,(R2) ;SEQUENCE ERROR?
10665 036456 001023 BNE FOVER ;BR TO ERROR HALT ON SEQ ERROR
10666
10667 036460 012767 036526 141316 MOV #FOVER,4 ;SET UP OVERFLOW POINTER
10668 036466 012706 001002 MOV #1002,%6
10669 036472 005746 TST -(6) ;SHOULD NOT OVERFLOW
10670 036474 012706 002002 MOV #2002,%6
10671 036500 005746 TST -(6) ;SHOULD NOT OVERFLOW
10672 036502 012706 004002 MOV #4002,%6
10673 036506 005746 TST -(6) ;SHOULD NOT OVERFLOW
10674 036510 012706 010002 MOV #10002,%6
10675 036514 005746 TST -(6)
10676 036516 012706 020000 MOV #20000,%6 ;SHOULD NOT OVERFLOW
10677 036522 005746 TST -(6)
10678 036524 000404 BR STP
10679 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10680 ; CONDITIONAL BRANCH INST. AND <====
10681 ; REPLACE THE MOVE INSTRUCTION <====
10682 ; WHICH FOLLOWS W/ 754 <====
10683 036526 FOVER:
10684 036526 012742 001104 MOV #1104,-(R2) ;MOVE TO MAILBOX # ***** 1104 *****
10685 036532 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10686 036534 000000 HALT ;IT OVRFLOWED,OR WRONG $STNM
10687
10688 036536 012767 000006 141240 STP: MOV #6,4
10689 036544 005067 141236 CLR 6
10690
10691 ;*****
10692 ;TEST 364 TEST THAT BIT 4 PSW WILL CAUSE A TRAP TO 14
10693 ;*****
10693 036550 005212 TS364: INC (R2) ;UPDATE TEST NUMBER
10694 036552 022712 000364 CMP #364,(R2) ;SEQUENCE ERROR?
10695 036556 001013 BNE TS365-10 ;BR TO ERROR HALT ON SEQ ERROR
10696 036560 012706 001000 MOV #BUFF,SP
10697 036564 012767 036616 141222 MOV #RETAT,RTRAP4 ;SET UP TO TRAP TO 14
10698 036572 012746 000020 MOV #20,-(SP) ;PUSH T BIT
10699 036576 012746 036604 MOV #.+6,-(SP) ;PUSH PC
10700 036602 000002 RTI ;SET T BIT
10701 036604 000240 NOP ;TRAP HERE
10702 036606 012742 001105 MOV #1105,-(R2) ;MOVE TO MAILBOX # ***** 1105 *****
10703 036612 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10704 036614 000000 HALT ;TRACE BIT DID NOT TRAP!,OR WRONG $TESTN
10705 036616 RETAT:
10706 ;*****
10707 ;TEST 365 TEST STACK POINTER DECREMENTS
10708 ;*****
10709 036616 005212 TS365: INC (R2) ;UPDATE TEST NUMBER
10710 036620 022712 000365 CMP #365,(R2) ;SEQUENCE ERROR?
10711 036624 001022 BNE TS366-10 ;BR TO ERROR HALT ON SEQ ERROR
10712 036626 012706 001000 MOV #BUFF,SP
10713 036632 012767 036664 141154 MOV #RETBT,RTRAP4
    
```

```
10714 036640 012746 000020      MOV    #20,-(SP)      ;PUSH T BIT
10715 036644 012746 036652      MOV    #.+6,-(SP)    ;PUSH PC
10716 036650 000002      RTI                    ;SET T BIT
10717 036652 000240      NOP                    ;TRAP HERE
10718 036654 012742 001106      MOV    #1106,-(R2)   ;MOVE TO MAILBOX # ***** 1106 *****
10719 036660 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
10720 036662 000000      HALT                   ;TRACE BIT DID NOT TRAP.
10721 036664 020627 000774      RETBT:  CMP    SP,#BUFF-4
10722 036670 001404      BEQ    TS366
10723
10724
10725
10726
10727 036672 012742 001107      MOV    #1107,-(R2)   ;MOVE TO MAILBOX # ***** 1107 *****
10728 036676 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
10729 036700 000000      HALT                   ;STACK POINTER WAS NOT PUSHED BY TRAP,OR WRONG $TESTN
10730
10731
10732
10733
10734 036702 005212      TS366:  INC    (R2)          ;UPDATE TEST NUMBER
10735 036704 022712 000366      CMP    #366,(R2)     ;SEQUENCE ERROR?
10736 036710 001016      BNE    TS367-10      ;BR TO ERROR HALT ON SEQ ERROR
10737 036712 012706 001000      MOV    #BUFF,SP
10738 036716 012767 036736 141070      MOV    #RETCT,RTRAP4
10739 036724 012746 000020      MOV    #20,-(SP)    ;PUSH T BIT
10740 036730 012746 036736      MOV    #.+6,-(SP)    ;PUSH PC
10741 036734 000002      RTI                    ;SET T BIT
10742
10743 036736 022767 036736 142030  RETCT:  CMP    #.BUFF-4
10744 036744 001404      BEQ    TS367
10745
10746
10747
10748
10749 036746 012742 001110      MOV    #1110,-(R2)   ;MOVE TO MAILBOX # ***** 1110 *****
10750 036752 005242      INC    -(R2)         ;SET MSGTYP TO FATAL ERROR
10751 036754 000000      HALT                   ;CORRECT PC WAS NOT SAVED ON STACK,OR WRONG $TESTN
10752
10753
10754
10755
10756
10757
10758 036756 005212      TS367:  INC    (R2)          ;UPDATE TEST NUMBER
10759 036760 022712 000367      CMP    #367,(R2)     ;SEQUENCE ERROR?
10760 036764 001015      BNE    TS370-10      ;BR TO ERROR HALT ON SEQ ERROR
10761
10762 036766 012706 001000      MOV    #BUFF,SP
10763 036772 005001      CLR    R1             ;CLEAR R1
10764 036774 012746 000020      MOV    #20,-(SP)
10765 037000 012746 037014      MOV    #RTT1,-(SP)
10766 037004 012767 037030 141002      MOV    #RTT2,14
10767 037012 000006      RTT
10768 037014 000240      RTT1:  NOP
10769 037016 001404      BEQ    TS370
```

```
10770
10771
10772
10773
10774 037020 012742 001111      MOV    #1111,-(R2)
10775 037024 005242              INC    -(R2)
10776 037026 000000              HALT
10777
10778
10779 037030
10780
10781
10782
10783 037030 005212
10784 037032 022712 000370      TS370: INC    (R2)
10785 037036 001030              CMP    #370,(R2)
10786 037040 012705 177777      BNE   TS371-10
10787 037044 012706 001000      MOV    #177777,%5
10788 037050 012746 000020      RTT5: MOV    #BUFF,SP
10789 037054 012746 037072      MOV    #20,-(SP)
10790 037060 012767 037110 140726  MOV    #RTT3,-(SP)
10791 037066 005001              MOV    #RTT4,14
10792 037070 000006              CLR    R1
10793 037072 005201      RTT3: RTT
10794 037074 005205              INC    R1
10795 037076 001762              INC    %5
10796 037100 012742 001112      BEQ   RTT5
10797 037104 005242      MOV    #1112,-(R2)
10798 037106 000000      INC    -(R2)
10799 037110 005301      HALT
10800 037112 001406      RTT4: DEC    R1
10801 037114 005205      BEQ   RTT6
10802 037116 001752      INC    %5
10803
10804
10805
10806
10807 037120 012742 001113      BEQ   RTT5
10808 037124 005242      MOV    #1113,-(R2)
10809 037126 000000      INC    -(R2)
10810 037130      HALT
10811
10812
10813
10814 037130 005212
10815 037132 022712 000371      TS371: INC    (R2)
10816 037136 001022      CMP    #371,(R2)
10817 037140 012706 001000      BNE   TS372-10
10818 037144 012746 000020      MOV    #BUFF,SP
10819 037150 012746 037166      MOV    #20,-(SP)
10820 037154 012767 037200 140632  MOV    #RTI1,-(SP)
10821 037162 005001      MOV    #RTI2,14
10822 037164 000002      CLR    R1
10823 037166 005201      RTI
10824 037170 012742 001114      RTI1: INC    R1
10825 037174 005242      MOV    #1114,-(R2)
10825              INC    -(R2)
```

```
: 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
```

```
RTT2:
:*****
:TEST 370 TEST THAT RTT ALLOWS ONE INST. BEFORE TRAP
:*****
```

```
TS370: INC (R2) ;UPDATE TEST NUMBER
CMP #370,(R2) ;SEQUENCE ERROR?
BNE TS371-10 ;BR TO ERROR HALT ON SEQ ERROR
```

```
RTT5: MOV #BUFF,SP
MOV #20,-(SP)
MOV #RTT3,-(SP)
MOV #RTT4,14
CLR R1 ;CLEAR R0
RTT ;SET T-BIT
```

```
RTT3: INC R1
INC %5
BEQ RTT5 ;DO THIS TEST NO MORE THAN 2 TIMES
MOV #1112,-(R2) ;MOVE TO MAILBOX # ***** 1112 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DID NOT TRAP
```

```
RTT4: DEC R1 ;SEE IF RTT ALLOWS 1 INST.
BEQ RTT6
INC %5 ;DO THIS TEST NO MORE THAN TWO TIMES
BEQ RTT5
```

```
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
:          CONDITIONAL BRANCH INST. AND <
:          REPLACE THE MOVE INSTRUCTION <-
:          WHICH FOLLOWS W/ 747 <
: MOVE TO MAILBOX # ***** 1113 *****
: SET MSGTYP TO FATAL ERROR
: RTT DID NOT ALLOW 1 INST.,OR WRONG $TESTN
```

```
RTT6:
:*****
:TEST 371 TEST THAT RTI DOES NOT ALLOW 1 INST.
:*****
```

```
TS371: INC (R2) ;UPDATE TEST NUMBER
CMP #371,(R2) ;SEQUENCE ERROR?
BNE TS372-10 ;BR TO ERROR HALT ON SEQ ERROR
```

```
RTI1: MOV #BUFF,SP
MOV #20,-(SP)
MOV #RTI1,-(SP)
MOV #RTI2,14
CLR R1 ;SET T-BIT
RTI ;RTI SHOULD NOT ALLOW THIS
```

```
MOV #1114,-(R2) ;MOVE TO MAILBOX # ***** 1114 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
```



```
10826 037176 000000          HALT          ;T- BIT DID NOT CAUSE TRAP
10827 037200 005701          RTI2: TST      R1
10828                                ;RTI SHOULD NOT ALLOW 1 INST. BEFORE TRAP
10829 037202 001404          BEQ      TS372
10830                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <-- =
10831                                ;          CONDITIONAL BRANCH INST. AND  <-- =
10832                                ;          REPLACE THE MOVE INSTRUCTION  <-- =
10833                                ;          WHICH FOLLOWS W/ 755          <-- =
10834 037204 012742 001115          MOV      #1115,-(R2) ;MOVE TO MAILBOX # ***** 1115 *****
10835 037210 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10836 037212 000000          HALT          ;RTI DID ALLOW 1 INST. BEFORE TRAP,OR WRONG $TESTN
10837                                ; OR SEQUENCE ERROR
10838
10839
10840
10841
```

```
*****
;TEST 372 TEST TRAP ON TRAP THAT TRACE BIT TRAPS ARE INHIBITED v TRAP INST
*****
```

```
10842 037214 005212 000372          TS372: INC      (R2)      ;UPDATE TEST NUMBER
10843 037216 022712 000372          CMP      #372,(R2)   ;SEQUENCE ERROR?
10844 037222 001026          BNE      BR70        ;BR TO ERROR HALT ON SEQ ERROR
10845
10846 037224 012706 001000          MOV      #BUFF,%6
10847 037230 012767 037270 140556          MOV      #TRACE,14 ;TRACE TRAP
10848 037236 005027 000016          CLR      #16
10849 037242 005027 000022          CLR      #22
10850 037246 012767 037310 140544          MOV      #TONT1,20 ;IOT TRAP
10851 037254 012746 000020          MOV      #20,-(SP)  ;PUSH T BIT
10852 037260 012746 037266          MOV      #.+6,-(SP) ;PUSH PC
10853 037264 000006          RTT
10854 037266 000004          IOT          ;TRAP, NEW CC HAVE TRACE RESET
10855 037270
10856 037270 012742 001116          TRACE: MOV      #1116,-(R2) ;MOVE TO MAILBOX # ***** 1116 *****
10857 037274 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10858 037276 000000          HALT          ;TRACE TRAP WAS NOT INHIBITED
10859 037300
10860 037300 012742 001117          BR70: MOV      #1117,-(R2) ;MOVE TO MAILBOX # ***** 1117 *****
10861 037304 005242          INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
10862 037306 000000          HALT          ;WRONG TSTNM,OR WRONG $TSTNM
10863 037310 012767 000016 140476          TONT1: MOV      #16,14
10864 037316 012767 000022 140474          MOV      #22,20
```

```
*****
;TEST 373 TEST THAT THE TRACE BIT IS SAVED IN THE STACK
*****
```

```
10868 037324 005212 000373          TS373: INC      (R2)      ;UPDATE TEST NUMBER
10869 037326 022712 000373          CMP      #373,(R2)   ;SEQUENCE ERROR?
10870 037332 001020          BNE      STP3        ;BR TO ERROR HALT ON SEQ ERROR
10871 037334 012706 001000          MOV      #BUFF,%6
10872 037340 012767 037364 140446          MOV      #TRC1,14 ;TRACE TRAP RETURN
10873 037346 005067 140444          CLR      16
10874 037352 012746 000020          MOV      #20,-(SP)  ;SET THE T BIT
10875 037356 012746 037364          MOV      #TRC1,-(SP)
10876 037362 000002          RTI
10877 037364 036727 141406 000020          TRC1: BIT      BUFF-2,#20 ;CHECK FOR T BIT ON STACK
10878 037372 001004          BNE      STP3D
10879                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <= -
10880                                ;          CONDITIONAL BRANCH INST. AND <
10881                                ;          REPLACE THE MOVE INSTRUCTION < --
```

```
10882 ; WHICH FOLLOWS W/ 757 <---
10883 037374 STP3: ;
10884 037374 012742 001120 MOV #1120,-(R2) ;MOVE TO MAILBOX # ***** 1120 *****
10885 037400 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10886 037402 000000 HALT ;T BIT NOT SAVED ON THE STACK,OR WRONG $TSTNM
10887 037404 012767 000016 140402 STP3D: MOV #16,14
10888
10889 ;*****
10890 ;THIS ROUTINE TESTS THAT NO LEGAL ADDRESS TRAPS AND THAT AN ILLEGAL
10891 ;ADDRESS TRAPS TO LOCATION 4. THIS WILL RUN ON 30K SYSTEM. BUT IF
10892 ;SWITCH REGISTER BIT 1=0, THEN THE MEMORY FROM 28K-30K IS NOT LOOKED
10893 ;AT, SINCE IT MAY HAVE I/O DEVICES. IF SWR BIT 1=1, THEN THAT AREA IS
10894 ;CHECKED. (IT SHOULD EITHER ALL TRAP OR ALL NOT TRAP). LOC 160000
10895 ;IS NO LONGER GUARANTEED TO TRAP, SINCE IT MAY CONTAIN MEMORY. LOCATION
10896 ;177700 (THE UNIBUS ADDRESS FOR R0 ON OLDER SYSTEMS) IS USED FOR FORCING
10897 ;A TIMEOUT IN THE EVENT THAT THERE WAS NO TIMEOUT FROM 0K-28K OR 30K.
10898 ;THIS ROUTINE TESTS MEMORY UNTIL IT DOES A NXM STOP
10899 ;*****
10900 ;TEST 374 TEST NON-EXISTENT ADDRESS TRAPS
10901 ;*****
10902 037412 005212 TS374: INC (R2) ;UPDATE TEST NUMBER
10903 037414 022712 000374 CMP #374,(R2) ;SEQUENCE ERROR?
10904 037420 001150 BNE TS375-10 ;BR TO ERROR HALT ON SEQ ERROR
10905 037422 042737 010000 037472 BIC #10000,@#HICORE ;SET HIGHT CORE LIMIT TO 160000
10906 037430 032737 000002 000322 BIT #2,@$$SWREG ;CHECK IF BIT 1 IS SET
10907 037436 001403 BEQ 1$ ;BRANCH IF IT IS, LEAVE LIMIT=160000
10908 037440 052737 010000 037472 BIS #10000,@#HICORE ;SET UPPER CORE LIMIT TO 30K (170000)
10909 037446 005000 1$: CLR R0 ;
10910 037450 005067 140332 CLR 6 ;
10911 037454 012767 037562 140322 MOV #ATRAP,4 ;SET UP ADDRESS TRAP ENTRANCE
10912 037462 012706 001000 MOV #BUFF,SP ;SET STACK POINTER
10913 037466 105720 NOR: TSTB (0)+ ;IF OUTSIDE OF CORE, TRAP TO 4
10914 037470 020027 CMP R0,(PC)+ ;IS POINTER INSIDE 28K (30K) CORE
10915 037472 160000 HICORE: .WORD 160000 ;MAY BE CHANGED TO 170000 IF 30K
10916 037474 103774 BLO NOR ;TEST THE REST OF CORE
10917 037476 012737 037520 000004 MOV #ROTRAP,@#4 ;SET UP NEW VECTOR POINTER
10918 037504 105737 177700 TSTB @#177700 ;SHOULD CAUSE A TRAP
10919 037510 TRPADR:
10920 037510 012742 001121 MOV #1121,-(R2) ;MOVE TO MAILBOX # ***** 1121 *****
10921 037514 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10922 037516 000000 HALT ;SHOULD HAVE TRAPED
10923 ;TRAP TO HERE IF FORCING TRAP BY TESTING 177700
10924 037520 106767 140252 ROTRAP: MFPS STATUS ;
10925 037524 005767 140246 TST STATUS ;TEST PSW
10926 037530 001404 BEQ 1$ ;
10927 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10928 ; CONDITIONAL BRANCH INST. AND <====
10929 ; REPLACE THE MOVE INSTRUCTION <====
10930 ; WHICH FOLLOWS W/ 733 <====
10931 037532 012742 001122 MOV #1122,-(R2) ;MOVE TO MAILBOX # ***** 1122 *****
10932 037536 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
10933 037540 000000 HALT ;NEW PSW SHOULD HAVE BEEN ZERO
10934 037542 026727 141226 037510 1$: CMP BUFF-4,#TRPADR ;TEST OLD PC AT STACK
10935 037550 001453 BEQ TRAPB ;
10936 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
10937 ; CONDITIONAL BRANCH INST. AND <====
```

```

10938
10939
10940 037552 012742 001123      MOV    #1123,-(R2)      ;
10941 037556 005242      INC    -(R2)           ; REPLACE THE MOVE INSTRUCTION <---
10942 037560 000000      HALT                    ; WHICH FOLLOWS W/ 723 <---
10943      ;RETURN HERE ON AN ADDRESS TRAP FROM MEMORY BELOW 28K (OR 30K)
10944 037562 005300      ATRAP: DEC    RO        ;
10945 037564 010067 000032      MOV    RO,CORH        ; MOVE THE FIRST NXM LOCATION IN CORH
10946      ;THIS ROUTINE DOES NXM TRAPS UNTIL IT FINDS AN EXISTENT MEMORY LOCATION
10947 037570 013700 037472      MOV    @#HICORE,RO    ; SET UP THE HIGHEST MEM LOCATION
10948 037574 005300      DEC    RO              ; MAKE 1 LESS THAN THE HIGHEST CORE BOUNDARY
10949 037576 000402      BR     NOSUB          ; DON'T SUBTRACT 1K FIRST TIME
10950 037600 162700 001000      CTRAP: SUB   #1000,RO ; SUBTRACT 1K OCTAL BYTE FROM ADDRESS
10951      ;TO SPEED UP TESTING
10952 037604 012767 037636 140172 NOSUB: MOV    #BTRAP,4 ; SET UP THE VECTOR
10953 037612 012706 001000      MOV    #BUFF,SP
10954 037616 005710      TST    (RO)           ; DOES THIS MEMORY EXIST?
10955      ;IF NXM, TRAP TO BTRAP
10956 037620 020027      DTRAP1: CMP   RO,(PC)+ ; IF EXISTS, IS THIS THE SAME TRAP THAT CAUSED
10957      ;TRAP TO ATRAP
10958 037622 000000      CORH:  .WORD  0
10959 037624 101425      BLOS   TRAPB
10960
10961      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
10962      ; CONDITIONAL BRANCH INST. AND < =
10963      ; REPLACE THE MOVE INSTRUCTION < - =
10964      ; WHICH FOLLOWS W/ 675 <---
10964 037626 012742 001124      MOV    #1124,-(R2)    ; MOVE TO MAILBOX # ***** 1124 *****
10965 037632 005242      INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
10966 037634 000000      HALT                    ; CONTENTS OF RO SHOULD BE LESS THAN OR EQUAL TO CORH
10967      ; IF THIS COMPARISON FAILS IT MEANS
10968      ; THAT SOME LEGAL ADDRESS TRAPPED, OR
10969      ; THAT AN ILLEGAL ADDRESS DID NOT TRAP
10970 037636 106767 140134      BTRAP: MFPS   STATUS
10971 037642 005767 140130      TST   STATUS
10972 037646 001404      BEQ   1$
10973
10974      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <
10975      ; CONDITIONAL BRANCH INST. AND <-
10976      ; REPLACE THE MOVE INSTRUCTION <-
10977      ; WHICH FOLLOWS W/ 664 <- -
10977 037650 012742 001125      MOV    #1125,-(R2)    ; MOVE TO MAILBOX # ***** 1125 *****
10978 037654 005242      INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
10979 037656 000000      HALT                    ; NEW PSW SHOULD HAVE BEEN ZERO
10980 037660 026727 141110 037620 1$: CMP   BUFF-4,#DTRAP1 ; CHECK IF TRAP PC IS OK
10981 037666 001744      BEQ   CTRAP
10982
10983      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <- -
10984      ; CONDITIONAL BRANCH INST. AND <- -=
10985      ; REPLACE THE MOVE INSTRUCTION <- -
10986      ; WHICH FOLLOWS W/ 654 < =--
10986 037670      AUTO1:
10987 037670 012742 001126      MOV    #1126,-(R2)    ; MOVE TO MAILBOX # ***** 1126 *****
10988 037674 005242      INC    -(R2)           ; SET MSGTYP TO FATAL ERROR
10989 037676 000000      HALT                    ; OLD PC WAS NOT SAVED OR WRONG $TESTN
10990 037700 012767 000006 140076 TRAPB: MOV    #6,4
10991 037706 005067 140074      CLR   6                ; RESET TRAP CATCHER
10992
10993
    
```

```

10994 ;THIS ROUTINE WILL FIGURE OUT IF YOU HAVE A DL11W
10995 037712 005067 000020 CLR PROFTE
10996 037716 012706 001000 MOV #BUFF,%6 ;SET UP THE STACK POINTER
10997 037722 012767 037740 140054 MOV #DL11W,4 ;SET UP THE TRAP VECTOR
10998 037730 005767 137630 TST TPS ;TEST THE PUNCH STATUS REGISTER
10999 037734 000403 BR DL11W1
11000 037736 000000 PROFTE: 000000
11001 037740 005267 177772 DL11W: INC PROFTE ;INCR IF NO DL11W
11002 037744 012767 000006 140032 DL11W1: MOV #6,4
11003
11004 037752 SKP104:
11005 ;*****
11006 ;TEST 375 TEST THAT A TTY INTERRUPT CAUSES AN OVERFLOW TRAP
11007 ;*****
11008 037752 005212 TS375: INC (R2) ;UPDATE TEST NUMBER
11009 037754 022712 000375 CMP #375,(R2) ;SEQUENCE ERROR?
11010 037760 001037 BNE TDEC8 ;BR TO ERROR HALT ON SEQ ERROR
11011 037762 005767 177750 TST PROFTE
11012 037766 001047 BNE R7TRX
11013 037770 122767 000001 140322 CMPB #APTENV,$ENV ;RUNING IN APT MODE?
11014 037776 001003 BNE 2$ ;IF NOT, DO THIS TEST
11015 040000 005767 140302 TST $PASS ;IS THIS THE FIRST PASS?
11016 040004 001040 BNE R7TRX ;IF NOT FIRST PASS, SKIP TEST
11017 040006 2$:
11018 040006 000005 RESET
11019 040010 012767 000340 137760 MOV #340,STATUS ;LOCK OUT INTERRUPT
11020 040016 012706 000400 MOV #400,%6 ;SET UP STACK TO OVERFLOW
11021 040022 012767 040070 137754 MOV #TDEC77,4 ;SET UP OVERFLOW TRAP
11022 040030 012767 040060 140026 MOV #TDEC8,64 ;SET UP INTERRUPT VECTOR
11023 040036 012767 000100 137520 MOV #100,TTCSR ;SET INTERRUPT ENABLE
11024 040044 005067 137726 CLR STATUS ;ALLOW INTEKRUPT TO OCCUR
11025 040050 012742 001127 MOV #1127,-(R2) ;MOVE TO MAILBOX # ***** 1127 *****
11026 040054 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11027 040056 000000 HALT ;NO INTERRUPT OCCURRED
11028 040060 TDEC8:
11029 040060 012742 001130 MOV #1130,-(R2) ;MOVE TO MAILBOX # ***** 1130 *****
11030 040064 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11031 040066 000000 HALT ;OVERFLOW TRAP DID NOT OCCUR OR WRONG $STNM
11032 040070 005067 137470 TDEC77: CLR TTCSR ;CLEAR INTERRUPT ENABLE
11033 040074 012767 000006 137702 MOV #6,4
11034 040102 005067 137700 CLR 6
11035 040106 R7TRX:
11036 ;*****
11037 ;TEST 376 TEST THAT A PENDING INTERRUPT OCCURS BEFORE TRAP
11038 ;*****
11039 040106 005212 TS376: INC (R2) ;UPDATE TEST NUMBER
11040 040110 022712 000376 CMP #376,(R2) ;SEQUENCE ERROR?
11041 040114 001045 BNE BR71 ;BR TO ERROR HALT ON SEQ ERROR
11042 040116 005767 177614 TST PROFTE
11043 040122 001053 BNE NODL
11044 040124 122767 000001 140166 CMPB #APTENV,$ENV ;RUNING IN APT MODE?
11045 040132 001003 BNE 2$ ;IF NOT, DO THIS TEST
11046 040134 005767 140146 TST $PASS ;IS THIS THE FIRST PASS?
11047 040140 001044 BNE NODL ;IF NOT FIRST PASS, SKIP TEST
11048 040142 2$:
11049 040142 012706 001000 MOV #BUFF,%6
    
```

```

11050 040146 012767 000340 137622      MOV    #340,STATUS      ;SET TO A HIGH PRIORITY LEVEL
11051 040154 012767 040220 137702      MOV    #TR0,64
11052 040162 012767 000100 137374      MOV    #100,TTCSR      ;INTERRUPT FOR TTY PUNCH/PRINTER
11053 040170 012767 040230 137636      MOV    #BR71,34        ;TRAP VECTOR
11054 040176 012767 040240 137660      MOV    #TR2,64        ;TTY VECTOR
11055 040204 012767 000340 137624      MOV    #340,36        ;IF TRAP TRAPS, MOVE 340 TO PRIORITY
11056 040212 005067 137560      CLR    STATUS          ;SHOULD INTERRUPT AT END OF CLR INST
11057 040216 104400      TRAP                    ;TTY INTERRUPT SHOULD OVERRIDE TRAP
11058 040220      TR0:
11059 040220 012742 001131      MOV    #1131,-(R2)     ;MOVE TO MAILBOX # ***** 1131 *****
11060 040224 005242      INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
11061 040226 000000      HALT                    ;TTY SHOULDN'T HAVE INTERRUPTED
11062 040230      BR71:
11063 040230 012742 001132      MOV    #1132,-(R2)     ;MOVE TO MAILBOX # ***** 1132 *****
11064 040234 005242      INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
11065 040236 000000      HALT                    ;TRAP OCCURRED FIRST,OR WRONG $STNM
11066 040240 005067 137572      TR2:  CLR    36
11067 040244 042767 000100 137312      BIC    #100,TTCSR
11068 040252      NODL:
11069      ;*****
11070      ;TEST 377      TEST THAT A PENDING INTERRUPT, INTERRUPTS BETWEEN TRAPS
11071      ;*****
11072 040252 005212      TS377: INC    (R2)      ;UPDATE TEST NUMBER
11073 040254 022712 000377      CMP    #377,(R2)      ;SEQUENCE ERROR?
11074 040260 001043      BNE    TR5             ;BR TO ERROR HALT ON SEQ ERROR
11075 040262 005767 177450      TST    PROFTE
11076 040266 001063      BNE    NODL1
11077 040270 122767 000001 140022      CMPB   #APTENV,$ENV   ;RUNING IN APT MODE?
11078 040276 001003      BNE    2$             ;IF NOT, DO THIS TEST
11079 040300 005767 140002      TST    $PASS          ;IS THIS THE FIRST PASS?
11080 040304 001054      BNE    NODL1          ;IF NOT FIRST PASS, SKIP TEST
11081 040306      2$:
11082 040306 012706 001000      MOV    #BUFF,%6
11083 040312 012767 000340 137456      MOV    #340,STATUS
11084 040320 012767 000100 137236      MOV    #100,TTCSR
11085 040326 012767 040366 137500      MOV    #TR3,34        ;TRAP
11086 040334 012767 040400 137522      MOV    #TR4,64        ;TTY OUTPUT
11087 040342 012767 000340 137516      MOV    #340,66        ;TTY OUTPUT PRIORITY
11088 040350 012767 040370 137442      MOV    #TR5,20        ;IOT
11089 040356 012767 000340 137436      MOV    #340,22        ;IOT PRIORITY
11090 040364 104400      TRAP                    ;THE ACT OF TRAPPING LOWER PRIORITY
11091 040366 000004      TR3:  IOT              ;INTERRUPT SHOULD OCCUR IN PLACE OF IOT TRAP
11092 040370      TR5:
11093 040370 012742 001133      MOV    #1133,-(R2)     ;MOVE TO MAILBOX # ***** 1133 *****
11094 040374 005242      INC    -(R2)           ;SET MSGTYP TO FATAL ERROR
11095 040376 000000      HALT                    ;NO INTERRUPT BETWEEN TRAPS,OR WRONG $STNM
11096 040400 005067 137416      TR4:  CLR    22
11097 040404 005067 137456      CLR    66
11098 040410 012767 000036 137416      MOV    #36,34
11099 040416 012767 000066 137440      MOV    #66,64
11100 040424 012767 000022 137366      MOV    #22,20
11101 040432 005067 137126      CLR    TTCSR
11102 040436      NODL1:
11103      ;*****
11104      ;TEST 400      TEST THAT 'RESET' GOES TO OUTSIDE WORLD
11105

```

```
11106
11107 040436 005212
11108 040440 022712 000400
11109 040444 001021
11110 040446 005767 177264
11111 040452 001022
11112 040454 016700 137102
11113 040460 012767 000100 137072
11114 040466 000005
11115 040470 032767 000100 137062
11116 040476 001410
11117
11118
11119
11120
11121 040500 012742 001134
11122 040504 005242
11123 040506 000000
11124
11125 040510 012742 001135
11126 040514 005242
11127 040516 000000
11128 040520
11129
11130
11131
11132 040520 005212
11133 040522 022712 000401
11134 040526 001014
11135 040530 012706 001000
11136 040534 012767 040570 137252
11137 040542 012746 000020
11138 040546 012746 040554
11139 040552 000006
11140 040554 000005
11141 040556 000005
11142 040560
11143 040560 012742 001136
11144 040564 005242
11145 040566 000000
11146 040570 005067 137202
11147 040574 005067 137216
11148 040600 012767 000016 137206
11149 040606
11150
11151
11152
11153 040606 005212
11154 040610 022712 000402
11155 040614 001057
11156 040616 005767 177114
11157 040622 001062
11158 040624 122767 000001 137466
11159 040632 001003
11160 040634 005767 137446
11161 040640 001053

:*****
TS400: INC (R2) ;UPDATE TEST NUMBER
      CMP #400,(R2) ;SEQUENCE ERROR?
      BNE TS401-10 ;BR TO ERROR HALT ON SEQ ERROR
      TST PROFTE
      BNE NODL2
      MOV TKB,R0 ;MAKE SURE RECEIVER DONE IS SET
      MOV #100,TRCSR ;SET INTERRUPT ENABLE
      RESET ;SHOULD CLEAR INTERRUPT ENABLE
      BIT #100,TRCSR ;TEST FOR CLEAR
      BEQ TS401
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 762 <====
      MOV #1134,-(R2) ;MOVE TO MAILBOX # ***** 1134 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;RESET FAILED TO CLEAR TRCSR
      ; OR SEQUENCE ERROR
      MOV #1135,-(R2) ;MOVE TO MAILBOX # ***** 1135 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ; WRONG $STSTM

NODL2:
:*****
:TEST 401 TEST THAT RESET HAS NO EFFECT ON THE TRACE TRAP
:*****
TS401: INC (R2) ;UPDATE TEST NUMBER
      CMP #401,(R2) ;SEQUENCE ERROR?
      BNE RESET3 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #BUFF,%6 ;SET STACK
      MOV #RESET2,14 ;SET UP TRACE VECTOR
      MOV #20,-(R6) ;SET THE T-BIT ON STACK
      MOV #1$,-(R6) ;MOVE NEW PC ON STACK
      RTT
      RESET ;SHOULD HAVE NO EFFECT
      RESET ;NO EFFECT
      RE ET3:
      MOV #1136,-(R2) ;MOVE TO MAILBOX # ***** 1136 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;TRACE TRAP FAILED,OR WRONG $STSTM
      RESET2: CLR STATUS ;CLEAR TRACK
      CLR 16 ;TRACE STATUS
      MOV #16,14
      SKTST2:
:*****
:TEST 402 TEST THAT WHEN TTY INTERRUPTS IT POPS NEW STATUS
:*****
TS402: INC (R2) ;UPDATE TEST NUMBER
      CMP #402,(R2) ;SEQUENCE ERROR?
      BNE TTY11 ;BR TO ERROR HALT ON SEQ ERROR
      TST PROFIE
      BNE NODL3
      CMPB #APTENV,$ENV ;RUNING IN APT MODE?
      BNE 2$ ;IF NOT, DO THIS TEST
      TST $PASS ;IS THIS THE FIRST PASS?
      BNE NODL3 ;IF NOT FIRST PASS, SKIP TEST
```

```
11162 040642          2$:
11163 040642 000005          RESET
11164 040644 012706 001000          MOV #BUFF,%6          ;SET UP STACK
11165 040650 012767 040674 137206          MOV #TTY3,64          ;INTERRUPT VECTOR
11166 040656 005067 137114          CLR STATUS          ;DROP PROCESSOR PRIORITY
11167 040662 012767 000357 137176          MOV #357,66          ;HIGH PRIORITY ON INTERRUPT
11168 040670 105167 136670          COMB TTCSR          ;SHOULD SET INTERRUPT ENABLE & INTERRUPT
11169 040674 026727 137076 000357 TTY3:  CMP STATUS,#357
11170 040702 001404          BEQ 1$
11171
11172          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11173          ;          CONDITIONAL BRANCH INST. AND <====
11174          ;          REPLACE THE MOVE INSTRUCTION <====
11175          ;          WHICH FOLLOWS W/ 744 <====
11175 040704 012742 001137          MOV #1137,-(R2)      ;MOVE TO MAILBOX # ***** 1137 *****
11176 040710 005242          INC -(R2)          ;SET MSGTYP TO FATAL ERROR
11177 040712 000000          HALT          ;INTERRUPT DID NOT POP CORRECT STATUS
11178 040714 000005          1$: RESET          ;CLR INTERRUPT ENABLE
11179 040716 012706 001000          MOV #BUFF,%6          ;STACK SET UP
11180 040722 012767 040746 137134          MOV #TTY4,64          ;INTERRUPT VECTOR
11181 040730 005067 137132          CLR 66          ;CLR NEW STATUS
11182 040734 012767 000157 137034          MOV #157,STATUS      ;PROCESSOR STATUS
11183 040742 105167 136616          COMB TTCSR          ;SET INTERRUPT ENABLE
11184 040746 005767 137024 TTY4:  TST STATUS
11185 040752 001404          BEQ TTT37
11186
11187          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11188          ;          CONDITIONAL BRANCH INST. AND <====
11189          ;          REPLACE THE MOVE INSTRUCTION <====
11190          ;          WHICH FOLLOWS W/ 720 <====
11190 040754          TTY11:
11191 040754 012742 001140          MOV #1140,-(R2)      ;MOVE TO MAILBOX # ***** 1140 *****
11192 040760 005242          INC -(R2)          ;SET MSGTYP TO FATAL ERROR
11193 040762 000000          HALT          ;INCORRECT STATUS,OR WRONG $STNM
11194 040764 105167 136574 TTT37: COMB TTCSR
11195 040770          NODL3:
11196
11197          ;*****
11198          ;TEST 403 TEST THE 'WAIT' INSTRUCTION
11199          ;*****
11200 040770 005212          TS403: INC (R2)          ;UPDATE TEST NUMBER
11201 040772 022712 000403          CMP #403,(R2)        ;SEQUENCE ERROR?
11202 040776 001062          BNE STP4          ;BR TO ERROR HALT ON SEQ ERROR
11203 041000 122767 000001 137312          CMPB #APTENV,$ENV      ;RUNING IN APT MODE?
11204 041006 001003          BNE 1$          ;IF NOT, DO THIS TEST
11205 041010 005767 137272          TST $PASS          ;IS THIS THE FIRST PASS?
11206 041014 001057          BNE STP4E          ;IF NOT FIRST PASS, SKIP TEST
11207 041016          1$:
11208 041016 042767 000100 136540          BIC #100,TPS          ;CLEAR INTERRUPT ENABLE
11209 041024 012706 001000          MOV #BUFF,SP          ;SET UP THE STACK
11210 041030 012767 041116 137026          MOV #WATE,64          ;SET UP THE INTERRUPT VECTOR
11211 041036 005067 137024          CLR 66
11212 041042 105767 136516          WATE1: TSTB TPS          ;WAIT FOR READY
11213 041046 100375          BPL WATE1          ;TO BE UP
11214 041050 012767 000015 136510          MOV #15,TPB          ;DO A CARRIAGE RETURN
11215 041056 105767 136502          WATE2: TSTB TPS          ;WAIT FOR READY TO COME UP
11216 041062 100375          BPL WATE2
11217 041064 012767 000015 136474          MOV #15,TPB          ;DO ANOTHER CARRIAGE RETURN
```

```

11218 041072 052767 000100 136464      BIS      #100,TPS      ;SET THE INTERRUPT ENABLE
11219 041100 005067 136672      CLR      STATUS      ;CLEAR THE PSW
11220 041104 000001      WATE3:  WAIT        ;WAIT FOR THE INTERRUPT
11221 041106 012742 001141      MOV      #1141,-(R2)  ;MOVE TO MAILBOX # ***** 1141 *****
11222 041112 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
11223 041114 000000      HALT                    ;WAIT INSTRUCTION DID NOT LOOP
11224 041116 005767 136654      WATE:   TST      STATUS ;IS THE PSW CORRECT?
11225 041122 001404      BEQ      1$
11226                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11227                                     ;          CONDITIONAL BRANCH INST. AND <====
11228                                     ;          REPLACE THE MOVE INSTRUCTION <====
11229                                     ;          WHICH FOLLOWS W/ 725 <====
11230 041124 012742 001142      MOV      #1142,-(R2)  ;MOVE TO MAILBOX # ***** 1142 *****
11231 041130 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
11232 041132 000000      HALT                    ;NEW PSW SHOULD HAVE BEEN ZERO
11233 041134 026727 137634 041106 1$:  CMP      BUFF-4,#WATE3+2 ;IS THE OLD PC SAVED
11234 041142 001404      BEQ      STP4E
11235                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11236                                     ;          CONDITIONAL BRANCH INST. AND <====
11237                                     ;          REPLACE THE MOVE INSTRUCTION <====
11238                                     ;          WHICH FOLLOWS W/ 715 <====
11239 041144      STP4:  MOV      #1143,-(R2) ;MOVE TO MAILBOX # ***** 1143 *****
11240 041144 012742 001143      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
11241 041150 005242      HALT                    ;OLD PC WAS NOT SAVED OR WRONG $TESTN
11242 041152 000000
11243 041154      STP4E:
11244      ;*****
11245      ;TEST 404 TEST THAT USING REGISTER ADDR (177700) CAUSES TIME OUT.
11246      ;*****
11247 041154 005212      TS404: INC      (R2)      ;UPDATE TEST NUMBER
11248 041156 022712 000404      CMP      #404,(R2)    ;SEQUENCE ERROR?
11249 041162 001017      BNE      TS405-10     ;BR TO ERROR HALT ON SEQ ERROR
11250
11251      ;REGISTER ADDRESS (177700-177717) CAUSE TIME OUT WHEN USED
11252      ;AS PROGRAM ADDRESS BY THE CPU.
11253
11254 041164 012706 001000      MOV      #BUFF,SP     ;SET STACK POINTER
11255 041170 012737 041212 000004      MOV      #RETR1,#RTRAPS ;SET TRAP RETURN ADDR
11256 041176 005737 177700      PCN1:  TST      @#177700 ;BAD ADDR REFERENCE, TRAP TO 4
11257 041202 012742 001144      MOV      #1144,-(R2)  ;MOVE TO MAILBOX # ***** 1144 *****
11258 041206 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
11259 041210 000000      HALT                    ;REFERENCING 177700 DID NOT CAUSE TIME OUT
11260 041212 022767 041202 137554 RETR1:  CMP      #PCN1+4,BUFF-4 ;PROPER PC STORED ON STACK?
11261 041220 001404      BEQ      TS405
11262                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
11263                                     ;          CONDITIONAL BRANCH INST. AND <===
11264                                     ;          REPLACE THE MOVE INSTRUCTION <===
11265                                     ;          WHICH FOLLOWS W/ 760 <===
11266 041222 012742 001145      MOV      #1145,-(R2)  ;MOVE TO MAILBOX # ***** 1145 *****
11267 041226 005242      INC      -(R2)        ;SET MSGTYP TO FATAL
11268 041230 000000      HALT                    ;OLD PC WAS NOT SAVED IN STACK
11269                                     ; OR SEQUENCE ERROR
11270
11271      ;*****
11272      ;ODD ADDRESS USED BY A 'WORD' INSTRUCTION SHOULD NOT
11273      ;CAUSE A TRAP, BUT THE LOW ORDER ADDRESS BIT WOULD BE IGNORED.
    
```



```
11274 :*****
11275 :TEST 405 TEST ODD ADDRESS TRAP IS NOT IMPLEMENTED.
11276 :*****
11277 041232 005212 TS405: INC (R2) ;UPDATE TEST NUMBER
11278 041234 022712 000405 CMP #405,(R2) ;SEQUENCE ERROR?
11279 041240 001013 BNE TS406-10 ;BR TO ERROR HALT ON SEQ ERROR
11280 :
11281 041242 012737 041270 000004 : MOV #RETR2,@#RTRAP5 ;SET TRAP RETURN ADDR
11282 041250 005037 000000 CLR @#0 ;PUT ALL 0 IN LOC 0
11283 041254 005337 000001 DEC @#1 ;DECREMENT ODD ADDRESS, SHOULD NOT TRAP
11284 041260 022737 177777 000000 CMP #-1,@#0 ;WORD LOC 0 HAS ALL ONES?
11285 041266 001404 BEQ TS406
11286 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--
11287 : CONDITIONAL BRANCH INST. AND <= --
11288 : REPLACE THE MOVE INSTRUCTION < --
11289 : WHICH FOLLOWS W/ 764 <== --
11290 041270 RETR2:
11291 041270 012742 001146 MOV #1146,-(R2) ;MOVE TO MAILBOX # ***** 1146 *****
11292 041274 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11293 041276 000000 HALT ;LOC 0 DID NOT STORE -1,OR ODD ADDR REFERENCE CAUSE TRAP
11294 : OR SEQUENCE ERROR
11295 :
11296 :*****
11297 :
11298 :USING ADDRESS 177700 IN MODE 2, CAUSES BUS ERROR, BUT
11299 :THE REGISTER IN USE WILL BE INCREMENTED.
11300 :
11301 :*****
11302 :TEST 406 TEST THAT IN MODE 2, BAD ADDRESS REFERENCE CAUSES BUS ERROR.
11303 :*****
11304 041300 005212 TS406: INC (R2) ;UPDATE TEST NUMBER
11305 041302 022712 000406 CMP #406,(R2) ;SEQUENCE ERROR?
11306 041306 001016 BNE TS407-10 ;BR TO ERROR HALT ON SEQ ERROR
11307 041310 012737 041336 000004 MOV #RETR3,@#RTRAP5 ;SET TRAP RETURN ADDR
11308 041316 012700 177700 MOV #177700,R0 ;STORES BAD MEMORY REFERENCE
11309 041322 012720 001234 MOV #1234,(R0)+ ;BAD ADDR REFERENCE, TRAP TO LOC 4
11310 041326 012742 001147 MOV #1147,-(R2) ;MOVE TO MAILBOX # ***** 1147 *****
11311 041332 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11312 041334 000000 HALT ;ADDRESSING 177700 DID NOT CAUSE TRAP
11313 041336 022700 177702 RETR3: CMP #177702,R0 ;WAS R0 INCREMENTED?
11314 041342 001404 BEQ TS407
11315 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <--
11316 : CONDITIONAL BRANCH INST. AND <= --
11317 : REPLACE THE MOVE INSTRUCTION < --
11318 : WHICH FOLLOWS W/ 761 <= --
11319 041344 012742 001150 MOV #1150,-(R2) ;MOVE TO MAILBOX # ***** 1150 *****
11320 041350 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11321 041352 000000 HALT ;R0 WAS NOT INCREMENTED
11322 : OR SEQUENCE ERROR
11323 :*****
11324 :
11325 :AFTER THE FIRST BUS ERROR WAS ENCOUNTERED, AN ATTEMPT WAS MADE
11326 :TO PUSH PC AND PS INTO THE STACK. HOWEVER, IF THE STACK POINTER
11327 :WAS BAD, A DOUBLE BUS ERROR OCCURED. THE STACK POINTER WOULD
11328 :THEN BE SET TO LOCATION 4, OLD PC AND PS WERE PUSHED INTO
11329 :LOCATIONS 0 AND 2. THE PROCESSOR WOULD TRAP TO 4 AND CONTINUE
```

```
11330 ;EXECUTION.
11331 ;
11332 ;*****
11333 ;TEST 407 TEST FOR DOUBLE BUS ERROR.
11334 ;*****
11335 041354 005212 TS407: INC (R2) ;UPDATE TEST NUMBER
11336 041356 022712 000407 CMP #407,(R2) ;SEQUENCE ERROR?
11337 041362 001062 BNE TS410-10 ;BR TO ERROR HALT ON SEQ ERROR
11338 041364 012737 041442 000004 MOV #DBE1,@#RTRAP5 ;SET TRAP RETURN ADDR
11339 041372 012737 000340 000006 MOV #340,@#6 ;SET UP PS
11340 041400 012737 041432 000010 MOV #DBE2,@#RTRAP ;SET TRAP RETURN ADDR
11341 041406 012737 000340 000012 MOV #340,@#12 ;SET UP PS
11342 041414 012706 177700 MOV #177700,SP ;SET ILLEGAL SP
11343 041420 000077 DBE: TRAPA ;ILLEGAL INSTRUCTION
11344 041422 012742 001151 MOV #1151,-(R2) ;MOVE TO MAILBOX # ***** 1151 *****
11345 041426 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11346 041430 000000 HALT ;DOUBLE BUS ERROR DID NOT CAUSE TRAP
11347 041432 DBE2:
11348 041432 012742 001152 MOV #1152,-(R2) ;MOVE TO MAILBOX # ***** 1152 *****
11349 041436 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11350 041440 000000 HALT ;TRAP TO WRONG LOCATION
11351 041442 022737 041422 000000 DBE1: CMP #DBE+2,@#0 ;OLD PC GOT SAVED?
11352 041450 001404 BEQ DBE3
11353 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11354 ; CONDITIONAL BRANCH INST. AND <====
11355 ; REPLACE THE MOVE INSTRUCTION <====
11356 ; WHICH FOLLOWS W/ 744 <====
11357 041452 012742 001153 MOV #1153,-(R2) ;MOVE TO MAILBOX # ***** 1153 *****
11358 041456 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11359 041460 000000 HALT ;OLD PC DID NOT GET SAVEDD
11360 041462 022737 000340 000002 DBE3: CMP #340,@#2 ;CORRECT PS SAVFD?
11361 041470 001404 BEQ DBE4
11362 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11363 ; CONDITIONAL BRANCH INST. AND <====
11364 ; REPLACE THE MOVE INSTRUCTION <====
11365 ; WHICH FOLLOWS W/ 734 <====
11366 041472 012742 001154 MOV #1154,-(R2) ;MOVE TO MAILBOX # ***** 1154 *****
11367 041476 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11368 041500 000000 HALT ;CORRECT PS DID NOT GET SAVE
11369 041502 022706 000000 DBE4: CMP #0,SP ;SP POINTS TO LOC 0?
11370 041506 001404 BEQ DBE5
11371 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11372 ; CONDITIONAL BRANCH INST. AND <====
11373 ; REPLACE THE MOVE INSTRUCTION <====
11374 ; WHICH FOLLOWS W/ 725 <====
11375 041510 012742 001155 MOV #1155,-(R2) ;MOVE TO MAILBOX # ***** 1155 *****
11376 041514 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11377 041516 000000 HALT ;SP IS NOT POINTING TO LOC 0
11378 041520 012737 027324 000004 DBE5: MOV #T04,@#RTRAP5 ;RESET TIMEOUT VECTOR
11379 041526 012737 027334 000010 MOV #T010,@#RTRAP ;RESET ILLEGAL INST. VECTOR
11380 041534 012706 001000 MOV #STBOT,SP ;RESET SP
11381
11382 ;*****
11383 ;TEST 410 TEST MFPT
11384 ;*****
11385
```

```

11386 041540 005212 TS410: INC (R2) ;UPDATE TEST NUMBER
11387 041542 022712 000410 CMP #410,(R2) ;SEQUENCE ERROR?
11388 041546 001023 BNE TS411-10 ;BR TO ERROR HALT ON SEQ ERROR
11389 ;THIS TESTS THE MFPT INSTRUCTION- MOVE FROM PROCESSOR TYPE
11390 ;UPON EXECUTION, R0 WILL RECEIVE THE PROCESSOR MODEL CODE
11391 ;WHICH IS '000003' FOR F11.
11392 MFPT=000007
11393 041550 012706 001000 MOV #STBOT,R6 ;INIT. SP
11394 041554 013746 000010 MOV @#10,-(SP) ;SAVE TRAP VECTOR
11395 041560 012737 041610 000010 MOV #1$,@#10 ;SET UP ILLEGAL INSTRUCTION TRAP
11396 041566 010046 MOV R0,-(SP) ;SAVE R0
11397 041570 000007 MFPT ;GET PROCESSOR MODEL
11398 041572 010037 041620 MOV R0,@#CPUTYP ;STORE IT
11399 041576 012600 MOV (SP)+,R0 ;RESTORE R0
11400 041600 022737 000003 04162C CMP #3,@#CPUTYP ;CHECK MODEL TYPE
11401 041606 001405 BEQ XXT
11402 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
11403 ; CONDITIONAL BRANCH INST. AND <====
11404 ; REPLACE THE MOVE INSTRUCTION <====
11405 ; WHICH FOLLOWS W/ 757 <====
11406 041610 1$:
11407 041610 012742 001156 MOV #1156,-(R2) ;MOVE TO MAILBOX # ***** 1156 *****
11408 041614 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
11409 041616 000000 HALT ;ILLEGAL INSTR TRAP OR WRONG MODEL TYPE
11410
11411 041620 000000 CPUTYP: .WORD 0
11412 041622 XXT:
11413 041622 012637 000010 MOV (SP)+,@#10 ;RESTORE TRAP VECTOR
11414 ;*****
11415 ;THIS TEST WILL CHECK THE SERVICE ROUTINE FOR A CONTROL CHIP ERROR.
11416 ;THIS IS DONE BY EXECUTING INSTRUCTIONS WHICH JUMP TO NON-EXISTENT
11417 ;CONTROL-CHIP. ON THE KDF11-A, THIS INCLUDES: FIS(CTL3), CIS AND WCS
11418 ;INSTRUCTIONS. A CTLERR TRAPS TO LOCATION 10.
11419 ;THE RESET LINE IS ALSO ASSERTED FOR 1 CYCLE.
11420 ;*****
11421 ;TEST 411 TEST CTLERR SERVICE ROUTINE
11422 ;*****
11423 041626 005212 TS411: INC (R2) ;UPDATE TEST NUMBER
11424 041630 022712 000411 CMP #411,(R2) ;SEQUENCE ERROR?
11425 041634 001022 BNE TS412-10 ;BR TO ERROR HALT ON SEQ ERROR
11426
11427 041636 012706 001000 MOV #STBOT,R6 ;INIT STACK POINTER
11428 041642 012737 041662 000010 MOV #T$,@#10 ;SET UP RETURN ADDR FROM TRAP
11429 041650 012737 000340 000012 MOV #340,@#12 ;SET TRAP PRIORITY-7
11430 041656 075006 FADD R6 ;EXECUTE FIS INSTR..SHOULD CAUSE CTLERR
11431 041660 000000 HALT ;DID NOT TRAP..CHECK CSEL LINE
11432 041662 012737 041674 000010 1$: MOV #2$,@#10 ;SET UP RETURN ADDR FROM TRAP
11433 041670 076000 HALT ;EXECUTE CIS INSTR..SHOULD TRAP
11434 041672 000000 HALT ;DID NOT TRAP
11435 041674 012737 041706 000010 2$: MOV #3$,@#10 ;SET UP RETURN ADDR FROM TRAP
11436 041702 076700 HALT ;EXECUTE WCS INSTR..SHOULD CAUSE CTLERR AND TRAP
11437 041704 000000 HALT ;DID NOT TRAP
11438 041706 012706 001000 3$: MOV #STBOT,R6 ;RE-INIT STACK POINTER
11439
11440 ;*****
11441 ;TEST 412 TEST THAT ALL RESERVED INSTRUCTIONS TRAP
    
```

```

11442
11443 041712 005212
11444 041714 022712 000412
11445 041720 001116
11446 041722 042767 000100 135634
11447 041730 012737 041756 000244
11448 041736 013767 000010 000024
11449 041744 012737 041766 000010
11450 041752 170007
11451 041754 000406
11452 041756
11453 041756 013767 042310 000334
11454 041764 000002
11455 041766
11456 041766 000002
11457 041770 000000
11458
11459 041772
11460 041772 012737 000246 000244
11461 042000 016737 177764 000010
11462 042006 012703 042200
11463 042012 012305
11464 042014 012301
11465 042016 020537 042254
11466 042022 001007
11467 042024 032777 000002 165422
11468 042032 001403
11469 042034 012703 042310
11470 042040 000764
11471 042042 020567 000252
11472 042046 001415
11473 042050 010567 000246
11474 042054 005267 000242
11475 042060 012767 042112 135722
11476 042066 012706 001000
11477 042072 005067 135700
11478 042076 000167 000220
11479 042102 012700 000370
11480 042106 000167 000320
11481
11482
11483 042112 020627 000774
11484 042116 001404
11485 042120 012742 001157
11486 042124 005242
11487 042126 000000
11488 042130 026727 136640 042324
11489 042136 001404
11490 042140 012742 001160
11491 042144 005242
11492 042146 000000
11493 042150 005767 136622
11494 042154 001404
11495
11496
11497

```

```

:*****
TS412: INC (R2) ;UPDATE TEST NUMBER
      CMP #412,(R2) ;SEQUENCE ERROR?
      BNE RET4 ;BR TO ERROR HALT ON SEQ ERROR
      BIC #100,TPS
      MOV #TRAP244,@#244 ; SET UP TO SEE IF
      MOV @#10,TENSAVE ; THIS PROCESSOR HAS THE
      MOV #TRAP10,@#10 ; FLOATING POINT OPTION
      .WORD 170007 ; AN ILLEGAL FPP INSTRUCTION
      BR AROUND ; THE FOLLOWING
TRAP244: MOV @#FPP,FINISH ; IF FPP IN--
      RTI ; RESET END OF TABLE POINTER
TRAP10: ; AND RETURN
      RTI ; LEAVE THE TABLE ALONE
TENSAVE: .WORD 0 ; AND RETURN
      ; A PLACE TO STORE CONTENTS OF 10
AROUND: ; CONTINUATION POINT
      MOV #246,@#244 ; RESTORE THE TRAP VECTOR
      MOV TENSAVE,@#10 ; RESTORE THE ILLEGAL INST. VECTOR
      MOV #TABLE,TAB ;TABLE POINTER
GIN1: MOV (TAB)+,FIRST ;FIRST OR CURRENT INSTRUCTION
      MOV (TAB)+,LAST ;LAST INSTRUCTION OR GROUP
      CMP FIRST,@#CIS ;HAVE WE TESTED UP TO THE CIS INSTRUCTION SPACE
      BNE 1$ ;IF NO CONTINUE TESTING
      BIT #2,@SWR ;IF YES,CHECK IF CIS CHIPS PRESENT
      BEQ 1$ ;IF THEY ARE NOT CONTINUE TESTING
      MOV #FPP,TAB ;IF CIS HERE SKIP CIS INSTRUCTION SPACE
      BR GIN1
1$: CMP FIRST,FINISH ;TESTED ALL
      BEQ GIN3 ;YES BRANCH
      MOV FIRST,INST ;SET UP INST
GIN2: INC INST
      MOV #RET,10 ;SET UP RETURN FROM TRAP
      MOV #BUFF,SP ;SET UP STACK POINTER
      CLR CC ;CLEAR PRIORITY
      JMP INST ;EXECUTE RESERVED INSTRUCTION
GIN3: MOV #370,R0 ;RESET RESERVED AREA 370-402
      JMP THRPT ;JUMP TO EIS TEST
;TRAPPING SHOULD SEND YOU HERE
RET: CMP SP,#BUFF-4 ;TEST DECREMENT OF SP
      BEQ RET1
      MOV #1157,-(R2) ;MOVE TO MAILBOX # ***** 1157 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;WRONG DECREMENT
RET1: CMP BUFF-4,#INST+2 ;LOC OF INST UNINCREMENTED
      BEQ RET2
      MOV #1160,-(R2) ;MOVE TO MAILBOX # ***** 1160 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;INST INC ON TRAP
RET2: TST BUFF-2
      BEQ RET3
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <---
; CONDITIONAL BRANCH INST. AND <---
; REPLACE THE MOVE INSTRUCTION <---

```

```
11498
11499 042156          RET4:          ;          WHICH FOLLOWS W/ 661          <====
11500 042156 012742 001161      MOV      #1161,-(R2)      ;MOVE TO MAILBOX # ***** 1161 *****
11501 042162 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
11502 042164 000000          HALT                    ;CONDITION CODES SET ON TRAP OR WRONG $STNM
11503 042166 026701 000130      RET3:  CMP      INST, LAST
11504 042172 001707          BEQ      GIN1          ;SET UP NEW GROUP
11505 042174 000167 177654      JMP      GIN2          ;FINISH OLD GROUP
11506
11507 042200 000007      TABLE: 7          ;END OF INSTRUCTION GROUP
11508 042202 000077          77                    ;END OF OPERATE
11509 042204 000207          207                  ;RTS,RT1,JMP
11510 042206 000227          227
11511 042210 006777          6777
11512 042212 007777          7777
11513 042214 075037          075037
11514 042216 076017          76017
11515 042220 076032          76032
11516 042222 076037          76037
11517 042224 076045          76045
11518 042226 076047          76047
11519 042230 076077          76077
11520 042232 076127          76127
11521 042234 076132          76132
11522 042236 076137          76137
11523 042240 076145          76145
11524 042242 076147          76147
11525 042244 076157          76157
11526 042246 076167          76167
11527 042250 076177          76177
11528 042252 076777          76777
11529 042254 076017      CIS:  76017
11530 042256 076032          76032
11531 042260 076037          76037
11532 042262 076045          76045
11533 042264 076047          76047
11534 042266 076077          76077
11535 042270 076127          76127
11536 042272 076132          76132
11537 042274 076137          76137
11538 042276 076145          76145
11539 042300 076147          76147
11540 042302 076157          76157
11541 042304 076167          76167
11542 042306 076177          76177
11543 042310 167777      FPP:  167777          ; START OF THE FPP INSTRUCTIONS
11544 042312 177700          177700
11545 042314 177716          177716
11546 042316 177777          177777
11547 042320 042320      FINISH: .          ;END FLAG
11548 042322 000000      INST:  HALT          ;WILL CONTINUE RESERVED INST
11549 042324 000000          HALT          ;SHOULD TRAP TO LOC 10
11550 042326 000000          HALT          ;LOC 10 SHOULD SEND YOU TO
11551 042330 000000          HALT          ;RET
11552 042332 000000          HALT
```

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

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 20000 IN
LOCATION 320

.ENDR

```
11591
11592
11593          000000          DUMMY=  0
11594          000001          ERRNM=  1
11595          000051          F=      51
11596          000176          N=     176
11597          001000          SW09=  1000
11598          002000          SW10=  2000
11599          004000          SW11=  4000
11600          010000          SW12= 10000
11601
11602 042334          COUNT:
11603          042336          .=COUNT+2
11604 042336          PSWORD:
11605          042340          .=PSWORD+2
11606 042340          TEMP1:
11607          042342          .=TEMP1+2
11608 042342          TEMP2:
11609          042344          .=TEMP2+2
11610 042344          TEMP3:
11611          042346          .-TEMP3+2
11612 042346          TEMP4:
11613          042350          .-TEMP4+2
11614 042350 000000          TEMP5: .WORD
11615 042352 000000          TEMP6: .WORD
11616 042354          000          TYPcnt: .BYTE
11617 042355          000          $TPCNT: .BYTE
11618 042356 000007          S0:     7
11619 042360 177771          S1:    -7
11620 042362 042360          S2:     S1
11621 042364 177772          S3:    -6
11622 042366 177777          S4:    -1
11623 042370 040000          S5:   40000
11624 042372 042370          S6:     S5
11625 042374 040000          S7:   40000
11626 042376 177776          S8:    -2
11627 042400 000002          S9:     2
11628 042402 042400          S10:    S9
11629 042404 000002          S11:     2
11630 042406 000064          TTYOUT: 64
11631 042410 177566          $TPB:  177566
11632 042412 177564          $TPS:  177564
11633 042414 005015 020040 000040          $CRLF: .ASCIIZ <15><12>/ /
11634 042422 006412 047520 042527          POWER: .ASCIIZ <12><15>/POWER/
11635 042430 000122
11636
11637
11638
11639
11640
11641
11642
11643
11644
11645
11646 042432 012705 000304          THRPRt: MOV #STESTN,R5          :MAKE R5 POINT TO THE LOCATION $TESTN
```

11647	042436	005037	042334		CLR	@#COUNT	:CLEAR THE COUNTER
11648	042442	012715	000001		MOV	#1,(R5)	:INITIALIZE TEST NUMBER
11649	042446	012706	001000		MOV	#STBOT,SP	:** STACK AT STBOT **
11650	042452	012737	000001	042340	MOV	#1,@#TEMP1	:TEMP1=1
11651	042460	005037	042342		CLR	@#TEMP2	:TEMP2=0
11652	042464	012737	000001	042344	MOV	#1,@#TEMP3	:TEMP3=1
11653	042472	005037	042346		CLR	@#TEMP4	:TEMP4=0
11654	042476	012737	062320	000034	MOV	#\$TRAP,@#34	:SET UP TRAP INSTRUCTION VECTOR
11655	042504	012737	000340	000036	MOV	#340,@#36	:AND VECTOR PLUS TWO
11656	042512	106427	000000		MTPS	#0	:CLEAR PROCESSOR PRIORITY BITS IN PSW
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

042516 010767 135450
042522 013700 042340
042526 032737 000001 000306
042534 001004
042536 013701 042342
042542 072001
042544 000402
042546 072067 177570
042552 106737 042336
042556 123737 042346 042336
042564 001403
042566 004767 016020
042572 000001
042574 005237 042334
042600 023700 042344
042604 001403
042606
042606 004767 016000
042612 000002
042614 021537 042334
042620 001372
042622 005215
042624 010767 135342
042630 021527 000037
042634 002011

ASTART:
000306
2\$:
4\$:
6\$:
6\$:
8\$

```
ASTART: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV @#TEMP1,%0 ;LOAD R0 WITH THE CONTENTS OF TEMP1
BIT #1,@#$PASS ;IS IT AN EVEN PASS ?
BNE 2$ ;IF NOT THEN GO TO 2$
MOV @#TEMP2,R1 ;OTHERWISE EXECUTE THE INSTRUCTION
;IN MODE 0 USING R1
ASH R1,R0
BR 4$
2$: ASH TEMP2,%0 ;SHIFT R0 BY THE NUMBER SPECIFIED BY TEMP2
4$: MFPS @#PSWORD ;SAVE PS
CMPB @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;THE PS IS NOT EQUAL TO 0
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
INC @#COUNT ;INCREMENT THE COUNTER
CMP @#TEMP3,%0 ;IS THE RESULT IN R0 EQUAL TO TEMP3?
BEQ .+10
6$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;EITHER INCORRECT R0 OR INCORRECT SEQUENCE
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
CMP (R5),@#COUNT ;IS THE TEST NUMBER EQUAL TO THE
;COUNTER?
BNE 6$ ;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$
```

: ASH INSTRUCTION TESTS
:*****

: TESTS 1-36
:*****

11706	042636	005237	042342	INC	@TEMP2	
11707	042642	006367	177476	ASL	TEMP3	;SHIFT TEMP3 LEFT.
11708	042646	021527	000020	CMP	(R5),#20	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11709	042652	001004		BNE	REGR1	
11710	042654	000167	001010	JMP	NEGAT	;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11711	042660	004767	001032	8\$: JSR	PC,TST37	;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11712	042664	010767	135302	REGR1: MOV	PC,LPADR	;STORE ERROR LOOP ADDRESS
11713	042670	013701	042340	MOV	@TEMP1,%1	;LOAD R1 WITH THE CONTENTS OF TEMP1
11714	042674	032737	000001	000306 BIT	#1,@\$PASS	;IS IT AN EVEN PASS ?
11715	042702	001004		BNE	2\$;IF NOT THEN GO TO 2\$
11716	042704	013702	042342	MOV	@TEMP2,R2	;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11717	042710	072102		ASH	R2,R1	;USING R1
11718	042712	000402		BR	4\$	
11719	042714	072167	177422	2\$: ASH	TEMP2,%1	;SHIFT R1 BY THE NUMBER SPECIFIED BY TEMP2
11720	042720	106737	042336	4\$: MFPS	@PSWORD	;SAVE PS
11721	042724	123737	042346	042336 CMPB	@TEMP4,@PSWORD	;IS THE PS = TEMP4 ?
11722	042732	001403		BEQ	.+10	
11723	042734	004767	015652	JSR	PC,\$HLT	;SEEN AN ERROR, GO TO THE HALT ROUTINE
11724						;THE PS IS NOT EQUAL TO 0
11725	042740	000003		3		;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11726						;BY (013746 000172 000207)
11727						
11728	042742	005237	042334	INC	@COUNT	;INCREMENT THE COUNTER
11729	042746	023701	042344	CMP	@TEMP3,%1	;IS THE RESULT IN R1 EQUAL TO TEMP3?
11730	042752	001403		BEQ	.+10	
11731	042754			6\$: JSR	PC,\$HLT	;SEEN AN ERROR, GO TO THE HALT ROUTINE
11732	042754	004767	015632			;EITHER INCORRECT R1 OR INCORRECT SEQUENCE
11733						;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11734	042760	000004		4		;BY (013746 000172 000207)
11735						
11736						
11737	042762	021537	042334	CMP	(R5),@COUNT	;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11738	042766	001372		BNE	6\$;IF NOT GO TO THE HLT ABOVE
11739	042770	005215		INC	(R5)	
11740	042772	010767	135174	MOV	PC,LPADR	;STORE ERROR LOOP ADDRESS
11741	042776	021527	000037	CMP	(R5),#37	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT
11742						;BY 14. AND RIGHT BY 14.?
11743	043002	002011		BGE	8\$	
11744	043004	005237	042342	INC	@TEMP2	
11745	043010	006367	177330	ASL	TEMP3	;SHIFT TEMP3 LEFT
11746	043014	021527	000020	CMP	(R5),#20	;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11747	043020	001004		BNE	REGR2	
11748	043022	000167	000642	JMP	NEGAT	;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11749	043026	004767	000664	8\$: JSR	PC,TST37	;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11750	043032	010767	135134	REGR2: MOV	PC,LPADR	;STORE ERROR LOOP ADDRESS
11751	043036	013702	042340	MOV	@TEMP1,%2	;LOAD R2 WITH THE CONTENTS OF TEMP1
11752	043042	032737	000001	000306 BIT	#1,@\$PASS	;IS IT AN EVEN PASS ?
11753	043050	001004		BNE	2\$;IF NOT THEN GO TO 2\$
11754	043052	013703	042342	MOV	@TEMP2,R3	;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11755	043056	072203		ASH	R3,R2	;USING R2
11756	043060	000402		BR	4\$	
11757	043062	072267	177254	2\$: ASH	TEMP2,%2	;SHIFT R2 BY THE NUMBER SPECIFIED BY TEMP2
11758	043066	106737	042336	4\$: MFPS	@PSWORD	;SAVE PS
11759	043072	123737	042346	042336 CMPB	@TEMP4,@PSWORD	;IS THE PS = TEMP4 ?
11760	043100	001403		BEQ	.+10	
11761	043102	004767	015504	JSR	PC,\$HLT	;SEEN AN ERROR, GO TO THE HALT ROUTINE

```
11762                                     ;THE PS IS NOT EQUAL TO 0
11763 043106 000005                        5      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11764                                     ;BY (013746 000172 000207)
11765
11766 043110 005237 042334                INC    @#COUNT
11767 043114 023702 042344                CMP    @#TEMP3,%2      ;IS THE RESULT IN R2 EQUAL TO TEMP3?
11768 043120 001403                        BEQ    .+10
11769 043122
11770 043122 004767 015464                6$:    JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11771                                     ;EITHER INCORRECT R2 OR INCORRECT SEQUENCE
11772 043126 000006                        6      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11773                                     ;BY (013746 000172 000207)
11774
11775 043130 021537 042334                CMP    (R5),@#COUNT  ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11776 043134 001372                        BNE    6$              ;IF NOT GO TO THE HLT ABOVE
11777 043136 005215                        INC    (R5)
11778 043140 010767 135026                MOV    PC,LPADR        ;STORE ERROR LOOP ADDRESS
11779 043144 021527 000037                CMP    (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11780                                     ;LEFT BY 14, AND RIGHT BY 14.?
11781 043150 002011                        BGE    8$
11782 043152 005237 042342                INC    @#TEMP2
11783 043156 006367 177162                ASL    TEMP3           ;SHIFTED TEMPS LEFT
11784 043162 021527 000020                CMP    (R5),#20        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11785 043166 001004                        BNE    REGR3
11786 043170 000167 000474                JMP    NEGAT           ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11787 043174 004767 000516                8$:    JSR    PC,TST37   ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11788 043200 010767 134766                REGR3: MOV    PC,LPADR  ;STORE ERROR LOOP ADDRESS
11789 043204 013703 042340                MOV    @#TEMP1,%3     ;LOAD R3 WITH THE CONTENTS OF TEMP1
11790 043210 032737 000001 000306        BIT    #1,@#SPASS     ;IS IT AN EVEN PASS ?
11791 043216 001004                        BNE    2$              ;IF NOT THEN GO TO 2$
11792 043220 013704 042342                MOV    @#TEMP2,R4     ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
11793 043224 072304                        ASH    R4,R3           ;USING R3
11794 043226 000402                        BR     4$
11795 043230 072367 177106                2$:    ASH    TEMP2,%3   ;SHIFT R3 BY THE NUMBER SPECIFIED BY TEMP2
11796 043234 106737 042336                4$:    MFPS    @#PSWORD  ;SAVE PS
11797 043240 123737 042346 042336        CMPB   @#TEMP4,@#PSWORD;IS THE PS = TEMP4 ?
11798 043246 001403                        BEQ    .+10
11799 043250 004767 015336                JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11800                                     ;THE PS IS NOT EQUAL TO 0.
11801 043254 000007                        7      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11802                                     ;BY (013746 000172 000207)
11803
11804 043256 005237 042334                INC    @#COUNT
11805 043262 023703 042344                CMP    @#TEMP3,%3     ;IS THE RESULT IN R3 EQUAL TO TEMP3?
11806 043266 001403                        BEQ    .+10
11807 043270
11808 043270 004767 015316                6$:    JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11809                                     ;EITHER INCORRECT R3 OR INCORRECT SEQUENCE
11810 043274 000010                        10     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11811                                     ;BY (013746 000172 000207)
11812
11813 043276 021537 042334                CMP    (R5),@#COUNT  ;IS THE TEST NUMBER EQUAL TO THE COUNTER?
11814 043302 001372                        BNE    6$              ;IF NOT GO TO THE HLT ABOVE
11815 043304 005215                        INC    (R5)
11816 043306 010767 134660                MOV    PC,LPADR        ;STORE ERROR LOOP ADDRESS
11817 043312 021527 000037                CMP    (R5),#37        ;HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
```

```

11818                                     ;LEFT BY 14, AND RIGHT BY 14.?
11819 043316 002010                      BGE      8$
11820 043320 005237 042342              INC      @#TEMP2
11821 043324 006367 177014              ASL      TEMP3
11822 043330 021527 000020              CMP      (R5),#20
11823 043334 001003                      BNE      REGR4
11824 043336 000554                      BR       NEGAT
11825 043340 004767 000352              8$: JSR   PC,TST37
11826 043344 010767 134622              REGR4: MOV  PC,LPADR
11827 043350 013704 042340              MOV     @#TEMP1,%4
11828 043354 010501                      MOV     R5,R1
11829 043356 032737 0000C1 000306      BIT     #1,@#SPASS
11830 043364 001004                      BNE     2$
11831 043366 013705 042342              MOV     @#TEMP2,R5
11832 043372 072405                      ASH     R5,R4
11833 043374 000402                      BR      4$
11834 043376 072467 176740              2$: ASH  TEMP2,%4
11835 043402 106737 042336              4$: MFPS @#PSWORD
11836 043406 123737 042346 042336      CMPB   @#TEMP4,@#PSWORD
11837 043414 001403                      BEQ     .+10
11838 043416 004767 015170              JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11839                                     ;THE PS IS NOT EQUAL TO 0
11840 043422 000011                      11
11841                                     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11842                                     ;BY (013746 000172 000207)
11843 043424 005237 042334              INC     @#COUNT
11844 043430 023704 042344              CMP     @#TEMP3,%4
11845 043434 001403                      BEQ     .+10
11846 043436                                     ;IS THE RESULT IN R4 EQUAL TO TEMP3?
11847 043436 004767 015150              6$: JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11848                                     ;EITHER INCORRECT R4 OR INCORRECT SEQUENCE
11849 043442 000012                      12
11850                                     ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11851                                     ;BY (013746 000172 000207)
11852 043444 010105                      MOV     R1,R5
11853 043446 021537 042334              CMP     (R5),@#COUNT
11854 043452 001371                      BNE     6$
11855 043454 005215                      INC     (R5)
11856 043456 010767 134510              MOV     PC,LPADR
11857 043462 021527 000037              CMP     (R5),#37
11858                                     ;STORE ERROR LOOP ADDRESS
11859 043466 002010                      BGE     8$
11860 043470 005237 042342              INC     @#TEMP2
11861 043474 006367 176644              ASL     TEMP3
11862 043500 021527 000020              CMP     (R5),#20
11863 043504 001003                      BNE     REGR5
11864 043506 000470                      BR      NEGAT
11865 043510 004767 000202              8$: JSR   PC,TST37
11866 043514 010767 134452              REGR5: MOV  PC,LPADR
11867 043520 010501                      MOV     R5,R1
11868 043522 013705 042340              MOV     @#TEMP1,%5
11869 043526 032737 000001 000306      BIT     #1,@#SPASS
11870 043534 001004                      BNE     2$
11871 043536 013700 042342              MOV     @#TEMP2,R0
11872 043542 072500                      ASH     R0,R5
11873 043544 000402                      BR      4$
                                     ;OTHERWISE EXECUTE ASH INSTRUCTION IN MODE 0
                                     ;USING R5
                                     ;SHIFTE LEFT BY 14. AND RIGHT BY 14.?
                                     ;HAS THE CONTENTS OF REGISTER BEEN SHIFTE BY 14.?
                                     ;IF SO GO TO NEGAT AND INITIATE RIGHT SHIFTE
                                     ;IF SO GO AND CONTINUE THE REST OF THE PROGRAM
                                     ;STORE ERROR LOOP ADDRESS
                                     ;SAVE R5
                                     ;LOAD R5 WITH THE CONTENTS OF TEMP1
                                     ;IS IT AN EVEN PASS ?
                                     ;IF NOT THEN GO TO 2$
                                     ;USING R5

```

11874	043546	072567	176570		2\$:	ASH	TEMP2,%5		:SHIFT R5 BY THE NUMBER SPECIFIED BY TEMP2
11875	043552	106737	042336		4\$:	MFPS	@#PSWORD		:SAVE PS
11876	043556	123737	042346	042336		CMPB	@#TEMP4,@#PSWORD		:IS PS = TEMP4 ?
11877	043564	001403				BEQ	.+10		
11878	043566	004767	015020			JSR	PC,\$HLT		:SEEN AN ERROR, GO TO THE HALT ROUTINE
11879									:THE PS IS NOT EQUAL TO 0.
11880	043572	000013				13			:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11881									:BY (013746 000172 000207)
11882									
11883	043574	005237	042334			INC	@#COUNT		
11884	043600	023705	042344			CMP	@#TEMP3,%5		:IS THE RESULT IN R5 EQUAL TO TEMP3?
11885	043604	001403				BEQ	.+10		
11886	043606				6\$:				
11887	043606	004767	015000			JSR	PC,\$HLT		:SEEN AN ERROR, GO TO THE HALT ROUTINE
11888									:EITHER INCORRECT R5 OR INCORRECT SEQUENCE
11889	043612	000014				14			:TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11890									:BY (013746 000172 000207)
11891									
11892	043614	021137	042334			CMP	(R1),@#COUNT		:IS THE TEST NUMBER EQUAL TO THE COUNTER?
11893	043620	001372				BNE	6\$:IF NOT GO TO THE HLT ABOVE
11894	043622	010105				MOV	R1,R5		:RESTORE R5
11895	043624	005215				INC	(R5)		
11896	043626	010767	134340			MOV	PC,LPADR		:STORE ERROR LOOP ADDRESS
11897	043632	021527	000037			CMP	(R5),#37		:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED
11898									:LEFT BY 14. AND RIGHT BY 14.?
11899									
11900	043636	002010				BGE	8\$:IF SO GO AND CONTINUE THE REST OF THE PROGRAM
11901	043640	005237	042342			INC	@#TEMP2		
11902	043644	006367	176474			ASL	TEMP3		:SHIFT TEMP3 LEFT
11903	043650	021527	000020			CMP	(R5),#20		:HAS THE CONTENTS OF REGISTERS BEEN SHIFTED LEFT BY 14.?
11904	043654	001405				BEQ	NEGAT		:IF SO GO TO NEGAT AND INITIATE RIGHT SHIFT
11905	043656	000402				BR	10\$		
11906	043660	004767	000032		8\$:	JSR	PC,TST37		
11907	043664	000167	176626		10\$:	JMP	ASTART		:GO BACK TO START
11908	043670	012737	040000	042340	NEGAT:	MOV	#40000,@#TEMP1		:TEMP1=40000
11909	043676	012737	177762	042342		MOV	#177762,@#TEMP2		:TEMP2=177762
11910	043704	012737	000001	042344		MOV	#1,@#TEMP3		:TEMP3=1
11911	043712	000167	176600			JMP	ASTART		
11912	043716	021527	000037		TST37:	CMP	(R5),#37		:IS IT TEST 37?
11913	043722	001013				BNE	TST40		:IF NOT THEN TRY TEST 40
11914	043724	005037	042340			CLR	@#TEMP1		:0
11915	043730	012737	000020	042342		MOV	#16,@#TEMP2		:SHIFTED BY 16
11916	043736	005037	042344			CLR	@#TEMP3		:IS=0
11917	043742	012737	000004	042346		MOV	#4,@#TEMP4		:AND PS=4
11918	043750	000207				RTS	PC		
11919	043752	021527	000040		TST40:	CMP	(R5),#40		:IS IT TEST 40?
11920	043756	001003				BNE	TST41		:IF NOT THEN TRY TEST 41
11921	043760	005037	042342			CLR	@#TEMP2		:0 SHIFTED BY 0=0 AND PS=4
11922	043764	000207				RTS	PC		
11923	043766	021527	000041		TST41:	CMP	(R5),#41		:IS IT TEST 41?
11924	043772	001004				BNE	TST42		:IF NOT THEN TRY TEST 42
11925	043774	012737	177760	042342		MOV	#-16,@#TEMP2		:0 SHIFTED BY -16.=0 AND PS-4
11926	044002	000207				RTS	PC		
11927	044004	021527	000042		TST42:	CMP	(R5),#42		:IS IT TEST 42?
11928	044010	001013				BNE	TST43		:IF NOT THEN TRY TEST 43
11929	044012	012737	100000	042340		MOV	#100000,@#TEMP1		:100000

```

11930 044020 005237 042342      INC      @#TEMP2      ;SHIFTED BY -15
11931 044024 005337 042344      DEC      @#TEMP3      ;IS=-1
11932 044030 012737 000010 042346  MOV      #10,@#TEMP4      ;AND PS=10
11933 044036 000207                RTS      PC
11934 044040 021527 000045      TST43:  CMP      (R5),#43      ;IS IT TEST 43?
11935 044044 001012                BNE      TST44      ;IF NOT THEN IF NOT THEN TRY TEST 44
11936 044046 012737 125252 042340  MOV      #125252,@#TEMP1 ;125252
11937 044054 012737 177777 042342  MOV      #-1,@#TEMP2      ;SHIFTED BY -1
11938 044062 012737 152525 042344  MOV      #152525,@#TEMP3 ;IS=152525 AND PS=10
11939 044070 000207                RTS      PC
11940 044072 021527 000044      TST44:  CMP      (R5),#44      ;IS IT TEST 44?
11941 044076 001012                BNE      TST45      ;IF NOT THEN TRY TEST 45
11942 044100 012737 000001 042342  MOV      #1,@#TEMP2      ;125252 SHIFTED BY 1
11943 044106 012737 052524 042344  MOV      #52524,@#TEMP3 ;IS=52524
11944 044114 012737 000003 042346  MOV      #3,@#TEMP4      ;AND PS=3
11945 044122 000207                RTS      PC
11946 044124 021527 000045      TST45:  CMP      (R5),#45      ;IS IT TEST 45?
11947 044130 001012                BNE      TST46      ;IF NOT THEN TRY TEST 46
11948 044132 012737 177776 042342  MOV      #-2,@#TEMP2      ;125252 SHIFTED BY -2
11949 044140 012737 165252 042344  MOV      #165252,@#TEMP3 ;IS=165252
11950 044146 012737 000011 042346  MOV      #11,@#TEMP4      ;AND PS=11
11951 044154 000207                RTS      PC
11952 044156 021527 000046      TST46:  CMP      (R5),#46      ;IS IT TEST 46?
11953 044162 001014                BNE      TST47      ;IF NOT THEN TRY TEST 47
11954 044164 012737 177777 042340  MOV      #-1,@#TEMP1      ;-1
11955 044172 012737 000020 042342  MOV      #16,@#TEMP2      ;SHIFTED BY 15.
11956 044200 005037 042344      CLR      @#TEMP3      ;IS=0
11957 044204 012737 000007 042346  MOV      #7,@#TEMP4      ;AND PS=7
11958 044212 000207                RTS      PC
11959 044214 021527 000047      TST47:  CMP      (R5),#47      ;IS IT TEST 47?
11960 044220 001011                BNE      TST50      ;IF NOT THEN TRY TEST 50
11961 044222 005337 042342      DEC      @#TEMP2      ;-1 SHIFTED BY 15
11962 044226 012737 100000 042344  MOV      #100000,@#TEMP3 ;IS=100000
11963 044234 012737 000011 042346  MOV      #11,@#TEMP4      ;AND PS=11
11964 044242 000207                RTS      PC
11965 044244 021527 000050      TST50:  CMP      (R5),#50      ;IS IT TEST 50
11966 044250 001007                BNE      ENT51      ;IF NOT THEN TRY TEST 51
11967 044252 012737 137777 042340  MOV      #137777,@#TEMP1 ;137777 SHIFTED BY 15. IS=100000
11968 044260 012737 000013 042346  MOV      #13,@#TEMP4      ;AND PS=13
11969 044266 000207                RTS      PC
11970 044270 021527 000051      ENT51:  CMP      (R5),#51      ;IS IT ENTERING TEST 51?
11971 044274 001403                BEQ      .+10
11972 044276 004767 014310      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11973                                ;TEST NUMBER GOOFED
11974 044302 000015                15                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11975                                ;BY (013746 000172 000207)
11976
11977
11978 044304 005726                TST      (SP)+      ;RESTORE STACK POINTER
11979 044306 012704 177771      MOV      #-7,%4
11980 044312 012702 042360      MOV      #S1,%2
11981 044316 012703 042362      MOV      #S2,%3
11982
11983 :*****
11984 :TEST:51      ASH      125252 SHIFTED BY #5 = 52500 PS = 3
11985 :*****

```

```
11986 044322 010767 133644 TST51: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
11987 044326 012701 125252 MOV #125252,%1 ;LOAD R1 WITH 125252
11988 044332 072127 000005 ASH #5,%1 ;SHIFT R1 BY #5
11989 044336 106737 042336 MFPS @#PSWORD ;SAVE PS
11990 044342 122737 000003 042336 CMPB #3,@#PSWORD ;IS THE PS 3?
11991 044350 001403 BEQ .+10
11992 044352 004767 014234 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
11993 ;THE PS IS NOT EQUAL TO 3
11994 044356 000016 16 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
11995 ;BY (013746 000172 000207)
11996
11997 044360 022701 052500 CMP #52500,%1 ;IS THE RESULT 52500?
11998 044364 001403 BEQ .+10
11999 044366 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12000 044366 004767 014220 ;R1 IS NOT EQUAL TO 52500 OR INCORRECT SEQUENCE
12001 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12002 044372 000017 17 ;BY (013746 000172 000207)
12003
12004
12005 044374 021527 000051 CMP (R5),#51 ;IS $TESTN = #51
12006 044400 001372 BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
12007 044402 005215 INC (R5)
12008
12009
12010
12011
12012 ;*****
12013 ;TEST:52 ASH 125252 SHIFTED BY @S2 = 177525 PS = 10
12014 ;*****
12015 044404 010767 133562 TST52: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12016 044410 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
12017 044414 072077 175742 ASH @S2,%0 ;SHIFT R0 BY @S2
12018 044420 106737 042336 MFPS @#PSWORD ;SAVE PS
12019 044424 122737 000010 042336 CMPB #10,@#PSWORD ;IS THE PS 10?
12020 044432 001403 BEQ .+10
12021 044434 004767 014152 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12022 ;THE PS IS NOT EQUAL TO 10
12023 044440 000020 20 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12024 ;BY (013746 000172 000207)
12025
12026 044442 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
12027 044446 001403 BEQ .+10
12028 044450 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12029 044450 004767 014136 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12030 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12031 044454 000021 21 ;BY (013746 000172 000207)
12032
12033
12034 044456 021527 000052 CMP (R5),#52 ;IS $TESTN = #52
12035 044462 001372 BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
12036 044464 005215 INC (R5)
12037
12038
12039
12040
12041 ;*****
;TEST:53 ASH 125252 SHIFTED BY @S1 - 177525 PS - 10
```

```
12042 ;*****
12043 ;*****
12044 044466 010767 133500 TST53: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12045 044472 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
12046 044476 072037 042360 ASH @#S1,%0 ;SHIFT R0 BY @#S1
12047 044502 106737 042336 MFPS @#PSWORD ;SAVE PS
12048 044506 122737 000010 042336 CMPB #10,@#PSWORD ;IS THE PS 10?
12049 044514 001403 BEQ .+10
12050 044516 004767 014070 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12051 ;THE PS IS NOT EQUAL TO 10
12052 044522 000022 22 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12053 ;BY (013746 000172 000207)
12054
12055 044524 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
12056 044530 001403 BEQ .+10
12057 044532 1$:
12058 044532 004767 014054 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12059 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12060 044536 000023 23 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12061 ;BY (013746 000172 000207)
12062
12063 044540 021527 000053 CMP (R5),#53 ;IS $TESTN = #53
12064 044544 001372 BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
12065 044546 005215 INC (R5)
12066
12067
12068
12069 ;*****
12070 ;TEST:54 ASH 125252 SHIFTED BY (2) = 177525 PS 10
12071 ;*****
12072
12073 044550 010767 133416 TST54: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12074 044554 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
12075 044560 072012 ASH (2),%0 ;SHIFT R0 BY (2)
12076 044562 106737 042336 MFPS @#PSWORD ;SAVE PS
12077 044566 122737 000010 042336 CMPB #10,@#PSWORD ;IS THE PS 10?
12078 044574 001403 BEQ .+10
12079 044576 004767 014010 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12080 ;THE PS IS NOT EQUAL TO 10
12081 044602 000024 24 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12082 ;BY (013746 000172 000207)
12083
12084 044604 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
12085 044610 001403 BEQ .+10
12086 044612 1$:
12087 044612 004767 013774 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12088 ;R0 IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12089 044616 000025 25 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12090 ;BY (013746 000172 000207)
12091
12092 044620 021527 000054 CMP (R5),#54 ;IS $TESTN = #54
12093 044624 001372 BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
12094 044626 005215 INC (R5)
12095
12096
12097
```



```
12098
12099 :*****
12100 :TEST:55      ASH      125252 SHIFTED BY (2)+ = 177525 PS = 10
12101 :*****
12102 044630 010767 133336 TST55: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
12103 044634 012700 125252      MOV      #125252,%0      ;LOAD R0 WITH 125252
12104 044640 072022      ASH      (2)+,%0        ;SHIFT R0 BY (2)+
12105 044642 106737 042336      MFPS     @#PSWORD      ;SAVE PS
12106 044646 122737 000010 042336      CMPB    #10,@#PSWORD   ;IS THE PS 10?
12107 044654 001403      BEQ     .+10
12108 044656 004767 013730      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12109      ;THE PS IS NOT EQUAL TO 10
12110 044662 000026      26      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12111      ;BY (013746 000172 000207)
12112
12113 044664 022700 177525      CMP     #177525,%0      ;IS THE RESULT 177525?
12114 044670 001403      BEQ     .+10
12115 044672
12116 044672 004767 013714 1$:      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12117      ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12118 044676 000027      27      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12119      ;BY (013746 000172 000207)
12120
12121 044700 021527 000055      CMP     (R5),#55        ;IS $TESTN = #55
12122 044704 001372      BNE    1$              ;IF NOT THEN GO TO HLT ABOVE
12123 044706 005215      INC     (R5)
12124
12125
12126
12127 :*****
12128 :TEST:56      ASH      125252 SHIFTED BY -(2) = 177525 PS = 10
12129 :*****
12130
12131 044710 010767 133256 TST56: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
12132 044714 012700 125252      MOV      #125252,%0      ;LOAD R0 WITH 125252
12133 044720 072042      ASH      -(2),%0        ;SHIFT R0 BY -(2)
12134 044722 106737 042336      MFPS     @#PSWORD      ;SAVE PS
12135 044726 122737 000010 042336      CMPB    #10,@#PSWORD   ;IS THE PS 10?
12136 044734 001403      BEQ     .+10
12137 044736 004767 013650      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12138      ;THE PS IS NOT EQUAL TO 10
12139 044742 000030      30      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12140      ;BY (013746 000172 000207)
12141
12142 044744 022700 177525      CMP     #177525,%0      ;IS THE RESULT 177525?
12143 044750 001403      BEQ     .+10
12144 044752
12145 044752 004767 013634 1$:      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12146      ;RO IS NOT EQUAL TO 177525 OR INCORRECT SEQUENCE
12147 044756 000031      31      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12148      ;BY (013746 000172 000207)
12149
12150 044760 021527 000056      CMP     (R5),#56        ;IS $TESTN = #56
12151 044764 001372      BNE    1$              ;IF NOT THEN GO TO HLT ABOVE
12152 044766 005215      INC     (R5)
12153
```

12154
12155
12156
12157
12158
12159
12160
12161
12162
12163
12164
12165
12166
12167
12168
12169
12170
12171
12172
12173
12174
12175
12176
12177
12178
12179
12180
12181
12182
12183
12184
12185
12186
12187
12188
12189
12190
12191
12192
12193
12194
12195
12196
12197
12198
12199
12200
12201
12202
12203
12204
12205
12206
12207
12208
12209

044770 010767 133176
044774 012700 125252
045000 072063 000002
045004 106737 042336
045010 122737 000011 042336
045016 001403
045020 004767 013566

045024 000032

045026 022700 177252
045032 001403
045034
045034 004767 013552

045040 000033

045042 021527 000057
045046 001372
045050 005215

045052 010767 133114
045056 012700 125252
045062 072073 000000
045066 106737 042336
045072 122737 000010 042336
045100 001403
045102 004767 013504

045106 000034

045110 022700 177525
045114 001403
045116
045116 004767 013470

045122 000035

045124 021527 000060
045130 001372

```
*****  
: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)  
  
1$:     CMP      #177252,%0     ;IS THE RESULT 177252?  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;R0 IS NOT EQUAL TO 177252 OR INCORRECT SEQUENCE  
        33      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     (R5),#57        ;IS $TESTN - #57  
        BNE    1$              ;IF NOT THEN GO TO HLT ABOVE  
        INC     (R5)  
  
*****  
:TEST:60      ASH      125252 SHIFTED BY @ (3) = 177525 PS 10  
*****  
TST60:  MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS  
        MOV      #125252,%0     ;LOAD R0 WITH 125252  
        ASH      @ (3),%0       ;SHIFT R0 BY @ (3)  
        MFPS     @#PSWORD      ;SAVE PS  
        CMPB    #10,@#PSWORD    ;IS THE PS 10?  
        BEQ     .+10  
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
        ;THE PS IS NOT EQUAL TO 10  
        34      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
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  
        35      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
        ;BY (013746 000172 000207)  
  
        CMP     (R5),#60        ;IS $TESTN = #60  
        BNE    1$              ;IF NOT THEN GO TO HLT ABOVE
```

12210 045132 005215

INC (R5)

12211

12212

12213

12214

12215

12216

12217

12218 045134 010767 133032

12219 045140 012700 125252

12220 045144 072033

12221 045146 106737 042336

12222 045152 122737 000010 042336

12223 045160 001403

12224 045162 004767 013424

12225

12226 045166 000036

12227

12228

12229 045170 022700 177525

12230 045174 001403

12231 045176

12232 045176 004767 013410

12233

12234 045202 000037

12235

12236

12237 045204 021527 000061

12238 045210 001372

12239 045212 005215

12240

12241

12242

12243

12244

12245

12246 045214 010767 132752

12247 045220 012700 125252

12248 045224 072053

12249 045226 106737 042336

12250 045232 122737 000010 042336

12251 045240 001403

12252 045242 004767 013344

12253

12254

12255 045246 000040

12256

12257

12258 045250 022700 177525

12259 045254 001403

12260 045256

12261 045256 004767 013330

12262

12263 045262 000041

12264

12265

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

TST61: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #125252,%0 ;LOAD R0 WITH 125252
ASH @ (3)+,%0 ;SHIFT R0 BY @ (3)+
MFPS @#PSWORD ;SAVE PS
CMPB #10,@#PSWORD ;IS THE PS 10?
BEQ .+10
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
36 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

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

CJKDB-C DCF11-AA CPU DIAG.
CJKDBC.P11 07-MAR-80 12:17

MACY11 30A(1052) 07-MAR-80 12:18 H 2 PAGE 226
ASH INSTRUCTION TESTS

SEQ 0226

12266 045264 021527 000062
12267 045270 001372
12268 045272 005215
12269
12270
12271

CMP (R5),#62
BNE 1\$
INC (R5)

;IS \$TESTN = #62
;IF NOT THEN GO TO HLT ABOVE

```
12272
12273
12274
12275
12276
12277
12278
12279
12280
12281
12282
12283
12284 045274 010767 132672          MOV    PC,LPADR          ;STORE ERROR LOOP ADDRESS
12285 045300 012737 000062 042334    MOV    #62,@#COUNT
12286 045306 005037 042340          CLR    @#TEMP1          ;TEMP1=0
12287 045312 012737 000001 042342    MOV    #1,@#TEMP2      ;TEMP2=1
12288 045320 005037 042344          CLR    @#TEMP3          ;TEMP3=0
12289 045324 005037 042346          CLR    @#TEMP4          ;TEMP4=0
12290 045330 012737 000001 042350    MOV    #1,@#TEMP5      ;TEMP5=1
12291 045336 005037 042352          CLR    @#TEMP6          ;0 1 SHIFTED BY 0=0 1, PS=0
12292
12293 045342 010502          REG01: MOV    R5,R2          ;SAVE R5
12294 045344 013700 042340    MOV    @#TEMP1,%0      ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 0
12295 045350 013701 042342    MOV    @#TEMP2,%0.1    ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 1
12296 045354 000241          CLC
12297 045356 032737 000001 000306    BIT    #1,@#SPASS      ;IS IT AN EVEN PASS ?
12298 045364 001004          BNE    2$              ;IF NOT THEN GO TO 2$
12299 045366 013705 042344    MOV    @#TEMP3,R5      ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12300 045372 073005          ASHC  R5,R0            ;USING R0
12301 045374 000402          BR     4$
12302 045376 073067 174742          2$: ASHC  TEMP3,%0        ;ASHC REGISTER 0 BY THE CONTENTS OF TEMP3
12303 045402 106737 042336          4$: MFPS  @#PSWORD      ;SAVE PS
12304 045406 123737 042352 042336    CMPB  @#TEMP6,@#PSWORD;COMPARE PS WITH THE CONTENTS OF TEMP6
12305 045414 001403          BEQ   .+10
12306 045416 004767 013170          JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12307
12308 045422 000042          42   ;WRONG PS
12309
12310
12311 045424 005237 042334          INC   @#COUNT
12312 045430 023700 042346          CMP   @#TEMP4,%0      ;IS THE RESULT IN R0 SAME AS TEMP4?
12313 045434 001403          BEQ   .+10
12314 045436 004767 013150          JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12315
12316 045442 000043          43   ;WRONG RESULT IN R0
12317
12318
12319 045444 023701 042350          CMP   @#TEMP5,%1      ;IS THE RESULT IN R1 SAME AS TEMP5?
12320 045450 001403          BEQ   .+10            ;TEMP1 TEMP2 SHIFTED BY TEMP3 TEMP4 TEMP5
12321
12322 045452 004767 013134          JSR   PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12323
12324 045456 000044          44   ;WRONG RESULT IN R1
12325
12326
12327 045460 010205          MOV   R2,R5           ;RESTORE R5
```

```
12328 045462 021537 042334      CMP      (R5),@#COUNT      ;IS TEST NUMBER=COUNTER?
12329 045466 001403              BEQ      .+10
12330 045470 004767 013116      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12331                                ;NO
12332 045474 000045              45          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12333                                ;BY (013746 000172 000207)
12334
12335 045476 005215              INC      (R5)
12336 045500 021527 000160      CMP      (R5),#160          ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
12337 045504 002014              BGE      6$                  ;YES
12338 045506 005237 042344      INC      @#TEMP3
12339 045512 000241              CLC
12340 045514 006137 042350      ROL      @#TEMP5            ;ROTATE TEMP5 LEFT BY 1 PLACE
12341 045520 006137 042346      ROL      @#TEMP4            ;INTRODUCE CARRY FROM TEMP4 IN TEMP5
12342 045524 021527 000121      CMP      (R5),#121          ;IS IT TEST 121?
12343 045530 001004              BNE      REGR23
12344 045532 004467 000414      JSR      R4,RITSH           ;IF SO THEN GO AND INITIATE RIGHT SHIFT
12345 045536 004767 000444      JSR      %7,TST160
12346 045542 010767 132424      MOV      PC,LPADR           ;STORE ERROR LOOP ADDRESS
12347 045546 013702 042340      MOV      @#TEMP1,%2         ;PLACE THE CONTENTS OF TEMP1 IN REGISTER 2
12348 045552 013703 042342      MOV      @#TEMP2,%2!1       ;PLACE THE CONTENTS OF TEMP2 IN REGISTER 3
12349 045556 000241              CLC
12350 045560 032737 000001 000306  BIT      #1,@#$PASS          ;IS IT AN EVEN PASS ?
12351 045566 001004              BNE      2$                  ;IF NOT THEN GO TO 2$
12352 045570 013704 042344      MOV      @#TEMP3,R4         ;OTHERWISE EXECUTE ASHC INSTRUCTION IN MODE 0
12353 045574 073204              ASHC     R4,R2              ;USING R2
12354 045576 000402              BR       4$
12355 045600 073267 174540      ASHC     TEMP3,%2           ;ASHC REGISTER 2 BY THE CONTEINS OF TEMP3
12356 045604 106737 042336      MFPS     @#PSWORD           ;SAVE PS
12357 045610 123737 042352 042336  CMPB     @#TEMP6,@#PSWORD    ;COMPARE PS WITH THE CONTENTS OF TEMP6
12358 045616 001403              BEQ      .+10
12359 045620 004767 012766      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12360                                ;WRONG PS
12361 045624 000046              46          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12362                                ;BY (013746 000172 000207)
12363
12364 045626 005237 042334      INC      @#COUNT
12365 045632 023702 042346      CMP      @#TEMP4,%2         ;IS THE RESULT IN R2 SAME AS TEMP4?
12366 045636 001403              BEQ      .+10
12367 045640 004767 012746      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12368                                ;WRONG RESULT IN R2
12369 045644 000047              47          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12370                                ;BY (013746 000172 000207)
12371
12372 045646 023703 042350      CMP      @#TEMP5,%3         ;IS THE RESULT IN R3 SAME AS TEMP5?
12373 045652 001403              BEQ      .+10               ;TEMP1 TEMP2 SHIFTED BY TEMP3-TEMP4 TEMP5
12374                                ;AND PS=TEMP6
12375 045654 004767 012732      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12376                                ;WRONG RESULT IN R1
12377 045660 000050              50          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12378                                ;BY (013746 000172 000207)
12379
12380 045662 021537 042334      CMP      (R5),@#COUNT      ;IS TEST NUMBER=COUNTER?
12381 045666 001403              BEQ      .+10
12382 045670 004767 012716      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12383                                ;NO
```



```

12440 046100 010105          MOV    R1,R5          ;RESTORE R5
12441 046102 005215          INC    (R5)
12442 046104 021527 000160          CMP    (R5),#160     ;HAVE THE FIRST 159 TEST BEEN EXECUTED?
12443 046110 002014          BGE   6$             ;YES
12444 046112 005237 042344          INC    @#TEMP3
12445 046116 000241          CLC
12446 046120 006137 042350          ROL    @#TEMP5       ;ROTATE TEMP5 LEFT BY 1 PLACE
12447 046124 006137 042346          ROL    @#TEMP4       ;INTRODUCE CARRY FROM TEMP5 IN TEMP4
12448 046130 021527 000121          CMP    (R5),#121    ;IS IT TEST 121?
12449 046134 001004          BNE   8$
12450 046136 004467 000010          JSR   R4,RITSH      ;IF SO THEN GO AND INITIATE RIGHT SHIFT
12451 046142 004767 000040          6$: JSR   %7,TST160
12452 046146 000167 177170          8$: JMP   REG01
12453 046152 022424          RITSH: CMP   (R4)+,(R4)+ ;MAKE R4 POINT TO THE NEXT REG TAG
12454 046154 012737 040000 04234C          MOV   #40000,@#TEMP1 ;TEMP1=4000
12455 046162 005037 042342          CLR   @#TEMP2       ;TEMP2=0
12456 046166 012737 177742 042344          MOV   #-30,@#TEMP3  ;TEMP3=-30
12457 046174 005037 042346          CLR   @#TEMP4       ;TEMP4=0
12458 046200 005237 042350          INC   @#TEMP5       ;TEMP5=1
12459 046204 000204          RTS   R4
12460 046206 021527 000160          TST160: CMP  (R5),#160 ;IS IT TEST 160
12461 046212 001010          BNE   TST161        ;IF NOT THEN TRY TEST 161
12462 046214 005037 042340          CLR   @#TEMP1       ;0 0 SHIFTED BY 0
12463 046220 005037 042346          CLR   @#TEMP4       ;IS EQUAL TO 0 0
12464 046224 012737 000004 042352          MOV   #4,@#TEMP6    ;AND PS=4
12465 046232 000207          RTS   %7
12466 046234 021527 000161          TST161: CMP  (R5),#161 ;IS IT TEST 161
12467 046240 001004          BNE   TST162        ;IF NOT THEN TRY TEST 162
12468 046242 012737 177746 042344          MOV   #-32,@#TEMP3  ;0 0 SHIFTED BY -32=0 0, PS=4
12469 046250 000207          RTS   %7
12470 046252 021527 000162          TST162: CMP  (R5),#162 ;IS IT TEST 162
12471 046256 001004          BNE   TST163        ;IF NOT THEN TRY TEST 163
12472 046260 012737 000032 042344          MOV   #32,@#TEMP3  ;0 0 SHIFTED BY 32=0 0, PS=4
12473 046266 000207          RTS   %7
12474 046270 021527 000163          TST163: CMP  (R5),#163 ;IS IT TEST 163?
12475 046274 001016          BNE   TST164        ;IF NOT THEN TRY TEST 164
12476 046276 012737 052525 042340          MOV   #52525,@#TEMP1 ;52525 0
12477 046304 012737 177760 042344          MOV   #-16,@#TEMP3 ;SHIFTED BY -16.
12478 046312 005037 042346          CLR   @#TEMP4
12479 046316 012737 052525 042350          MOV   #52525,@#TEMP5 ;IS EQUAL TO 0 52525
12480 046324 005037 042352          CLR   @#TEMP6       ;AND PS = 0
12481 046330 000207          RTS   %7
12482 046332 021527 000164          TST164: CMP  (R5),#164 ;IS IT TEST 164?
12483 046336 001014          BNE   TST165        ;IF NOT THEN TRY TEST 165
12484 046340 012737 125252 042340          MOV   #125252,@#TEMP1 ;125252 0 SHIFTED BY -16.
12485 046346 005337 042346          DEC   @#TEMP4
12486 046352 012737 125252 042350          MOV   #125252,@#TEMP5 ;IS EQUAL TO -1 125252
12487 046360 012737 000010 042352          MOV   #10,@#TEMP6   ;AND PS=10
12488 046366 000207          RTS   %7
12489 046370 021527 000165          TST165: CMP  (R5),#165 ;IS IT TEST 165?
12490 046374 001007          BNE   TST166        ;IF NOT THEN TRY TEST 166
12491 046376 012737 177777 042340          MOV   #-1,@#TEMP1  ;-1 0 SHIFTED BY -16
12492 046404 012737 177777 042350          MOV   #-1,@#TEMP5  ;IS EQUAL TO -1 -1, AND PS=10
12493 046412 000207          RTS   %7
12494 046414 021527 000166          TST166: CMP  (R5),#166 ;IS IT TEST 166?
12495 046420 001011          BNE   TST167        ;IF NOT THEN TRY TEST 167

```



```
12496 046422 012737 100000 042340 MOV #100000,@#TEMP1 ;100000 0
12497 046430 012737 177740 042344 MOV #-32,@#TEMP3 ;SHIFTED BY -32 IS EQUAL TO -1 -1
12498 046436 005237 042352 INC @#TEMP6 ;AND PS=11
12499 046442 000207 RTS %7
12500 046444 021527 000167 TST167: CMP (R5),#167 ;IS IT TEST 167?
12501 046450 001014 BNE TST170 ;IF NOT THEN TRY TEST 170
12502 046452 005037 042340 CLR @#TEMP1
12503 046456 005337 042342 DEC @#TEMP2 ;0 -1
12504 046462 012737 000020 042344 MOV #16,@#TEMP3 ;SHIFTED BY 16.
12505 046470 005037 042350 CLR @#TEMP5 ;IS EQUAL TO -1 0
12506 046474 005237 042352 INC @#TEMP6 ;AND PS=12
12507 046500 000207 RTS %7
12508 046502 021527 000170 TST170: CMP (R5),#170 ;IS IT TEST 170?
12509 046506 001007 BNE TST171 ;IF NOT THEN TRY TEST 171
12510 046510 012737 125252 042342 MOV #125252,@#TEMP2 ;0 125252 SHIFTED BY 16
12511 046516 012737 125252 042346 MOV #125252,@#TEMP4 ;IS EQUAL TO 125252 0, AND PS=12
12512 046524 000207 RTS %7
12513 046526 021527 000171 TST171: CMP (R5),#171 ;IS IT TEST 171?
12514 046532 001010 BNE TST172 ;IF NOT THEN TRY TEST 172
12515 046534 005337 042344 DEC @#TEMP3 ;0 125252 SHIFTED BY 15
12516 046540 012737 052525 042346 MOV #52525,@#TEMP4 ;IS EQUAL TO 52525 0
12517 046546 005037 042352 CLR @#TEMP6 ;AND PS=0
12518 046552 000207 RTS %7
12519 046554 021527 000172 TST172: CMP (R5),#172 ;IS IT TEST 172?
12520 046560 001006 BNE TST173 ;IF NOT THEN TRY TEST 173
12521 046562 012737 052525 042342 MOV #52525,@#TEMP2 ;0 52525
12522 046570 005237 042344 INC @#TEMP3 ;SHIFTED BY 16. IS EQUAL TO 52525 0, AND PS=0
12523 046574 000207 RTS %7
12524 046576 021527 000173 TST173: CMP (R5),#173 ;IS IT TEST 173?
12525 046602 001014 BNE TST174 ;IF NOT THEN TRY TEST 174
12526 046604 012737 177777 042342 MOV #-1,@#TEMP2 ;0 -1
12527 046612 005337 042344 DEC @#TEMP3 ;SHIFTED BY 15.
12528 046616 012737 077777 042346 MOV #77777,@#TEMP4
12529 046624 012737 100000 042350 MOV #100000,@#TEMP5 ;IS EQUAL TO 77777 100000, AND PS=0
12530 046632 000207 RTS %7
12531 046634 021527 000174 TST174: CMP (R5),#174 ;IS IT TEST 174?
12532 046640 001013 BNE TST175 ;IF NOT THEN TRY TEST 175
12533 046642 012737 100000 042340 MOV #100000,@#TEMP1
12534 046650 005337 042342 DEC @#TEMP2 ;100000 -2 SHIFTED BY 15.
12535 046654 005037 042350 CLR @#TEMP5 ;IS EQUAL TO 77777 0
12536 046660 012737 000002 042352 MOV #2,@#TEMP6 ;AND PS-2
12537 046666 000207 RTS %7
12538 046670 021527 000175 TST175: CMP (R5),#175 ;IS IT TEST 175?
12539 046674 001015 BNE ENT176 ;IF NOT THEN TRY TEST 176
12540 046676 012737 177777 042340 MOV #-1,@#TEMP1
12541 046704 005037 042342 CLR @#TEMP2 ;-1 0
12542 046710 005237 042344 INC @#TEMP3 ;SHIFTED BY 16.
12543 046714 005037 042346 CLR @#TEMP4 ;IS EQUAL TO 0 0
12544 046720 012737 000007 042352 MOV #7,@#TEMP6 ;AND PS=7
12545 046726 000207 RTS %7
12546 046730 021527 000176 ENT176: CMP (R5),#176 ;IS THE PROGRAM ENTERING TEST 176?
12547 046734 001403 BEQ .+10
12548 046736 004767 011650 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12549 ;TEST NUMBER GOOFED
12550 046742 000056 56 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12551 ;BY (013746 000172 000207)
```

```
12552
12553
12554 046744 005726          TST      (SP)+          ;RESTORE STACK POINTER
12555
12556 :*****
12557 :TEST:176          1 SHIFTED BY 8. = 400 PS = 0
12558 :*****
12559
12560 046746 010767 131220    TST176: MOV      PC,LPADR          ;STORE ERROR LOOP ADDRESS
12561 046752 012701 000000    MOV      #DUMMY,%1          ;LOAD R1 WITH DUMMY
12562 046756 012701 000001    MOV      #1,%1!1          ;LOAD R1!1 WITH 1
12563 046762 000241          CLC
12564 046764 073127 000010    ASHC     #8,%1          ;SHIFT R1,R1!1 BY 8.
12565 046770 106737 042336    MFPS     @#PSWORD          ;SAVE PS
12566 046774 122737 000000    CMPB     #0,@#PSWORD        ;IS THE PS 0?
12567 047002 001403          BEQ      .+10
12568 047004 004767 011602    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12569                                     ;THE PS IS NOT EQUAL TO 0
12570 047010 000057          57          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12571                                     ;BY (013746 000172 000207)
12572
12573 047012 022701 000400    CMP      #400,%1          ;IS THE RESULT 400?
12574 047016 001403          BEQ      .+10
12575 047020 004767 011566    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12576                                     ;R1 IS NOT EQUAL TO 400
12577 047024 000060          60          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12578                                     ;BY (013746 000172 000207)
12579
12580 047026 021527 000176    CMP      (R5),#176          ;IS $TESTN = #176?
12581 047032 001403          BEQ      .+10          ;IF NOT THEN GO TO HLT
12582 047034 004767 011552    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12583                                     ;TEST IS IN WRONG SEQUENCE
12584 047040 000061          61          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12585                                     ;BY (013746 000172 000207)
12586
12587 047042 005215          INC      (R5)
12588
12589
12590 :*****
12591 :TEST:177          -1 SHIFTED BY 15. = 100000 PS = 11
12592 :*****
12593
12594 047044 010767 131122    TST177: MOV      PC,LPADR          ;STORE ERROR LOOP ADDRESS
12595 047050 012703 000000    MOV      #DUMMY,%3          ;LOAD R3 WITH DUMMY
12596 047054 012703 177777    MOV      #-1,%3!1          ;LOAD R3!1 WITH -1
12597 047060 000241          CLC
12598 047062 073327 000017    ASHC     #15,%3          ;SHIFT R3,R3!1 BY 15.
12599 047066 106737 042336    MFPS     @#PSWORD          ;SAVE PS
12600 047072 122737 000011    CMPB     #11,@#PSWORD        ;IS THE PS 11?
12601 047100 001403          BEQ      .+10
12602 047102 004767 011504    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12603                                     ;THE PS IS NOT EQUAL TO 11
12604 047106 000062          62          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12605                                     ;BY (013746 000172 000207)
12606
12607 047110 022703 100000    CMP      #100000,%3          ;IS THE RESULT 100000?
```

```
12608 047114 001403      BEQ      .+10
12609 047116 004767 011470  JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12610                                     ;R3 IS NOT EQUAL TO 100000
12611 047122 000063      63      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12612                                     ;BY (013746 000172 000207)
12613
12614 047124 021527 000177  CMP      (R5),#177      ;IS $TESTN - #177?
12615 047130 001403      BEQ      .+10      ;IF NOT THEN GO TO HLT
12616 047132 004767 011454  JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12617                                     ;TEST IS IN WRONG SEQUENCE
12618 047136 000064      64      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12619                                     ;BY (013746 000172 000207)
12620
12621 047140 005215      INC      (R5)
12622
12623
12624
12625 ;*****
12626 ;TEST:200      52525 SHIFTED BY 0 - 52525 PS = 0
12627 ;*****
12628 047142 010767 131024  TST200: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
12629 047146 010501      MOV    R5,R1      ;SAVE R5
12630 047150 012705 000000  MOV    #DUMMY,%5      ;LOAD R5 WITH DUMMY
12631 047154 012705 052525  MOV    #52525,%5!1      ;LOAD R5!1 WITH 52525
12632 047160 000241      CLC
12633 047162 073527 000000  ASHC   #0,%5      ;SHIFT R5,R5!1 BY 0
12634 047166 106737 042336  MFPS   @#PSWORD      ;SAVE PS
12635 047172 122737 000000 042336  CMPB   #0,@#PSWORD      ;IS THE PS 0?
12636 047200 001403      BEQ    .+10
12637 047202 004767 011404  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12638                                     ;THE PS IS NOT EQUAL TO 0
12639 047206 000065      65      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12640                                     ;BY (013746 000172 000207)
12641
12642 047210 022705 052525  CMP    #52525,%5      ;IS THE RESULT 52525?
12643 047214 001403      BEQ    .+10
12644 047216 004767 011370  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12645                                     ;R5 IS NOT EQUAL TO 52525
12646 047222 000066      66      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12647                                     ;BY (013746 000172 000207)
12648
12649 047224 010105      MOV    R1,R5      ;RESTORE R5
12650 047226 021527 000200  CMP    (R5),#200      ;IS $TESTN #200?
12651 047232 001403      BEQ    .+10      ;IF NOT THEN GO TO HLT
12652 047234 004767 011352  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12653                                     ;TEST IS IN WRONG SEQUENCE
12654 047240 000067      67      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12655                                     ;BY (013746 000172 000207)
12656
12657 047242 005215      INC    (R5)
12658
12659
12660 ;*****
12661 ;TEST:201      20010 SHIFTED BY -13. = 101 PS = 0
12662 ;*****
12663
```

```
12664 047244 010767 130722      TST201: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
12665 047250 012701 000000      MOV      #DUMMY,%1          ;LOAD R1 WITH DUMMY
12666 047254 012701 020010      MOV      #20010,%1.1        ;LOAD R1!1 WITH 20010
12667 047260 000241                CLC
12668 047262 073127 177763      ASHC     #-13,%1            ;SHIFT R1,R1!1 BY -13.
12669 047266 106737 042336      MFPS     @#PSWORD           ;SAVE PS
12670 047272 122737 000000 042336  CMPB     #0,@#PSWORD        ;IS THE PS 0?
12671 047300 001403                BEQ     .+10
12672 047302 004767 011304      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12673                                ;THE PS IS NOT EQUAL TO 0
12674 047306 000070                70      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12675                                ;BY (013746 000172 000207)
12676
12677 047310 022701 000101      CMP     #101,%1            ;IS THE RESULT 101?
12678 047314 001403                BEQ     .+10
12679 047316 004767 011270      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12680                                ;R1 IS NOT EQUAL TO 101
12681 047322 000071                71      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12682                                ;BY (013746 000172 000207)
12683
12684 047324 021527 000201      CMP     (R5),#201          ;IS $TESTN = #201?
12685 047330 001403                BEQ     .+10                ;IF NOT THEN GO TO HLT
12686 047332 004767 011254      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12687                                ;TEST IS IN WRONG SEQUENCE
12688 047336 000072                72      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12689                                ;BY (013746 000172 000207)
12690
12691 047340 005215                INC     (R5)
12692
12693
12694
12695 :*****
12696 :TEST:202      -1 SHIFTED BY 16.  0 PS - 11
12697 :*****
12698 047342 010767 130624      TST202: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
12699 047346 012703 000000      MOV      #DUMMY,%3          ;LOAD R3 WITH DUMMY
12700 047352 012703 177777      MOV      #-1,%3!1          ;LOAD R3!1 WITH -1
12701 047356 000241                CLC
12702 047360 073327 000020      ASHC     #16,%3            ;SHIFT R3,R3!1 BY 16.
12703 047364 106737 042336      MFPS     @#PSWORD           ;SAVE PS
12704 047370 122737 000011 042336  CMPB     #11,@#PSWORD        ;IS THE PS 11?
12705 047376 001403                BEQ     .+10
12706 047400 004767 011206      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12707                                ;THE PS IS NOT EQUAL TO 11
12708 047404 000073                73      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12709                                ;BY (013746 000172 000207)
12710
12711 047406 022703 000000      CMP     #0,%3            ;IS THE RESULT 0?
12712 047412 001403                BEQ     .+10
12713 047414 004767 011172      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12714                                ;R3 IS NOT EQUAL TO 0
12715 047420 000074                74      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12716                                ;BY (013746 000172 000207)
12717
12718 047422 021527 000202      CMP     (R5),#202          ;IS $TESTN = #202?
12719 047426 001403                BEQ     .+10                ;IF NOT THEN GO TO HLT
```

```
12720 047430 004767 011156 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12721 :TEST IS IN WRONG SEQUENCE
12722 047434 000075 75 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403),
12723 ;BY (013746 000172 000207)
12724
12725 047436 005215 INC (R5)
12726
12727
12728 :*****
12729 :TEST:203 1 SHIFTED BY -1 = 100000 PS 1
12730 :*****
12731
12732 047440 010767 130526 TST203: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12733 047444 010501 MOV R5,R1 ;SAVE R5
12734 047446 012705 000000 MOV #DUMMY,%5 ;LOAD R5 WITH DUMMY
12735 047452 012705 000001 MOV #1,%5!1 ;LOAD R5!1 WITH 1
12736 047456 000241 CLC
12737 047460 073527 177777 ASHC #-1,%5 ;SHIFT R5,R5.1 BY -1
12738 047464 106737 042336 MFPS @#PSWORD ;SAVE PS
12739 047470 122737 000001 042336 CMPB #1,@#PSWORD ;IS THE PS 1?
12740 047476 001403 BEQ .+10
12741 047500 004767 011106 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12742 ;THE PS IS NOT EQUAL TO 1
12743 047504 000076 76 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12744 ;BY (013746 000172 000207)
12745
12746 047506 022705 100000 CMP #100000,%5 ;IS THE RESULT 100000?
12747 047512 001403 BEQ .+10
12748 047514 004767 011072 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12749 ;R5 IS NOT EQUAL TO 100000
12750 047520 000077 77 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12751 ;BY (013746 000172 000207)
12752
12753 047522 010105 MOV R1,R5 ;RESTORE R5
12754 047524 021527 000203 CMP (R5),#203 ;IS $TESTN - #203?
12755 047530 001403 BEQ .+10 ;IF NOT THEN GO TO HLT
12756 047532 004767 011054 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12757 ;TEST IS IN WRONG SEQUENCE
12758 047536 000100 100 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12759 ;BY (013746 000172 000207)
12760
12761 047540 005215 INC (R5)
12762
12763
12764 :*****
12765 :TEST:204 125252 SHIFTED BY -16. = 125252 PS 11
12766 :*****
12767
12768 047542 010767 130424 TST204: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12769 047546 012701 000000 MOV #DUMMY,%1 ;LOAD R1 WITH DUMMY
12770 047552 012701 125252 MOV #125252,%1.1 ;LOAD R1!1 WITH 125252
12771 047556 000241 CLC
12772 047560 073127 177760 ASHC #-16.,%1 ;SHIFT R1,R1.1 BY -16.
12773 047564 106737 042336 MFPS @#PSWORD ;SAVE PS
12774 047570 122737 000011 042336 CMPB #11,@#PSWORD ;IS THE PS 11?
12775 047576 001403 BEQ .+10
```

```
12776 047600 004767 011006 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12777 ;THE PS IS NOT EQUAL TO 11
12778 047604 000101 101 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12779 ;BY (013746 000172 000207)
12780
12781 047606 022701 125252 CMP #125252,%1 ;IS THE RESULT 125252?
12782 047612 001403 BEQ .+10
12783 047614 004767 010772 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12784 ;R1 IS NOT EQUAL TO 125252
12785 047620 000102 102 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12786 ;BY (013746 000172 000207)
12787
12788 047622 021527 000204 CMP (R5),#204 ;IS $TESTN = #204?
12789 047626 001403 BEQ .+10 ;IF NOT THEN GO TO HLT
12790 047630 004767 010756 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12791 ;TEST IS IN WRONG SEQUENCE
12792 047634 000103 103 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12793 ;BY (013746 000172 000207)
12794
12795 047636 005215 INC (R5)
12796
12797
12798
12799 ;*****
12800 ;TEST:205 125252 125252 SHIFTED BY 21. = 52500 000000 PS 3
12801 ;*****
12802 047640 010767 130326 TST205: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
12803 047644 012702 125252 MOV #125252,%2 ;LOAD R2 WITH 125252
12804 047650 012703 125252 MOV #125252,%2:1 ;LOAD R2:1 WITH 125252
12805 047654 000241 CLC
12806 047656 073227 000025 ASHC #21,%2 ;SHIFT R2,R2.1 BY 21.
12807 047662 106737 042336 MFPS @#PSWORD ;SAVE PS
12808 047666 122737 000003 042336 CMPB #3,@#PSWORD ;IS THE PS 3?
12809 047674 001403 BEQ .+10
12810 047676 004767 010710 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12811 ;THE PS IS NOT EQUAL TO 3
12812 047702 000104 104 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12813 ;BY (013746 000172 000207)
12814
12815 047704 022702 052500 CMP #52500,%2 ;IS THE RESULT 52500?
12816 047710 001403 BEQ .+10
12817 047712 004767 010674 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12818 ;R2 IS NOT EQUAL TO 52500
12819 047716 000105 105 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12820 ;BY (013746 000172 000207)
12821
12822 047720 022703 000000 CMP #000000,%2.1 ;IS THE RESULT 000000?
12823 047724 001403 BEQ .+10
12824 047726 004767 010660 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12825 ;R2.1 IS NOT EQUAL TO 000000
12826 047732 000106 106 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12827 ;BY (013746 000172 000207)
12828
12829 047734 021527 000205 CMP (R5),#205 ;IS $TESTN - #205?
12830 047740 001403 BEQ .+10 ;IF NOT THEN GO TO HLT
12831 047742 004767 010644 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
```

```
12832                                     ;TEST IS IN WRONG SEQUENCE
12833 047746 000107                       107      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12834                                     ;BY (013746 000172 000207)
12835
12836 047750 005215                       INC      (R5)
12837
12838
12839
12840 047752 012702 177771                 MOV      #-7,%2
12841 047756 012703 042360                 MOV      #S1,%3
12842 047762 012704 042362                 MOV      #S2,%4
12843
12844                                     ;*****
12845                                     ;TEST:206      125252 125252 SHIFTED BY S1 177525 52525 PS = 10
12846                                     ;*****
12847
12848 047766 010767 130200                 TST206: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
12849 047772 012700 125252                 MOV      #125252,%0        ;LOAD R0 WITH 125252
12850 047776 012701 125252                 MOV      #125252,%0.1     ;LOAD R0!1 WITH 125252
12851 050002 000241                         CLC
12852 050004 073067 172350                 ASHC    S1,%0             ;SHIFT R0,R0.1 BY S1
12853 050010 106737 042336                 MFPS    @#PSWORD         ;SAVE PS
12854 050014 122737 000010 042336         CMPB    #10,@#PSWORD     ;IS THE PS 10?
12855 050022 001403                         BEQ     .+10
12856 050024 004767 010562                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12857                                     ;THE PS IS NOT EQUAL TO 10
12858 050030 000110                       110      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12859                                     ;BY (013746 000172 000207)
12860
12861 050032 022700 177525                 CMP     #177525,%0       ;IS THE RESULT 177525?
12862 050036 001403                         BEQ     .+10
12863 050040 004767 010546                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12864                                     ;R0 IS NOT EQUAL TO 177525
12865 050044 000111                       111      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12866                                     ;BY (013746 000172 000207)
12867
12868 050046 022701 052525                 CMP     #52525,%0.1     ;IS THE RESULT 52525?
12869 050052 001403                         BEQ     .+10
12870 050054                                     1$:
12871 050054 004767 010532                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12872                                     ;R0.1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12873 050060 000112                       112      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12874                                     ;BY (013746 000172 000207)
12875
12876 050062 021527 000206                 CMP     (R5),#206        ;IS THE $TESTN - #206?
12877 050066 001372                         BNE     1$              ;IF NOT THEN GO TO HLT ABOVE
12878 050070 005215                       INC     (R5)
12879
12880
12881                                     ;*****
12882                                     ;TEST:207      125252 125252 SHIFTED BY @S2 = 177525 52525 PS = 10
12883                                     ;*****
12884
12885 050072 010767 130074                 TST207: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
12886 050076 012700 125252                 MOV      #125252,%0        ;LOAD R0 WITH 125252
12887 050102 012701 125252                 MOV      #125252,%0.1     ;LOAD R0!1 WITH 125252
```

```

12888 050106 000241          CLC
12889 050110 073077 172246  ASHC @S2,%0          ;SHIFT R0,R0.1 BY @S2
12890 050114 106737 042336  MFPS @#PSWORD       ;SAVE PS
12891 050120 122737 000010 042336  CMPB #10,@#PSWORD   ;IS THE PS 10?
12892 050126 001403          BEQ .+10
12893 050130 004767 010456  JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12894                                ;THE PS IS NOT EQUAL TO 10
12895 050134 000113          113          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12896                                ;BY (013746 000172 000207)
12897
12898 050136 022700 177525  CMP #177525,%0      ;IS THE RESULT 177525?
12899 050142 001403          BEQ .+10
12900 050144 004767 010442  JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12901                                ;RO IS NOT EQUAL TO 177525
12902 050150 000114          114          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12903                                ;BY (013746 000172 000207)
12904
12905 050152 022701 052525  CMP #52525,%0.1    ;IS THE RESULT 52525?
12906 050156 001403          BEQ .+10
12907 050160          1$:
12908 050160 004767 010426  JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12909                                ;RO.1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
12910 050164 000115          115          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12911                                ;BY (013746 000172 000207)
12912
12913 050166 021527 000207  CMP (R5),#207      ;IS THE $TESTN - #207?
12914 050172 001372          BNE 1$             ;IF NOT THEN GO TO HLT ABOVE
12915 050174 005215          INC (R5)
12916
12917
12918
12919          ;*****
12920          ;TEST:210      125252 125252 SHIFTED BY @#S1 = 177525 52525 PS 10
12921          ;*****
12922 050176 010767 127770  TST210: MOV PC,LPADR      ;STORE ERROR LOOP ADDRESS
12923 050202 012700 125252  MOV #125252,%0      ;LOAD R0 WITH 125252
12924 050206 012701 125252  MOV #125252,%0!1    ;LOAD R0!1 WITH 125252
12925 050212 000241          CLC
12926 050214 073037 042360  ASHC @#S1,%0        ;SHIFT R0,R0!1 BY @#S1
12927 050220 106737 042336  MFPS @#PSWORD       ;SAVE PS
12928 050224 122737 000010 042336  CMPB #10,@#PSWORD   ;IS THE PS 10?
12929 050232 001403          BEQ .+10
12930 050234 004767 010352  JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12931                                ;THE PS IS NOT EQUAL TO 10
12932 050240 000116          116          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12933                                ;BY (013746 000172 000207)
12934
12935 050242 022700 177525  CMP #177525,%0      ;IS THE RESULT 177525?
12936 050246 001403          BEQ .+10
12937 050250 004767 010336  JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
12938                                ;RO IS NOT EQUAL TO 177525
12939 050254 000117          117          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
12940                                ;BY (013746 000172 000207)
12941
12942 050256 022701 052525  CMP #52525,%0.1    ;IS THE RESULT 52525?
12943 050262 001403          BEQ .+10

```


12944 050264
12945 050264 004767 010322
12946
12947 050270 000120
12948
12949
12950 050272 021527 000210
12951 050276 001372
12952 050300 005215

1\$:
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;RO.1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
120 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)
CMP (R5),#210 ;IS THE \$TESTN = #210?
BNE 1\$;IF NOT THEN GO TO HLT ABOVE
INC (R5)

12953
12954
12955

:*****
:TEST:211 125252 125252 SHIFTED BY (3) = 177525 52525 PS = 10
:*****

12956
12957
12958

12959 050302 010767 127664
12960 050306 012700 125252
12961 050312 012701 125252
12962 050316 000241
12963 050320 073013
12964 050322 106737 042336
12965 050326 122737 000010 042336
12966 050334 001403
12967 050336 004767 010250

TST211: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #125252,%0 ;LOAD R0 WITH 125252
MOV #125252,%0!1 ;LOAD R0!1 WITH 125252
CLC
ASHC (3),%0 ;SHIFT R0,R0!1 BY (3)
MFPS @#PSWORD ;SAVE PS
CMPB #10,@#PSWORD ;IS THE PS 10?
BEQ .+10
JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;THE PS IS NOT EQUAL TO 10
121 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

12968
12969 050342 000121
12970
12971

12972 050344 022700 177525
12973 050350 001403
12974 050352 004767 010234

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)

12975
12976 050356 000122
12977
12978

12979 050360 022701 052525
12980 050364 001403
12981 050366

CMP #52525,%0!1 ;IS THE RESULT 52525?
BEQ .+10

12982 050366 004767 010220
12983
12984 050372 000123
12985
12986

1\$:
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)

12987 050374 021527 000211
12988 050400 001372
12989 050402 005215

CMP (R5),#211 ;IS THE \$TESTN = #211?
BNE 1\$;IF NOT THEN GO TO HLT ABOVE
INC (R5)

12990
12991
12992

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

12993
12994
12995

12996 050404 010767 127562
12997 050410 012700 125252
12998 050414 012701 125252
12999 050420 000241

TST212: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #125252,%0 ;LOAD R0 WITH 125252
MOV #125252,%0!1 ;LOAD R0!1 WITH 125252
CLC

```
13000 050422 073023          ASHC    (3)+,%0          ;SHIFT R0,R0.1 BY (3)+
13001 050424 106737 042336  MFPS    @#PSWORD        ;SAVE PS
13002 050430 122737 000010 042336  CMPB   #10,@#PSWORD     ;IS THE PS 10?
13003 050436 001403          BEQ     .+10
13004 050440 004767 010146  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13005                                     ;THE PS IS NOT EQUAL TO 10
13006 050444 000124          124    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13007                                     ;BY (013746 000172 000207)
13008
13009 050446 022700 177525  CMP    #177525,%0      ;IS THE RESULT 177525?
13010 050452 001403          BEQ     .+10
13011 050454 004767 010132  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13012                                     ;R0 IS NOT EQUAL TO 177525
13013 050460 000125          125    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13014                                     ;BY (013746 000172 000207)
13015
13016 050462 022701 052525  CMP    #52525,%0!1     ;IS THE RESULT 52525?
13017 050466 001403          BEQ     .+10
13018 050470          1$:
13019 050470 004767 010116  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13020                                     ;R0!1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13021 050474 000126          126    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13022                                     ;BY (013746 000172 000207)
13023
13024 050476 021527 000212  CMP    (R5),#212       ;IS THE $TESTN - #212?
13025 050502 001372          BNE    1$              ;IF NOT THEN GO TO HLT ABOVE
13026 050504 005215          INC    (R5)
13027
13028
13029
13030          :*****
13031          :TEST:213      125252 125252 SHIFTED BY -(3) = 177525 52525 PS - 10
13032          :*****
13033 050506 010767 127460  TST213: MOV    PC,LPADR      ;STORE ERROR LOQP ADDRESS
13034 050512 012700 125252  MOV    #125252,%0      ;LOAD R0 WITH 125252
13035 050516 012701 125252  MOV    #125252,%0!1   ;LOAD R0!1 WITH 125252
13036 050522 000241          CLC
13037 050524 073043          ASHC   -(3),%0        ;SHIFT R0,R0!1 BY -(3)
13038 050526 106737 042336  MFPS   @#PSWORD        ;SAVE PS
13039 050532 122737 000010 042336  CMPB   #10,@#PSWORD     ;IS THE PS 10?
13040 050540 001403          BEQ     .+10
13041 050542 004767 010044  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13042                                     ;THE PS IS NOT EQUAL TO 10
13043 050546 000127          127    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13044                                     ;BY (013746 000172 000207)
13045
13046 050550 022700 177525  CMP    #177525,%0      ;IS THE RESULT 177525?
13047 050554 001403          BEQ     .+10
13048 050556 004767 010030  JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13049                                     ;R0 IS NOT EQUAL TO 177525
13050 050562 000130          130    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13051                                     ;BY (013746 000172 000207)
13052
13053 050564 022701 052525  CMP    #52525,%0!1     ;IS THE RESULT 52525?
13054 050570 001403          BEQ     .+10
13055 050572          1$:
```

13056 050572 004767 010014 JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13057 ;R0.1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13058 050576 000131 131 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13059 ;BY (013746 000172 000207)
13060
13061 050600 021527 000213 CMP (R5),#213 ;IS THE \$TESTN = #213?
13062 050604 001372 BNE 1\$;IF NOT THEN GO TO HLT ABOVE
13063 050606 005215 INC (R5)

13064
13065
13066
13067 :*****
13068 :TEST:214 125252 125252 SHIFTED BY 2(4) = 177252 125252 PS 11
13069 :*****

13070 050610 010767 127356 TST214: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13071 050614 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
13072 050620 012701 125252 MOV #125252,%0.1 ;LOAD R0.1 WITH 125252
13073 050624 000241 CLC
13074 050626 073064 000002 ASHC 2(4),%0 ;SHIFT R0,R0.1 BY 2(4)
13075 050632 106737 042336 MFPS @#PSWORD ;SAVE PS
13076 050636 122737 000011 042336 CMPS #11,@#PSWORD ;IS THE PS 11?
13077 050644 001403 BEQ .+10
13078 050646 004767 007740 JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13079 ;THE PS IS NOT EQUAL TO 11
13080 050652 000132 132 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13081 ;BY (013746 000172 000207)

13082
13083 050654 022700 177252 CMP #177252,%0 ;IS THE RESULT 177252?
13084 050660 001403 BEQ .+10
13085 050662 004767 007724 JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13086 ;R0 IS NOT EQUAL TO 177252
13087 050666 000133 133 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13088 ;BY (013746 000172 000207)

13089
13090 050670 022701 125252 CMP #125252,%0.1 ;IS THE RESULT 125252?
13091 050674 001403 BEQ .+10
13092 050676 004767 007710 1\$: JSR PC,\$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13093 ;R0.1 IS NOT EQUAL TO 125252 OR INCORRECT SEQUENCE
13094 050702 000134 134 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13095 ;BY (013746 000172 000207)

13096
13097
13098 050704 021527 000214 CMP (R5),#214 ;IS THE \$TESTN #214?
13099 050710 001372 BNE 1\$;IF NOT THEN GO TO HLT ABOVE
13100 050712 005215 INC (R5)

13101
13102
13103 :*****
13104 :TEST:215 125252 125252 SHIFTED BY @ (4) = 177525 52525 PS 10
13105 :*****

13106
13107 050714 010767 127252 TST215: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13108 050720 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
13109 050724 012701 125252 MOV #125252,%0.1 ;LOAD R0.1 WITH 125252
13110 050730 000241 CLC
13111 050732 073074 000000 ASHC @ (4),%0 ;SHIFT R0,R0.1 BY @ (4)

```

13112 050736 106737 042336 MFPS @#PSWORD ;SAVE PS
13113 050742 122737 000010 042336 CMPB #10,@#PSWORD ;IS THE PS 10?
13114 050750 001403 BEQ .+10
13115 050752 004767 007634 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13116 ;THE PS IS NOT EQUAL TO 10
13117 050756 000135 135 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13118 ;BY (013746 000172 000207)
13119
13120 050760 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
13121 050764 001403 BEQ .+10
13122 050766 004767 007620 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13123 ;RO IS NOT EQUAL TO 177525
13124 050772 000136 136 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13125 ;BY (013746 000172 000207)
13126
13127 050774 022701 052525 CMP #52525,%0.1 ;IS THE RESULT 52525?
13128 051000 001403 BEQ .+10
13129 051002 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13130 051002 004767 007604 ;R0.1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13131 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13132 051006 000137 137 ;BY (013746 000172 000207)
13133
13134
13135 051010 021527 000215 CMP (R5),#215 ;IS THE $TESTN - #215?
13136 051014 001372 BNE 1$ ;IF NOT THEN GO TO HLT ABOVE
13137 051016 005215 INC (R5)
13138
13139
13140
13141 ;*****
13142 ;TEST:216 125252 125252 SHIFTED BY @ (4)+ = 177525 52525 PS = 10
13143 ;*****
13144 051020 010767 127146 TST216: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13145 051024 012700 125252 MOV #125252,%0 ;LOAD R0 WITH 125252
13146 051030 012701 125252 MOV #125252,%0.1 ;LOAD R0.1 WITH 125252
13147 051034 000241 CLC
13148 051036 073034 ASHC @ (4)+,%0 ;SHIFT R0,R0.1 BY @ (4)+
13149 051040 106737 042336 MFPS @#PSWORD ;SAVE PS
13150 051044 122737 000010 042336 CMPB #10,@#PSWORD ;IS THE PS 10?
13151 051052 001403 BEQ .+10
13152 051054 004767 007532 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13153 ;THE PS IS NOT EQUAL TO 10
13154 051060 000140 140 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13155 ;BY (013746 000172 000207)
13156
13157 051062 022700 177525 CMP #177525,%0 ;IS THE RESULT 177525?
13158 051066 001403 BEQ .+10
13159 051070 004767 007516 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13160 ;RO IS NOT EQUAL TO 177525
13161 051074 000141 141 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13162 ;BY (013746 000172 000207)
13163
13164 051076 022701 052525 CMP #52525,%0.1 ;IS THE RESULT 52525?
13165 051102 001403 BEQ .+10
13166 051104 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13167 051104 004767 007502

```

```
13168                                     ;R0.1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13169 051110 000142                       142      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13170                                     ;BY (013746 000172 000207)
13171
13172 051112 021527 000216                 CMP      (R5),#216      ;IS THE $TESTN - #216?
13173 051116 001372                       BNE      1$            ;IF NOT THEN GO TO HLT ABOVE
13174 051120 005215                       INC      (R5)
13175
13176
13177                                     :*****
13178 :TEST:217      125252 125252 SHIFTED BY @-(4) = 177525 52525 PS 10
13179 :*****
13180
13181 051122 010767 127044                 TST217: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
13182 051126 012700 125252                 MOV      #125252,%0      ;LOAD R0 WITH 125252
13183 051132 012701 125252                 MOV      #125252,%0!1    ;LOAD R0.1 WITH 125252
13184 051136 000241                       CLC
13185 051140 073054                       ASHC     @-(4),%0        ;SHIFT R0,R0!1 BY @-(4)
13186 051142 106737 042336                 MFPS    @#PSWORD        ;SAVE PS
13187 051146 122737 000010 042336        CMPB    #10,@#PSWORD     ;IS THE PS 10?
13188 051154 001403                       BEQ     .+10
13189 051156 004767 007430                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13190                                     ;THE PS IS NOT EQUAL TO 10
13191 051162 000143                       143      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13192                                     ;BY (013746 000172 000207)
13193
13194 051164 022700 177525                 CMP      #177525,%0      ;IS THE RESULT 177525?
13195 051170 001403                       BEQ     .+10
13196 051172 004767 007414                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13197                                     ;R0 IS NOT EQUAL TO 177525
13198 051176 000144                       144      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13199                                     ;BY (013746 000172 000207)
13200
13201 051200 022701 052525                 CMP      #52525,%0!1    ;IS THE RESULT 52525?
13202 051204 001403                       BEQ     .+10
13203 051206                                     1$:
13204 051206 004767 007400                 JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13205                                     ;R0.1 IS NOT EQUAL TO 52525 OR INCORRECT SEQUENCE
13206 051212 000145                       145      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13207                                     ;BY (013746 000172 000207)
13208
13209 051214 021527 000217                 CMP      (R5),#217      ;IS THE $TESTN = #217?
13210 051220 001372                       BNE      1$            ;IF NOT THEN GO TO HLT ABOVE
13211 051222 005215                       INC      (R5)
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
13249
13250
13251
13252
13253
13254
13255
13256
13257
13258
13259
13260
13261
13262
13263
13264
13265
13266
13267
13268
13269
13270
13271
13272
13273
13274
13275
13276

: 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
1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;LOW ORDER IS WRONG OR WRONG SEQUENCE
;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP (R5),#220
BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
INC (R5)
```

042336
042336

: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)
```

042336

```
13277
13278 051360 022700 177777      CMP    #-1,%0      ;IS HIGH ORDER = -1
13279 051364 001403      BEQ    .+10
13280 051366 004767 007220      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13281                                     ;HIGH ORDER IS WRONG
13282 051372 000152      152    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13283                                     ;BY (013746 000172 000207)
13284
13285 051374 022701 177777      CMP    #-1,%0.1    ;IS LOW ORDER = -1
13286 051400 001403      BEQ    .+10
13287 051402      1$:
13288 051402 004767 007204      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13289                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13290 051406 000153      153    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13291                                     ;BY (013746 000172 000207)
13292
13293 051410 021527 000221      CMP    (R5),#221
13294 051414 001372      BNE    1$
13295 051416 005215      INC    (R5)      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13296
13297
13298
13299
13300      :*****
13301      :TEST:222      MUL    2 * #2 - 0 4      PS = 0
13302      :*****
13303      TST222: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
13304      MOV    #2,%2      ;LOAD MULTIPLICAND WITH 2
13305      MUL    #2,%2      ;MULTIPLY 2 * #2
13306      MFPS   @#PSWORD      ;SAVE PS
13307      CMPB  #0,@#PSWORD      ;IS PS = 0
13308      BEQ    .+10
13309      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13310      PS IS WRONG
13311      154    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13312      ;BY (013746 000172 000207)
13313 051456 022702 000000      CMP    #0,%2      ;IS HIGH ORDER = 0
13314 051462 001403      BEQ    .+10
13315 051464 004767 007122      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13316      ;HIGH ORDER IS WRONG
13317 051470 000155      155    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13318      ;BY (013746 000172 000207)
13319
13320 051472 022703 000004      CMP    #4,%2.1    ;IS LOW ORDER - 4
13321 051476 001403      BEQ    .+10
13322 051500      1$:
13323 051500 004767 007106      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13324      ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13325 051504 000156      156    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13326      ;BY (013746 000172 000207)
13327
13328 051506 021527 000222      CMP    (R5),#222
13329 051512 001372      BNE    1$
13330 051514 005215      INC    (R5)      ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13331
13332
```

```
13333
13334
13335
13336
13337 051516 010767 126450
13338 051522 010501
13339 051524 012704 001000
13340 051530 070427 000200
13341 051534 106737 042336
13342 051540 122737 000001 042336
13343 051546 001403
13344 051550 004767 007036
13345
13346 051554 000157
13347
13348
13349 051556 022704 000001
13350 051562 001403
13351 051564 004767 007022
13352
13353 051570 000160
13354
13355
13356 051572 022705 000000
13357 051576 001403
13358 051600
13359 051600 004767 007006
13360
13361 051604 000161
13362
13363
13364 051606 021127 000223
13365 051612 001372
13366 051614 010105
13367 051616 005215
13368
13369
13370
13371
13372
13373
13374 051620 010767 126346
13375 051624 012700 000002
13376 051630 070027 077777
13377 051634 106737 042336
13378 051640 122737 000001 042336
13379 051646 001403
13380 051650 004767 006736
13381
13382 051654 000162
13383
13384
13385 051656 022700 000000
13386 051662 001403
13387 051664 004767 006722
13388

;*****
;TEST:223      MUL      1000 * #200 = 1 0      PS = 1
;*****

TST223: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      R5,R1        ;SAVE R5
        MOV      #1000,%4     ;LOAD MULTIPLICAND WITH 1000
        MUL      #200,%4     ;MULTIPLY 1000 * #200
        MFPS     @#PSWORD     ;SAVE PS
        CMPB    #1,@#PSWORD   ;IS PS = 1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;PS IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP      #1,%4        ;IS HIGH ORDER = 1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;HIGH ORDER IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMF     #0,%4!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      (R1),#223     ;CHECK THE TEST NUMBER
        BNE     1$           ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
        MOV      R1,R5        ;RESTORE R5
        INC     (R5)

;*****
;TEST:224      MUL      2 * #77777 = 0 177776      PS - 1
;*****

TST224: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #2,%0        ;LOAD MULTIPLICAND WITH 2
        MUL      #77777,%0    ;MULTIPLY 2 * #77777
        MFPS     @#PSWORD     ;SAVE PS
        CMPB    #1,@#PSWORD   ;IS PS = 1
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;PS IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

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



```
13389 051670 000163          163          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13390                                ;BY (013746 000172 000207)
13391
13392 051672 022701 177776          CMP      #177776,%0 1      ;IS LOW ORDER = 177776
13393 051676 001403          BEQ      .+10
13394 051700
13395 051700 004767 006706          1$:      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13396                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13397 051704 000164          164          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13398                                ;BY (013746 000172 000207)
13399
13400 051706 021527 000224          CMP      (R5),#224
13401 051712 001372          BNE      1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13402 051714 005215          INC      (R5)
13403
13404
13405
13406                                ;*****
13407                                ;TEST:225      MUL      7777 * #10 = 0 77770      PS = 0
13408                                ;*****
13409 051716 010767 126250          TST225: MOV     PC,LPADR      ;STORE ERROR LOOP ADDRESS
13410 051722 012702 007777          MOV     #7777,%2          ;LOAD MULTIPLICAND WITH 7777
13411 051726 070227 000010          MUL     #10,%2           ;MULTIPLY 7777 * #10
13412 051732 106737 042336          MFPS   @#PSWORD         ;SAVE PS
13413 051736 122737 000000          CMPB   #0,@#PSWORD      ;IS PS 0
13414 051744 001403          BEQ     .+10
13415 051746 004767 006640          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13416                                ;PS IS WRONG
13417 051752 000165          165          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13418                                ;BY (013746 000172 000207)
13419
13420 051754 022702 000000          CMP     #0,%2           ;IS HIGH ORDER = 0
13421 051760 001403          BEQ     .+10
13422 051762 004767 006624          JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13423                                ;HIGH ORDER IS WRONG
13424 051766 000166          166          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13425                                ;BY (013746 000172 000207)
13426
13427 051770 022703 077770          CMP     #77770,%2!1     ;IS LOW ORDER = 77770
13428 051774 001403          BEQ     .+10
13429 051776
13430 051776 004767 006610          1$:      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13431                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13432 052002 000167          167          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13433                                ;BY (013746 000172 000207)
13434
13435 052004 021527 000225          CMP     (R5),#225
13436 052010 001372          BNE     1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13437 052012 005215          INC     (R5)
13438
13439
13440                                ;*****
13441                                ;TEST:226      MUL      77777 * #77777 37777 1      PS 1
13442                                ;*****
13443
13444 052014 0 767 126152          TST226: MOV     PC,LPADR      ;STORE ERROR LOOP ADDRESS
```

```
13445 052020 010501          MOV      R5,R1          ;SAVE R5
13446 052022 012704 077777    MOV      #77777,%4      ;LOAD MULTIPLICAND WITH 77777
13447 052026 070427 077777    MUL      #77777,%4      ;MULTIPLY 77777 * #77777
13448 052032 106737 042336    MFPS    @#PSWORD        ;SAVE PS
13449 052036 122737 000001 042336  CMPB    #1,@#PSWORD     ;IS PS = 1
13450 052044 001403          BEQ      .+10
13451 052046 004767 006540    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13452                                ;PS IS WRONG
13453 052052 000170          170          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13454                                ;BY (013746 000172 000207)
13455
13456 052054 022704 037777    CMP      #37777,%4      ;I HIGH ORDER = 37777
13457 052060 001403          BEQ      .+10
13458 052062 004767 006524    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13459                                ;HIGH ORDER IS WRONG
13460 052066 000171          171          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13461                                ;BY (013746 000172 000207)
13462
13463 052070 022705 000001    CMP      #1,%4.1        ;IS LOW ORDER = 1
13464 052074 001403          BEQ      .+10
13465 052076          1$:
13466 052076 004767 006510    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13467                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13468 052102 000172          172          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13469                                ;BY (013746 000172 000207)
13470
13471 052104 021127 000226    CMP      (R1),#226      ;CHECK THE TEST NUMBER
13472 052110 001372          BNE      1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13473 052112 010105          MOV      R1,R5          ;RESTORE R5
13474 052114 005215          INC      (R5)
13475
13476
13477
13478          ;*****
13479          ;TEST:227      MUL      -1 * #77777 - -1 100001      PS 10
13480          ;*****
13481 052116 010767 126050    TST227: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
13482 052122 012702 177777    MOV      #-1,%2        ;LOAD MULTIPLICAND WITH -1
13483 052126 070227 077777    MUL      #77777,%2      ;MULTIPLY -1 * #77777
13484 052132 106737 042336    MFPS    @#PSWORD        ;SAVE PS
13485 052136 122737 000010 042336  CMPB    #10,@#PSWORD    ;IS PS - 10
13486 052144 001403          BEQ      .+10
13487 052146 004767 006440    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13488                                ;PS IS WRONG
13489 052152 000173          173          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13490                                ;BY (013746 000172 000207)
13491
13492 052154 022702 177777    CMP      #-1,%2        ;IS HIGH ORDER = -1
13493 052160 001403          BEQ      .+10
13494 052162 004767 006424    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13495                                ;HIGH ORDER IS WRONG
13496 052166 000174          174          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13497                                ;BY (013746 000172 000207)
13498
13499 052170 022703 100001    CMP      #100001,%2.1    ;IS LOW ORDER = 100001
13500 052174 001403          BEQ      .+10
```

```
13501 052176  
13502 052176 004767 006410  
13503  
13504 052202 000175  
13505  
13506  
13507 052204 021527 000227  
13508 052210 001372  
13509 052212 005215  
13510  
13511  
13512  
13513  
13514  
13515  
13516 052214 010767 125752  
13517 052220 012700 177776  
13518 052224 070027 077777  
13519 052230 106737 042336  
13520 052234 122737 000011 042336  
13521 052242 001403  
13522 052244 004767 006342  
13523  
13524 052250 000176  
13525  
13526  
13527 052252 022700 177777  
13528 052256 001403  
13529 052260 004767 006326  
13530  
13531 052264 000177  
13532  
13533  
13534 052266 022701 000002  
13535 052272 001403  
13536 052274  
13537 052274 004767 006312  
13538  
13539 052300 000200  
13540  
13541  
13542 052302 021527 000230  
13543 052306 001372  
13544 052310 005215  
13545  
13546  
13547  
13548  
13549  
13550  
13551 052312 010767 125654  
13552 052316 012702 125252  
13553 052322 070227 000002  
13554 052326 106737 042336  
13555 052332 122737 000011 042336  
13556 052340 001403
```

```
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  
*****
```

```
TST230: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #-2,%0 ;LOAD MULTIPLICAND WITH -2  
MUL #77777,%0 ;MULTIPLY -2 * #77777  
MFPS @#PSWORD ;SAVE PS  
CMPB #11,@#PSWORD ;IS PS = 11  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;PS IS WRONG  
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)
```

```
1$: CMP #2,%0!1 ;IS LOW ORDER = 2  
BEQ .+10  
JSR PC,$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  
*****
```

```
TST231: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS  
MOV #125252,%2 ;LOAD MULTIPLICAND WITH 125252  
MUL #2,%2 ;MULTIPLY 125252 * #2  
MFPS @#PSWORD ;SAVE PS  
CMPB #11,@#PSWORD ;IS PS = 11  
BEQ .+10
```

```
13557 052342 004767 006244 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13558 :PS IS WRONG
13559 052346 000201 201 :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13560 :BY (013746 000172 000207)
13561
13562 052350 022702 177777 CMP #-1,%2 ;IS HIGH ORDER = -1
13563 052354 001403 BEQ .+10
13564 052356 004767 006230 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13565 :HIGH ORDER IS WRONG
13566 052362 000202 202 :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13567 :BY (013746 000172 000207)
13568
13569 052364 022703 052524 CMP #52524,%2!1 ;IS LOW ORDER = 52524
13570 052370 001403 BEQ .+10
13571 052372 1$:
13572 052372 004767 006214 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13573 :LOW ORDER IS WRONG OR WRONG SEQUENCE
13574 052376 000203 203 :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13575 :BY (013746 000172 000207)
13576
13577 052400 021527 000231 CMP (R5),#231
13578 052404 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13579 052406 005215 INC (R5)
13580
13581
13582 :*****
13583 :TEST:232 MUL 125252 * #40000 - 165252 100000 PS 11
13584 :*****
13585
13586 052410 010767 125556 TST232: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13587 052414 010501 MOV R5,R1 ;SAVE R5
13588 052416 012704 125252 MOV #125252,%4 ;LOAD MULTIPLICAND WITH 125252
13589 052422 070427 040000 MUL #40000,%4 ;MULTIPLY 125252 * #40000
13590 052426 106737 042336 MFPS @#PSWORD ;SAVE PS
13591 052432 122737 000011 042336 CMPB #11,@#PSWORD ;IS PS = 11
13592 052440 001403 BEQ .+10
13593 052442 004767 006144 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13594 :PS IS WRONG
13595 052446 000204 204 :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13596 :BY (013746 000172 000207)
13597
13598 052450 022704 165252 CMP #165252,%4 ;IS HIGH ORDER = 165252
13599 052454 001403 BEQ .+10
13600 052456 004767 006130 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13601 :HIGH ORDER IS WRONG
13602 052462 000205 205 :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13603 :BY (013746 000172 000207)
13604
13605 052464 022705 100000 CMP #100000,%4.1 ;IS LOW ORDER = 100000
13606 052470 001403 BEQ .+10
13607 052472 1$:
13608 052472 004767 006114 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13609 :LOW ORDER IS WRONG OR WRONG SEQUENCE
13610 052476 000206 206 :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13611 :BY (013746 000172 000207)
13612
```

```
13613 052500 021127 000232      CMP      (R1),#232      ;CHECK THE TEST NUMBER
13614 052504 001372      BNE      1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13615 052506 010105      MOV      R1,R5        ;RESTORE R5
13616 052510 005215      INC      (R5)
13617
13618
13619
13620      ;*****
13620      ;TEST:233      MUL      107070 * #107070 = 31222 26100      PS 1
13621      ;*****
13622
13623 052512 010767 125454      TST233: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
13624 052516 012700 107070      MOV      #107070,%0      ;LOAD MULTIPLICAND WITH 107070
13625 052522 070027 107070      MUL      #107070,%0      ;MULTIPLY 107070 * #107070
13626 052526 106737 042336      MFPS     @#PSWORD        ;SAVE PS
13627 052532 122737 000001 042336      CMPB     #1,@#PSWORD      ;IS PS = 1
13628 052540 001403      BEQ      .+10
13629 052542 004767 006044      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13630
13631 052546 000207      207      ;PS IS WRONG
13632      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13633      ;BY (013746 000172 000207)
13634 052550 022700 031222      CMP      #31222,%0      ;IS HIGH ORDER = 31222
13635 052554 001403      BEQ      .+10
13636 052556 004767 006030      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13637      ;HIGH ORDER IS WRONG
13638 052562 000210      210      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13639      ;BY (013746 000172 000207)
13640
13641 052564 022701 026100      CMP      #26100,%0.1    ;IS LOW ORDER = 26100
13642 052570 001403      BEQ      .+10
13643 052572      1$:
13644 052572 004767 006014      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13645      ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13646 052576 000211      211      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13647      ;BY (013746 000172 000207)
13648
13649 052600 021527 000233      CMP      (R5),#233
13650 052604 001372      BNE      1$            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13651 052606 005215      INC      (R5)
13652
13653
13654      ;*****
13655      ;TEST:234      MUL      -1 * #1 -1 -1      PS 10
13656      ;*****
13657
13658 052610 010767 125356      TST234: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
13659 052614 012701 177777      MOV      #-1,%1         ;LOAD MULTIPLICAND WITH -1
13660 052620 070127 000001      MUL      #1,%1         ;MULTIPLY -1 * #1
13661 052624 106737 042336      MFPS     @#PSWORD        ;SAVE PS
13662 052630 122737 000010 042336      CMPB     #10,@#PSWORD    ;IS PS - 10
13663 052636 001403      BEQ      .+10
13664 052640 004767 005746      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13665      ;PS IS WRONG
13666 052644 000212      212      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13667      ;BY (013746 000172 000207)
13668
```

```
13669 052646 022701 177777      CMP    #-1,%1          ;IS HIGH ORDER = -1
13670 052652 001403      BEQ    .+10
13671 052654 004767 005732      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13672                                ;HIGH ORDER IS WRONG
13673 052660 000213      213    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13674                                ;BY (013746 000172 000207)
13675
13676 052662 022701 177777      CMP    #-1,%1.1        ;IS LOW ORDER = -1
13677 052666 001403      BEQ    .+10
13678 052670      1$:
13679 052670 004767 005716      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13680                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13681 052674 000214      214    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13682                                ;BY (013746 000172 000207)
13683
13684 052676 021527 000234      CMP    (R5),#234
13685 052702 001372      BNE    1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13686 052704 005215      INC    (R5)
13687
13688
13689
13690      ;*****
13691      ;TEST:235      MUL    -1 * #0 - 0 0      PS 4
13692      ;*****
13693 052706 010767 125260      TST235: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
13694 052712 012703 177777      MOV    #-1,%3          ;LOAD MULTIPLICAND WITH -1
13695 052716 070327 000000      MUL    #0,%3           ;MULTIPLY -1 * #0
13696 052722 106737 042336      MFPS   @#PSWORD        ;SAVE PS
13697 052726 122737 000004 042336      CMPB   #4,@#PSWORD     ;IS PS - 4
13698 052734 001403      BEQ    .+10
13699 052736 004767 005650      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13700                                ;PS IS WRONG
13701 052742 000215      215    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13702                                ;BY (013746 000172 000207)
13703
13704 052744 022703 000000      CMP    #0,%3           ;IS HIGH ORDER = 0
13705 052750 001403      BEQ    .+10
13706 052752 004767 005634      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13707                                ;HIGH ORDER IS WRONG
13708 052756 000216      216    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13709                                ;BY (013746 000172 000207)
13710
13711 052760 022703 000000      CMP    #0,%3.1         ;IS LOW ORDER - 0
13712 052764 001403      BEQ    .+10
13713 052766      1$:
13714 052766 004767 005620      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13715                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13716 052772 000217      217    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13717                                ;BY (013746 000172 000207)
13718
13719 052774 021527 000235      CMP    (R5),#235
13720 053000 001372      BNE    1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13721 053002 005215      INC    (R5)
13722
13723
13724      ;*****
```

```
13725 :TEST:236 MUL 77777 * #100000 = 100000 100000 PS = 11
13726 :*****
13727
13728 053004 010767 125162 TST236: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13729 053010 010501 MOV R5,R1 ;SAVE R5
13730 053012 012705 077777 MOV #77777,%5 ;LOAD MULTIPLICAND WITH 77777
13731 053016 070527 100000 MUL #100000,%5 ;MULTIPLY 77777 * #100000
13732 053022 106737 042336 MFPS @#PSWORD ;SAVE PS
13733 053026 122737 000011 042336 CMPB #11,@#PSWORD ;IS PS = 11
13734 053034 001403 BEQ .+10
13735 053036 004767 005550 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13736 ;PS IS WRONG
13737 053042 000220 220 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13738 ;BY (013746 000172 000207)
13739
13740 053044 022705 100000 CMP #100000,%5 ;IS HIGH ORDER = 100000
13741 053050 001403 BEQ .+10
13742 053052 004767 005534 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13743 ;HIGH ORDER IS WRONG
13744 053056 000221 221 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13745 ;BY (013746 000172 000207)
13746
13747 053060 022705 100000 CMP #100000,%5:1 ;IS LOW ORDER = 100000
13748 053064 001403 BEQ .+10
13749 053066 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13750 053066 004767 005520 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13751 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13752 053072 000222 222 ;BY (013746 000172 000207)
13753
13754
13755 053074 021127 000236 CMP (R1),#236 ;CHECK THE TEST NUMBER
13756 053100 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13757 053102 010105 MOV R1,R5 ;RESTORE R5
13758 053104 005215 INC (R5)
13759
13760
13761 :*****
13762 :TEST:237 MUL -1 * #77777 = 100001 100001 PS - 10
13763 :*****
13764
13765 053106 010767 125060 TST237: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13766 053112 012701 177777 MOV #-1,%1 ;LOAD MULTIPLICAND WITH -1
13767 053116 070127 077777 MUL #77777,%1 ;MULTIPLY -1 * #77777
13768 053122 106737 042336 MFPS @#PSWORD ;SAVE PS
13769 053126 122737 000010 042336 CMPB #10,@#PSWORD ;IS PS = 10
13770 053134 001403 BEQ .+10
13771 053136 004767 005450 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13772 ;PS IS WRONG
13773 053142 000223 223 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13774 ;BY (013746 000172 000207)
13775
13776 053144 022701 100001 CMP #100001,%1 ;IS HIGH ORDER = 100001
13777 053150 001403 BEQ .+10
13778 053152 004767 005434 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13779 ;HIGH ORDER IS WRONG
13780 053156 000224 224 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
```

```
13781                                     ;BY (013746 000172 000207)
13782
13783 053160 022701 100001             CMP    #100001,%1!1    ;IS LOW ORDER = 100001
13784 053164 001403                   BEQ    .+10
13785 053166                           1$:
13786 053166 004767 005420             JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13787                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13788 053172 000225                   225    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13789                                     ;BY (013746 000172 000207)
13790
13791 053174 021527 000237             CMP    (R5),#237
13792 053200 001372                   BNE    1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13793 053202 005215                   INC    (R5)
13794
13795
13796                                     ;*****
13797 :TEST:240      MUL      77777 * #77777 - 1 1      PS - 1
13798                                     ;*****
13799
13800 053204 010767 124762             TST240: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
13801 053210 012703 077777             MOV    #77777,%3      ;LOAD MULTIPLICAND WITH 77777
13802 053214 070327 077777             MUL    #77777,%3      ;MULTIPLY 77777 * #77777
13803 053220 106737 042336             MFPS   @#PSWORD      ;SAVE PS
13804 053224 122737 000001 042336     CMPB   #1,@#PSWORD    ;IS PS = 1
13805 053232 001403                   BEQ    .+10
13806 053234 004767 005352             JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13807                                     ;PS IS WRONG
13808 053240 000226                   226    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13809                                     ;BY (013746 000172 000207)
13810
13811 053242 022703 000001             CMP    #1,%3          ;IS HIGH ORDER - 1
13812 053246 001403                   BEQ    .+10
13813 053250 004767 005336             JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13814                                     ;HIGH ORDER IS WRONG
13815 053254 000227                   227    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13816                                     ;BY (013746 000172 000207)
13817
13818 053256 022703 000001             CMP    #1,%3!1       ;IS LOW ORDER = 1
13819 053262 001403                   BEQ    .+10
13820 053264                           1$:
13821 053264 004767 005322             JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13822                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13823 053270 000230                   230    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13824                                     ;BY (013746 000172 000207)
13825
13826 053272 021527 000240             CMP    (R5),#240
13827 053276 001372                   BNE    1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13828 053300 005215                   INC    (R5)
13829
13830
13831                                     ;*****
13832 :TEST:241      MUL      2 * #2 = 4 4      PS 0
13833                                     ;*****
13834
13835 053302 010767 124664             TST241: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
13836 053306 010501                   MOV    R5,R1          ;SAVE R5
```



```

CJKDB-C DCF11-AA CPU DIAG. MACY11 30A(1052) 07-MAR-80 12:18 K 4 PAGE 255
CJKDBC.P11 07-MAR-80 12:17 MUL INSTRUCTION TESTS SEQ 0255

13837 053310 012705 000002 MOV #2,%5 ;LOAD MULTIPLICAND WITH 2
13838 053314 070527 000002 MUL #2,%5 ;MULTIPLY 2 * #2
13839 053320 106737 042336 MFPS @#PSWORD ;SAVE PS
13840 053324 122737 000000 042336 CMPB #0,@#PSWORD ;IS PS = 0
13841 053332 001403 BEQ .+10
13842 053334 004767 005252 JSP PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13843 ;PS IS WRONG
13844 053340 000231 231 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13845 ;BY (013746 000172 000207)
13846
13847 053342 022705 000004 CMP #4,%5 ;IS HIGH ORDER = 4
13848 053346 001403 BEQ .+10
13849 053350 004767 005236 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13850 ;HIGH ORDER IS WRONG
13851 053354 000232 232 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13852 ;BY (013746 000172 000207)
13853
13854 053356 022705 000004 CMP #4,%5.1 ;IS LOW ORDER - 4
13855 053362 001403 BEQ .+10
13856 053364 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13857 053364 004767 005222 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13858 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13859 053370 000233 233 ;BY (013746 000172 000207)
13860
13861
13862 053372 021127 000241 CMP (R1),#241 ;CHECK THE TEST NUMBER
13863 053376 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13864 053400 010105 MOV R1,R5 ;RESTORE R5
13865 053402 005215 INC (R5)
13866
13867
13868 053404 012702 040000 MOV #40000,%2
13869 053410 012703 042370 MOV #S5,%3
13870 053414 012704 042372 MOV #S6,%4
13871
13872 ;*****
13873 ;TEST:242 MUL 125252 * S5 = 165252 100000 PS - 11
13874 ;*****
13875
13876 053420 010767 124546 TST242: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13877 053424 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
13878 053430 070067 166734 MUL S5,%0 ;MULTIPLY 125252 * S5
13879 053434 106737 042336 MFPS @#PSWORD ;SAVE PS
13880 053440 122737 000011 042336 CMPB #11,@#PSWORD ;IS PS = 11
13881 053446 001403 BEQ .+10
13882 053450 004767 005136 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13883 ;PS IS WRONG
13884 053454 000234 234 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13885 ;BY (013746 000172 000207)
13886
13887 053456 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
13888 053462 001403 BEQ .+10
13889 053464 004767 005122 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13890 ;HIGH ORDER IS WRONG
13891 053470 000235 235 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13892 ;BY (013746 000172 000207)

```

```
13893
13894 053472 022701 100000          CMP    #100000,%0!1    ;IS LOW ORDER = 100000
13895 053476 001403          BEQ    .+10
13896 053500          1$:
13897 053500 004767 005106          JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13898          ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13899 053504 000236          236          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13900          ;BY (013746 000172 000207)
13901
13902 053506 021527 000242          CMP    (R5),#242
13903 053512 001372          BNE    1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13904 053514 005215          INC    (R5)
13905
13906
13907          ;*****
13908          ;TEST:243          MUL    125252 * @S6 = 165252 100000          PS = 11
13909          ;*****
13910
13911 053516 010767 124450          TST243: MOV    PC,LPADR          ;STORE ERROR LOOP ADDRESS
13912 053522 012700 125252          MOV    #125252,%0          ;LOAD MULTIPLICAND WITH 125252
13913 053526 070077 166640          MUL    @S6,%0          ;MULTIPLY 125252 * @S6
13914 053532 106737 042336          MFPS   @#PSWORD          ;SAVE PS
13915 053536 122737 000011          042336      CMPB   #11,@#PSWORD          ;IS PS = 11
13916 053544 001403          BEQ    .+10
13917 053546 004767 005040          JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13918          ;PS IS WRONG
13919 053552 000237          237          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13920          ;BY (013746 000172 000207)
13921
13922 053554 022700 165252          CMP    #165252,%0          ;IS HIGH ORDER = 165252
13923 053560 001403          BEQ    .+10
13924 053562 004767 005024          JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13925          ;HIGH ORDER IS WRONG
13926 053566 000240          240          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13927          ;BY (013746 000172 000207)
13928
13929 053570 022701 100000          CMP    #100000,%0!1    ;IS LOW ORDER = 100000
13930 053574 001403          BEQ    .+10
13931 053576          1$:
13932 053576 004767 005010          JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13933          ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13934 053602 000241          241          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13935          ;BY (013746 000172 000207)
13936
13937 053604 021527 000243          CMP    (R5),#243
13938 053610 001372          BNE    1$          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13939 053612 005215          INC    (R5)
13940
13941
13942          ;*****
13943          ;TEST:244          MUL    125252 * @#S5 = 165252 100000          PS = 11
13944          ;*****
13945
13946 053614 010767 124352          TST244: MOV    PC,LPADR          ;STORE ERROR LOOP ADDRESS
13947 053620 012700 125252          MOV    #125252,%0          ;LOAD MULTIPLICAND WITH 125252
13948 053624 070037 042370          MUL    @#S5,%0          ;MULTIPLY 125252 * @#S5
```

```
13949 053630 106737 042336 MFPS @#PSWORD ;SAVE PS
13950 053634 122737 000011 042336 CMPB #11,@#PSWORD ;IS PS = 11
13951 053642 001403 BEQ .+10
13952 053644 004767 004742 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13953 ;PS IS WRONG
13954 053650 000242 242 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13955 ;BY (013746 000172 000207)
13956
13957 053652 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
13958 053656 001403 BEQ .+10
13959 053660 004767 004726 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13960 ;HIGH ORDER IS WRONG
13961 053664 000243 243 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13962 ;BY (013746 000172 000207)
13963
13964 053666 022701 100000 CMP #100000,%0.1 ;IS LOW ORDER = 100000
13965 053672 001403 BEQ .+10
13966 053674 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13967 053674 004767 004712 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
13968 053700 000244 244 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13969 ;BY (013746 000172 000207)
13970
13971
13972 053702 021527 000244 CMP (R5),#244
13973 053706 001372 BNE 1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
13974 053710 005215 INC (R5)
13975
13976
13977
13978 ;*****
13979 ;TEST:245 MUL 125252 * %2 = 165252 100000 PS - 11
13980 ;*****
13981 053712 010767 124254 TST245: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
13982 053716 012700 125252 MOV #125252,%0 ;LOAD MULTIPLICAND WITH 125252
13983 053722 070002 MUL %2,%0 ;MULTIPLY 125252 * %2
13984 053724 106737 042336 MFPS @#PSWORD ;SAVE PS
13985 053730 122737 000011 042336 CMPB #11,@#PSWORD ;IS PS = 11
13986 053736 001403 BEQ .+10
13987 053740 004767 004646 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13988 ;PS IS WRONG
13989 053744 000245 245 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13990 ;BY (013746 000172 000207)
13991
13992 053746 022700 165252 CMP #165252,%0 ;IS HIGH ORDER = 165252
13993 053752 001403 BEQ .+10
13994 053754 004767 004632 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
13995 ;HIGH ORDER IS WRONG
13996 053760 000246 246 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
13997 ;BY (013746 000172 000207)
13998
13999 053762 022701 100000 CMP #100000,%0.1 ;IS LOW ORDER = 100000
14000 053766 001403 BEQ .+10
14001 053770 1$: JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14002 053770 004767 004616 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14003 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14004 053774 000247 247
```

```
14005 ;BY (013746 000172 000207)
14006
14007 053776 021527 000245      CMP      (R5),#245
14008 054002 001372              BNE      1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14009 054004 005215              INC      (R5)
14010
14011
14012 ;*****
14013 ;TEST:246      MUL      125252 * (3)+ - 165252 100000      PS - 11
14014 ;*****
14015
14016 054006 010767 124160      TST246: MOV      PC,LPADR ;STORE ERROR LOOP ADDRESS
14017 054012 012700 125252      MOV      #125252,%0 ;LOAD MULTIPLICAND WITH 125252
14018 054016 070023              MUL      (3)+,%0 ;MULTIPLY 125252 * (3)+
14019 054020 106737 042336      MFPS    @#PSWORD ;SAVE PS
14020 054024 122737 000011 042336      CMPS    #11,@#PSWORD ;IS PS = 11
14021 054032 001403              BEQ     .+10
14022 054034 004767 004552      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14023 ;PS IS WRONG
14024 054040 000250              250 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14025 ;BY (013746 000172 000207)
14026
14027 054042 022700 165252      CMP     #165252,%0 ;IS HIGH ORDER = 165252
14028 054046 001403              BEQ     .+10
14029 054050 004767 004536      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14030 ;HIGH ORDER IS WRONG
14031 054054 000251              251 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14032 ;BY (013746 000172 000207)
14033
14034 054056 022701 100000      CMP     #100000,%0.1 ;IS LOW ORDER = 100000
14035 054062 001403              BEQ     .+10
14036 054064 ;1$:
14037 054064 004767 004522      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14038 ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14039 054070 000252              252 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14040 ;BY (013746 000172 000207)
14041
14042 054072 021527 000246      CMP     (R5),#246
14043 054076 001372              BNE     1$ ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14044 054100 005215              INC     (R5)
14045
14046 ;*****
14047 ;TEST:247      MUL      125252 * -(3) = 165252 100000      PS = 11
14048 ;*****
14049
14050
14051 054102 010767 124064      TST247: MOV      PC,LPADR ;STORE ERROR LOOP ADDRESS
14052 054106 012700 125252      MOV      #125252,%0 ;LOAD MULTIPLICAND WITH 125252
14053 054112 070043              MUL      -(3),%0 ;MULTIPLY 125252 * -(3)
14054 054114 106737 042336      MFPS    @#PSWORD ;SAVE PS
14055 054120 122737 000011 042336      CMPS    #11,@#PSWORD ;IS PS = 11
14056 054126 001403              BEQ     .+10
14057 054130 004767 004456      JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14058 ;PS IS WRNG
14059 054134 000253              253 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14060 ;BY (013746 000172 000207)
```

```

14061
14062 054136 022700 165252      CMP    #165252,%0      ;IS HIGH ORDER = 165252
14063 054142 001403      BEQ    .+10
14064 054144 004767 004442      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14065                                     ;HIGH ORDER IS WRONG
14066 054150 000254      254    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14067                                     ;BY (013746 000172 000207)
14068
14069 054152 022701 100000      CMP    #100000,%0!1   ;IS LOW ORDER = 100000
14070 054156 001403      BEQ    .+10
14071 054160      1$:
14072 054160 004767 004426      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14073                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14074 054164 000255      255    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14075                                     ;BY (013746 000172 000207)
14076
14077 054166 021527 000247      CMP    (R5),#247
14078 054172 001372      BNE    1$
14079 054174 005215      INC    (R5)          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14080
14081
14082
14083      ;*****
14084      ;TEST:250      MUL    125252 * 2(4) = 165252 100000      PS 11
14085      ;*****
14086 054176 010767 123770      TST250: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
14087 054202 012700 125252      MOV    #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14088 054206 070064 000002      MUL    2(4),%0          ;MULTIPLY 125252 * 2(4)
14089 054212 106737 042336      MFPS   @#PSWORD        ;SAVE PS
14090 054216 122737 000011 042336      CMPB   #11,@#PSWORD    ;IS PS = 11
14091 054224 001403      BEQ    .+10
14092 054226 004767 004360      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14093                                     ;PS IS WRONG
14094 054232 000256      256    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14095                                     ;BY (013746 000172 000207)
14096
14097 054234 022700 165252      CMP    #165252,%0      ;IS HIGH ORDER = 165252
14098 054240 001403      BEQ    .+10
14099 054242 004767 004344      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14100                                     ;HIGH ORDER IS WRONG
14101 054246 000257      257    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14102                                     ;BY (013746 000172 000207)
14103
14104 054250 022701 100000      CMP    #100000,%0!1   ;IS LOW ORDER = 100000
14105 054254 001403      BEQ    .+10
14106 054256      1$:
14107 054256 004767 004330      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14108                                     ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14109 054262 000260      260    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14110                                     ;BY (013746 000172 000207)
14111
14112 054264 021527 000250      CMP    (R5),#250
14113 054270 001372      BNE    1$
14114 054272 005215      INC    (R5)          ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14115
14116

```

```
14117
14118
14119
14120
14121 054274 010767 123672
14122 054300 012700 125252
14123 054304 070074 000000
14124 054310 106737 042336
14125 054314 122737 000011 042336
14126 054322 001403
14127 054324 004767 004262
14128
14129 054330 000261
14130
14131
14132 054332 022700 165252
14133 054336 001403
14134 054340 004767 004246
14135
14136 054344 000262
14137
14138
14139 054346 022701 100000
14140 054352 001403
14141 054354
14142 054354 004767 004232
14143
14144 054360 000263
14145
14146
14147 054362 021527 000251
14148 054366 001372
14149 054370 005215
14150
14151
14152
14153
14154
14155
14156 054372 010767 123574
14157 054376 012700 125252
14158 054402 070034
14159 054404 106737 042336
14160 054410 122737 000011 042336
14161 054416 001403
14162 054420 004767 004166
14163
14164 054424 000264
14165
14166
14167 054426 022700 165252
14168 054432 001403
14169 054434 004767 004152
14170
14171 054440 000265
14172

:*****
:TEST:251      MUL      125252 * @ (4) = 165252 100000      PS  11
:*****

TST251: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0    ;LOAD MULTIPLICAND WITH 125252
        MUL      @ (4),%0      ;MULTIPLY 125252 * @ (4)
        MFPS     @#PSWORD      ;SAVE PS
        CMPB    #11,@#PSWORD   ;IS PS = 11
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HAIT ROUTINE
                ;PS IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP     #165252,%0      ;IS HIGH ORDER = 165252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;HIGH ORDER IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP     #100000,%0.1    ;IS LOW ORDER - 100000
        BEQ     .+10
1$:     JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP     (R5),#251
        BNE    1$
        INC     (R5)            ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE

:*****
:TEST:252      MUL      125252 * @ (4)+ = 165252 100000      PS  11
:*****

TST252: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
        MOV      #125252,%0    ;LOAD MULTIPLICAND WITH 125252
        MUL      @ (4)+,%0     ;MULTIPLY 125252 * @ (4)+
        MFPS     @#PSWORD      ;SAVE PS
        CMPB    #11,@#PSWORD   ;IS PS = 11
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;PS IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)

        CMP     #165252,%0      ;IS HIGH ORDER = 165252
        BEQ     .+10
        JSR     PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
                ;HIGH ORDER IS WRONG
                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
                ;BY (013746 000172 000207)
```

```
14173
14174 054442 022701 100000      CMP    #100000,%0.1    ;IS LOW ORDER - 100000
14175 054446 001403              BEQ    .+10
14176 054450              1$:
14177 054450 004767 004136      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14178                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14179 054454 000266              266    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14180                                ;BY (013746 000172 000207)
14181
14182 054456 021527 000252      CMP    (R5),#252
14183 054462 001372              BNE    1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14184 054464 005215              INC    (R5)
14185
14186
14187
14188                                :*****
14189                                :TEST:253      MUL    125252 * @-(4) - 165252 100000      PS  11
14190                                :*****
14191 054466 010767 123500      TST253: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
14192 054472 012700 125252      MOV    #125252,%0      ;LOAD MULTIPLICAND WITH 125252
14193 054476 070054              MUL    @-(4),%0        ;MULTIPLY 125252 * @-(4)
14194 054500 106737 042336      MFPS   @#PSWORD        ;SAVE PS
14195 054504 122737 000011 042336  CMPB   #11,@#PSWORD    ;IS PS = 11
14196 054512 001403              BEQ    .+10
14197 054514 004767 004072      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14198                                ;PS IS WRONG
14199 054520 000267              267    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14200                                ;BY (013746 000172 000207)
14201
14202 054522 022700 165252      CMP    #165252,%0      ;IS HIGH ORDER = 165252
14203 054526 001403              BEQ    .+10
14204 054530 004767 004056      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14205                                ;HIGH ORDER IS WRONG
14206 054534 000270              270    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14207                                ;BY (013746 000172 000207)
14208
14209 054536 022701 100000      CMP    #100000,%0.1    ;IS LOW ORDER - 100000
14210 054542 001403              BEQ    .+10
14211 054544              1$:
14212 054544 004767 004042      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14213                                ;LOW ORDER IS WRONG OR WRONG SEQUENCE
14214 054550 000271              271    ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14215                                ;BY (013746 000172 000207)
14216
14217 054552 021527 000253      CMP    (R5),#253
14218 054556 001372              BNE    1$              ;IF IN WRONG SEQUENCE GO TO THE HLT ABOVE
14219 054560 005215              INC    (R5)
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
14250
14251
14252
14253
14254
14255
14256
14257
14258
14259
14260
14261
14262
14263
14264
14265
14266
14267
14268
14269
14270
14271
14272
14273
14274
14275
14276
14277

054562 010767 123404
054566 012700 000000
054572 012701 000004
054576 071027 000002
054602 106737 042336

054606 122737 000000 042336
054614 001403
054616 004767 003770

054622 000272

054624 022700 000002
054630 001403
054632 004767 003754

054636 000273

054640 022701 000000
054644 001403
054646 004767 003740

054652 000274

054654 021527 000254
054660 001403
054662 004767 003724

054666 000275

054670 005215

054672 010767 123274
054676 012702 177777
054702 012703 177767
054706 071227 000003

```
*****  
: 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 @APSWORD ;SAVE PS  
  
CMPB #0,@APSWORD ;IS PS - 0  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;PS IS WRONG  
272 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #2,%0 ;IS QUOTIENT - 2  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;QUOTIENT IS WRONG  
273 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP #0,%0+1 ;IS REMAINDER = 0  
BEQ .+10  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;WRONG REMAINDER  
274 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
CMP (R5),#254  
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT  
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE  
;TEST IS IN WRONG SEQUENCE  
275 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)  
;BY (013746 000172 000207)  
  
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
```



```

14278 054712 106737 042336 MFPS @#PSWORD ;SAVE PS
14279
14280 054716 122737 000010 042336 CMPB #10,@#PSWORD ;IS PS - 10
14281 054724 001403 BEQ .+10
14282 054726 004767 003660 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14283 ;PS IS WRONG
14284 054732 000276 276 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14285 ;BY (013746 000172 000207)
14286
14287
14288 054734 022702 177775 CMP #-3,%2 ;IS QUOTIENT = -3
14289 054740 001403 BEQ .+10
14290 054742 004767 003644 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14291 ;QUOTIENT IS WRONG
14292 054746 000277 277 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14293 ;BY (013746 000172 000207)
14294
14295
14296 054750 022703 000000 CMP #0,%2+1 ;IS REMAINDER = 0
14297 054754 001403 BEQ .+10
14298 054756 004767 003630 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14299 ;WRONG REMAINDER
14300 054762 000300 300 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14301 ;BY (013746 000172 000207)
14302
14303 054764 021527 000255 CMP (R5),#255
14304 054770 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
14305 054772 004767 003614 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14306 ;TEST IS IN WRONG SEQUENCE
14307 054776 000301 301 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14308 ;BY (013746 000172 000207)
14309
14310 055000 005215 INC (R5)
14311
14312
14313 :*****
14314 :TEST:256 DIV 0 9. / #2 4 REM - 1 PS 0
14315 :*****
14316 055002 010767 123164 TST256: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
14317 055006 010501 MOV R5,R1 ;SAVE R5
14318 055010 012704 000000 MOV #0,%4 ;LOAD HIGH ORDER WITH 0
14319 055014 012705 000011 MOV #9,%4+1 ;LOAD LOW ORDER WITH 9.
14320 055020 071427 000002 DIV #2,%4 ;DIVIDE BY #2
14321 055024 106737 042336 MFPS @#PSWORD ;SAVE PS
14322
14323 055030 122737 000000 042336 CMPB #0,@#PSWORD ;IS PS = 0
14324 055036 001403 BEQ .+10
14325 055040 004767 003546 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14326 ;PS IS WRONG
14327 055044 000302 302 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14328 ;BY (013746 000172 000207)
14329
14330
14331 055046 022704 000004 CMP #4,%4 ;IS QUOTIENT = 4
14332 055052 001403 BEQ .+10
14333 055054 004767 003532 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE

```

```
14334
14335 055060 000303          303          :QUOTIENT IS WRONG
14336                      :TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14337                      :BY (013746 000172 000207)
14338
14339 055062 022705 000001    CMP      #1,%4+1      :IS REMAINDER = 1
14340 055066 001403          BEQ      .+10
14341 055070 004767 003516    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14342                      :WRONG REMAINDER
14343 055074 000304          304          :TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14344                      :BY (013746 000172 000207)
14345
14346 055076 010105          MOV      R1,R5        :RESTORE R5
14347 055100 021527 000256    CMP      (R5),#256
14348 055104 001403          BEQ      .+10        :IF IN WRONG SEQUENCE GO TO THE HLT
14349 055106 004767 003500    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14350                      :TEST IS IN WRONG SEQUENCE
14351 055112 000305          305          :TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14352                      :BY (013746 000172 000207)
14353
14354 055114 005215          INC      (R5)
14355
14356                      :*****
14357                      :TEST:257      DIV      -1 -9. / #2 - -4      REM = -1      PS  10
14358                      :*****
14359
14360 055116 010767 123050    TST257: MOV      PC,LPADR      :STORE ERROR LOOP ADDRESS
14361 055122 012700 177777    MOV      #-1,%0        :LOAD HIGH ORDER WITH -1
14362 055126 012701 177767    MOV      #-9,%0+1      :LOAD LOW ORDER WITH -9.
14363 055132 071027 000002    DIV      #2,%0         :DIVIDE BY #2
14364 055136 106737 042336    MFPS     @#PSWORD      :SAVE PS
14365
14366 055142 122737 000010 042336  CMPB     #10,@#PSWORD   :IS PS - 10
14367 055150 001403          BEQ      .+10
14368 055152 004767 003434    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14369                      :PS IS WRONG
14370 055156 000306          306          :TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14371                      :BY (013746 000172 000207)
14372
14373
14374 055160 022700 177774    CMP      #-4,%0        :IS QUOTIENT = -4
14375 055164 001403          BEQ      .+10
14376 055166 004767 003420    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14377                      :QUOTIENT IS WRONG
14378 055172 000307          307          :TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14379                      :BY (013746 000172 000207)
14380
14381
14382 055174 022701 177777    CMP      #-1,%0+1      :IS REMAINDER = -1
14383 055200 001403          BEQ      .+10
14384 055202 004767 003404    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14385                      :WRONG REMAINDER
14386 055206 000310          310          :TO SCOPE, REPLACE LAST: BEQ  .+10 (001403)
14387                      :BY (013746 000172 000207)
14388
14389 055210 021527 000257    CMP      (R5),#257
```

```
14390 055214 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
14391 055216 004767 003370 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14392 ;TEST IS IN WRONG SEQUENCE
14393 055222 000311 311 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14394 ;BY (013746 000172 000207)
14395
14396 055224 005215 INC (R5)
14397
14398 ;*****
14399 ;TEST:260 DIV 0 2 / #-3 = 0 REM - 2 PS = 4
14400 ;*****
14401
14402 055226 010767 122740 TST260: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
14403 055232 012702 000000 MOV #0,%2 ;LOAD HIGH ORDER WITH 0
14404 055236 012703 000002 MOV #2,%2+1 ;LOAD LOW ORDER WITH 2
14405 055242 071227 177775 DIV #-3,%2 ;DIVIDE BY #-3
14406 055246 106737 042336 MFPS @#PSWORD ;SAVE PS
14407
14408 055252 122737 000004 042336 CMPB #4,@#PSWORD ;IS PS = 4
14409 055260 001403 BEQ .+10
14410 055262 004767 003324 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14411 ;PS IS WRONG
14412 055266 000312 312 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14413 ;BY (013746 000172 000207)
14414
14415
14416 055270 022702 000000 CMP #0,%2 ;IS QUOTIENT = 0
14417 055274 001403 BEQ .+10
14418 055276 004767 003310 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14419 ;QUOTIENT IS WRONG
14420 055302 000313 313 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14421 ;BY (013746 000172 000207)
14422
14423
14424 055304 022703 000002 CMP #2,%2+1 ;IS REMAINDER = 2
14425 055310 001403 BEQ .+10
14426 055312 004767 003274 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14427 ;WRONG REMAINDER
14428 055316 000314 314 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14429 ;BY (013746 000172 000207)
14430
14431 055320 021527 000260 CMP (R5),#260
14432 055324 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
14433 055326 004767 003260 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14434 ;TEST IS IN WRONG SEQUENCE
14435 055332 000315 315 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14436 ;BY (013746 000172 000207)
14437
14438 055334 005215 INC (R5)
14439
14440 ;*****
14441 ;TEST:261 DIV -1 -2 / #3 = 0 REM = -2 PS - 4
14442 ;*****
14443
14444 055336 010767 122630 TST261: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
14445 055342 010501 MOV R5,R1 ;SAVE R5
```

```

14446 055344 012704 177777      MOV    #-1,%4      ;LOAD HIGH ORDER WITH -1
14447 055350 012705 177776      MOV    #-2,%4+1    ;LOAD LOW ORDER WITH -2
14448 055354 071427 000003      DIV    #3,%4       ;DIVIDE BY #3
14449 055360 106737 042336      MFPS   @#PSWORD    ;SAVE PS
14450
14451 055364 122737 000004 042336  CMPB   #4,@#PSWORD ;IS PS = 4
14452 055372 001403                BEQ    .+10         ;PS IS WRONG
14453 055374 004767 003212      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14454                                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14455 055400 000316                ;BY (013746 000172 000207)
14456
14457
14458
14459 055402 022704 000000      CMP    #0,%4       ;IS QUOTIENT = 0
14460 055406 001403                BEQ    .+10
14461 055410 004767 003176      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14462                                ;QUOTIENT IS WRONG
14463 055414 000317                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14464                                ;BY (013746 000172 000207)
14465
14466
14467 055416 022705 177776      CMP    #-2,%4+1    ;IS REMAINDER = -2
14468 055422 001403                BEQ    .+10
14469 055424 004767 003162      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14470                                ;WRONG REMAINDER
14471 055430 000320                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14472                                ;BY (013746 000172 000207)
14473
14474 055432 010105                MOV    R1,R5       ;RESTORE R5
14475 055434 021527 000261      CMP    (R5),#261
14476 055440 001403                BEQ    .+10         ;IF IN WRONG SEQUENCE GO TO THE HLT
14477 055442 004767 003144      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14478                                ;TEST IS IN WRONG SEQUENCE
14479 055446 000321                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14480                                ;BY (013746 000172 000207)
14481
14482 055450 005215                INC    (R5)
14483
14484
14485
14486
14487
14488 055452 010767 122514      TST262: MOV    PC,LPADR ;STORE ERROR LOOP ADDRESS
14489 055456 012700 177777      MOV    #-1,%0      ;LOAD HIGH ORDER WITH -1
14490 055462 012701 177777      MOV    #-1,%0+1    ;LOAD LOW ORDER WITH -1
14491 055466 071027 000001      DIV    #1,%0       ;DIVIDE BY #1
14492 055472 106737 042336      MFPS   @#PSWORD    ;SAVE PS
14493
14494 055476 122737 000010 042336  CMPB   #10,@#PSWORD ;IS PS = 10
14495 055504 001403                BEQ    .+10
14496 055506 004767 003100      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14497                                ;PS IS WRONG
14498 055512 000322                ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14499                                ;BY (013746 000172 000207)
14500
14501

```

```

:*****
:TEST:262      DIV    -1 -1 / #1 = -1      REM = 0      PS  10
:*****

```



```
14558
14559 055654 021527 000263      CMP      (R5),#263
14560 055660 001403      BEQ      .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT
14561 055662 004767 002724      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14562                                     ;TEST IS IN WRONG SEQUENCE
14563 055666 000331      331      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14564                                     ;BY (013746 000172 000207)
14565
14566 055670 005215      INC      (R5)
14567
14568
14569 ;*****
14570 ;TEST:264      DIV      -1 125252 / #2 = 152525      REM = 0      PS = 10
14571 ;*****
14572 055672 010767 122274      TST264: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
14573 055676 012702 177777      MOV      #-1,%2      ;LOAD HIGH ORDER WITH -1
14574 055702 012703 125252      MOV      #125252,%2+1 ;LOAD LOW ORDER WITH 125252
14575 055706 071227 000002      DIV      #2,%2      ;DIVIDE BY #2
14576 055712 106737 042336      MFPS     @#PSWORD      ;SAVE PS
14577
14578 055716 122737 000010 042336      CMPB     #10,@#PSWORD ;IS PS = 10
14579 055724 001403      BEQ      .+10
14580 055726 004767 002660      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14581                                     ;PS IS WRONG
14582 055732 000332      332      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14583                                     ;BY (013746 000172 000207)
14584
14585
14586 055734 022702 152525      CMP      #152525,%2 ;IS QUOTIENT - 152525
14587 055740 001403      BEQ      .+10
14588 055742 004767 002644      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14589                                     ;QUOTIENT IS WRONG
14590 055746 000333      333      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14591                                     ;BY (013746 000172 000207)
14592
14593
14594 055750 022703 000000      CMP      #0,%2+1 ;IS REMAINDER = 0
14595 055754 001403      BEQ      .+10
14596 055756 004767 002630      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14597                                     ;WRONG REMAINDER
14598 055762 000334      334      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14599                                     ;BY (013746 000172 000207)
14600
14601 055764 021527 000264      CMP      (R5),#264
14602 055770 001403      BEQ      .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT
14603 055772 004767 002614      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14604                                     ;TEST IS IN WRONG SEQUENCE
14605 055776 000335      335      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14606                                     ;BY (013746 000172 000207)
14607
14608 056000 005215      INC      (R5)
14609
14610
14611 ;*****
14612 ;TEST:265      DIV      -1 -1 / #-1 - 1      REM = 0      PS = 0
14613 ;*****
```



```
14726 056316 000350          350          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14727                                     ;BY (013746 000172 000207)
14728
14729 056320 021527 000267      CMP      (R5),#267
14730 056324 001403          BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
14731 056326 004767 002260      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14732                                     ;TEST IS IN WRONG SEQUENCE
14733 056332 000351          351          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14734                                     ;BY (013746 000172 000207)
14735
14736 056334 005215          INC      (R5)
14737
14738                                     ;*****
14739                                     ;TEST:270      DIV      0 100000 / #2 = 40000      REM = 0      PS 0
14740                                     ;*****
14741
14742 056336 010767 121630      TST270: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
14743 056342 010501          MOV    R5,R1      ;SAVE R5
14744 056344 012704 000000      MOV    #0,%4      ;LOAD HIGH ORDER WITH 0
14745 056350 012705 100000      MOV    #100000,%4+1 ;LOAD LOW ORDER WITH 100000
14746 056354 071427 000002      DIV    #2,%4      ;DIVIDE BY #2
14747 056360 106737 042336      MFPS   @#PSWORD   ;SAVE PS
14748
14749 056364 122737 000000 042336  CMPB   #0,@#PSWORD ;IS PS = 0
14750 056372 001403          BEQ    .+10
14751 056374 004767 002212      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14752                                     ;PS IS WRONG
14753 056400 000352          352          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14754                                     ;BY (013746 000172 000207)
14755
14756
14757 056402 022704 040000      CMP    #40000,%4   ;IS QUOTIENT = 40000
14758 056406 001403          BEQ    .+10
14759 056410 004767 002176      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14760                                     ;QUOTIENT IS WRONG
14761 056414 000353          353          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14762                                     ;BY (013746 000172 000207)
14763
14764
14765 056416 022705 000000      CMP    #0,%4+1     ;IS REMAINDER = 0
14766 056422 001403          BEQ    .+10
14767 056424 004767 002162      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14768                                     ;WRONG REMAINDER
14769 056430 000354          354          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14770                                     ;BY (013746 000172 000207)
14771
14772 056432 010105          MOV    R1,R5      ;RESTORE R5
14773 056434 021527 000270      CMP    (R5),#270
14774 056440 001403          BEQ    .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
14775 056442 004767 002144      JSR    PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14776                                     ;TEST IS IN WRONG SEQUENCE
14777 056446 000355          355          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14778                                     ;BY (013746 000172 000207)
14779
14780 056450 005215          INC    (R5)
14781
```

```
14782
14783
14784
14785
14786 056452 010767 121514
14787 056456 012700 177777
14788 056462 012701 077777
14789 056466 071027 177776
14790 056472 106737 042336
14791
14792 056476 122737 000000 042336
14793 056504 001403
14794 056506 004767 002100
14795
14796 056512 000356
14797
14798
14799
14800 056514 022700 040000
14801 056520 001403
14802 056522 004767 002064
14803
14804 056526 000357
14805
14806
14807
14808 056530 022701 177777
14809 056534 001403
14810 056536 004767 002050
14811
14812 056542 000360
14813
14814
14815 056544 021527 000271
14816 056550 001403
14817 056552 004767 002034
14818
14819 056556 000361
14820
14821
14822 056560 005215
14823
14824
14825
14826
14827
14828 056562 010767 121404
14829 056566 012702 000000
14830 056572 012703 052525
14831 056576 071227 052525
14832 056602 106737 042336
14833
14834 056606 122737 000000 042336
14835 056614 001403
14836 056616 004767 001770
14837
```

```
*****
:TEST:271 DIV 177777 77777 / #177776 = 40000 REM = 177777 PS = 0
*****

TST271: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #177777,%0 ;LOAD HIGH ORDER WITH 177777
MOV #77777,%0+1 ;LOAD LOW ORDER WITH 77777
DIV #177776,%0 ;DIVIDE BY #177776
MFPS @#PSWORD ;SAVE PS

CMPB #0,@#PSWORD ;IS PS = 0
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;PS IS WRONG
356 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #40000,%0 ;IS QUOTIENT = 40000
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;QUOTIENT IS WRONG
357 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP #177777,%0+1 ;IS REMAINDER = 177777
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;WRONG REMAINDER
360 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

CMP (R5),#271
BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;TEST IS IN WRONG SEQUENCE
361 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
;BY (013746 000172 000207)

INC (R5)

*****
:TEST:272 DIV 0 52525 / #52525 = 1 REM = 0 PS 0
*****

TST272: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
MOV #0,%2 ;LOAD HIGH ORDER WITH 0
MOV #52525,%2+1 ;LOAD LOW ORDER WITH 52525
DIV #52525,%2 ;DIVIDE BY #52525
MFPS @#PSWORD ;SAVE PS

CMPB #0,@#PSWORD ;IS PS = 0
BEQ .+10
JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
;PS IS WRONG
```



```

14950 057142 000373          373          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14951                                     ;BY (013746 000172 000207)
14952
14953
14954 057144 022701 000001      CMP      #1,%0+1          ;IS REMAINDER = 1
14955 057150 001403          BEQ      .+10
14956 057152 004767 001434      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14957                                     ;WRONG REMAINDER
14958 057156 000374          374          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14959                                     ;BY (013746 000172 000207)
14960
14961 057160 021527 000275      CMP      (R5),#275
14962 057164 001403          BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
14963 057166 004767 001420      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14964                                     ;TEST IS IN WRONG SEQUENCE
14965 057172 000375          375          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14966                                     ;BY (013746 000172 000207)
14967
14968 057174 005215          INC      (R5)
14969
14970
14971                                     :*****
14972                                     :TEST:276      DIV      0 52525 / @S10 = 25252      REM = 1      PS  0
14973                                     :*****
14974 057176 010767 120770      TST276: MOV      PC,LPADR          ;STORE ERROR LOOP ADDRESS
14975 057202 012700 000000      MOV      #0,%0          ;LOAD HIGH ORDER WITH 0
14976 057206 012701 052525      MOV      #52525,%0+1    ;LOAD LOW ORDER WITH 52525
14977 057212 071077 163164      DIV      @S10,%0        ;DIVIDE BY @S10
14978 057216 106737 042336      MFPS     @#PSWORD       ;SAVE PS
14979
14980 057222 122737 000000 042336  CMPB     #0,@#PSWORD     ;IS PS = 0
14981 057230 001403          BEQ      .+10
14982 057232 004767 001354      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14983                                     ;PS IS WRONG
14984 057236 000376          376          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14985                                     ;BY (013746 000172 000207)
14986
14987
14988 057240 022700 025252      CMP      #25252,%0      ;IS QUOTIENT = 25252
14989 057244 001403          BEQ      .+10
14990 057246 004767 001340      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14991                                     ;QUOTIENT IS WRONG
14992 057252 000377          377          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
14993                                     ;BY (013746 000172 000207)
14994
14995
14996 057254 022701 000001      CMP      #1,%0+1          ;IS REMAINDER = 1
14997 057260 001403          BEQ      .+10
14998 057262 004767 001324      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
14999                                     ;WRONG REMAINDER
15000 057266 000400          400          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15001                                     ;BY (013746 000172 000207)
15002
15003 057270 021527 000276      CMP      (R5),#276
15004 057274 001403          BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
15005 057276 004767 001310      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE

```



```

15062 057434 106737 042336 MFPS @#PSWORD ;SAVE PS
15063
15064 057440 122737 000000 042336 CMPB #0,@#PSWORD ;IS PS = 0
15065 057446 001403 BEQ .+10
15066 057450 004767 001136 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15067 ;PS IS WRONG
15068 057454 000406 406 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15069 ;BY (013746 000172 000207)
15070
15071
15072 057456 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
15073 057462 001403 BEQ .+10
15074 057464 004767 001122 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15075 ;QUOTIENT IS WRONG
15076 057470 000407 407 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15077 ;BY (013746 000172 000207)
15078
15079
15080 057472 022701 000001 CMP #1,%0+1 ;IS REMAINDER 1
15081 057476 001403 BEQ .+10
15082 057500 004767 001106 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15083 ;WRONG REMAINDER
15084 057504 000410 410 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15085 ;BY (013746 000172 000207)
15086
15087 057506 021527 000300 CMP (R5),#300
15088 057512 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
15089 057514 004767 001072 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15090 ;TEST IS IN WRONG SEQUENCE
15091 057520 000411 411 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15092 ;BY (013746 000172 000207)
15093
15094 057522 005215 INC (R5)
15095
15096
15097 ;*****
15098 ;TEST:301 DIV 0 52525 / (3)+ - 25252 REM 1 PS 0
15099 ;*****
15100 057524 010767 120442 TST301: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
15101 057530 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
15102 057534 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
15103 057540 071023 DIV (3)+,%0 ;DIVIDE BY (3)+
15104 057542 106737 042336 MFPS @#PSWORD ;SAVE PS
15105
15106 057546 122737 000000 042336 CMPB #0,@#PSWORD ;IS PS = 0
15107 057554 001403 BEQ .+10
15108 057556 004767 001030 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15109 ;PS IS WRONG
15110 057562 000412 412 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15111 ;BY (013746 000172 000207)
15112
15113
15114 057564 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
15115 057570 001403 BEQ .+10
15116 057572 004767 001014 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15117 ;QUOTIENT IS WRONG

```

```

15118 057576 000413          413          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15119                                     :BY (013746 000172 000207)
15120
15121
15122 057600 022701 000001    CMP      #1,%0+1          :IS REMAINDER = 1
15123 057604 001403          BEQ      .+10
15124 057606 004767 001000    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15125                                     :WRONG REMAINDER
15126 057612 000414          414          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15127                                     :BY (013746 000172 000207)
15128
15129 057614 021527 000301    CMP      (R5),#301
15130 057620 001403          BEQ      .+10          :IF IN WRONG SEQUENCE GO TO THE HLT
15131 057622 004767 000764    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15132                                     :TEST IS IN WRONG SEQUENCE
15133 057626 000415          415          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15134                                     :BY (013746 000172 000207)
15135
15136 057630 005215          INC      (R5)
15137
15138 :*****
15139 :TEST:302      DIV      0 52525 / -(3) = 25252      REM 1      PS 0
15140 :*****
15141
15142 057632 010767 120334    TST302: MOV      PC,LPADR          ;STORE ERROR LOOP ADDRESS
15143 057636 012700 000000    MOV      #0,%0          ;LOAD HIGH ORDER WITH 0
15144 057642 012701 052525    MOV      #52525,%0+1    ;LOAD LOW ORDER WITH 52525
15145 057646 071043          DIV      -(3),%0        ;DIVIDE BY -(3)
15146 057650 106737 042336    MFPS     @#PSWORD        ;SAVE PS
15147
15148 057654 122737 000000 042336    CMPB     #0,@#PSWORD     ;IS PS - 0
15149 057662 001403          BEQ      .+10
15150 057664 004767 000722    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15151                                     :PS IS WRONG
15152 057670 000416          416          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15153                                     :BY (013746 000172 000207)
15154
15155
15156 057672 022700 025252    CMP      #25252,%0      ;IS QUOTIENT = 25252
15157 057676 001403          BEQ      .+10
15158 057700 004767 000706    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15159                                     :QUOTIENT IS WRONG
15160 057704 000417          417          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15161                                     :BY (013746 000172 000207)
15162
15163
15164 057706 022701 000001    CMP      #1,%0+1          :IS REMAINDER = 1
15165 057712 001403          BEQ      .+10
15166 057714 004767 000672    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15167                                     :WRONG REMAINDER
15168 057720 000420          420          :TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15169                                     :BY (013746 000172 000207)
15170
15171 057722 021527 000302    CMP      (R5),#302
15172 057726 001403          BEQ      .+10          :IF IN WRONG SEQUENCE GO TO THE HLT
15173 057730 004767 000656    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE

```



```
15174                                     ;TEST IS IN WRONG SEQUENCE
15175 057734 000421                       421      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15176                                     ;BY (013746 000172 000207)
15177
15178 057736 005215                       INC      (R5)
15179
15180                                     ;*****
15181 ;TEST:303      DIV      0 52525 / 2(4) - 25252      REM - 1      PS  0
15182 ;*****
15183
15184 057740 010767 120226      TST303: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
15185 057744 012700 000000      MOV      #0,%0      ;LOAD HIGH ORDER WITH 0
15186 057750 012701 052525      MOV      #52525,%0+1 ;LOAD LOW ORDER WITH 52525
15187 057754 071064 000002      DIV      2(4),%0     ;DIVIDE BY 2(4)
15188 057760 106737 042336      MFPS     @#PS'WORD   ;SAVE PS
15189
15190 057764 122737 000000 042336      CMPB     #0,@#PSWORD ;IS PS = 0
15191 057772 001403      BEQ      .+10
15192 057774 004767 000612      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15193                                     ;PS IS WRONG
15194 060000 000422                       422      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15195                                     ;BY (013746 000172 000207)
15196
15197
15198 060002 022700 025252      CMP      #25252,%0   ;IS QUOTIENT - 25252
15199 060006 001403      BEQ      .+10
15200 060010 004767 000576      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15201                                     ;QUOTIENT IS WRONG
15202 060014 000423                       423      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15203                                     ;BY (013746 000172 000207)
15204
15205
15206 060016 022701 000001      CMP      #1,%0+1    ;IS REMAINDER = 1
15207 060022 001403      BEQ      .+10
15208 060024 004767 000562      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15209                                     ;WRONG REMAINDER
15210 060030 000424                       424      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15211                                     ;BY (013746 000172 000207)
15212
15213 060032 021527 000303      CMP      (R5),#303
15214 060036 001403      BEQ      .+10      ;IF IN WRONG SEQUENCE GO TO THE HLT
15215 060040 004767 000546      JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15216                                     ;TEST IS IN WRONG SEQUENCE
15217 060044 000425                       425      ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15218                                     ;BY (013746 000172 000207)
15219
15220 060046 005215                       INC      (R5)
15221
15222                                     ;*****
15223 ;TEST:304      DIV      0 52525 / @ (4) = 25252      REM  1      PS  0
15224 ;*****
15225
15226 060050 010767 120116      TST304: MOV      PC,LPADR      ;STORE ERROR LOOP ADDRESS
15227 060054 012700 000000      MOV      #0,%0      ;LOAD HIGH ORDER WITH 0
15228 060060 012701 052525      MOV      #52525,%0+1 ;LOAD LOW ORDER WITH 52525
15229 060064 071074 000000      DIV      @ (4),%0    ;DIVIDE BY @ (4)
```

J 6
PAGE 280

CJKDB-C DCF11-AA CPU DIAG. MACY11 30A(1052) 07-MAR-80 12:18
 CJKDBC.P11 07-MAR-80 12:17 DIV INSTRUCTION TESTS

SEQ 0280

```

15230 060070 106737 042336 MFPS @#PSWORD ;SAVE PS
15231
15232 060074 122737 000000 042336 CMPB #0,@#PSWORD ;IS PS = 0
15233 060102 001403 BEQ .+10
15234 060104 004767 000502 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15235 ;PS IS WRONG
15236 060110 000426 426 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15237 ;BY (013746 000172 000207)
15238
15239
15240 060112 022700 025252 CMP #25252,%0 ;IS QUOTIENT = 25252
15241 060116 001403 BEQ .+10
15242 060120 004767 000466 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15243 ;QUOTIENT IS WRONG
15244 060124 000427 427 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15245 ;BY (013746 000172 000207)
15246
15247
15248 060126 022701 000001 CMP #1,%0+1 ;IS REMAINDER = 1
15249 060132 001403 BEQ .+10
15250 060134 004767 000452 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15251 ;WRONG REMAINDER
15252 060140 000430 430 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15253 ;BY (013746 000172 000207)
15254
15255 060142 021527 000304 CMP (R5),#304
15256 060146 001403 BEQ .+10 ;IF IN WRONG SEQUENCE GO TO THE HLT
15257 060150 004767 000436 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15258 ;TEST IS IN WRONG SEQUENCE
15259 060154 000431 431 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15260 ;BY (013746 000172 000207)
15261
15262 060156 005215 INC (R5)
15263
15264 ;*****
15265 ;TEST:305 DIV 0 52525 / @ (4)+ = 25252 REM 1 PS 0
15266 ;*****
15267
15268 060160 010767 120006 TST305: MOV PC,LPADR ;STORE ERROR LOOP ADDRESS
15269 060164 012700 000000 MOV #0,%0 ;LOAD HIGH ORDER WITH 0
15270 060170 012701 052525 MOV #52525,%0+1 ;LOAD LOW ORDER WITH 52525
15271 060174 071034 DIV @ (4)+,%0 ;DIVIDE BY @ (4)+
15272 060176 106737 042336 MFPS @#PSWORD ;SAVE PS
15273
15274 060202 122737 000000 042336 CMPB #0,@#PSWORD ;IS PS = 0
15275 060210 001403 BEQ .+10
15276 060212 004767 000374 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15277 ;PS IS WRONG
15278 060216 000432 432 ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15279 ;BY (013746 000172 000207)
15280
15281
15282 060220 022700 025252 CMP #25252,%0 ;IS QUOTIENT - 25252
15283 060224 001403 BEQ .+10
15284 060226 004767 000360 JSR PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15285 ;QUOTIENT IS WRONG

```

```

15286 060232 000433          433          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15287                                     ;BY (013746 000172 000207)
15288
15289
15290 060234 022701 000001    CMP      #1,%0+1          ;IS REMAINDER = 1
15291 060240 001403          BEQ      .+10
15292 060242 004767 000344    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15293                                     ;WRONG REMAINDER
15294 060246 000434          434          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15295                                     ;BY (013746 000172 000207)
15296
15297 060250 021527 000305    CMP      (R5),#305
15298 060254 001403          BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
15299 060256 004767 000330    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15300                                     ;TEST IS IN WRONG SEQUENCE
15301 060262 000435          435          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15302                                     ;BY (013746 000172 000207)
15303
15304 060264 005215          INC      (R5)
15305
15306
15307
15308
15309
15310 060266 010767 117700    TST306: MOV    PC,LPADR      ;STORE ERROR LOOP ADDRESS
15311 060272 012700 000000    MOV      #0,%0          ;LOAD HIGH ORDER WITH 0
15312 060276 012701 052525    MOV      #52525,%0+1    ;LOAD LOW ORDER WITH 52525
15313 060302 071054          DIV      @-(4),%0        ;DIVIDE BY @-(4)
15314 060304 106737 042336    MFPS     @#PSWORD       ;SAVE PS
15315
15316 060310 122737 000000 042336  CMPB     #0,@#PSWORD     ;IS PS = 0
15317 060316 001403          BEQ      .+10
15318 060320 004767 000266    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15319                                     ;PS IS WRONG
15320 060324 000436          436          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15321                                     ;BY (013746 000172 000207)
15322
15323
15324 060326 022700 025252    CMP      #25252,%0      ;IS QUOTIENT = 25252
15325 060332 001403          BEQ      .+10
15326 060334 004767 000252    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15327                                     ;QUOTIENT IS WRONG
15328 060340 000437          437          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15329                                     ;BY (013746 000172 000207)
15330
15331
15332 060342 022701 000001    CMP      #1,%0+1          ;IS REMAINDER = 1
15333 060346 001403          BEQ      .+10
15334 060350 004767 000236    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE
15335                                     ;WRONG REMAINDER
15336 060354 000440          440          ;TO SCOPE, REPLACE LAST: BEQ .+10 (001403)
15337                                     ;BY (013746 000172 000207)
15338
15339 060356 021527 000306    CMP      (R5),#306
15340 060362 001403          BEQ      .+10          ;IF IN WRONG SEQUENCE GO TO THE HLT
15341 060364 004767 000222    JSR      PC,$HLT;SEEN AN ERROR, GO TO THE HALT ROUTINE

```

```

;*****
;TEST:306 DIV 0 52525 / @-(4) = 25252 REM - 1 PS 0
;*****

```

15342
15343 060370 000441
15344
15345
15346 060372 005215
15347

441

INC (R5)

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

```

15348
15349
15350
15351
15352
15353
15354
15355
15356 060374 005227 177777
15357 060400 001002
15358 060402 104401 060550
15359 060406 005267 117674
15360 060412 042767 100000 117666
15361 060420 005327
15362 060422 000001
15363 060424 003037
15364 060426 012737
15365 060430 000016
15366 060432 060422
15367 060434 104401 060533
15368 060440 016746 117642
15369 060444 104405
15370 060446 104401 060530
15371 060452 013700 000042
15372 060456 001405
15373 060460 000005
15374 060462 004710
15375 060464 000240
15376 060466 000240
15377 060470 000240
15378 060472 013746 000004
15379 060476 012737 060514 000004
15380 060504 012737 000001 164000
15381 060512 000402
15382 060514 062706 000004
15383 060520 012637 000004
15384 060524 000137
15385 060526 001024
15386 060530 377 377 000
15387 060533 015 042412 042116
15388 060540 050040 051501 020123
15389 060546 000043
15390 060550 005015 045103 042113
15391 060556 041502 020060 041504
15392 060564 030506 026461 040501
15393 060572 041440 052520 042040
15394 060600 040511 047107 051517
15395 060606 044524 000103
15396

```

```

.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 , $ENULL ;TYPE A FEW NULL CHARACTERS
$GET42: MOV @#42,R0 ;GET MONITOR ADDRESS
        BEQ DOAGIN ;BRANCH IF NO MONITOR
        RESET ;CLEAR THE WORLD
$ENDAD: JSR PC,(R0) ;GO TO MONITOR
        NOP ;SAVE ROOM
        NOP ;FOR
        NOP ;ACT11
DOAGIN: MOV @#4,-(SP) ;SAVE CONTENTS OF LOCATION 4
        MOV #1$,@#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: .ASCIIZ <15><12>/END PASS #/
MSG1: .ASCIIZ <15><12>/CJKDBC DCF11-AA CPU DIAGNOSTIC/
.EVEN

```

```

15397      .SBTTL  HALT ROUTINE
15398
15399      :*      HALT ROUTINE
15400      :*      -----
15401      :*
15402      :*
15403      :*      PROGRAM COMES HERE ON ENCOUNTERING ANY ERROR
15404      :*
15405
15406 060612 017637 000000 000302 $HLT:  MOV      @($P),@#$FATAL  ;PLACE THE ERROR NUMBER AT LOCATION $FATAL
15407 060620 011637 061000          MOV      ($P),@#CONTIN ;SAVE ERROR NUMBER ADDRESS
15408 060624 032777 020000 146622  BIT      #20000,@$SWR  ;HAS THE OPERATOR ASKED TO SUPRESS ERROR TYPE OUTS
15409 060632 001021          BNE      6$
15410 060634 104401 042414          TYPE     , $CR LF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15411 060640 104401 060740          TYPE     ,MSGERR          ;INFORM ERROR
15412 060644 104401 042414          TYPE     , $CR LF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15413 060650 013746 061000          MOV      @#CONTIN,-($P) ;RETREIVE ERROR NUMBER ADDR
15414 060654 162716 000004          SUB      #4,($P)        ;CALCULATE ERROR PC
15415 060660 104402          TYPOC          ;TYPE PC
15416 060662 104401 042414          TYPE     , $CR LF          ;GO AND TYPE ^ CR, LF, FOLLOWED BY 3 SPACES
15417 060666 013746 000302          MOV      @#$FATAL,-($P) ;RETREIVE ERROR NUMBER
15418 060672 104403          TYPOS          ;TYPE ERROR NUMBER
15419 060674          003          .BYTE     3
15420 060675          000          .BYTE     0
15421 060676 105767 117416          6$:  TSTB    $ENV          ;IF WE ARE NOT UNDER APT. THEN GO TO
15422 060702 001403          BEQ      8$          ;8$
15423 060704 005237 000300          INC      @#$MSGTY      ;OTHERWISE INFORM APT. ABOUT SEEING THE ERROR
15424 060710 000777          BR        ;AND LOOP
15425 060712 104401 042414          8$:  TYPE     , $CR LF          ;GO AND TYPE A CR, LF, FOLLOWED BY 3 SPACES
15426 060716 005777 146532          TST      @$SWR          ;IS IT REQUIRED TO HALT ON ERROR ?
15427 060722 100001          BPL      10$          ;IF NOT THEN GO TO 10$
15428 060724 000000          HALT
15429 060726 013746 061000          10$:  MOV      @#CONTIN,-($P) ;
15430 060732 062716 000002          ADD      #2,($P)        ;CALCULATE RETURN ADDRESS
15431 060736 000207          RTS      PC            ;RETURN
15432 060740 042440 051122 051117 MSGERR: .ASCIZ / ERROR. PC, AND ERROR # ARE: /
15433 060746 020041 020040 041520
15434 060754 020054 047101 020104
15435 060762 051105 047522 020122
15436 060770 020043 051101 035105
15437 060776 000040
15438
15439 061000 000000          .EVEN
15440          CONTIN: .WORD 0
15441
15442 061002 012767 061012 117014 .SBTTL  POWER FAIL ROUTINE
15443 061010 000000          PWRDN:  MOV      #PWRUP,24
15444          HALT
15445 061012 012767 061002 117004 PWRUP:  MOV      #PWRDN,24
15446 061020 012706 001000          MOV      #BUFF,$P
15447 061024 132767 000040 117267  BITB    #40,$ENVM      ;WILL APT ALLOW PRINTING?
15448 061032 001013          BNE      PFRES          ;NO
15449 061034 012700 061066          MOV      #MSGPWF,$RO   ;GET MSG ADDR.
15450 061040 105737 177564          PWAIT:  TSTB    @#TPS    ;TTY READY
15451 061044 100375          BPL      PWAIT          ;NO WAIT
15452 061046 112037 177566          MOV      (RO)+,@#TPB   ;PRINT CHARACTER

```

```
15453 061052 001372  
15454 061054 105737 177564  
15455 061060 100375  
15456 061062 000167 117736  
15457 061066 005015 047520 042527  
15458 061074 020122 040506 046111  
15459 061102 042105 000041  
15460  
15461  
15462  
15463  
15464  
15465  
15466  
15467  
15468  
15469  
15470  
15471  
15472  
15473  
15474  
15475  
15476  
15477  
15478 061106 105767 000261  
15479 061112 100002  
15480 061114 000000  
15481 061116 000430  
15482 061120 010046  
15483 061122 017600 000002  
15484 061126 122767 000001 117164  
15485 061134 001011  
15486 061136 132767 000100 117155  
15487 061144 001405  
15488 061146 010067 000004  
15489 061152 004767 000230  
15490 061156 000000  
15491 061160 132767 000040 117133  
15492 061166 001003  
15493 061170 112046  
15494 061172 001005  
15495 061174 005726  
15496 061176 012600  
15497 061200 062716 000002  
15498 061204 000002  
15499 061206 122716 000011  
15500 061212 001430  
15501 061214 122716 000200  
15502 061220 001006  
15503 061222 005726  
15504 061224 104401  
15505 061226 042414  
15506 061230 105067 000130  
15507 061234 000755  
15508 061236 004767 000056
```

```
          BNE      PWAIT          ;NEXT IF NOT DONE.  
PWAIT1: TSTB     @4TPS  
          BPL     PWAIT1  
PFRES:  JMP     RESTR  
MSGPWF: .ASCIZ  <15><12>.POWER FAILED!.  
  
.EVEN  
.SBTTL  TYPE ROUTINE  
  
;*****  
;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.  
;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.  
;*NOTE1:          $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.  
;*NOTE2:          $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.  
;*NOTE3:          $FILLC CONTAINS THE CHARACTER TO FILL AFTER.  
;*  
;*CALL:  
;*1) USING A TRAP INSTRUCTION  
;*      TYPE      ,MESADR          ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING  
;*OR  
;*      TYPE  
;*      MESADR  
;*  
$TYPE:  TSTB     $TPFLG          ;;IS THERE A TERMINAL?  
          BPL     1$              ;;BR IF YES  
          HALT    HERE           ;;HA. T 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  
4$:     CMPB   #HT,(SP)         ;;BRANCH IF <HT>  
          BEQ   8$              ;;BRANCH IF NOT <CRLF>  
          CMPB   #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>  
          BNE   5$              ;;BRANCH IF NOT <CRLF>  
          TST   (SP)+            ;;POP <CR><LF> EQUIV  
          TYPE  
          ;;TYPE A CR AND LF  
5$:     CLRB   $CHARCNT         ;;CLEAR CHARACTER COUNT  
          BR    2$              ;;GET NEXT CHARACTER  
          JSR   PC,$TYPEC       ;;GO TYPE THIS CHARACTER
```

```

15509 061242 126726 000124      6$:  CMPB  $FILLC,(SP)+  ;; IS IT TIME FOR FILLER CHARS.?
15510 061246 001350              BNE  2$                ;; IF NO GO GET NEXT CHAR.
15511 061250 016746 000114      MOV  $NULL,-(SP)      ;; GET # OF FILLER CHARS. NEEDED
15512                                ;; AND THE NULL CHAR.
15513 061254 105366 000001      7$:  DECB  1(SP)        ;; DOES A NULL NEED TO BE TYPED?
15514 061260 002770              BLT  6$                ;; BR IF NO--GO POP THE NULL OFF OF STACK
15515 061262 004767 000032      JSR  PC,$TYPEC       ;; GO TYPE A NULL
15516 061266 105367 000072      DECB  $CHARCNT       ;; DO NOT COUNT AS A COUNT
15517 061272 000770              BR   7$                ;; LOOP

```

:HORIZONTAL TAB PROCESSOR

```

15521 061274 112716 000040      8$:  MOVB  #' ,(SP)    ;; REPLACE TAB WITH SPACE
15522 061300 004767 000014      9$:  JSR  PC,$TYPEC    ;; TYPE A SPACE
15523 061304 132767 000007 000052  BITB  #7,$CHARCNT    ;; BRANCH IF NOT AT
15524 061312 001372              BNE  9$                ;; TAB STOP
15525 061314 005726              TST  (SP)+            ;; POP SPACE OFF STACK
15526 061316 000724              BR   2$                ;; GET NEXT CHARACTER
15527 061320 105777 161066      $TYPEC: TSTB @2$TPS    ;; WAIT UNTIL PRINTER IS READY
15528 061324 100375              BPL  $TYPEC
15529 061326 116677 000002 161054  MOVB  2(SP),@2$TPB    ;; LOAD CHAR TO BE TYPED INTO DATA REG.
15530 061334 122766 000015 000002  CMPB  #CR,2(SP)      ;; IS CHARACTER A CARRIAGE RETURN?
15531 061342 001003              BNE  1$                ;; BRANCH IF NO
15532 061344 105067 000014      CLRB  $CHARCNT       ;; YES--CLEAR CHARACTER COUNT
15533 061350 000406              BR   $TYPEX           ;; EXIT
15534 061352 122766 000012 000002  1$:  CMPB  #LF,2(SP)    ;; IS CHARACTER A LINE FEED?
15535 061360 001402              BEQ  $TYPEX           ;; BRANCH IF YES
15536 061362 105227              INCB (PC)+           ;; COUNT THE CHARACTER
15537 061364 000000      $CHARCNT: .WORD 0    ;; CHARACTER COUNT STORAGE
15538 061366 000207      $TYPEX:  RTS  PC

```

```

15539
15540 061370      000      $NULL:  .BYTE 0      ;; CONTAINS NULL CHARACTER FOR FILLS
15541 061371      002      $FILLS:  .BYTE 2      ;; CONTAINS # OF FILLER CHARACTERS REQUIRED
15542 061372      012      $FILLC:  .BYTE 12     ;; INSERT FILL CHARS. AFTER A 'LINE FEED'
15543 061373      000      $TPFLG:  .BYTE 0      ;; 'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
15544 061374      077      $QUES:  .ASCII '?'    ;; QUESTION MARK
15545 061375      012      000      $LF:  .ASCIIZ <12>  ;; LINEFEED
15546      061400

```

.SBTTL APT COMMUNICATIONS ROUTINE

```

15549
15550 061400 112767 000001 000236  $ATY1: MOVB  #1,$FFLG    ;; TO REPORT FATAL ERROR
15551 061406 112767 000001 000226  $ATY3: MOVB  #1,$MFLG    ;; TO TYPE A MESSAGE
15552 061414 000403              BR   $ATYC
15553 061416 112767 000001 000220  $ATY4: MOVB  #1,$FFLG    ;; TO ONLY REPORT FATAL ERROR
15554 061424
15555 061424 010046              MOV  R0,-(SP)        ;; PUSH R0 ON STACK
15556 061426 010146              MOV  R1,-(SP)        ;; PUSH R1 ON STACK
15557 061430 105767 000206      TSTB  $MFLG          ;; SHOULD TYPE A MESSAGE?
15558 061434 001450              BEQ  5$              ;; IF NOT: BR
15559 061436 122767 000001 116654  CMPB  #APTENV,$ENV    ;; OPERATING UNDER APT?
15560 061444 001031              BNE  3$              ;; IF NOT: BR
15561 061446 132767 000100 116645  BITB  #APTSPOOL,$ENVM ;; SHOULD SPOOL MESSAGES?
15562 061454 001425              BEQ  3$              ;; IF NOT: BR
15563 061456 017600 000004      MOV  @4(SP),R0       ;; GET MESSAGE ADDR.
15564 061462 062766 000002 000004  ADD  #2,4(SP)        ;; BUMP RETURN ADDR.

```



```

15565 061470 005767 116604      1$:   TST   $MSGTYPE      ;;SEE IF DONE W/ LAST XMISSION?
15566 061474 001375                BNE   1$                ;;IF NOT: WAIT
15567 061476 010067 116612      MOV   R0,$MSGAD        ;;PUT ADDR IN MAILBOX
15568 061502 105720                TSTB  (R0)+            ;;FIND END OF MESSAGE
15569 061504 001376                BNE   2$                ;;
15570 061506 166700 116602      SUB   $MSGAD,R0        ;;SUB START OF MESSAGE
15571 061512 006200                ASR   R0                ;;GET MESSAGE LNGTH !N WORDS
15572 061514 010067 116576      MOV   R0,$MSGGLT       ;;PUT LENGTH IN MAILBOX
15573 061520 012767 000004 116552  MOV   #4,$MSGTYPE      ;;TELL APT TO TAKE MSG.
15574 061526 000413                BR    5$                ;;
15575 061530 017667 000004 000016 3$:   MOV   @4(SP),4$        ;;PUT MSG ADDR IN JSR LINKAGE
15576 061536 062766 000002 000004  ADD   #2,4(SP)         ;;BUMP RETURN ADDRESS
15577 061544 016746 116226      MOV   177776,-(SP)    ;;PUSH 177776 ON STACK
15578 061550 004767 177332      JSR   PC,$TYPE        ;;CALL TYPE MACRO
15579 061554 000000                4$:   .WORD 0
15580 061556                5$:
15581 061556 105767 000062      10$:  TSTB  $FFLG           ;;SHOULD REPORT FATAL ERROR?
15582 061562 001416                BEQ   12$              ;;IF NOT: BR
15583 061564 005767 116530      TST   $ENV            ;;RUNNING UNDER APT?
15584 061570 001413                BEQ   12$              ;;IF NOT: BR
15585 061572 005767 116502      11$:  TST   $MSGTYPE       ;;FINISHED LAST MESSAGE?
15586 061576 001375                BNE   11$              ;;IF NOT: WAIT
15587 061600 017667 000004 116474  MOV   @4(SP),$FATAL    ;;GET ERROR #
15588 061606 062766 000002 000004  ADD   #2,4(SP)         ;;BUMP RETURN ADDR.
15589 061614 005267 116460      INC   $MSGTYPE        ;;TELL APT TO TAKE ERROR
15590 061620 105067 000020      12$:  CLRB  $FFLG           ;;CLEAR FATAL FLAG
15591 061624 105067 000013      CLRB  $LFLG           ;;CLEAR LOG FLAG
15592 061630 105067 000006      CLRB  $MFLG          ;;CLEAR MESSAGE FLAG
15593 061634 012601                MOV   (SP)+,R1        ;;POP STACK INTO R1
15594 061636 012600                MOV   (SP)+,R0        ;;POP STACK INTO R0
15595 061640 000207                RTS   PC              ;;RETURN
15596 061642 000                $MFLG: .BYTE 0        ;;MESSG. FLAG
15597 061643 000                $LFLG: .BYTE 0        ;;LOG FLAG
15598 061644 000                $FFLG: .BYTE 0        ;;FATAL FLAG
15599                .EVEN
15600                APTSIZE=200
15601                APTENV=001
15602                APTSPOOL=100
15603                APTCSUP=040
15604                .SBTTL CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
15605
15606                ;*****
15607                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
15608                ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
15609                ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
15610                ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
15611                ;*REPLACED WITH SPACES.
15612                ;*CALL:
15613                ;*   MOV   NUM,-(SP)      ;;PUT THE BINARY NUMBER ON THE STACK
15614                ;*   TYPDS                ;;GO TO THE ROUTINE
15615
15616                $TYPDS:
15617                MOV   R0,-(SP)        ;;PUSH R0 ON STACK
15618                MOV   R1,-(SP)        ;;PUSH R1 ON STACK
15619                MOV   R2,-(SP)        ;;PUSH R2 ON STACK
15620                MOV   R3,-(SP)        ;;PUSH R3 ON STACK

```

```

15621 061656 010546      MOV      R5,-(SP)      ;;PUSH R5 ON STACK
15622 061660 012746 020200  MOV      #20200,-(SP)  ;;SET BLANK SWITCH AND SIGN
15623 061664 016605 000020  MOV      20(SP),R5    ;;GET THE INPUT NUMBER
15624 061670 100004      BPL      1$           ;;BR IF INPUT IS POS.
15625 061672 005405      NEG      R5           ;;MAKE THE BINARY NUMBER POS.
15626 061674 112766 000055 000001  MOVVB   #'-,1(SP)     ;;MAKE THE ASCII NUMBER NEG.
15627 061702 005000      CLR      R0           ;;ZERO THE CONSTANTS INDEX
15628 061704 012703 062062  MOV      #SDBLK,R3    ;;SETUP THE OUTPUT POINTER
15629 061710 112723 000040  MOVVB   #' ,(R3)+     ;;SET THE FIRST CHARACTER TO A BLANK
15630 061714 005002      CLR      R2           ;;CLEAR THE BCD NUMBER
15631 061716 016001 062052  MOV      $DTBL(R0),R1 ;;GET THE CONSTANT
15632 061722 160105      SUB      R1,R5        ;;FORM THIS BCD DIGIT
15633 061724 002402      BLT     4$           ;;BR IF DONE
15634 061726 005202      INC     R2           ;;INCREASE THE BCD DIGIT BY 1
15635 061730 000774      BR      3$
15636 061732 060105      ADD     R1,R5        ;;ADD BACK THE CONSTANT
15637 061734 005702      TST     R2           ;;CHECK IF BCD DIGIT=0
15638 061736 001002      BNE     5$           ;;FALL THROUGH IF 0
15639 061740 105716      TSTB   (SP)         ;;STILL DOING LEADING 0'S?
15640 061742 100407      BMI     7$           ;;BR IF YES
15641 061744 106316      ASLB   (SP)         ;;MSD?
15642 061746 103003      BCC     6$           ;;BR IF NO
15643 061750 116663 000001 177777  MOVVB   1(SP),-1(R3)  ;;YES--SET THE SIGN
15644 061756 052702 000060 6$:     BIS     #'0,R2      ;;MAKE THE BCD DIGIT ASCII
15645 061762 052702 000040 7$:     BIS     #' ,R2      ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
15646 061766 110223      MOVVB   R2,(R3)+     ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
15647 061770 005720      TST    (R0)+        ;;JUST INCREMENTING
15648 061772 020027 000010  CMP     R0,#10      ;;CHECK THE TABLE INDEX
15649 061776 002746      BLT    2$           ;;GO DO THE NEXT DIGIT
15650 062000 003002      BGT    8$           ;;GO TO EXIT
15651 062002 010502      MOV     R5,R2       ;;GET THE LSD
15652 062004 000764      BR     6$           ;;GO CHANGE TO ASCII
15653 062006 105726      TSTB   (SP)+        ;;WAS THE LSD THE FIRST NON-ZERO?
15654 062010 100003      BPL    9$           ;;BR IF NO
15655 062012 116663 177777 177776  MOVVB   -1(SP),-2(R3) ;;YES--SET THE SIGN FOR TYPING
15656 062020 105013      CLRB   (R3)         ;;SET THE TERMINATOR
15657 062022 012605      MOV    (SP)+,R5     ;;POP STACK INTO R5
15658 062024 012603      MOV    (SP)+,R3     ;;POP STACK INTO R3
15659 062026 012602      MOV    (SP)+,R2     ;;POP STACK INTO R2
15660 062030 012601      MOV    (SP)+,R1     ;;POP STACK INTO R1
15661 062032 012600      MOV    (SP)+,R0     ;;POP STACK INTO R0
15662 062034 104401 062062  TYPE    ,SDBLK      ;;NOW TYPE THE NUMBER
15663 062040 016666 000002 000004  MOV     2(SP),4(SP)  ;;ADJUST THE STACK
15664 062046 012616      MOV    (SP)+,(SP)
15665 062050 000002      RTI
15666 062052 023420      $DTBL: 10000.
15667 062054 001750      1000.
15668 062056 000144      100.
15669 062060 000012      10.
15670 062062 000004      $SDBLK: .BLKW 4
15671      .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
15672
15673      ;*****
15674      ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
15675      ;*OCTAL (ASCII) NUMBER AND TYPE IT.
15676      ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE

```

```
15677 : *CALL:
15678 : *      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
15679 : *      TYPOS    ;;CALL FOR TYPEOUT
15680 : *      .BYTE   N                ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
15681 : *      .BYTE   M                ;;M=1 OR 0
15682 : *                                     ;;1=TYPE LEADING ZEROS
15683 : *                                     ;;0=SUPPRESS LEADING ZEROS
15684 :
15685 : *$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
15686 : *$TYPOS OR $TYPOC
15687 : *CALL:
15688 : *      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
15689 : *      TYPON    ;;CALL FOR TYPEOUT
15690 :
15691 : *$TYPOC----ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
15692 : *CALL:
15693 : *      MOV      NUM,-(SP)      ;;NUMBER TO BE TYPED
15694 : *      TYPOC    ;;CALL FOR TYPEOUT
15695 :
15696 062072 017646 000000 $TYPOS: MOV      @ (SP),-(SP)      ;;PICKUP THE MODE
15697 062076 116667 000001 000211 MOVBS 1(SP), $OFILL      ;;LOAD ZERO FILL SWITCH
15698 062004 12667 000207 MOVBS (SP)+, $OMODE+1    ;;NUMBER OF DIGITS TO TYPE
15699 062110 062716 000002 ADD      #2,(SP)        ;;ADJUST RETURN ADDRESS
15700 062114 000406 BR      $TYPON
15701 062116 112767 000001 000171 $TYPOC: MOVBS #1, $OFILL      ;;SET THE ZERO FILL SWITCH
15702 062124 112767 000006 000165 MOVBS #6, $OMODE+1    ;;SET FOR SIX(6) DIGITS
15703 062132 112767 000005 000154 $TYPON: MOVBS #5, $OCNT      ;;SET THE ITERATION COUNT
15704 062140 010346 MOV      R3,-(SP)      ;;SAVE R3
15705 062142 010446 MOV      R4,-(SP)      ;;SAVE R4
15706 062144 010546 MOV      R5,-(SP)      ;;SAVE R5
15707 062146 116704 000145 MOVBS $OMODE+1, R4    ;;GET THE NUMBER OF DIGITS TO TYPE
15708 062152 005404 NEG      R4
15709 062154 062704 000006 ADD      #6, R4        ;;SUBTRACT IT FOR MAX. ALLOWED
15710 062160 110467 000132 MOVBS  R4, $OMODE      ;;SAVE IT FOR USE
15711 062164 116704 000125 MOVBS  $OFILL, R4      ;;GET THE ZERO FILL SWITCH
15712 062170 016605 000012 MOV      12(SP), R5    ;;PICKUP THE INPUT NUMBER
15713 062174 005003 CLR      R3            ;;CLEAR THE OUTPUT WORD
15714 062176 006105 1$: ROL      R5            ;;ROTATE MSB INTO 'C'
15715 062200 000404 BR      3$           ;;GO DO MSB
15716 062202 006105 2$: ROL      R5            ;;FORM THIS DIGIT
15717 062204 006105 ROL      R5
15718 062206 006105 ROL      R5
15719 062210 010503 MOV      R5, R3
15720 062212 006103 3$: ROL      R3            ;;GET LSB OF THIS DIGIT
15721 062214 105367 000076 DECB   $OMODE          ;;TYPE THIS DIGIT?
15722 062220 100016 BPL    7$             ;;BR IF NO
15723 062222 042703 177770 BIC    #177770, R3    ;;GET RID OF JUNK
15724 062226 001002 BNE    4$             ;;TEST FOR 0
15725 062230 005704 TST    R4             ;;SUPPRESS THIS 0?
15726 062232 001403 BEQ    5$             ;;BR IF YES
15727 062234 005204 4$: INC      R4            ;;DON'T SUPPRESS ANYMORE 0'S
15728 062236 052703 000060 BIS    #'0, R3        ;;MAKE THIS DIGIT ASCII
15729 062242 052703 000040 5$: BIS    #' , R3      ;;MAKE ASCII IF NOT ALREADY
15730 062246 110367 000040 MOVBS  R3, 8$         ;;SAVE FOR TYPING
15731 062252 104401 062312 TYPE   , 8$           ;;GO TYPE THIS DIGIT
15732 062256 105367 000032 7$: DECB   $OCNT          ;;COUNT BY 1
```

```

15733 062262 003347          BGT      2$          ;;BR IF MORE TO DO
15734 062264 002402          BLT      6$          ;;BR IF DONE
15735 062266 005204          INC      R4          ;;INSURE LAST DIGIT ISN'T A BLANK
15736 062270 000744          BR       2$          ;;GO DO THE LAST DIGIT
15737 062272 012605          6$:      MOV      (SP)+,R5      ;;RESTORE R5
15738 062274 012604          MOV      (SP)+,R4      ;;RESTORE R4
15739 062276 012603          MOV      (SP)+,R3      ;;RESTORE R3
15740 062300 016666 000002 000004  MOV      2(SP),4(SP)    ;;SET THE STACK FOR RETURNING
15741 062306 012616          MOV      (SP)+,(SP)
15742 062310 000002          RTI
15743 062312 000          8$:      .BYTE   0          ;;RETURN
15744 062313 000          .BYTE   0          ;;STORAGE FOR ASCII DIGIT
15745 062314 000          $OCNT:  .BYTE   0          ;;TERMINATOR FOR TYPE ROUTINE
15746 062315 000          $OFILL: .BYTE   0          ;;OCTAL DIGIT COUNTER
15747 062316 000000          $OMODE: .WORD   0          ;;ZERO FILL SWITCH
15748          .SBTTL  TRAP DECODER      ;;NUMBER OF DIGITS TO TYPE
15749
15750          ;*****
15751          ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE 'TRAP' INSTRUCTION
15752          ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
15753          ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
15754          ;*GO TO THAT ROUTINE.
15755
15756 062320 010046          $TRAP:  MOV      R0,-(SP)      ;;SAVE R0
15757 062322 016600 000002      MOV      2(SP),R0          ;;GET TRAP ADDRESS
15758 062326 005740          TST     -(R0)             ;;BACKUP BY 2
15759 062330 111000          MOV     (R0),R0          ;;GET RIGHT BYTE OF TRAP
15760 062332 006300          ASL    R0                ;;POSITION FOR INDEXING
15761 062334 016000 062354      MOV     $TRPAD(R0),R0     ;;INDEX TO TABLE
15762 062340 000200          RTS    R0                ;;GO TO ROUTINE
15763
15764
15765          ;;THIS IS USE TO HANDLE THE 'GETPRI' MACRO
15766
15767 062342 011646          $TRAP2: MOV     (SP),-(SP)   ;;MOVE THE PC DOWN
15768 062344 016666 000004 000002  MOV     4(SP),2(SP)      ;;MOVE THE PSW DOWN
15769 062352 000002          RTI     ;;RESTORE THE PSW
15770
15771          .SBTTL  TRAP TABLE
15772
15773          ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED
15774          ;*BY THE 'TRAP' INSTRUCTION.
15775
15776          :      ROUTINE
15777          :      -----
15778 062354 062342          $TRPAD: .WORD   $TRAP2
15779 062356 061106          $TYPE  ;;CALL=TYPE      TRAP+1(104401)  TTY TYPEOUT ROUTINE
15780 062360 062116          $TYPOC ;;CALL=TYPOC     TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
15781 062362 062072          $TYPOS ;;CALL=TYPOS     TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
15782 062364 062132          $TYPON ;;CALL=TYPON      TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
15783 062366 061646          $TYPDS ;;CALL=TYPDS     TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
15784
15785
15786 062370 177777          PASSPT: -1
15787          .END

```


AMTYP4=	000000	345							
APASS =	000000	345	350						
APRIOR=	000000	345							
APTCSU=	000040	15491	15603#						
APTENV=	000001	11013	11044	11077	11158	11203	15484	15559	15601#
APTSIZ=	000200	15600#							
APTSPO=	000100	15486	15561	15602#					
AROUN	025300	7995	7998#						
AROUND	041772	11451	11459#						
ASL1	021716	6925	6926	6927	6933#				
ASL2	021726	6928	6937#						
ASL3	021744	6940	6941	6942	6948#				
ASL4	021754	6943	6952#						
ASL5	021770	6955	6956	6962#					
ASL6	022000	6957	6966#						
ASL7	022024	6969	6970	6971	6972	6979#			
ASR1	022066	6995	6996	6997	7003#				
ASR2	022076	6998	7007#						
ASR3	022120	7011	7012	7013	7019#				
ASR4	022130	7014	7023#						
ASR5	022144	7026	7027	7028	7034#				
ASR6	022154	7029	7038#						
ASR7	022204	7042	7043	7044	7045	7052#			
ASTART	042516	11672#	11907	11911					
ASWREG=	000000	345	358						
ATESTN=	000000	345	349						
ATRAP	037562	10911	10944#						
AUNIT =	000000	345	352						
AUSWR =	000000	345	359						
AUTO1	037670	10986#							
AVECT1=	000000	345							
AVECT2=	000000	345							
BELL =	000240	8664#							
BIC1	017346	6016	6017	6023#					
BIC2	017356	6018	6027#						
BIC3	017374	6030	6031	6037#					
BIS1	017436	6052	6053	6054	6060#				
BIS2	017446	6055	6064#						
BIS3	017466	6067	6068	6069	6075#				
BITCHK	025466	8051#							
BITCLR	025416	8038#							
BITCON	025550	8064	8073#						
BITSET	025432	8044#							
BIT1	017256	5979	5980	5986#					
BIT2	017266	5981	5991#						
BIT3	017304	5994	5995	6001#					
BRA1	001074	457	463#						
BRA2	001104	458	467#						
BRA3	001116	468	476#						
BRA4	001124	477	483#						
BRA5	001134	478	487#						
BRCT	025336	7982*	8012#	8018*					
BRC1	003350	1341	1347#						
BRC2	003360	1342	1352#						
BRC3	003370	1354	1360#						
BRH	025310	7983*	8003#						

BRMFPD	027120	8519	8523#											
BRN1	003230	1257	1263#											
BRN2	003240	1258	1268#											
BRN3	003250	1270	1276#											
BRTAB	027214	7980	8549#											
BRV1	003300	1299	1305#											
BRV2	003310	1300	1310#											
BRV3	003320	1312	1318#											
BRZ1	003160	1210	1216#											
BRZ2	003170	1211	1221#											
BRZ3	003200	1223	1229#											
BR1	027544	8709	8718#											
BR10	027760	8797	8806#											
BR11	030064	8830	8839#											
BR12	030132	8845	8854#											
BR13	030200	8860	8869#											
BR14	030246	8875	8884#											
BR15	030344	8906	8915#											
BR16	030364	8916	8925#											
BR17	030404	8926	8935#											
BR2	027572	8721	8730#											
BR20	030424	8936	8944#											
BR21	030444	8945	8954#											
BR22	030464	8955	8964#											
BR23	030504	8965	8974#											
BR3	027614	8733	8742#											
BR33	030554	8994	9002#											
BR34	030566	9003	9011#											
BR35	030600	9012	9020#											
BR36	030612	9021	9029#											
BR37	030634	9032	9040#											
BR4	027640	8746	8755#											
BR40	030646	9041	9049#											
BR41	030660	9050	9058#											
BR45	032206	9463	9469#											
BR46	032754	9569	9667#											
BR46A	032764	9662	9671#											
BR47	033544	9870	9876#											
BR5	027664	8759	8768#											
BR51	034312	9978	10076#											
BR51A	034322	10071	10080#											
BR6	027710	8772	8780#											
BR7	027734	8784	8793#											
BR70	037300	10844	10859#											
BR71	040230	11041	11053	11062#										
BTCON	025624	8092	8097#											
BTERR	025610	8087	8093#											
BTRAP	037636	10952	10970#											
BUFF	001000	8672#	9074	9088	9091	9107	9110	9126	9131	9140	9145	9161	9210	9272
		9285	9288	9304	9307	9323	9328	9337	9341	9357	9406	9467	9484	9497
		9500	9516	9519	9535	9540	9549	9554	9570	9619	9679	9692	9695	9711
		9714	9730	9735	9744	9749	9765	9814	9874	9892	9905	9908	9924	9927
		9943	9948	9958	9963	9979	10028	10095	10108	10111	10127	10130	10146	10151
		10160	10165	10181	10230	10287	10300	10303	10319	10322	10339	10344	10353	10358
		10374	10423	10696	10712	10721	10737	10743	10762	10787	10817	10846	10871	10877
		10912	10934	10953	10980	10996	11049	11082	11135	11164	11179	11209	11233	11254

DNMB3B	011462	3957	3966#															
DNMB3C	011500	3967	3975#															
DNMB3D	011516	3978	3979	3985#														
DNMB3E	011526	3980	3989#															
DNMB4A	011716	4057	4058	4059	4065#													
DNMB4B	011726	4060	4069#															
DNMB4C	011744	4070	4078#															
DNMB4D	011754	4079	4085#															
DNMB4E	011764	4080	4089#															
DNMB4F	012000	4090	4098#															
DNM03A	010136	3373	3374	3375	3381#													
DNM03B	010146	3376	3385#															
DNM03C	010156	3386	3393#															
DNM1	010010	3307	3315#															
DNM1A	011122	3807	3808	3809	3815#													
DNM1B	011132	3810	3819#															
DNM2	010024	3316	3324#															
DNM2A	011200	3840	3841	3847#														
DNM2B	011210	3842	3851#															
DNM2C	011216	3858#																
DNM2D	011226	3853	3862#															
DNM3	010042	3326	3334#															
DNM4	010064	3338	3346#															
DNM4A	011610	4013	4014	4015	4021#													
DNM4B	011620	4016	4025#															
DNM4C	011634	4026	4034#															
DNM5A	012060	4121	4122	4123	4129#													
DNM5B	012070	4124	4133#															
DNM5C	012106	4134	4142#															
DNM6A	012166	4164	4165	4166	4172#													
DNM6B	012176	4167	4176#															
DNM6C	012214	4177	4185#															
DNM7A	012276	4207	4208	4209	4215#													
DNM7B	012306	4210	4219#															
DNM7C	012324	4220	4228#															
DOAGIN	060472	15372	15378#															
DOPB2A	010550	3624	3632#															
DOPB2B	010626	3663	3671#															
DOP0A	007552	3189	3197#															
DOP0B	007576	3202	3210#															
DOP0C	007616	3213	3221#															
DOP0D	007646	3228	3236#															
DOP03A	007724	3267	3268	3269	3275#													
DOP03B	007734	3270	3279#															
DOP1	010362	3513	3521#															
DOP2	010474	3583	3591#															
DOP4	014000	4798	4810#															
DOP5	014064	4836	4848#															
DTRAP1	037620	10956#	10980															
DUMMY	000000	11593#	12561	12595	12630	12665	12699	12734	12769									
END1	027204	8530	8543#															
ENT176	046730	12539	12546#															
ENT51	044270	11966	11970#															
ER	025322	7979	7994	7996	8008#													
ERRNM =	000442	11594#	11684	11689#	11693	11698#	11723	11728#	11732	11737#	11761	11766#	11770	11775#				
		11799	11804#	11808	11813#	11838	11843#	11847	11852#	11878	11883#	11887	11892#	11972				

11977#	11992	11997#	12000	12005#	12021	12026#	12029	12034#	12050	12055#	12058	12063#
12079	12084#	12087	12092#	12108	12113#	12116	12121#	12137	12142#	12145	12150#	12166
12171#	12174	12179#	12195	12200#	12203	12208#	12224	12229#	12232	12237#	12253	12258#
12261	12266#	12306	12311#	12314	12319#	12322	12327#	12330	12335#	12359	12364#	12367
12372#	12375	12380#	12382	12387#	12412	12417#	12420	12425#	12428	12433#	12435	12440#
12548	12553#	12568	12573#	12575	12580#	12582	12587#	12602	12607#	12609	12614#	12616
12621#	12637	12642#	12644	12649#	12652	12657#	12672	12677#	12679	12684#	12686	12691#
12706	12711#	12713	12718#	12720	12725#	12741	12746#	12748	12753#	12756	12761#	12776
12781#	12783	12788#	12790	12795#	12810	12815#	12817	12822#	12824	12829#	12831	12836#
12856	12861#	12863	12868#	12871	12876#	12893	12898#	12900	12905#	12908	12913#	12930
12935#	12937	12942#	12945	12950#	12967	12972#	12974	12979#	12982	12987#	13004	13009#
13011	13016#	13019	13024#	13041	13046#	13048	13053#	13056	13061#	13078	13083#	13085
13090#	13093	13098#	13115	13120#	13122	13127#	13130	13135#	13152	13157#	13159	13164#
13167	13172#	13189	13194#	13196	13201#	13204	13209#	13238	13243#	13245	13250#	13253
13258#	13273	13278#	13280	13285#	13288	13293#	13308	13313#	13315	13320#	13323	13328#
13344	13349#	13351	13356#	13359	13364#	13380	13385#	13387	13392#	13395	13400#	13415
13420#	13422	13427#	13430	13435#	13451	13456#	13458	13463#	13466	13471#	13487	13492#
13494	13499#	13502	13507#	13522	13527#	13529	13534#	13537	13542#	13557	13562#	13564
13569#	13572	13577#	13593	13598#	13600	13605#	13608	13613#	13629	13634#	13636	13641#
13644	13649#	13664	13669#	13671	13676#	13679	13684#	13699	13704#	13706	13711#	13714
13719#	13735	13740#	13742	13747#	13750	13755#	13771	13776#	13778	13783#	13786	13791#
13806	13811#	13813	13818#	13821	13826#	13842	13847#	13849	13854#	13857	13862#	13882
13887#	13889	13894#	13897	13902#	13917	13922#	13924	13929#	13932	13937#	13952	13957#
13959	13964#	13967	13972#	13987	13992#	13994	13999#	14002	14007#	14022	14027#	14029
14034#	14037	14042#	14057	14062#	14064	14069#	14072	14077#	14092	14097#	14099	14104#
14107	14112#	14127	14132#	14134	14139#	14142	14147#	14162	14167#	14169	14174#	14177
14182#	14197	14202#	14204	14209#	14212	14217#	14240	14245#	14248	14253#	14256	14261#
14263	14268#	14282	14287#	14290	14295#	14298	14303#	14305	14310#	14325	14330#	14333
14338#	14341	14346#	14349	14354#	14368	14373#	14376	14381#	14384	14389#	14391	14396#
14410	14415#	14418	14423#	14426	14431#	14433	14438#	14453	14458#	14461	14466#	14469
14474#	14477	14482#	14496	14501#	14504	14509#	14512	14517#	14519	14524#	14538	14543#
14546	14551#	14554	14559#	14561	14566#	14580	14585#	14588	14593#	14596	14601#	14603
14608#	14623	14628#	14631	14636#	14639	14644#	14647	14652#	14666	14671#	14674	14679#
14682	14687#	14689	14694#	14708	14713#	14716	14721#	14724	14729#	14731	14736#	14751
14756#	14759	14764#	14767	14772#	14775	14780#	14794	14799#	14802	14807#	14810	14815#
14817	14822#	14836	14841#	14844	14849#	14852	14857#	14859	14864#	14880	14885#	14889
14894#	14909	14914#	14917	14922#	14940	14945#	14948	14953#	14956	14961#	14963	14968#
14982	14987#	14990	14995#	14998	15003#	15005	15010#	15024	15029#	15032	15037#	15040
15045#	15047	15052#	15066	15071#	15074	15079#	15082	15087#	15089	15094#	15108	15113#
15116	15121#	15124	15129#	15131	15136#	15150	15155#	15158	15163#	15166	15171#	15173
15178#	15192	15197#	15200	15205#	15208	15213#	15215	15220#	15234	15239#	15242	15247#
15250	15255#	15257	15262#	15276	15281#	15284	15289#	15292	15297#	15299	15304#	15318
15323#	15326	15331#	15334	15339#	15341	15346#						
11595#	11982	12011#	12040#	12069#	12098#	12127#	12156#	12185#	12214#	12243#	12272#	
11453*	11471	11547#										
10665	10667	10683#										
11453	11469	11543#										
GIN1	11463#	11470	11504									
GIN2	11474#	11505										
GIN3	11472	11479#										
GNS =	***** U	15779	15780	15781	15782	15783						
HERE =	000000	8678#	8707*	8719*	8807							
HICORE	037472	10905*	10908*	10915#	10947							
HLT =	000000	8652#										
HT -	000011	15499	15540									
IJMP	016362	5702	5705#									

F 000063
FINISH 042320
FOVER 036526
FPP 042310
GIN1 042012
GIN2 042054
GIN3 042102
GNS = ***** U
HERE = 000000
HICORE 037472
HLT = 000000
HT - 000011
IJMP 016362

MDM6D	013520	4684	4692#			
MDM6E	013550	4695	4703#			
MDM7A	013624	4731	4732	4738#		
MDM7B	013634	4733	4742#			
MDM7C	013652	4743	4751#			
MDM7D	013672	4752	4760#			
MDM7E	013716	4762	4770#			
MFPI0	025006	7907	7912#			
MFPI0A	025034	7914	7918#			
MFPS1	023532	7518	7527#			
MFPS2A	023622	7552	7553	7554	7560#	
MFPS2B	023632	7555	7564#			
MFPS2C	023652	7565	7573#			
MFPS3A	023730	7594	7595	7596	7602#	
MFPS3B	023740	7597	7606#			
MFPS3C	023760	7607	7615#			
MFPS4A	024036	7636	7637	7638	7644#	
MFPS4B	024046	7639	7648#			
MFPS4C	024066	7649	7657#			
MFPS5A	024144	7678	7679	7680	7686#	
MFPS5B	024154	7681	7690#			
MFPS5C	024174	7691	7699#			
MFPS6A	024254	7720	7721	7722	7728#	
MFPS6B	024264	7723	7732#			
MFPS6C	024304	7733	7741#			
MFPS7A	024364	7762	7763	7764	7770#	
MFPS7B	024374	7765	7774#			
MFPS7C	024414	7775	7783#			
MFPT	000007	11392#	11397			
MOR0	026106	8193	8204#			
MOR1	026146	8216	8225#			
MOR2	026226	8249	8258#			
MOR3	026320	8284	8293#			
MOR4	026340	8294	8303#			
MOR5	026360	8305	8314#			
MOR6	026450	8338	8347#			
MOR7	026470	8349	8359#			
MOR8	026510	8360	8369#			
MOV1	017166	5942	5943	5949#		
MOV2	017176	5944	5954#			
MOV3	017214	5957	5958	5964#		
MRK1	022572	7228	7235#			
MRK2	022614	7235	7236	7237	7239	7246#
MRK3	022624	7241	7250#			
MRK4	022646	7253	7255#			
MRK5	022660	7254	7259#			
MRK6	022674	7260	7267#			
MSGERR	060740	15411	15432#			
MSGPWF	061066	15449	15457#			
MSG1	060550	15358	15390#			
MTP10	025122	7933	7938#			
MTPS1	022744	7293	7301#			
MTPS1A	022764	7305	7306	7307	7313#	
MTPS2	023040	7330	7338#			
MTPS3	023130	7361	7369#			
MTPS4	023216	7391	7399#			

REG4E	002430	961	969#	
REG45	045742	12395	12398#	
REG5	002532	1010#	1013	
REG5A	002576	1034#	1037	
REG5E	002544	1011	1019#	
REG6	002654	1060#	1063	
REG6A	002720	1084#	1087	
REG6E	002666	1061	1069#	
RESET2	040570	11136	11146#	
RESET3	040560	11134	11142#	
REST	024472	7820#		
RESTR1	001024	445#	15385	15456
RET	042112	11475	11483#	
RETA	030716	9073	9077#	
RETAH	030726	9075	9081#	
RETAT	036616	10697	10705#	
RETA1	031472	9273	9278#	
RETA2	032300	9485	9490#	
RETA3	033032	9680	9685#	
RETA4	033636	9893	9898#	
RETA5	034370	10096	10101#	
RETB	030752	9089	9091#	
RETB1	036664	10713	10721#	
RETB2	031516	9286	9288#	
RETB3	032324	9498	9500#	
RETB4	033056	9693	9695#	
RETB5	033662	9906	9908#	
RETC	034414	10109	10111#	
RETC	031014	9108	9110#	
RETC1	036736	10738	10743#	
RETC1	031560	9305	9307#	
RETC2	032366	9517	9519#	
RETC3	033120	9712	9714#	
RETC4	033724	9925	9927#	
RETC5	034456	10128	10130#	
RETD	031066	9127	9131#	
RETD1	031632	9324	9328#	
RETD2	032440	9536	9540#	
RETD3	033172	9731	9735#	
RETD4	033776	9944	9948#	
RETD5	034530	10147	10151#	
RETE	031132	9141	9145#	
RETE1	031674	9338	9341#	
RETE2	032504	9550	9554#	
RETE3	033236	9745	9749#	
RETE4	034042	9959	9963#	
RETE5	034574	10161	10165#	
RETF	031202	9162	9165#	
RETF1	031744	9358	9361#	
RETF2	032554	9571	9574#	
RETF3	033306	9766	9769#	
RETF4	034112	9980	9983#	
RETF5	034644	10182	10185#	
RETG	031314	9211	9214#	
RETG1	032056	9407	9410#	
RETG2	032666	9620	9623#	

RETG3	033420	9815	9818#								
RETG4	034224	10029	10032#								
RETG5	034756	10231	10234#								
RETH5	035104	10288	10293#								
RETJ	035130	10301	10303#								
RETK	035172	10320	10322#								
RETL	035244	10340	10344#								
RETM	035310	10354	10358#								
RETN	035360	10375	10378#								
RETO	035470	10424	10427#								
RETR1	041212	11255	11260#								
RETR2	041270	11281	11290#								
RETR3	041336	11307	11313#								
RET1	042130	11484	11488#								
RET2	042150	11489	11493#								
RET3	042166	11494	11503#								
RET4	042156	11445	11499#								
RITSH	046152	12344	12396	12450	12453#						
ROL1	021406	6790	6791	6792	6798#						
ROL2	021416	6793	6802#								
ROL3	021434	6805	6806	6807	6813#						
ROL4	021444	6808	6817#								
ROL5	021460	6820	6821	6827#							
ROL6	021470	6822	6831#								
ROL7	021512	6834	6835	6836	6843#						
ROR1	021554	6858	6859	6860	6866#						
ROR2	021564	6861	6870#								
ROR3	021602	6873	6874	6875	6881#						
ROR4	021612	6876	6885#								
ROR5	021630	6888	6889	6890	6896#						
ROR6	021640	6891	6900#								
ROR7	021654	6903	6904	6910#							
POTX	015172	5209	5211*	5215	5228#	5243*	5245*	5247	5273*	5274	5278
ROTXAD	015316	5274*	5276*	5289#							
ROTOA	014310	4938	4939	4946#							
ROTOB	014320	4941	4950#								
ROTOC	014342	4953	4954	4961#							
ROT1A	014410	4987	4988	4995#							
ROT1B	014420	4990	4999#								
ROT1C	014444	5002	5003	5010#							
ROT1D	014454	5005	5014#								
ROT1E	014504	5019	5020	5027#							
ROT2A	014556	5050	5052	5060#							
ROT2B	014566	5055	5064#								
ROT2C	014616	5068	5070	5077#							
ROT2D	014626	5072	5081#								
ROT2E	014662	5086	5088	5096#							
ROT3A	014730	5119	5126#								
ROT3B	014740	5121	5130#								
ROT3C	014766	5133	5140#								
ROT3D	014776	5135	5144#								
ROT3E	015024	5147	5154#								
ROT4	015100	5178	5180	5187#							
ROT5	015162	5208	5214	5216	5223#						
ROT6	015232	5246	5253#								
ROT7	015306	5272	5277	5284#							

SKPMSG	060406	15357	15359#		
SKP104	037752	11004#			
SKTST2	040606	11149#			
SNMBOA	006300	2601	2602	2608#	
SNMB1A	006404	2666	2667	2673#	
SNMB1B	006414	2668	2677#		
SNMB1C	006436	2682	2683	2684	2690#
SNMB2A	006560	2758	2759	2765#	
SNMB2B	006570	2760	2769#		
SNMB2C	006604	2770	2778#		
SNMB2D	006624	2782	2783	2784	2790#
SNMB2E	006634	2785	2794#		
SNMB3A	006776	2874	2875	2881#	
SNMB3B	007006	2876	2885#		
SNMB3C	007024	2888	2889	2890	2896#
SNMB3D	007034	2891	2900#		
SNM0A	006240	2568	2569	2570	2576#
SNM1A	006342	2633	2634	2635	2641#
SNM2A	006500	2714	2715	2716	2722#
SNM2B	006510	2717	2726#		
SNM3A	006710	2826	2827	2828	2834#
SNM3B	006720	2829	2838#		
SNM4A	007104	2931	2932	2938#	
SNM4B	007114	2933	2942#		
SNM5A	007166	2974	2975	2981#	
SNM5B	007176	2976	2985#		
SNM6A	007252	3017	3018	3024#	
SNM6B	007262	3019	3028#		
SNM7A	007334	3060	3061	3067#	
SNM7B	007344	3062	3071#		
SOB1	022454	7183#	7196		
SOB2	022462	7183	7184	7190#	
SOB3	022472	7185	7194#		
SOB4	022512	7197	7198	7199	7206#
SOPA	006156	2526	2534#		
SOPB	006176	2523	2536	2543#	
SOPBOA	003566	1493	1501#		
SOPBOB	003576	1502	1509#		
SOPB1A	003710	1571	1579#		
SOPB1B	003726	1580	1583	1590#	
SOPB1C	003772	1617	1625#		
SOPB1D	004012	1630	1637#		
SOPB2A	004146	1713	1721#		
SOPB2B	004166	1725	1733#		
SOPB2C	004236	1757	1765#		
SOPB2D	004262	1771	1779#		
SOPB3A	004750	2023	2031#		
SOPB3B	004774	2034	2036	2045#	
SOPB3C	005042	2076	2084#		
SOPB3D	005064	2090	2097#		
SOPX	006142	2525#	2534*	2535*	2548
SOPXAD	006206	2537*	2548#		
SOPZA	004060	1665	1673#		
SOPOA	003424	1404	1412#		
SOPOB	003444	1415	1423#		
SOPOC	003506	1452	1460#		

TST236	053004	13728#							
TST237	053106	13765#							
TST240	053204	13800#							
TST241	053302	13835#							
TST242	053420	13876#							
TST243	053516	13911#							
TST244	053614	13946#							
TST245	053712	13981#							
TST246	054006	14016#							
TST247	054102	14051#							
TST250	054176	14086#							
TST251	054274	14121#							
TST252	054372	14156#							
TST253	054466	14191#							
TST254	054562	14232#							
TST255	054672	14274#							
TST256	055002	14316#							
TST257	055116	14360#							
TST260	055226	14402#							
TST261	055336	14444#							
TST262	055452	14488#							
TST263	055562	14530#							
TST264	055672	14572#							
TST265	056002	14614#							
TST266	056116	14658#							
TST267	056226	14700#							
TST270	056336	14742#							
TST271	056452	14786#							
TST272	056562	14828#							
TST273	056672	14870#							
TST274	056764	14900#							
TST275	057066	14932#							
TST276	057176	14974#							
TST277	057306	15016#							
TST300	057416	15058#							
TST301	057524	15100#							
TST302	057632	15142#							
TST303	057740	15184#							
TST304	060050	15226#							
TST305	060160	15268#							
TST306	060266	15310#							
TST37	043716	11711	11749	11787	11825	11865	11906	11912#	
TST40	043752	11913	11919#						
TST41	043766	11920	11923#						
TST42	044004	11924	11927#						
TST43	044040	11928	11934#						
TST44	044072	11935	11940#						
TST45	044124	11941	11946#						
TST46	044156	11947	11952#						
TST47	044214	11953	11959#						
TST50	044244	11960	11965#						
TST51	044322	11986#							
TST52	044404	12015#							
TST53	044466	12044#							
TST54	044550	12073#							
TST55	044630	12102#							

TST56	044710	12131#			
TST57	044770	12160#			
TST60	045052	12189#			
TST61	045134	12218#			
TST62	045214	12247#			
TS1	001056	453#			
TS10	001456	623	627	637	650#
TS100	006446	2659	2685	2706#	
TS101	006526	2708	2728	2749#	
TS102	006652	2751	2796	2816#	
TS103	006736	2818	2840	2862#	
TS104	007052	2864	2902	2922#	
TS105	007130	2924	2943	2963#	
TS106	007214	2965	2987	3007#	
TS107	007276	3009	3029	3050#	
TS11	001526	652	660	674#	
TS110	007360	3052	3072	3092#	
TS111	007414	3094	3100	3117#	
TS112	007450	3119	3125	3142#	
TS113	007524	3144	3162	3184#	
TS114	007666	3186	3239	3258#	
TS115	007756	3260	3281	3300#	
TS116	010102	3302	3348	3365#	
TS117	010166	3367	3388	3409#	
TS12	001572	676	681	694#	
TS120	010224	3411	3418	3438#	
TS121	010262	3440	3447	3468#	
TS122	010320	3470	3477	3502#	
TS123	010400	3504	3523	3544#	
TS124	010440	3546	3554	3575#	
TS125	010512	3577	3593	3616#	
TS126	010566	3618	3633	3653#	
TS127	010644	3655	3673	3694#	
TS13	001642	696	704	738#	
TS130	010706	3696	3701	3720#	
TS131	010750	3722	3727	3747#	
TS132	011012	3749	3754	3767#	
TS133	011070	3769	3787	3800#	
TS134	011146	3802	3820	3833#	
TS135	011244	3835	3863	3876#	
TS136	011406	3878	3929	3943#	
TS137	011546	3945	3991	4004#	
TS14	001672	740	744	757#	
TS140	011654	4006	4035	4048#	
TS141	012016	4050	4099	4112#	
TS142	012126	4114	4143	4156#	
TS143	012234	4158	4186	4199#	
TS144	012344	4201	4229	4248#	
TS145	012420	4250	4268	4287#	
TS146	012514	4289	4318	4336#	
TS147	012650	4338	4384	4402#	
TS15	001724	759	762	775#	
TS150	013026	4404	4450	4469#	
TS151	013122	4471	4497	4518#	
TS152	013256	4520	4565	4587#	
TS153	013420	4589	4635	4656#	

TS154	013570	4658	4704	4724#
TS155	013736	4726	4771	4796#
TS156	014022	4805	4834#	
TS157	014106	4843	4872#	
TS16	001756	777	780	793#
TS160	014172	4874	4881	4903#
TS161	014256	4905	4912	4932#
TS162	014352	4934	4956	4980#
TS163	014514	4982	5022	5043#
TS164	014672	5045	5091	5113#
TS165	015034	5115	5149	5171#
TS166	015110	5173	5182	5206#
TS167	015174	5218	5240#	
TS17	002010	795	798	811#
TS170	015242	5242	5248	5270#
TS171	015320	5279	5302#	
TS172	015366	5304	5316	5336#
TS173	015430	5338	5343	5365#
TS174	015510	5367	5381	5402#
TS175	015552	5404	5408	5430#
TS176	015632	5432	5446	5470#
TS177	015720	5486	5513#	
TS2	001146	455	488	512#
TS20	002054	813	821	835#
TS200	015770	5520	5547#	
TS201	016042	5554	5603#	
TS202	016406	5708	5739#	
TS203	017064	5873	5896#	
TS204	017140	5898	5909	5936#
TS205	017224	5938	5959	5972#
TS206	017314	5974	5996	6009#
TS207	017404	6011	6032	6045#
TS21	002124	837	846	859#
TS210	017476	6047	6070	6096#
TS211	017614	6098	6136	6150#
TS212	017756	6152	6203	6228#
TS213	020014	6230	6237	6251#
TS214	020100	6253	6275	6288#
TS215	020166	6290	6312	6336#
TS216	020352	6338	6399	6413#
TS217	020470	6415	6452	6477#
TS22	002200	883#		
TS220	020614	6479	6519	6533#
TS221	020774	6535	6587	6601#
TS222	021034	6603	6610	6636#
TS223	021206	6638	6690	6703#
TS224	021354	6705	6758	6783#
TS225	021522	6785	6838	6851#
TS226	021664	6853	6905	6918#
TS227	022034	6920	6974	6988#
TS23	002260	885	905#	
TS230	022214	6990	7047	7071#
TS231	022322	7073	7103	7125#
TS232	022434	7127	7156	7177#
TS233	022522	7179	7201	7222#
TS234	022704	7224	7262	7286#

TS235	022774	7288	7308	7322#
TS236	023056	7324	7339	7352#
TS237	023146	7354	7370	7383#
TS24	002324	907	915	929#
TS240	023232	7385	7400	7413#
TS241	023314	7415	7430	7443#
TS242	023404	7445	7460	7473#
TS243	023474	7475	7490	7512#
TS244	023566	7514	7532	7545#
TS245	023670	7547	7574	7587#
TS246	023776	7589	7616	7629#
TS247	024104	7631	7658	7671#
TS25	002374	931	940	954#
TS250	024212	7673	7700	7713#
TS251	024322	7715	7742	7755#
TS252	024432	7757	7784	7805#
TS253	024472	7807	7811	7830#
TS254	024556	7832	7857#	
TS255	024730	7899#		
TS256	025034	7901	7923#	
TS257	025146	7925	7940	7977#
TS26	002440	956	964	978#
TS260	025406	8035#		
TS261	025554	8085#		
TS262	025630	8108#		
TS263	025670	8110	8117	8145#
TS264	026106	8208#		
TS265	026166	8210	8228	8241#
TS266	026246	8243	8261	8274#
TS267	026376	8276	8315	8328#
TS27	002510	980	989	1004#
TS270	026526	8330	8370	8390#
TS271	026562	8392	8396	8409#
TS272	026622	8411	8415	8428#
TS273	026670	8430	8435	8448#
TS274	026732	8450	8454	8467#
TS275	026772	8469	8473	8488#
TS276	027040	8490	8496	8509#
TS277	027120	8511	8528#	
TS3	001202	514	518	531#
TS30	002554	1006	1014	1028#
TS300	027510	8703#		
TS301	030006	8705	8809	8821#
TS302	030314	8823	8890	8902#
TS303	030524	8904	8975	8989#
TS304	030672	8991	9059	9071#
TS305	030726	9085#		
TS306	030770	9087	9092	9104#
TS307	031034	9106	9111	9123#
TS31	002624	1030	1039	1053#
TS310	031152	9125	9146	9158#
TS311	031424	9267#		
TS312	031472	9269	9282#	
TS313	031534	9284	9289	9301#
TS314	031600	9303	9308	9320#
TS315	031714	9322	9342	9354#

TS316	032154	9356	9449	9461#
TS317	032244	9481#		
TS32	002676	1055	1064	1078#
TS320	032300	9483	9494#	
TS321	032342	9496	9501	9513#
TS322	032406	9515	9520	9532#
TS323	032524	9534	9555	9567#
TS324	032776	9676#		
TS325	033032	9678	9689#	
TS326	033074	9691	9696	9708#
TS327	033140	9710	9715	9727#
TS33	002746	1080	1089	1117#
TS330	033256	9729	9750	9762#
TS331	033512	9764	9856	9868#
TS332	033602	9889#		
TS333	033636	9891	9902#	
TS334	033700	9904	9909	9921#
TS335	033744	9923	9928	9940#
TS336	034062	9942	9964	9976#
TS337	034334	10092#		
TS34	003006	1119	1123	1136#
TS340	034370	10094	10105#	
TS341	034432	10107	10112	10124#
TS342	034476	10126	10131	10143#
TS343	034614	10145	10166	10178#
TS344	035050	10180	10272	10284#
TS345	035104	10286	10297#	
TS346	035146	10299	10304	10316#
TS347	035212	10318	10323	10336#
TS35	003044	1138	1141	1154#
TS350	035330	10338	10359	10371#
TS351	035562	10373	10465	10478#
TS352	035616	10480	10492#	
TS353	035660	10494	10499	10512#
TS354	036002	10514	10547#	
TS355	036052	10562#		
TS356	036122	10577#		
TS357	036172	10592#		
TS36	003102	1156	1159	1172#
TS360	036242	10607#		
TS361	036312	10622#		
TS362	036400	10635	10647#	
TS363	036450	10663#		
TS364	036550	10693#		
TS365	036616	10695	10709#	
TS366	036702	10711	10722	10734#
TS367	036756	10736	10744	10758#
TS37	003140	1174	1177	1204#
TS370	037030	10760	10769	10783#
TS371	037130	10785	10814#	
TS372	037214	10816	10829	10842#
TS373	037324	10868#		
TS374	037412	10902#		
TS375	037752	10904	11008#	
TS376	040106	11039#		
TS377	040252	11072#		

\$EOPCT 060422
\$ERN = 001162

*5362#	15366												
317#	464	465#	473	474#	484	485#	493	494#	523	524#	542	543#	
561	562#	579	580#	613	614#	632	633#	642	643#	666	667#	686	
687#	710	711#	749	750#	767	768#	785	786#	803	804#	827	828#	
852	853#	875	876#	897	898#	921	922#	946	947#	970	971#	995	
996#	1020	1021#	1045	1046#	1070	1071#	1095	1096#	1128	1129#	1146	1147#	
1164	1165#	1182	1183#	1217	1218#	1230	1231#	1264	1265#	1277	1278#	1306	
1307#	1319	1320#	1348	1349#	1361	1362#	1409	1410#	1420	1421#	1429	1430#	
1457	1458#	1475	1476#	1498	1499#	1510	1511#	1536	1537#	1549	1550#	1576	
1577#	1591	1592#	1622	1623#	1638	1639#	1670	1671#	1686	1687#	1718	1719#	
1734	1735#	1762	1763#	1780	1781#	1808	1809#	1818	1819#	1833	1834#	1842	
1843#	1865	1866#	1875	1876#	1889	1890#	1898	1899#	1921	1922#	1935	1936#	
1944	1945#	1979	1980#	1993	1994#	2028	2029#	2046	2047#	2081	2082#	2098	
2099#	2125	2126#	2134	2135#	2145	2146#	2154	2155#	2165	2166#	2194	2195#	
2210	2211#	2250	2251#	2264	2265#	2292	2293#	2304	2305#	2334	2335#	2346	
2347#	2370	2371#	2379	2380#	2388	2389#	2415	2416#	2424	2425#	2434	2435#	
2460	2461#	2469	2470#	2493	2494#	2505	2506#	2531	2532#	2544	2545#	2577	
2578#	2609	2610#	2642	2643#	2674	2675#	2691	2692#	2723	2724#	2733	2734#	
2766	2767#	2775	2776#	2791	2792#	2801	2802#	2835	2836#	2845	2846#	2882	
2883#	2897	2898#	2907	2908#	2939	2940#	2948	2949#	2982	2983#	2992	2993#	
3025	3026#	3034	3035#	3068	3069#	3077	3078#	3105	3106#	3130	3131#	3158	
3159#	3167	3168#	3194	3195#	3207	3208#	3218	3219#	3233	3234#	3244	3245#	
3276	3277#	3286	3287#	3312	3313#	3321	3322#	3331	3332#	3343	3344#	3353	
3354#	3382	3383#	3394	3395#	3423	3424#	3452	3453#	3482	3483#	3518	3519#	
3528	3529#	3559	3560#	3588	3589#	3598	3599#	3629	3630#	3638	3639#	3668	
3669#	3678	3679#	3706	3707#	3732	3733#	3759	3760#	3783	3784#	3792	3793#	
3816	3817#	3825	3826#	3848	3849#	3859	3860#	3868	3869#	3893	3894#	3902	
3903#	3915	3916#	3925	3926#	3934	3935#	3963	3964#	3972	3973#	3986	3987#	
3996	3997#	4022	4023#	4031	4032#	4040	4041#	4066	4067#	4075	4076#	4086	
4087#	4095	4096#	4104	4105#	4130	4131#	4139	4140#	4148	4149#	4173	4174#	
4182	4183#	4191	4192#	4216	4217#	4225	4226#	4234	4235#	4264	4265#	4273	
4274#	4303	4304#	4314	4315#	4323	4324#	4350	4351#	4359	4360#	4371	4372#	
4380	4381#	4389	4390#	4417	4418#	4426	4427#	4435	4436#	4445	4446#	4455	
4456#	4484	4485#	4493	4494#	4502	4503#	4531	4532#	4540	4541#	4552	4553#	
4561	4562#	4570	4571#	4602	4603#	4611	4612#	4620	4621#	4631	4632#	4640	
4641#	4671	4672#	4680	4681#	4689	4690#	4700	4701#	4709	4710#	4739	4740#	
4748	4749#	4757	4758#	4767	4768#	4776	4777#	4811	4812#	4849	4850#	4886	
4887#	4917	4918#	4947	4948#	4962	4963#	4996	4997#	5011	5012#	5028	5029#	
5061	5062#	5078	5079#	5097	5098#	5127	5128#	5141	5142#	5155	5156#	5188	
5189#	5224	5225#	5254	5255#	5285	5286#	5312	5313#	5321	5322#	5348	5349#	
5377	5378#	5386	5387#	5413	5414#	5442	5443#	5451	5452#	5482	5483#	5492	
5493#	5526	5527#	5560	5561#	5615	5616#	5624	5625#	5638	5639#	5650	5651#	
5659	5660#	5672	5673#	5685	5686#	5699	5700#	5713	5714#	5754	5755#	5773	
5774#	5792	5793#	5810	5811#	5828	5829#	5844	5845#	5860	5861#	5879	5880#	
5905	5906#	5914	5915#	5950	5951#	5965	5966#	5987	5988#	6002	6003#	6024	
6025#	6038	6039#	6061	6062#	6076	6077#	6111	6112#	6127	6128#	6142	6143#	
6165	6166#	6179	6180#	6193	6194#	6209	6210#	6243	6244#	6266	6267#	6281	
6282#	6303	6304#	6318	6319#	6350	6351#	6364	6365#	6378	6379#	6390	6391#	
6405	6406#	6428	6429#	6443	6444#	6458	6459#	6493	6494#	6509	6510#	6525	
6526#	6548	6549#	6563	6564#	6579	6580#	6593	6594#	6616	6617#	6651	6652#	
6666	6667#	6681	6682#	6696	6697#	6719	6720#	6734	6735#	6749	6750#	6764	
6765#	6799	6800#	6814	6815#	6828	6829#	6844	6845#	6867	6868#	6882	6883#	
6897	6898#	6911	6912#	6934	6935#	6949	6950#	6963	6964#	6980	6981#	7004	
7005#	7020	7021#	7035	7036#	7053	7054#	7089	7090#	7109	7110#	7144	7145#	
7162	7163#	7191	7192#	7207	7208#	7232	7233#	7247	7248#	7256	7257#	7268	
7269#	7298	7299#	7314	7315#	7335	7336#	7344	7345#	7366	7367#	7375	7376#	

7396	7397#	7405	7406#	7426	7427#	7435	7436#	7456	7457#	7465	7466#	7486
7487#	7495	7496#	7523	7524#	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#	7771	7772#
7780	7781#	7789	7790#	7816	7817#	7841	7842#	7866	7867#	7876	7877#	7884
7885#	7909	7910#	7915	7916#	7935	7936#	7945	7946#	8009	8010#	8070	8071#
8094	8095#	8123	8124#	8163	8164#	8186	8187#	8221	8222#	8233	8234#	8254
8255#	8266	8267#	8289	8290#	8299	8300#	8310	8311#	8320	8321#	8343	8344#
8354	8355#	8365	8366#	8375	8376#	8401	8402#	8420	8421#	8440	8441#	8459
8460#	8478	8479#	8501	8502#	8520	8521#	8540	8541#	8544	8545#	8594	8595#
8598	8599#	8602	8603#	8606	8607#	8610	8611#	8614	8615#	8618	8619#	8622
8623#	8714	8715#	8726	8727#	8738	8739#	8751	8752#	8764	8765#	8777	8778#
8789	8790#	8802	8803#	8814	8815#	8835	8836#	8850	8851#	8865	8866#	8880
8881#	8895	8896#	8911	8912#	8921	8922#	8931	8932#	8941	8942#	8950	8951#
8960	8961#	8970	8971#	8980	8981#	8999	9000#	9008	9009#	9017	9018#	9026
9027#	9037	9038#	9046	9047#	9055	9056#	9064	9065#	9078	9079#	9097	9098#
9116	9117#	9137	9138#	9151	9152#	9171	9172#	9180	9181#	9189	9190#	9198
9199#	9207	9208#	9220	9221#	9229	9230#	9238	9239#	9247	9248#	9259	9260#
9275	9276#	9294	9295#	9313	9314#	9334	9335#	9347	9348#	9367	9368#	9376
9377#	9385	9386#	9394	9395#	9403	9404#	9416	9417#	9425	9426#	9434	9435#
9443	9444#	9454	9455#	9470	9471#	9487	9488#	9506	9507#	9525	9526#	9540
9547#	9560	9561#	9580	9581#	9589	9590#	9598	9599#	9607	9608#	9616	9617#
9629	9630#	9638	9639#	9647	9648#	9656	9657#	9668	9669#	9682	9683#	9701
9702#	9720	9721#	9741	9742#	9755	9756#	9775	9776#	9784	9785#	9793	9794#
9802	9803#	9811	9812#	9824	9825#	9833	9834#	9842	9843#	9851	9852#	9861
9862#	9877	9878#	9895	9896#	9914	9915#	9933	9934#	9955	9956#	9969	9970#
9989	9990#	9998	9999#	10007	10008#	10016	10017#	10025	10026#	10038	10039#	10047
10048#	10056	10057#	10065	10066#	10077	10078#	10098	10099#	10117	10118#	10136	10137#
10157	10158#	10171	10172#	10191	10192#	10200	10201#	10209	10210#	10218	10219#	10227
10228#	10240	10241#	10249	10250#	10258	10259#	10267	10268#	10277	10278#	10290	10291#
10309	10310#	10328	10329#	10350	10351#	10364	10365#	10384	10385#	10393	10394#	10402
10403#	10411	10412#	10420	10421#	10433	10434#	10442	10443#	10451	10452#	10460	10461#
10470	10471#	10484	10485#	10504	10505#	10525	10526#	10532	10533#	10539	10540#	10555
10556#	10570	10571#	10585	10586#	10600	10601#	10615	10616#	10630	10631#	10640	10641#
10655	10656#	10684	10685#	10702	10703#	10718	10719#	10727	10728#	10749	10750#	10774
10775#	10796	10797#	10807	10808#	10824	10825#	10834	10835#	10856	10857#	10860	10861#
10884	10885#	10920	10921#	10931	10932#	10940	10941#	10964	10965#	10977	10978#	10987
10988#	11025	11026#	11029	11030#	11059	11060#	11063	11064#	11093	11094#	11121	11122#
11125	11126#	11143	11144#	11175	11176#	11191	11192#	11221	11222#	11230	11231#	11240
11241#	11257	11258#	11266	11267#	11291	11292#	11310	11311#	11319	11320#	11344	11345#
11348	11349#	11357	11358#	11366	11367#	11375	11376#	11407	11408#	11485	11486#	11490
11491#	11500	11501#										
	428#	448*										
\$ERROR-	000302											
\$ETABL	000320	355#										
\$ETEND	000330	367#	390									
\$FATAL	000302	348#	428	15406*	15417	15587*						
\$FFLG	061644	15550*	15553*	15581	15590*	15598#						
\$FILLC	061372	15509	15542#									
\$FILLS	061371	15541#										
\$GET42	060452	15371#										
\$GTSWR=	***** U	15785										
\$HIBTS	000330	385#										
\$HLT	060612	11684	11693	11723	11732	11761	11770	11799	11808	11838	11847	11876
		11992	12000	12021	12029	12050	12058	12079	12087	12108	12116	12137
		12174	12195	12203	12224	12232	12253	12261	12306	12314	12322	12330
		12375	12382	12412	12420	12428	12435	12548	12568	12575	12582	12602
												11887
												11972
												12166
												12367
												12616

1606	1612#	1632	1653	1659#	1680	1701	1707#	1728	1744	1750#	1774	1790
1796#	1837	1846	1852#	1893	1902	1908#	1939	1963	1969#	1987	2010	2016#
2040	2064	2070#	2092	2102	2108#	2160	2179	2185#	2204	2230	2236#	2258
2277	2283#	2298	2318	2324#	2340	2351	2357#	2383	2392	2398#	2429	2438
2444#	2464	2473	2479#	2500	2518	2524#	2538	2558	2564#	2571	2590	2596#
2603	2622	2628#	2636	2654	2660#	2685	2703	2709#	2728	2746	2752#	2796
2813	2819#	2840	2859	2865#	2902	2919	2925#	2943	2960	2966#	2987	3004
3010#	3029	3047	3053#	3072	3089	3095#	3100	3114	3120#	3125	3139	3145#
3162	3181	3187#	3239	3255	3261#	3281	3297	3303#	3348	3362	3368#	3388
3406	3412#	3418	3435	3441#	3447	3465	3471#	3477	3499	3505#	3523	3541
3547#	3554	3572	3578#	3593	3613	3619#	3633	3650	3656#	3673	3691	3697#
3701	3717	3723#	3727	3744	3750#	3754	3764	3770#	3787	3797	3803#	3820
3830	3836#	3863	3873	3879#	3929	3940	3946#	3991	4001	4007#	4035	4045
4051#	4099	4109	4115#	4143	4153	4159#	4186	4196	4202#	4229	4245	4251#
4268	4284	4290#	4318	4333	4339#	4384	4399	4405#	4450	4466	4472#	4497
4515	4521#	4565	4584	4590#	4635	4653	4659#	4704	4721	4727#	4771	4793
4799#	4805	4831	4837#	4843	4869	4875#	4881	4900	4906#	4912	4929	4935#
4956	4977	4983#	5022	5040	5046#	5091	5110	5116#	5149	5168	5174#	5182
5203	5209#	5218	5237	5243#	5248	5267	5273#	5279	5299	5305#	5316	5333
5339#	5343	5362	5368#	5381	5399	5405#	5408	5427	5433#	5446	5467	5473#
5486	5510	5516#	5520	5544	5550#	5554	5600	5606#	5708	5736	5742#	5873
5893	5899#	5909	5933	5939#	5959	5969	5975#	5996	6006	6012#	6032	6042
6048#	6070	6093	6099#	6136	6147	6153#	6203	6225	6231#	6237	6248	6254#
6275	6285	6291#	6312	6333	6339#	6399	6410	6416#	6452	6474	6480#	6519
6530	6536#	6587	6598	6604#	6610	6633	6639#	6690	6700	6706#	6758	6780
6786#	6838	6848	6854#	6905	6915	6921#	6974	6985	6991#	7047	7068	7074#
7103	7122	7128#	7156	7174	7180#	7201	7219	7225#	7262	7283	7289#	7308
7319	7325#	7339	7349	7355#	7370	7380	7386#	7400	7410	7416#	7430	7440
7446#	7460	7470	7476#	7490	7509	7515#	7532	7542	7548#	7574	7584	7590#
7616	7626	7632#	7658	7668	7674#	7700	7710	7716#	7742	7752	7758#	7784
7802	7808#	7811	7827	7833#	7854	7860#	7896	7902#	7920	7926#	7940	7974
7980#	8032	8038#	8082	8088#	8105	8111#	8117	8142	8148#	8205	8211#	8228
8238	8244#	8261	8271	8277#	8315	8325	8331#	8370	8387	8393#	8396	8406
8412#	8415	8425	8431#	8435	8445	8451#	8454	8464	8470#	8473	8485	8491#
8496	8506	8512#	8525	8531#	8700	8706#	8809	8818	8824#	8890	8899	8905#
8975	8986	8992#	9059	9068	9074#	9082	9088#	9092	9101	9107#	9111	9120
9126#	9146	9155	9161#	9264	9270#	9279	9285#	9289	9298	9304#	9308	9317
9323#	9342	9351	9357#	9449	9458	9464#	9478	9484#	9491	9497#	9501	9510
9516#	9520	9529	9535#	9555	9564	9570#	9673	9679#	9686	9692#	9696	9705
9711#	9715	9724	9730#	9750	9759	9765#	9856	9865	9871#	9886	9892#	9899
9905#	9909	9918	9924#	9928	9937	9943#	9964	9973	9979#	10089	10095#	10102
10108#	10112	10121	10127#	10131	10140	10146#	10166	10175	10181#	10272	10281	10287#
10294	10300#	10304	10313	10319#	10323	10333	10339#	10359	10368	10374#	10465	10475
10481#	10489	10495#	10499	10509	10515#	10544	10550#	10559	10565#	10574	10580#	10589
10595#	10604	10610#	10619	10625#	10635	10644	10650#	10660	10666#	10690	10696#	10706
10712#	10722	10731	10737#	10744	10755	10761#	10769	10780	10786#	10811	10817#	10829
10839	10845#	10865	10871#	10399	10905#	11005	11011#	11036	11042#	11069	11075#	11104
11110#	11116	11129	11135#	11150	11156#	11197	11203#	11244	11250#	11261	11274	11280#
11285	11301	11307#	11314	11332	11338#	11383	11389#	11420	11426#	11440	11446#	
11631#	15529*	15540										
11617#												
15478	15543#											
11632#	15527	15540										
11654	15756#											
15767#	15778											
15771#	15780#	15781#	15782#	15783#	15784#							

\$TPE 042410
\$TPCNT 042355
\$TPFLG 061373
\$TPS 042412
\$TRAP 062320
\$TRAP2 062342
\$TRP - 000006

\$TRPAD	062354	*5761	15778#											
\$TSTM	000334	387#												
\$TSTM=	000304	429#	447*											
\$TYPBN=	***** U	15784												
\$TYPDS	061646	15616#	15783											
\$TYPE	061106	15478#	15578	15771	15779									
\$TYPEC	061320	15508	15515	15522	15527#	15528								
\$TYPEX	061366	15533	15535	15538#										
\$TYPOC	062116	15701#	15780											
\$TYPON	062132	15700	15703#	15782										
\$TYPOS	062072	15696#	15781											
\$UNIT	000312	352#												
\$UNITM	000340	389#												
\$USWR	000324	359#												
\$X -	041722	456#	471	491	515#	521	534#	540	553#	559	572#	577	604#	611
		624#	630	640	653#	663	677#	684	697#	707	741#	747	760#	765
		778#	783	796#	801	814#	824	828#	849	862#	886#	908#	918	932#
		943	957#	967	981#	992	1007#	1017	1031#	1042	1056#	1067	1081#	1092
		1120#	1126	1139#	1144	1157#	1162	1175#	1180	1207#	1214	1227	1254#	1261
		1274	1296#	1303	1316	1338#	1345	1358	1403#	1407	1418	1427	1450#	1455
		1472	1492#	1496	1507	1529#	1534	1546	1567#	1574	1588	1612#	1620	1635
		1659#	1668	1683	1707#	1716	1731	1750#	1760	1777	1796#	1805	1816	1830
		1840	1852#	1862	1873	1886	1896	1908#	1918	1933	1942	1969#	1977	1990
		2016#	2026	2043	2070#	2079	2095	2108#	2122	2132	2143	2152	2163	2185#
		2192	2207	2236#	2248	2261	2283#	2290	2301	2324#	2332	2343	2357#	2367
		2377	2386	2398#	2412	2422	2432	2444#	2457	2467	2479#	2490	2503	2524#
		2529	2541	2564#	2574	2596#	2606	2628#	2639	2660#	2671	2688	2709#	2720
		2731	2752#	2763	2773	2788	2799	2819#	2832	2843	2865#	2879	2894	2905
		2925#	2936	2946	2966#	2979	2990	3010#	3022	3032	3053#	3065	3075	3095#
		3103	3120#	3128	3145#	3155	3165	3187#	3192	3205	3216	3231	3242	3261#
		3273	3284	3303#	3310	3319	3329	3341	3351	3368#	3379	3391	3412#	3421
		3441#	3450	3471#	3480	3505#	3516	3526	3547#	3557	3578#	3586	3596	3619#
		3627	3636	3656#	3666	3676	3697#	3704	3723#	3730	3750#	3757	3770#	3780
		3790	3803#	3813	3823	3836#	3845	3856	3866	3879#	3890	3900	3912	3923
		3932	3946#	3960	3970	3983	3994	4007#	4019	4029	4038	4051#	4063	4073
		4083	4093	4102	4115#	4127	4137	4146	4159#	4170	4180	4189	4202#	4213
		4223	4232	4251#	4261	4271	4290#	4300	4311	4321	4339#	4347	4357	4368
		4378	4387	4403#	4414	4424	4433	4443	4453	4472#	4481	4491	4500	4521#
		4529	4538	4549	4559	4568	4590#	4599	4609	4618	4629	4638	4659#	4668
		4678	4687	4698	4707	4727#	4736	4746	4755	4765	4774	4799#	4808	4837#
		4846	4875#	4884	4906#	4915	4935#	4944	4959	4983#	4993	5008	5025	5046#
		5058	5075	5094	5116#	5124	5138	5152	5174#	5185	5209#	5221	5243#	5251
		5273#	5282	5305#	5310	5319	5339#	5346	5368#	5375	5384	5405#	5411	5433#
		5440	5449	5473#	5480	5489	5516#	5523	5550#	5557	5606#	5613	5622	5636
		5648	5657	5670	5683	5697	5711	5742#	5770	5789	5807	5825	5841	5857
		5876	5899#	5912	5939#	5947	5962	5975#	5984	5999	6012#	6021	6035	6048#
		6058	6073	6099#	6108	6124	6139	6153#	6162	6176	6190	6206	6231#	6240
		6254#	6263	6278	6291#	6300	6315	6339#	6347	6361	6375	6387	6402	6416#
		6425	6440	6455	6480#	6490	6506	6522	6536#	6545	6560	6576	6590	6604#
		6613	6639#	6648	6663	6678	6693	6706#	6716	6731	6746	6761	6786#	6796
		6811	6825	6841	6854#	6864	6879	6894	6908	6921#	6931	6946	6960	6977
		6991#	7001	7017	7032	7050	7074#	7086	7106	7128#	7141	7159	7180#	7188
		7204	7225#	7244	7265	7289#	7296	7311	7325#	7333	7342	7355#	7364	7373
		7386#	7394	7403	7416#	7424	7433	7446#	7454	7463	7476#	7484	7493	7515#
		7521	7535	7548#	7558	7568	7577	7590#	7600	7610	7619	7632#	7642	7652
		7661	7674#	7684	7694	7703	7716#	7726	7736	7745	7758#	7768	7778	7787

7808#	7814	7833#	7860#	7874	7902#	7926#	7943	7980#	8038#	8067	8088#	8111#
8120	8148#	8161	8183	8211#	8219	8231	8244#	8252	8264	8277#	8287	8297
8308	8318	8331#	8341	8352	8363	8373	8393#	8399	8412#	8418	8431#	8438
8451#	8457	8470#	8476	8491#	8499	8512#	8531#	8706#	8712	8724	8736	8749
8762	8775	8787	8800	8812	8824#	8833	8848	8863	8878	8893	8905#	8909
8919	8929	8939	8948	8958	8968	8978	8992#	8997	9006	9015	9024	9035
9044	9053	9062	9074#	9088#	9095	9107#	9114	9126#	9135	9149	9161#	9169
9178	9187	9196	9205	9218	9227	9236	9245	9256	9270#	9285#	9292	9304#
9311	9323#	9332	9345	9357#	9365	9374	9383	9392	9401	9414	9423	9432
9441	9452	9464#	9484#	9497#	9504	9516#	9523	9535#	9544	9558	9570#	9578
9587	9596	9605	9614	9627	9636	9645	9654	9665	9679#	9692#	9699	9711#
9718	9730#	9739	9753	9765#	9773	9782	9791	9800	9809	9822	9831	9840
9849	9859	9871#	9892#	9905#	9912	9924#	9931	9943#	9953	9967	9979#	9987
9996	10005	10014	10023	10036	10045	10054	10063	10074	10095#	10108#	10115	10127#
10134	10146#	10155	10169	10181#	10189	10198	10207	10216	10225	10238	10247	10256
10265	10275	10287#	10300#	10307	10319#	10326	10339#	10348	10362	10374#	10382	10391
10400	10409	10418	10431	10440	10449	10458	10468	10481#	10495#	10502	10515#	10523
10550#	10565#	10580#	10595#	10610#	10625#	10638	10650#	10666#	10681	10696#	10712#	10725
10737#	10747	10761#	10772	10786#	10805	10817#	10832	10845#	10871#	10881	10905#	10929
10938	10962	10975	10984	11011#	11042#	11075#	11110#	11119	11135#	11156#	11173	11188
11203#	11228	11237	11250#	11264	11280#	11288	11307#	11317	11338#	11355	11364	11373
11389#	11404	11426#	11446#	11497								
471#	491#	521#	540#	559#	577#	611#	630#	640#	663#	684#	707#	747#
765#	783#	801#	824#	849#	918#	943#	967#	992#	1017#	1042#	1067#	1092#
1126#	1144#	1162#	1180#	1214#	1227#	1261#	1274#	1303#	1316#	1345#	1358#	1407#
1418#	1427#	1455#	1472#	1496#	1507#	1534#	1546#	1574#	1588#	1620#	1635#	1668#
1683#	1716#	1731#	1760#	1777#	1805#	1816#	1830#	1840#	1862#	1873#	1886#	1896#
1918#	1933#	1942#	1977#	1990#	2026#	2043#	2079#	2095#	2122#	2132#	2143#	2152#
2163#	2192#	2207#	2248#	2261#	2290#	2301#	2332#	2343#	2367#	2377#	2386#	2412#
2422#	2432#	2457#	2467#	2490#	2503#	2529#	2541#	2574#	2606#	2639#	2671#	2688#
2720#	2731#	2763#	2773#	2788#	2799#	2832#	2843#	2879#	2894#	2905#	2936#	2946#
2979#	2990#	3022#	3032#	3065#	3075#	3103#	3128#	3155#	3165#	3192#	3205#	3216#
3231#	3242#	3273#	3284#	3310#	3319#	3329#	3341#	3351#	3379#	3391#	3421#	3450#
3480#	3516#	3526#	3557#	3586#	3596#	3627#	3636#	3666#	3676#	3704#	3730#	3757#
3780#	3790#	3813#	3823#	3845#	3856#	3866#	3890#	3900#	3912#	3923#	3932#	3960#
3970#	3983#	3994#	4019#	4029#	4038#	4063#	4073#	4083#	4093#	4102#	4127#	4137#
4146#	4170#	4180#	4189#	4213#	4223#	4232#	4261#	4271#	4300#	4311#	4321#	4347#
4357#	4368#	4378#	4387#	4414#	4424#	4433#	4443#	4453#	4481#	4491#	4500#	4529#
4538#	4549#	4559#	4568#	4599#	4609#	4618#	4629#	4638#	4668#	4678#	4687#	4698#
4707#	4736#	4746#	4755#	4765#	4774#	4808#	4846#	4884#	4915#	4944#	4959#	4993#
5008#	5025#	5058#	5075#	5094#	5124#	5138#	5152#	5185#	5221#	5251#	5282#	5310#
5319#	5346#	5375#	5384#	5411#	5440#	5449#	5480#	5489#	5523#	5557#	5613#	5622#
5636#	5648#	5657#	5670#	5683#	5697#	5711#	5770#	5789#	5807#	5825#	5841#	5857#
5876#	5912#	5947#	5962#	5984#	5999#	6021#	6035#	6058#	6073#	6108#	6124#	6139#
6162#	6176#	6190#	6206#	6240#	6263#	6278#	6300#	6315#	6347#	6361#	6375#	6387#
6402#	6425#	6440#	6455#	6490#	6506#	6522#	6545#	6560#	6576#	6590#	6613#	6648#
6663#	6678#	6693#	6716#	6731#	6746#	6761#	6796#	6811#	6825#	6841#	6864#	6879#
6894#	6908#	6931#	6946#	6960#	6977#	7001#	7017#	7032#	7050#	7086#	7106#	7141#
7159#	7188#	7204#	7244#	7265#	7296#	7311#	7333#	7342#	7364#	7373#	7394#	7403#
7424#	7433#	7454#	7463#	7484#	7493#	7521#	7535#	7558#	7568#	7577#	7600#	7610#
7619#	7642#	7652#	7661#	7684#	7694#	7703#	7726#	7736#	7745#	7768#	7778#	7787#
7814#	7874#	7943#	8067#	8120#	8161#	8183#	8219#	8231#	8252#	8264#	8287#	8297#
8308#	8318#	8341#	8352#	8363#	8373#	8399#	8418#	8438#	8457#	8476#	8499#	8712#
8724#	8736#	8749#	8762#	8775#	8787#	8800#	8812#	8833#	8848#	8863#	8878#	8893#
8909#	8919#	8929#	8939#	8948#	8958#	8968#	8978#	8997#	9006#	9015#	9024#	9035#
9044#	9053#	9062#	9095#	9114#	9135#	9149#	9169#	9178#	9187#	9196#	9205#	9218#

SXX 177662

SXXX 000661

9227#	9236#	9245#	9256#	9292#	9311#	9332#	9345#	9365#	9374#	9383#	9392#	9401#
9414#	9423#	9432#	9441#	9452#	9504#	9523#	9544#	9558#	9578#	9587#	9596#	9605#
9614#	9627#	9636#	9645#	9654#	9665#	9699#	9718#	9739#	9753#	9773#	9782#	9791#
9800#	9809#	9822#	9831#	9840#	9849#	9859#	9912#	9931#	9953#	9967#	9987#	9996#
10005#	10014#	10023#	10036#	10045#	10054#	10063#	10074#	10115#	10134#	10155#	10169#	10189#
10198#	10207#	10216#	10225#	10238#	10247#	10256#	10265#	10275#	10307#	10326#	10348#	10362#
10382#	10391#	10400#	10409#	10418#	10431#	10440#	10449#	10458#	10468#	10502#	10523#	10638#
10681#	10725#	10747#	10772#	10805#	10832#	10881#	10929#	10938#	10962#	10975#	10984#	11119#
11173#	11188#	11228#	11237#	11264#	11288#	11317#	11355#	11364#	11373#	11404#	11497#	
471#	491#	521#	540#	559#	577#	611#	630#	640#	663#	684#	707#	747#
765#	783#	801#	824#	849#	918#	943#	967#	992#	1017#	1042#	1067#	1092#
1126#	1144#	1162#	1180#	1214#	1227#	1261#	1274#	1303#	1316#	1345#	1358#	1407#
1418#	1427#	1455#	1472#	1496#	1507#	1534#	1546#	1574#	1588#	1620#	1635#	1668#
1683#	1716#	1731#	1760#	1777#	1805#	1816#	1830#	1840#	1862#	1873#	1886#	1896#
1918#	1933#	1942#	1977#	1990#	2026#	2043#	2079#	2095#	2122#	2132#	2143#	2152#
2163#	2192#	2207#	2248#	2261#	2290#	2301#	2332#	2343#	2367#	2377#	2386#	2412#
2422#	2432#	2457#	2467#	2490#	2503#	2529#	2541#	2574#	2606#	2639#	2671#	2688#
2720#	2731#	2763#	2773#	2788#	2799#	2832#	2843#	2879#	2894#	2905#	2936#	2946#
2979#	2990#	3022#	3032#	3065#	3075#	3103#	3128#	3155#	3165#	3192#	3205#	3216#
3231#	3242#	3273#	3284#	3310#	3319#	3329#	3341#	3351#	3379#	3391#	3421#	3450#
3480#	3516#	3526#	3557#	3586#	3596#	3627#	3636#	3666#	3676#	3704#	3730#	3757#
3780#	3790#	3813#	3823#	3845#	3856#	3866#	3890#	3900#	3912#	3923#	3932#	3960#
3970#	3983#	3994#	4019#	4029#	4038#	4063#	4073#	4083#	4093#	4102#	4127#	4137#
4146#	4170#	4180#	4189#	4213#	4223#	4232#	4261#	4271#	4300#	4311#	4321#	4347#
4357#	4368#	4378#	4387#	4414#	4424#	4433#	4443#	4453#	4481#	4491#	4500#	4529#
4538#	4549#	4559#	4568#	4599#	4609#	4618#	4629#	4638#	4668#	4678#	4687#	4698#
4707#	4736#	4746#	4755#	4765#	4774#	4808#	4846#	4884#	4915#	4944#	4959#	4993#
5008#	5025#	5058#	5075#	5094#	5124#	5138#	5152#	5185#	5221#	5251#	5282#	5310#
5319#	5346#	5375#	5384#	5411#	5440#	5449#	5480#	5489#	5523#	5557#	5613#	5622#
5636#	5648#	5657#	5670#	5683#	5697#	5711#	5770#	5789#	5807#	5825#	5841#	5857#
5876#	5912#	5947#	5962#	5984#	5999#	6021#	6035#	6058#	6073#	6108#	6124#	6139#
6162#	6176#	6190#	6206#	6240#	6263#	6278#	6300#	6315#	6347#	6361#	6375#	6387#
6402#	6425#	6440#	6455#	6490#	6506#	6522#	6545#	6560#	6576#	6590#	6613#	6648#
6663#	6678#	6693#	6716#	6731#	6746#	6761#	6796#	6811#	6825#	6841#	6864#	6879#
6894#	6908#	6931#	6946#	6960#	6977#	7001#	7017#	7032#	7050#	7086#	7106#	7141#
7159#	7188#	7204#	7244#	7265#	7296#	7311#	7333#	7342#	7364#	7373#	7394#	7403#
7424#	7433#	7454#	7463#	7484#	7493#	7521#	7535#	7558#	7568#	7577#	7600#	7610#
7619#	7642#	7652#	7661#	7684#	7694#	7703#	7726#	7736#	7745#	7768#	7778#	7787#
7814#	7874#	7943#	8067#	8120#	8161#	8183#	8219#	8231#	8252#	8264#	8287#	8297#
8308#	8318#	8341#	8352#	8363#	8373#	8399#	8418#	8438#	8457#	8476#	8499#	8712#
8724#	8736#	8749#	8762#	8775#	8787#	8800#	8812#	8833#	8848#	8863#	8878#	8893#
8909#	8919#	8929#	8939#	8948#	8958#	8968#	8978#	8997#	9006#	9015#	9024#	9035#
9044#	9053#	9062#	9095#	9114#	9135#	9149#	9169#	9178#	9187#	9196#	9205#	9218#
9227#	9236#	9245#	9256#	9292#	9311#	9332#	9345#	9365#	9374#	9383#	9392#	9401#
9414#	9423#	9432#	9441#	9452#	9504#	9523#	9544#	9558#	9578#	9587#	9596#	9605#
9614#	9627#	9636#	9645#	9654#	9665#	9699#	9718#	9739#	9753#	9773#	9782#	9791#
9800#	9809#	9822#	9831#	9840#	9849#	9859#	9912#	9931#	9953#	9967#	9987#	9996#
10005#	10014#	10023#	10036#	10045#	10054#	10063#	10074#	10115#	10134#	10155#	10169#	10189#
10198#	10207#	10216#	10225#	10238#	10247#	10256#	10265#	10275#	10307#	10326#	10348#	10362#
10382#	10391#	10400#	10409#	10418#	10431#	10440#	10449#	10458#	10468#	10502#	10523#	10638#
10681#	10725#	10747#	10772#	10805#	10832#	10881#	10929#	10938#	10962#	10975#	10984#	11119#
11173#	11188#	11228#	11237#	11264#	11288#	11317#	11355#	11364#	11373#	11404#	11497#	
15697#	15701#	15711	15746#									
331#	335	336#	338#	340#	341#	374	375#	377#	379#	394#	400#	405#
408#	414#	422#	430#	433#	440#	456	471	491	515	521	534	540
553	559	572	577	604	611	624	630	640	653	663	677	684

SOF ILL 062315
062372

697	707	741	747	760	765	778	783	796	801	814	824	838
849	862	886	908	918	932	943	957	967	981	992	1007	1017
1031	1042	1056	1067	1081	1092	1120	1126	1139	1144	1157	1162	1175
1180	1207	1214	1227	1254	1261	1274	1296	1303	1316	1338	1345	1358
1403	1407	1418	1427	1450	1455	1472	1492	1496	1507	1529	1534	1546
1567	1574	1588	1612	1620	1635	1659	1668	1683	1707	1716	1731	1750
1760	1777	1796	1805	1816	1830	1840	1852	1862	1873	1886	1896	1908
1918	1933	1942	1969	1977	1990	2016	2026	2043	2070	2079	2095	2108
2122	2132	2143	2152	2163	2185	2192	2207	2236	2248	2261	2283	2290
2301	2324	2332	2343	2357	2367	2377	2386	2398	2412	2422	2432	2444
2457	2467	2479	2490	2503	2524	2529	2541	2564	2574	2596	2606	2628
2639	2660	2671	2688	2709	2720	2731	2752	2763	2773	2788	2799	2819
2832	2843	2865	2879	2894	2905	2925	2936	2946	2966	2979	2990	3010
3022	3032	3053	3065	3075	3095	3103	3120	3128	3145	3155	3165	3187
3192	3205	3216	3231	3242	3261	3273	3284	3303	3310	3319	3329	3341
3351	3368	3379	3391	3412	3421	3441	3450	3471	3480	3505	3516	3526
3547	3557	3578	3586	3596	3619	3627	3636	3656	3666	3676	3697	3704
3723	3730	3750	3757	3770	3780	3790	3803	3813	3833	3845	3856	3885
3866	3879	3890	3900	3912	3923	3932	3946	3960	3970	3983	3994	4007
4019	4029	4038	4051	4063	4073	4083	4093	4102	4115	4127	4137	4146
4159	4170	4180	4189	4202	4213	4223	4232	4251	4261	4271	4290	4300
4311	4321	4339	4347	4357	4368	4378	4387	4405	4414	4424	4433	4443
4453	4472	4481	4491	4500	4521	4529	4538	4549	4559	4568	4590	4599
4609	4618	4629	4638	4659	4668	4678	4687	4698	4707	4727	4736	4746
4755	4765	4774	4799	4808	4837	4846	4875	4884	4906	4915	4935	4944
4959	4983	4993	5008	5025	5046	5058	5075	5094	5116	5124	5138	5152
5174	5185	5209	5221	5243	5251	5273	5282	5305	5310	5319	5339	5346
5368	5375	5384	5405	5411	5433	5440	5449	5473	5480	5489	5516	5523
5550	5557	5606	5609	5613	5622	5636	5648	5657	5670	5683	5697	5711
5742	5770	5789	5807	5825	5841	5857	5876	5899	5912	5939	5947	5962
5975	5984	5999	6012	6021	6035	6048	6058	6073	6099	6108	6124	6139
6153	6162	6176	6190	6206	6231	6240	6254	6263	6278	6291	6300	6315
6339	6347	6361	6375	6387	6402	6416	6425	6440	6455	6480	6490	6506
6522	6536	6545	6560	6576	6590	6604	6613	6639	6648	6663	6678	6693
6706	6716	6731	6746	6761	6786	6796	6811	6825	6841	6854	6864	6879
6894	6908	6921	6931	6946	6960	6977	6991	7001	7017	7032	7050	7074
7086	7106	7128	7141	7159	7180	7188	7204	7225	7244	7265	7289	7296
7311	7325	7333	7342	7355	7364	7373	7386	7394	7403	7416	7424	7433
7446	7454	7463	7476	7484	7493	7515	7521	7535	7548	7558	7568	7577
7590	7600	7610	7619	7632	7642	7652	7661	7674	7684	7694	7703	7716
7726	7736	7745	7758	7768	7778	7787	7808	7814	7833	7860	7874	7902
7926	7943	7980	7991	8016	8021	8038	8067	8088	8111	8120	8148	8161
8183	8211	8219	8231	8244	8252	8264	8277	8287	8297	8308	8318	8331
8341	8352	8363	8373	8393	8399	8412	8418	8431	8438	8451	8457	8470
8476	8491	8499	8512	8531	8549	8550	8551	8552	8553	8554	8555	8556
8557	8558	8559	8560	8561	8562	8563	8585#	8706	8712	8724	8736	8749
8762	8775	8787	8800	8812	8824	8833	8848	8863	8878	8893	8905	8909
8919	8929	8939	8948	8958	8968	8978	8992	8997	9006	9015	9024	9035
9044	9053	9062	9074	9088	9095	9107	9110	9114	9126	9135	9149	9161
9169	9178	9187	9196	9205	9218	9227	9236	9245	9256	9270	9285	9292
9304	9307	9311	9323	9332	9345	9357	9365	9374	9383	9392	9401	9414
9423	9432	9441	9452	9464	9484	9497	9504	9516	9519	9523	9535	9544
9558	9570	9578	9587	9596	9605	9614	9627	9636	9645	9654	9665	9679
9692	9699	9711	9714	9718	9730	9739	9753	9765	9773	9782	9791	9800
9809	9822	9831	9840	9849	9859	9871	9892	9905	9912	9924	9927	9931
9943	9953	9967	9979	9987	9996	10005	10014	10023	10036	10045	10054	10063

10074	10095	10108	10115	10127	10130	10134	10146	10155	10169	10181	10189	10198
10207	10216	10225	10238	10247	10256	10265	10275	10287	10300	10307	10319	10326
10339	10348	10362	10374	10382	10391	10400	10409	10418	10431	10440	10449	10458
10468	10481	10495	10502	10515	10523	10550	10565	10580	10595	10610	10625	10638
10650	10666	10681	10696	10699	10712	10715	10725	10737	10740	10743	10747	10761
10772	10786	10805	10817	10832	10845	10852	10871	10881	10905	10929	10938	10962
10975	10984	11011	11042	11075	11110	11119	11135	11156	11173	11188	11203	11228
11237	11250	11264	11280	11288	11307	11317	11338	11355	11364	11373	11389	11404
11426	11446	11497	11547	11603#	11605#	11607#	11609#	11611#	11613#	11683	11691	11722
11730	11760	11768	11798	11806	11837	11845	11877	11885	1971	11991	11998	12020
12027	12049	12056	12078	12085	12107	12114	12136	12143	12165	12172	12194	12201
12223	12230	12252	12259	12305	12313	12320	12329	12358	12366	12373	12381	12411
12419	12426	12434	12547	12567	12574	12581	12601	12608	12615	12636	12643	12651
12671	12678	12685	12705	12712	12719	12740	12747	12755	12775	12782	12789	12809
12816	12823	12830	12855	12862	12869	12892	12899	12906	12929	12936	12943	12966
12973	12980	13003	13010	13017	13040	13047	13054	13077	13084	13091	13114	13121
13128	13151	13158	13165	13188	13195	13202	13237	13244	13251	13272	13279	13286
13307	13314	13321	13343	13350	13357	13379	13386	13393	13414	13421	13428	13450
13457	13464	13486	13493	13500	13521	13528	13535	13556	13563	13570	13592	13599
13606	13628	13635	13642	13663	13670	13677	13698	13705	13712	13734	13741	13748
13770	13777	13784	13805	13812	13819	13841	13848	13855	13881	13888	13895	13916
13923	13930	13951	13958	13965	13986	13993	14000	14021	14028	14035	14056	14063
14070	14091	14098	14105	14126	14133	14140	14161	14168	14175	14196	14203	14210
14239	14247	14255	14262	14281	14289	14297	14304	14324	14332	14340	14348	14367
14375	14383	14390	14409	14417	14425	14432	14452	14460	14468	14476	14495	14503
14511	14518	14537	14545	14553	14560	14579	14587	14595	14602	14622	14630	14638
14646	14665	14673	14681	14688	14707	14715	14723	14730	14750	14758	14766	14774
14793	14801	14809	14816	14835	14843	14851	14858	14879	14888	14908	14916	14939
14947	14955	14962	14981	14989	14997	15004	15023	15031	15039	15046	15065	15073
15081	15088	15107	15115	15123	15130	15149	15157	15165	15172	15191	15199	15207
15214	15233	15241	15249	15256	15275	15283	15291	15298	15317	15325	15333	15340
15424	15540	15541	15542	15543	15544	15545	15546#	15599#	15670#			
15551	15554											

.\$ASTA= ***** U
.\$X - 000330

374# 379

ERROR	317#	463	467	483	487	518	537	556	574	608	627	637	660	681	704
	744	762	780	798	821	846	875	896	915	940	964	989	1014	1039	1064
	1089	1123	1141	1159	1177	1211	1224	1258	1271	1300	1313	1342	1355	1404	1415
	1424	1452	1469	1493	1504	1531	1543	1571	1585	1617	1632	1665	1680	1713	1728
	1757	1774	1802	1813	1827	1837	1859	1870	1883	1893	1915	1930	1939	1974	1987
	2023	2040	2076	2092	2119	2129	2140	2149	2160	2189	2204	2245	2258	2287	2298
	2329	2340	2364	2374	2383	2409	2419	2429	2454	2464	2487	2500	2526	2538	2571
	2603	2636	2668	2685	2717	2728	2760	2770	2785	2796	2829	2840	2876	2891	2902
	2933	2943	2976	2987	3019	3029	3062	3072	3100	3125	3152	3162	3189	3202	3213
	3228	3239	3270	3281	3307	3316	3326	3338	3348	3376	3388	3418	3447	3477	3513
	3523	3554	3583	3593	3624	3633	3663	3673	3701	3727	3754	3777	3787	3810	3820
	3842	3853	3863	3887	3897	3909	3920	3929	3957	3967	3980	3991	4016	4026	4035
	4060	4070	4080	4090	4099	4124	4134	4143	4167	4177	4186	4210	4220	4229	4258
	4268	4297	4308	4318	4344	4354	4365	4375	4384	4411	4421	4430	4440	4450	4478
	4488	4497	4526	4535	4546	4556	4565	4596	4606	4615	4626	4635	4665	4675	4684
	4695	4704	4733	4743	4752	4762	4771	4805	4843	4881	4912	4941	4956	4990	5005
	5022	5055	5072	5091	5121	5135	5149	5182	5218	5248	5279	5307	5316	5343	5372
	5381	5408	5437	5446	5477	5486	5520	5554	5610	5619	5633	5645	5654	5667	5680
	5694	5708	5753	5767	5786	5804	5822	5838	5854	5873	5905	5909	5944	5959	5981
	5996	6018	6032	6055	6070	6105	6121	6136	6159	6173	6187	6203	6237	6260	6275
	6297	6312	6344	6358	6372	6384	6399	6422	6437	6452	6487	6503	6519	6542	6557
	6573	6587	6610	6645	6660	6675	6690	6713	6728	6743	6758	6793	6808	6822	6838
	6861	6876	6891	6905	6928	6943	6957	6974	6998	7014	7029	7047	7083	7103	7138
	7156	7185	7201	7232	7241	7256	7262	7293	7308	7330	7339	7361	7370	7391	7400
	7421	7430	7451	7460	7481	7490	7518	7532	7555	7565	7574	7597	7607	7616	7639
	7649	7658	7681	7691	7700	7723	7733	7742	7765	7775	7784	7811	7841	7866	7871
	7884	7909	7915	7935	7940	8009	8064	8094	8117	8158	8180	8216	8228	8249	8261
	8284	8294	8305	8315	8338	8349	8360	8370	8396	8415	8435	8454	8473	8496	8520
	8540	8543	8593	8597	8601	8605	8609	8613	8617	8621	8709	8721	8733	8746	8759
	8772	8784	8797	8809	8830	8845	8860	8875	8890	8906	8916	8926	8936	8945	8955
	8965	8975	8994	9002	9011	9020	9032	9040	9049	9058	9077	9092	9111	9132	9146
	9166	9174	9183	9192	9202	9215	9223	9232	9241	9253	9275	9289	9308	9329	9342
	9362	9370	9379	9388	9398	9411	9419	9428	9437	9449	9469	9487	9501	9520	9541
	9555	9575	9583	9592	9601	9611	9624	9632	9641	9650	9662	9682	9696	9715	9736
	9750	9770	9778	9787	9796	9806	9819	9827	9836	9845	9856	9876	9895	9909	9928
	9950	9964	9984	9992	10001	10010	10020	10033	10041	10050	10059	10071	10098	10112	10131
	10152	10166	10186	10194	10203	10212	10222	10235	10243	10252	10261	10272	10290	10304	10323
	10345	10359	10379	10387	10396	10405	10415	10428	10436	10445	10454	10465	10484	10499	10520
	10532	10538	10554	10569	10584	10599	10614	10629	10635	10654	10678	10702	10718	10722	10744
	10769	10796	10802	10824	10829	10855	10859	10878	10919	10926	10935	10959	10972	10981	11025
	11028	11058	11062	11092	11116	11125	11142	11170	11185	11221	11225	11234	11257	11261	11285
	11310	11314	11344	11347	11352	11361	11370	11401	11485	11490	11494				
ERROR2	11642#	11684	11693	11723	11732	11761	11770	11799	11808	11838	11847	11878	11887	11972	11992
	12000	12021	12029	12050	12058	12079	12087	12108	12116	12137	12145	12166	12174	12195	12203
	12224	12232	12253	12261	12306	12314	12322	12330	12359	12367	12375	12382	12412	12420	12428
	12435	12548	12568	12575	12582	12602	12609	12616	12637	12644	12652	12672	12679	12686	12706
	12713	12720	12741	12748	12756	12776	12783	12790	12810	12817	12824	12831	12856	12863	12871
	12893	12900	12908	12930	12937	12945	12967	12974	12982	13004	13011	13019	13041	13048	13056
	13078	13085	13093	13115	13122	13130	13152	13159	13167	13189	13196	13204	13238	13245	13253
	13273	13280	13288	13308	13315	13323	13344	13351	13359	13380	13387	13395	13415	13422	13430
	13451	13458	13466	13487	13494	13502	13522	13529	13537	13557	13564	13572	13593	13600	13608
	13629	13636	13644	13664	13671	13679	13699	13706	13714	13735	13742	13750	13771	13778	13786
	13806	13813	13821	13842	13849	13857	13882	13889	13897	13917	13924	13932	13952	13959	13967
	13987	13994	14002	14022	14029	14037	14057	14064	14072	14092	14099	14107	14127	14134	14142
	14162	14169	14177	14197	14204	14212	14240	14248	14256	14263	14282	14290	14298	14305	14325
	14333	14341	14349	14368	14376	14384	14391	14410	14418	14426	14433	14453	14461	14469	14477

HLT

14496	14504	14512	14519	14538	14546	14554	14561	14580	14588	14596	14603	14623	14631	14639
14647	14666	14674	14682	14689	14708	14716	14724	14731	14751	14759	14767	14775	14794	14802
14810	14817	14836	14844	14852	14859	14880	14889	14909	14917	14940	14948	14956	14963	14982
14990	14998	15005	15024	15032	15040	15047	15066	15074	15082	15089	15108	15116	15124	15131
15150	15158	15166	15173	15192	15200	15208	15215	15234	15242	15250	15257	15276	15284	15292
15299	15318	15326	15334	15341										
11640#	11684	11692	11723	11731	11761	11769	11799	11807	11838	11846	11878	11886	11972	11992
11999	12021	12028	12050	12057	12079	12086	12108	12115	12137	12144	12166	12173	12195	12202
12224	12231	12253	12260	12306	12314	12322	12330	12359	12367	12375	12382	12412	12420	12428
12435	12548	12568	12575	12582	12602	12609	12616	12637	12644	12652	12672	12679	12686	12706
12713	12720	12741	12748	12756	12776	12783	12790	12810	12817	12824	12831	12856	12863	12870
12893	12900	12907	12930	12937	12944	12967	12974	12981	13004	13011	13018	13041	13048	13055
13078	13085	13092	13115	13122	13129	13152	13159	13166	13189	13196	13203	13238	13245	13252
13273	13280	13287	13308	13315	13322	13344	13351	13358	13380	13387	13394	13415	13422	13429
13451	13458	13465	13487	13494	13501	13522	13529	13536	13557	13564	13571	13593	13600	13607
13629	13636	13643	13664	13671	13678	13699	13706	13713	13735	13742	13749	13771	13778	13785
13806	13813	13820	13842	13849	13856	13882	13889	13896	13917	13924	13931	13952	13959	13966
13987	13994	14001	14022	14029	14036	14057	14064	14071	14092	14099	14106	14127	14134	14141
14162	14169	14176	14197	14204	14211	14240	14248	14256	14263	14282	14290	14298	14305	14325
14333	14341	14349	14368	14376	14384	14391	14410	14418	14426	14433	14453	14461	14469	14477
14496	14504	14512	14519	14538	14546	14554	14561	14580	14588	14596	14603	14623	14631	14639
14647	14666	14674	14682	14689	14708	14716	14724	14731	14751	14759	14767	14775	14794	14802
14810	14817	14836	14844	14852	14859	14880	14889	14909	14917	14940	14948	14956	14963	14982
14990	14998	15005	15024	15032	15040	15047	15066	15074	15082	15089	15108	15116	15124	15131
15150	15158	15166	15173	15192	15200	15208	15215	15234	15242	15250	15257	15276	15284	15292
15299	15318	15326	15334	15341										

JNE
LOOP

7057#	7989	8014	8019											
317#	471	491	521	540	559	577	611	630	640	663	684	707	747	765
783	801	824	849	918	943	967	992	1017	1042	1067	1092	1126	1144	1162
1180	1214	1227	1261	1274	1303	1316	1345	1358	1407	1418	1427	1455	1472	1496
1507	1534	1546	1574	1588	1620	1635	1668	1683	1716	1731	1760	1777	1805	1816
1830	1840	1862	1873	1886	1896	1918	1933	1942	1977	1990	2026	2043	2079	2095
2122	2132	2143	2152	2163	2192	2207	2248	2261	2290	2301	2332	2343	2367	2377
2386	2412	2422	2432	2457	2467	2490	2503	2529	2541	2574	2606	2639	2671	2688
2720	2731	2763	2773	2788	2799	2832	2843	2879	2894	2905	2936	2946	2979	2990
3022	3032	3065	3075	3103	3128	3155	3165	3192	3205	3216	3231	3242	3273	3284
3310	3319	3329	3341	3351	3379	3391	3421	3450	3480	3516	3526	3557	3586	3596
3627	3636	3666	3676	3704	3730	3757	3780	3790	3813	3823	3845	3856	3866	3890
3900	3912	3923	3932	3960	3970	3983	3994	4019	4029	4038	4063	4073	4083	4093
4102	4127	4137	4146	4170	4180	4189	4213	4223	4232	4261	4271	4300	4311	4321
4347	4357	4368	4378	4387	4414	4424	4433	4443	4453	4481	4491	4500	4529	4538
4549	4559	4568	4599	4609	4618	4629	4638	4668	4678	4687	4698	4707	4736	4746
4755	4765	4774	4708	4846	4884	4915	4944	4959	4993	5008	5025	5058	5075	5094
5124	5138	5152	5185	5221	5251	5282	5310	5319	5346	5375	5384	5411	5440	5449
5480	5489	5523	5557	5613	5622	5636	5648	5657	5670	5683	5697	5711	5770	5789
5807	5825	5841	5857	5876	5912	5947	5962	5984	5999	6021	6035	6058	6073	6108
6124	6139	6162	6176	6190	6206	6240	6263	6278	6300	6315	6347	6361	6375	6387
6402	6425	6440	6455	6490	6506	6522	6545	6560	6576	6590	6613	6648	6663	6678
6693	6716	6731	6746	6761	6796	6811	6825	6841	6864	6879	6894	6908	6931	6946
6960	6977	7001	7017	7032	7050	7086	7106	7141	7159	7188	7204	7244	7265	7296
7311	7333	7342	7364	7373	7394	7403	7424	7433	7454	7463	7484	7493	7521	7535
7558	7568	7577	7600	7610	7619	7642	7652	7661	7684	7694	7703	7726	7736	7745
7768	7778	7787	7814	7874	7943	8067	8120	8161	8183	8219	8231	8252	8264	8287
8297	8308	8318	8341	8352	8363	8373	8399	8418	8438	8457	8476	8499	8712	8724
8736	8749	8762	8775	8787	8800	8812	8833	8848	8863	8878	8893	8909	8919	8929
8939	8948	8958	8968	8978	8997	9006	9015	9024	9035	9044	9053	9062	9095	9114

9135	9149	9169	9178	9187	9196	9205	9218	9227	9236	9245	9256	9292	9311	9332	
9345	9365	9374	9383	9392	9401	9414	9423	9432	9441	9452	9504	9523	9544	9558	
9578	9587	9596	9605	9614	9627	9636	9645	9654	9665	9699	9718	9739	9753	9773	
9782	9791	9800	9809	9822	9831	9840	9849	9859	9912	9931	9953	9967	9987	9996	
10005	10014	10023	10036	10045	10054	10063	10074	10115	10134	10155	10169	10189	10198	10207	
10216	10225	10238	10247	10256	10265	10275	10307	10326	10348	10362	10382	10391	10400	10409	
10418	10431	10440	10449	10458	10468	10502	10523	10638	10681	10725	10747	10772	10805	10832	
10881	10929	10938	10962	10975	10984	11119	11173	11188	11228	11237	11264	11288	11317	11355	
11364	11373	11404	11497												
NEWST	317#	450	509	528	547	566	598	618	647	671	691	735	754	772	790
	808	832	856	880	902	926	951	975	1001	1025	1050	1075	1114	1133	1151
	1169	1201	1248	1290	1332	1397	1444	1486	1523	1561	1606	1653	1701	1744	1790
	1846	1902	1963	2010	2064	2102	2179	2230	2277	2318	2351	2392	2438	2473	2518
	2558	2590	2622	2654	2703	2746	2813	2859	2919	2960	3004	3047	3089	3114	3139
	3181	3255	3297	3362	3406	3435	3465	3499	3541	3572	3613	3650	3691	3717	3744
	3764	3797	3830	3873	3940	4001	4045	4109	4153	4196	4245	4284	4333	4399	4466
	4515	4584	4653	4721	4793	4831	4869	4900	4929	4977	5040	5110	5168	5203	5237
	5267	5299	5333	5362	5399	5427	5467	5510	5544	5600	5736	5893	5933	5969	6006
	6042	6093	6147	6225	6248	6285	6333	6410	6474	6530	6598	6633	6700	6780	6848
	6915	6985	7068	7122	7174	7219	7283	7319	7349	7380	7410	7440	7470	7509	7542
	7584	7626	7668	7710	7752	7802	7827	7854	7896	7920	7974	8032	8082	8105	8142
	8204	8238	8271	8325	8387	8406	8425	8445	8464	8485	8506	8525	8699	8818	8899
	8986	9068	9082	9101	9120	9155	9264	9279	9298	9317	9351	9458	9478	9491	9510
	9529	9564	9673	9686	9705	9724	9759	9865	9886	9899	9918	9937	9973	10089	10102
	10121	10140	10175	10281	10294	10313	10333	10368	10475	10489	10509	10544	10559	10574	10589
	10604	10619	10644	10660	10690	10706	10731	10755	10780	10811	10839	10865	10899	11004	11036
	11069	11104	11129	11150	11197	11243	11274	11301	11332	11383	11420	11440			
POP	331#	15593	15594	15657											
PUSH	331#	15554	15556	15577	15616										
SETTRA	15771#	15780	15781	15782	15783										
STARS	317#	333	344	371	373	380	391	393	419	421	426	450	452	498	509
	511	528	530	547	549	566	568	584	598	600	618	620	647	649	671
	673	691	693	715	735	737	754	756	772	774	790	792	808	810	832
	834	856	858	880	882	902	904	926	928	951	953	975	977	1001	1003
	1025	1027	1050	1052	1075	1077	1100	1114	1116	1133	1135	1151	1153	1169	1171
	1188	1201	1203	1235	1248	1250	1282	1290	1292	1324	1332	1334	1366	1381	1385
	1397	1399	1435	1444	1446	1480	1486	1488	1515	1523	1525	1554	1561	1563	1597
	1606	1608	1643	1653	1655	1691	1701	1703	1739	1744	1746	1785	1790	1792	1846
	1848	1902	1904	1949	1963	1965	1998	2010	2012	2051	2064	2066	2102	2104	2170
	2179	2181	2215	2230	2232	2269	2277	2279	2309	2318	2320	2351	2353	2392	2394
	2438	2440	2473	2475	2510	2518	2520	2550	2558	2560	2582	2590	2592	2614	2622
	2624	2647	2654	2656	2696	2703	2705	2738	2746	2748	2805	2813	2815	2850	2859
	2861	2911	2919	2921	2953	2960	2962	2997	3004	3006	3040	3047	3049	3083	3089
	3091	3110	3114	3116	3135	3139	3141	3173	3181	3183	3250	3255	3257	3291	3297
	3299	3357	3362	3364	3399	3406	3408	3429	3435	3437	3458	3465	3467	3488	3499
	3501	3534	3541	3543	3565	3572	3574	3604	3613	3615	3643	3650	3652	3683	3691
	3693	3710	3717	3719	3737	3744	3746	3764	3766	3797	3799	3830	3832	3873	3875
	3940	3942	4001	4003	4045	4047	4109	4111	4153	4155	4196	4198	4239	4245	4247
	4278	4284	4286	4328	4333	4335	4394	4399	4401	4459	4466	4468	4507	4515	4517
	4575	4584	4586	4645	4653	4655	4714	4721	4723	4781	4793	4795	4822	4831	4833
	4859	4869	4871	4890	4900	4902	4921	4929	4931	4968	4977	4979	5033	5040	5042
	5103	5110	5112	5160	5168	5170	5193	5203	5205	5230	5237	5239	5260	5267	5269
	5292	5299	5301	5326	5333	5335	5355	5362	5364	5392	5399	5401	5420	5427	5429
	5458	5467	5469	5501	5510	5512	5534	5544	5546	5569	5600	5602	5720	5736	5738
	5886	5893	5895	5919	5933	5935	5969	5971	6006	6008	6042	6044	6082	6093	6095
	6147	6149	6216	6225	6227	6248	6250	6285	6287	6323	6333	6335	6410	6412	6464

6474	6476	6530	6532	6598	6600	6623	6633	6635	6700	6702	6770	6780	6782	6848
6850	6915	6917	6985	6987	7060	7068	7070	7114	7122	7124	7167	7174	7176	7212
7219	7221	7274	7283	7285	7319	7321	7349	7351	7380	7382	7410	7412	7440	7442
7470	7472	7501	7509	7511	7542	7544	7584	7586	7626	7628	7668	7670	7710	7712
7752	7754	7794	7802	7804	7822	7827	7829	7847	7854	7856	7891	7896	7898	7920
7922	7952	7974	7976	8025	8032	8034	8076	8082	8084	8099	8105	8107	8128	8142
8144	8194	8205	8207	8238	8240	8271	8273	8325	8327	8380	8387	8389	8406	8408
8425	8427	8445	8447	8464	8466	8483	8485	8487	8506	8508	8525	8527	8587	8591
8625#	8700	8702	8818	8820	8899	8901	8986	8988	9068	9070	9082	9084	9101	9103
9120	9122	9155	9157	9264	9266	9279	9281	9298	9300	9317	9319	9351	9353	9458
9460	9478	9480	9491	9493	9510	9512	9529	9531	9564	9566	9673	9675	9686	9688
9705	9707	9724	9726	9759	9761	9865	9867	9886	9888	9899	9901	9918	9920	9937
9939	9973	9975	10089	10091	10102	10104	10121	10123	10140	10142	10175	10177	10281	10283
10294	10296	10313	10315	10333	10335	10368	10370	10475	10477	10489	10491	10509	10511	10544
10546	10559	10561	10574	10576	10589	10591	10604	10606	10619	10621	10644	10646	10660	10662
10690	10692	10706	10708	10731	10733	10755	10757	10780	10782	10811	10813	10839	10841	10865
10867	10899	10901	11005	11007	11036	11038	11069	11071	11104	11106	11129	11131	11150	11152
11197	11199	11244	11246	11274	11276	11301	11303	11332	11334	11383	11385	11420	11422	11440
11442	15349	15354	15463	15549	15606	15673	15750							
TRMTRP	15771#													
TYPDEC	331#													
VTRP	10543#	10544	10559	10574	10589	10604	10619	10644						
\$ASHC	11646#	12556	12590	12624	12660	12694	12728	12764	12798					
\$ASHC\$	11646#	12844	12881	12918	12955	12992	13029	13066	13103	13140	13177			
\$ASH\$	11646#	11982	12011	12040	12069	12098	12127	12156	12185	12214	12243			
\$DIV	13221#	14228	14270	14312	14356	14398	14440	14484	14526	14568	14610	14654	14696	14738
	14824	14866	14896	14928	14970	15012	15054	15096	15138	15180	15222	15264	15306	14782
\$MUL	13221#	13228	13263	13298	13333	13370	13405	13440	13477	13512	13547	13582	13619	13654
	13724	13761	13796	13831	13872	13907	13942	13977	14012	14047	14082	14117	14152	14187
\$SEPCD	317#	464	473	484	493	523	542	561	579	613	632	642	666	710
	749	767	785	803	827	852	875	897	921	946	970	995	1020	1045
	1095	1128	1146	1164	1182	1217	1230	1264	1277	1306	1319	1348	1361	1409
	1429	1457	1475	1498	1510	1536	1549	1576	1591	1622	1638	1670	1686	1718
	1762	1780	1808	1818	1833	1842	1865	1875	1889	1898	1921	1935	1944	1979
	2028	2046	2081	2098	2125	2134	2145	2154	2165	2194	2210	2250	2264	2292
	2334	2346	2370	2379	2388	2415	2424	2434	2460	2469	2493	2505	2531	2544
	2609	2642	2674	2691	2723	2733	2766	2775	2791	2801	2835	2845	2882	2897
	2939	2948	2982	2992	3025	3034	3068	3077	3105	3130	3158	3167	3194	3207
	3233	3244	3276	3286	3312	3321	3331	3343	3353	3382	3394	3423	3452	3482
	3528	3559	3588	3598	3629	3638	3668	3678	3706	3732	3759	3783	3792	3816
	3848	3859	3868	3893	3902	3915	3925	3934	3963	3972	3986	3996	4022	4031
	4066	4075	4086	4095	4104	4130	4139	4148	4173	4182	4191	4216	4225	4234
	4273	4303	4314	4323	4350	4359	4371	4380	4389	4417	4426	4435	4445	4455
	4493	4502	4531	4540	4552	4561	4570	4602	4611	4620	4631	4640	4671	4680
	4700	4709	4739	4748	4757	4767	4776	4811	4849	4886	4917	4947	4962	4996
	5028	5061	5078	5097	5127	5141	5155	5188	5224	5254	5285	5312	5321	5348
	5386	5413	5442	5451	5482	5492	5526	5560	5615	5624	5638	5650	5659	5672
	5699	5713	5754	5773	5792	5810	5828	5844	5860	5879	5905	5914	5950	5965
	6002	6024	6038	6061	6076	6111	6127	6142	6165	6179	6193	6209	6243	6266
	6303	6318	6350	6364	6378	6390	6405	6428	6443	6458	6493	6509	6525	6548
	6579	6593	6616	6651	6666	6681	6696	6719	6734	6749	6764	6799	6814	6828
	6867	6882	6897	6911	6934	6949	6963	6980	7004	7020	7035	7053	7089	7109
	7162	7191	7207	7232	7247	7256	7268	7298	7314	7335	7344	7366	7375	7396
	7426	7435	7456	7465	7486	7495	7523	7537	7561	7570	7579	7603	7612	7621
	7654	7663	7687	7696	7705	7729	7738	7747	7771	7780	7789	7816	7841	7866
	7884	7909	7915	7935	7945	8009	8070	8094	8123	8163	8186	8221	8233	8254
														8266

	8289	8299	8310	8320	8343	8354	8365	8375	8401	8420	8440	8459	8478	8501	8520
	8540	8544	8594	8598	8602	8606	8610	8614	8618	8622	8714	8726	8738	8751	8764
	8777	8789	8802	8814	8835	8850	8865	8880	8895	8911	8921	8931	8941	8950	8960
	8970	8980	8999	9008	9017	9026	9037	9046	9055	9064	9078	9097	9116	9137	9151
	9171	9180	9189	9198	9207	9220	9229	9238	9247	9259	9275	9294	9313	9334	9347
	9367	9376	9385	9394	9403	9416	9425	9434	9443	9454	9470	9487	9506	9525	9546
	9560	9580	9589	9598	9607	9616	9629	9638	9647	9656	9668	9682	9701	9720	9741
	9755	9775	9784	9793	9802	9811	9824	9833	9842	9851	9861	9877	9895	9914	9933
	9955	9969	9989	9998	10007	10016	10025	10038	10047	10056	10065	10077	10098	10117	10136
	10157	10171	10191	10200	10209	10218	10227	10240	10249	10258	10267	10277	10290	10309	10328
	10350	10364	10384	10393	10402	10411	10420	10433	10442	10451	10460	10470	10484	10504	10525
	10532	10539	10555	10570	10585	10600	10615	10630	10640	10655	10684	10702	10718	10727	10749
	10774	10796	10807	10824	10834	10856	10860	10884	10920	10931	10940	10964	10977	10987	11025
	11029	11059	11063	11093	11121	11125	11143	11175	11191	11221	11230	11240	11257	11266	11291
	11310	11319	11344	11348	11357	11366	11375	11407	11485	11490	11500				
\$SERNU	317#	462	473	484	493	523	542	561	579	613	632	642	666	686	710
	749	767	785	803	827	852	875	897	921	946	970	995	1020	1045	1070
	1095	1128	1146	1164	1182	1217	1230	1264	1277	1306	1319	1348	1361	1409	1420
	1429	1457	1475	1498	1510	1536	1549	1576	1591	1622	1638	1670	1686	1718	1734
	1762	1780	1808	1818	1833	1842	1865	1875	1889	1898	1921	1935	1944	1979	1993
	2028	2046	2081	2098	2125	2134	2145	2154	2165	2194	2210	2250	2264	2292	2304
	2334	2346	2370	2379	2388	2415	2424	2434	2460	2469	2493	2505	2531	2544	2577
	2609	2642	2674	2691	2723	2733	2766	2775	2791	2801	2835	2845	2882	2897	2907
	2939	2948	2982	2992	3025	3034	3068	3077	3105	3130	3158	3167	3194	3207	3218
	3233	3244	3276	3286	3312	3321	3331	3343	3353	3382	3394	3423	3452	3482	3518
	3528	3559	3588	3598	3629	3638	3668	3678	3706	3732	3759	3783	3792	3816	3825
	3848	3859	3868	3893	3902	3915	3925	3934	3963	3972	3986	3996	4022	4031	4040
	4066	4075	4086	4095	4104	4130	4139	4148	4173	4182	4191	4216	4225	4234	4264
	4273	4303	4314	4323	4350	4359	4371	4380	4389	4417	4426	4435	4445	4455	4484
	4493	4502	4531	4540	4552	4561	4570	4602	4611	4620	4631	4640	4671	4680	4689
	4700	4709	4739	4748	4757	4767	4776	4811	4849	4886	4917	4947	4962	4996	5011
	5028	5061	5078	5097	5127	5141	5155	5188	5224	5254	5285	5312	5321	5348	5377
	5386	5413	5442	5451	5482	5492	5526	5560	5615	5624	5638	5650	5659	5672	5685
	5699	5713	5754	5773	5792	5810	5828	5844	5860	5879	5905	5914	5950	5965	5987
	6002	6024	6038	6061	6076	6111	6127	6142	6165	6179	6193	6209	6243	6266	6281
	6303	6318	6350	6364	6378	6390	6405	6428	6443	6458	6493	6509	6525	6548	6563
	6579	6593	6616	6651	6666	6681	6696	6719	6734	6749	6764	6799	6814	6828	6844
	6867	6882	6897	6911	6934	6949	6963	6980	7004	7020	7035	7053	7089	7109	7144
	7162	7191	7207	7232	7247	7256	7268	7298	7314	7335	7344	7366	7375	7396	7405
	7426	7435	7456	7465	7486	7495	7523	7537	7561	7570	7579	7603	7612	7621	7645
	7654	7663	7687	7696	7705	7729	7738	7747	7771	7780	7789	7816	7841	7866	7876
	7884	7909	7915	7935	7945	8009	8070	8094	8123	8163	8186	8221	8233	8254	8266
	8289	8299	8310	8320	8343	8354	8365	8375	8401	8420	8440	8459	8478	8501	8520
	8540	8544	8594	8598	8602	8606	8610	8614	8618	8622	8714	8726	8738	8751	8764
	8777	8789	8802	8814	8835	8850	8865	8880	8895	8911	8921	8931	8941	8950	8960
	8970	8980	8999	9008	9017	9026	9037	9046	9055	9064	9078	9097	9116	9137	9151
	9171	9180	9189	9198	9207	9220	9229	9238	9247	9259	9275	9294	9313	9334	9347
	9367	9376	9385	9394	9403	9416	9425	9434	9443	9454	9470	9487	9506	9525	9546
	9560	9580	9589	9598	9607	9616	9629	9638	9647	9656	9668	9682	9701	9720	9741
	9755	9775	9784	9793	9802	9811	9824	9833	9842	9851	9861	9877	9895	9914	9933
	9955	9969	9989	9998	10007	10016	10025	10038	10047	10056	10065	10077	10098	10117	10136
	10157	10171	10191	10200	10209	10218	10227	10240	10249	10258	10267	10277	10290	10309	10328
	10350	10364	10384	10393	10402	10411	10420	10433	10442	10451	10460	10470	10484	10504	10525
	10532	10539	10555	10570	10585	10600	10615	10630	10640	10655	10684	10702	10718	10727	10749
	10774	10796	10807	10824	10834	10856	10860	10884	10920	10931	10940	10964	10977	10987	11025
	11029	11059	11063	11093	11121	11125	11143	11175	11191	11221	11230	11240	11257	11266	11291

														704	744	762	780
														681	681	681	681
														660	660	660	660
														1064	1089	1123	1141
														1585	1680	1728	1774
														2340	2429	2464	2500
														2987	3072	3100	3125
														3593	3633	3701	3727
														4229	4318	4384	4450
														5022	5091	5182	5218
														5708	5873	5959	5996
														6587	6610	6758	6838
														7400	7430	7490	7532
														8261	8315	8396	8415
														9146	9289	9342	9449
														10112	10131	10166	10272
														11261	11285	10304	10323
														611	640	663	684
														992	1042	1067	707
														1345	1407	1092	747
														1668	1716	1455	765
														1933	1942	1760	1144
														2248	2261	1777	1162
														2843	2879	1805	1472
														3165	3192	1855	1496
														3421	3450	1877	1516
														3780	3790	1899	1536
														3994	4019	1921	1556
														4213	4223	1943	1576
														4433	4443	1965	1596
														4653	4687	1987	1616
														4873	4884	2009	1636
														5093	5025	2031	1656
														5319	5375	2053	1676
														5657	5670	2075	1696
														5999	6021	2097	1716
														6340	6315	2119	1736
														6682	6648	2141	1756
														7024	6894	2163	1776
														7366	6908	2185	1796
														7708	7204	2207	1816
														8050	7244	2229	1836
														8392	7265	2251	1856
														8734	7287	2273	1876
														9076	7309	2295	1896
														9418	7331	2317	1916
														9760	7353	2339	1936
														10102	7375	2361	1956
														10444	7397	2383	1976
														10786	7419	2405	1996
														11128	7441	2427	2016
														7661	7463	2449	2036
														8120	7485	2471	2056
														8418	7507	2493	2076
														8716	7529	2515	2096
														9014	7551	2537	2116
														9312	7573	2559	2136
														9610	7595	2581	2156
														9908	7617	2603	2176
														10206	7639	2625	2196
														10504	7661	2647	2216
														10802	7683	2669	2236
														11100	7705	2691	2256
														627	8219	2713	2276
														608	8241	2735	2296
														989	8263	2757	2316
														1014	8285	2779	2336
														1543	8307	2801	2356
														2298	8329	2823	2376
														2943	8351	2845	2396
														3574	8373	2867	2416
														4186	8395	2889	2436
														4881	8417	2911	2456
														5520	8439	2933	2476
														6159	8461	2955	2496
														6819	8483	2977	2516
														7479	8505	2999	2536
														8117	8527	3021	2556
														8756	8549	3043	2576
														9395	8571	3065	2596
														10034	8593	3087	2616
														10673	8615	3109	2636
														11312	8637	3131	2656
														627	8659	3153	2676
														608	8681	3175	2696
														989	8703	3197	2716
														1014	8725	3219	2736
														1543	8747	3241	2756
														2298	8769	3263	2776
														2943	8791	3285	2796
														3574	8813	3307	2816
														4186	8835	3329	2836
														4881	8857	3351	2856
														5520	8879	3373	2876
														6159	8901	3395	2896
														6819	8923	3417	2916
														7479	8945	3439	2936
														8117	8967	3461	2956
														8756	8989	3483	2976
														9395	9011	3505	2996
														10034	9033	3527	3016
														10673	9055	3549	3036
														11312	9077	3571	3056
														627	9099	3593	3076
														608	9121	3615	3096
														989	9143	3637	3116
														1014	9165	3659	3136
														1543	9187	3681	3156
														2298	9209	3703	3176
														2943	9231	3725	3196
														3574	9253	3747	3216
														4186	9275	3769	3236
														4881	9297	3791	3256
														5520	9319	3813	3276
														6159	9341	3835	3296
														6819	9363	3857	3316
														7479	9385	3879	3336
														8117	9407	3901	3356
														8756	9429	3923	3376
														9395	9451	3945	3396
														10034	9473	3967	3416
														10673	9495	3989	3436
														11312	9517	4011	3456
														627	9539	4033	3476
														608	9561	4055	3496
														989	9583	4077	3516
														1014	9605	4099	3536
														1543	9627	4121	3556
														2298	9649	4143	3576
														2943	9671	4165	3596
														3574	9693	4187	3616
														4186	9715	4209	3636
														4881	9737	4231	3656
														5520	9759	4253	3676
														6159	9781	4275	3696
														6819	9803	4297	3716
														7479	9825	4319	3736
														8117	9847	4341	3756
														8756	9869	4363	3776
														9395	9891	4385	3796
														10034	9913	4407	3816
														10673	9935	4429	3836
														11312	9957	4451	3856
														627	9979	4473	3876
														608	10001	4495	3896
														989	10023	4517	3916
														1014	10045	4539	3936
														1543	10067	4561	3956
														2298	10089	4583	3976
														2943	10111	4605	3996
														3574	10133	4627	4016
														4186	10155	4649	4036
														4881	10177	4671	4056
														5520	10199	4693	4076
														6159	10221	4715	4096
														6819	10243	4737	4116
														7479	10265	4759	4136
														8117	10287	4781	4156
														8756	10309	4803	4176
														9395	10331	4825	4196
														10034	10353	4847	4216
														10673	10375	4869	4236
														11312	10397	4891	4256
														627	10419	4913	4276
														608	10441	4935	4296
														989	10463	4957	4316
														1014	10485	4979	4336
														1543	10507	5001	4356
														2298	10529	5023	4376
														2943	10551	5045	4396
														3574	10573	5067	4416
														4186	10595	5089	4436
														4881	10617	5111	4456
														5520	10639	5133	4476
														6159	10661	5155	4496
														6819	10683	5177	4516
														7479	10705	5199	4536
														8117	10727	5221	4556
														8756	10749	5243	4576
														9395	10771	5265	4596
														10034	10793	5287	4616
														10673	10815	5309	4636
														11312	10837	5331	4656
														627	10859	5353	4676
														608	10881	5375	4696
														989	10903	5397	4716
														1014	10925	5419	4736
														1543	10947	5441	4756
														2298	10969	5463	4776
														2943	10991	5485	4796
														3574	11013	5507	4816
														4186	11035	5529	4836
														4881	11057	5551	4856
														5520	11079	5573	4876
														6159	11101	5595	4896
														6819	11123	5617	4916
														7479	11145	5639	4936
														8117	11167	5661	4956
														8756	11189	5683	4976
														9395	11211	5705	4996
														10034	11233	5727	5016
														10673	11255	5749	5036
														11312	11277	5771	5056
														627	11299	5793	5076
														608	11321	5815	5096
														989	11343	5837	5116
														1014	11365	5859	5136
														1543	11387	5881	5156
														2298	11409	5903	5176
														2943	11431	5925	5196
														3574	11453	5947	5216
														4186	11475	5969	5236
														4881	11497	5991	5256
														5520	11519	6013	5276
														6159	11541	6035	5296
														6819	11563	6057	5316
														7479	11585	6079	5336
														8117	11607	6101	5356
														8756	11629	6123	5376
														9395	11651	6145	5396
														10034	11673	6167	5416
														10673	11695	6189	5436
														11312	11717	6211	5456
														627	11739	6233	5476
														608	11761	6255	5496
														989	11783	6277	5516
														1014	11805	6299	5536
														1543	11827	6321	5556
														2298	11849	6343	5576
														2943	11871	6365	5596
														3574	11893	6387	5616
														4186	11915	6409	5636
														4881	11937	6431	5656
														5520	11959	6453	5676
														6159	11981	6475	5696
														6819	12003	6497	5716
														7479	12025	6519	5736
														8117	12047	6541	5756
														8756	12069	6563	5776
														9395	12091	6585	5796
														10034	12113	6607	5816
														10673	12135	6629	5836
														11312	12157	6651	5856
														627	12179	6673	5876
														608	12201	6695	5896

1169	1201	1248	1290	1332	1397	1444	1486	1523	1561	1606	1653	1701	1744	1790
1846	1902	1963	2010	2064	2102	2179	2230	2277	2318	2351	2392	2438	2473	2518
2558	2590	2622	2654	2703	2746	2813	2859	2919	2960	3004	3047	3089	3114	3139
3181	3255	3297	3362	3406	3435	3465	3499	3541	3572	3613	3650	3691	3717	3744
3764	3797	3830	3873	3940	4001	4045	4109	4153	4196	4245	4284	4333	4399	4466
4515	4584	4653	4721	4793	4831	4869	4900	4929	4977	5040	5110	5168	5203	5237
5267	5299	5333	5362	5399	5427	5467	5510	5544	5600	5736	5893	5933	5969	6006
6042	6093	6147	6225	6248	6285	6333	6410	6474	6530	6598	6633	6700	6780	6848
6915	6985	7068	7122	7174	7219	7283	7319	7349	7380	7410	7440	7470	7509	7542
7584	7626	7668	7710	7752	7802	7827	7854	7896	7920	7974	8032	8082	8105	8142
8205	8238	8271	8325	8387	8406	8425	8445	8464	8485	8506	8525	8700	8818	8899
8986	9068	9082	9101	9120	9155	9264	9279	9298	9317	9351	9458	9478	9491	9510
9529	9564	9673	9686	9705	9724	9759	9865	9886	9899	9918	9937	9973	10089	10102
10121	10140	10175	10281	10294	10313	10333	10368	10475	10489	10509	10544	10559	10574	10589
10604	10619	10644	10660	10690	10706	10731	10755	10780	10811	10839	10865	10899	11005	11036
11069	11104	11129	11150	11197	11244	11274	11301	11332	11383	11420	11440			
\$\$SET	15771#	15780	15781	15782	15783									
.SETUP	331#	11637												
.\$ACT1	331#													
.\$APT8	331#	342												
.\$APTH	331#	369												
.\$APTY	331#	15547												
.\$EOP	331#													
.\$TRAP	331#	15748												
.\$YPD	331#	15604												
.\$TYPE	331#	15461												
.\$TYPO	331#	15671												

. ABS. 062372 000

ERRORS DETECTED: 0

CJKDBC.BIN,CJKDBC.LST/CRF/SOL/NL:TOC-CJKDBC.P11
RUN-TIME: 144 183 20 SECONDS
RUN-TIME RATIO: 623/348-1.7
CORE USED: 25K (49 PAGES)