

PDP11/34

BASIC INSTRUCTION TEST
MD-11-DFKAA-B

EP-DFKAA-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 2

NOV 1976
digital
MADE IN USA

PDP11/34

11/34 CPU TEST
MD-11-DFKAA-B

EP-DFKAA-B-DL-A
COPYRIGHT © 1976
FICHE 2 OF 2

NOV 1976
digital
MADE IN USA

A vertical strip of microfiche frames on the left side of the page. The frames contain data in a grid format, likely representing test results or system logs. The data is too small to read clearly but appears to be organized in columns and rows.

.REM %

IDENTIFICATION

PRODUCT CODE:	MAINDEC-11-DFKAA-B
PRODUCT NAME:	PDP-11/34 CPU TEST
DATE CREATED:	OCTOBER 1976
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	GLENN JOHNSON
REVISED BY:	MARY MCNALLY AUGUST 1976

COPYRIGHT (C) 1975,1976 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.

MAINDEC-11-DFKAA-B 11/34 CPU TEST
DFKAA.B.P11

50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78

* SUMMARY OF OPERATING INSTRUCTIONS *
*

THE FOLLOWING PROCEDURE CAN BE USED TO RUN THIS DIAGNOSTIC
IN A STANDARD CONFIGURATION WITH AT LEAST 4K OF MEMORY
AND A TELETYPE. IF THE PROGRAM DOES NOT RUN SUCCESSFULLY
CONSULT THE FOLLOWING DOCUMENT FOR ASSISTANCE.

OPERATING PROCEDURES:

1. LOAD THE PROGRAM USING NORMAL PROCEDURES
2. START THE PROGRAM AT LOCATION 200
3. PROGRAM SHOULD PRINT "END OF PASS" WITHIN
THE 1ST SECOND AND REPEATABLY THEREAFTER
AT APPROX. 10 SEC. INTERVALS UNTIL
EXTERNALLY HALTED.
4. IF THE PROGRAM DOES NOT RUN AS DESCRIBED ABOVE,
CONSULT THE FULL OPERATING INTRUCTIONS WHICH
FOLLOW.

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
128

1.0 GENERAL PROGRAM INFORMATION

1.1 PROGRAM PURPOSE

THIS DIAGNOSTIC PROGRAM IS DESIGNED TO BE A COMPREHENSIVE CHECK OF THE PDP-11/34 BASIC INSTRUCTION SET. THE PROGRAM EXERCISES ALL OF THE PROCESSOR LOGIC AND MICROCODE FOR ALL INSTRUCTIONS EXCEPT THE TRAP AND MEMORY MANAGEMENT INSTRUCTIONS. THE PROGRAM DOES NOT TEST INSTRUCTIONS OR HARDWARE RELATED TO THE TRAP OR INTERRUPT MECHANISMS OF THE 11/34 (E.G. RTT, RT1, WAIT, RESET, TRAP, EMT).

1.2 SYSTEM REQUIREMENTS

1.2.1 HARDWARE

PDP-11/34 PROCESSOR
BK MEMORY -- THE PROGRAM USES LOCATIONS 0 - 26214

1.2.2 SOFTWARE

THIS PROGRAM IS WRITTEN TO BE RUN AS A STAND-ALONE PROGRAM. HOWEVER, THE PROGRAM IS DESIGNED TO RUN UNDER AUTOMATED PRODUCT TEST SYSTEM (APT) IN ALL THREE MODES.

THE PROGRAM CAN ALSO BE RUN UNDER THE ACT 11 MONITOR

1.3 RELATED DOCUMENTS AND STANDARDS

- PDP-11/34 MICROCODE LISTING
- PDP-11/34 ELECTRICAL SCHEMATICS
- DIAGNOSTIC ENGINEERING PROJECT PLANFOR 11/34
- DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS PROGRAMMING PRACTICES
DOCUMENT NO. 175-003-009-00
- APT INTERFACE SPECIFICATION, REVISION 9.

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

NONE

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

1.5 FAILURE ASSUMPTIONS

NONE

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

2.1.1 LOADING

USE NORMAL PROCEDURES FOR LOADING ABSOLUTE BINARY TAPES.

2.1.2 NORMAL START

THIS IS THE PROCEDURE FOR NORMAL PROGRAM RUNNING (I.E., STARTING WITH TEST 1 AND EXECUTING ENTIRE DIAGNOSTIC).

LOAD ADDRESS = 200
START

2.1.3 SUBTEST START

THIS IS THE PROCEDURE FOR STARTING AT A SUBTEST OTHER THAN 1.

1. LOAD \$TESTN (IN MAILBOX SECTION) WITH THE NUMBER OF SUBTEST MINUS ONE (IN OCTAL) FOR EXAMPLE, TO START AT SUBTEST 100, \$TESTN=77.
2. LOAD STARTING ADDRESS OF SUBTEST IN LOC. 216
3. LOAD ADDRESS = 204
4. START

2.2 SPECIAL ENVIRONMENTS

THIS PROGRAM IS WRITTEN TO COMPLY WITH ALL THE REQUIREMENTS OF THE APT INTERFACE SPECIFICATION. IT WILL RUN UNDER APT IN EITHER QUICK VERIFY, PROGRAM OR RUN-TIME MODES.

THIS PROGRAM IS WRITTEN TO COMPLY WITH ALL OF THE REQUIREMENTS OF PROGRAMS TO RUN UNDER THE ACT11 MONITOR.



179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234

2.3 PROGRAM OPTIONS

THIS PROGRAM IS INTENDED TO BE A BASIC PROCESSOR TEST.
IT IS INTENDED TO BE THE LOWEST LEVEL DIAGNOSTIC RUN.
IT PROVIDES FOR NO SELECTABLE OPTIONS.

IN ORDER THAT THE TEST BE RUNNABLE ON A PROCESSOR WITHOUT A
TELETYPE, IT IS POSSIBLE TO SUPPRESS THE END OF PASS MESSAGE.
IF NO TELETYPE IS AVAILABLE, ALTER THE BYTE, SENVM, WHICH
IS LOCATED IN THE APT MAILBOX. SETTING SENVM TO 40(8) WILL
SUPPRESS ALL CONSOLE OUTPUT.
THE EXACT LOCATION OF THIS BYTE CAN BE FOUND IN THE SYMBOL
TABLE AT THE END OF THE LISTING.

2.4 EXECUTION TIMES

THE DIAGNOSTIC COMPLETES THE FIRST PASS IN LESS THAN 1 SEC.
SUBSEQUENT PASSES REQUIRE APPROXIMATELY 10 SECS. EACH.
THE PROGRAM WILL RUN CONTINUOUSLY UNTIL EXTERNALLY HALTED.

3.0 ERROR INFORMATION

3.1 ERROR TYPES

THERE ARE TWO BASIC TYPES OF ERRORS IN THE DIAGNOSTIC.

3.1.1 FUNCTIONAL ERRORS

THESE ARE ERRORS WHICH REPRESENT A MALFUNCTION OF AN
INSTRUCTION OR SEQUENCE OF INSTRUCTION. (E.G., THE PROPER
CONDITION CODE NOT SET OR IMPROPER RESULT OF AN ARITHMETIC
OR LOGICAL OPERATION).

3.1.2 SEQUENCE ERRORS

THE RESULT OF A TESTS BEING EXECUTED OUT OF SEQUENCE. (E.G.
WILD MACHINE OR IMPROPER BRANCH OR JUMP).

3.2 ERROR REPORTING PROCEDURES

THE DIAGNOSTIC RESPONDS TO THE DETECTION OF ALL ERRORS BY
STORING CERTAIN INFORMATION IN MEMORY AND HALTING THE PROCESSOR.
THE INFORMATION STORED IN MEMORY CAN BE USED BY THE OPERATOR
TO IDENTIFY THE ERROR DETECTED.

CERTAIN FAILURES WILL CAUSE THE PROESSOR TO HANG.
THIS TYPE OF FAILURE IS INDICATED IF THE PROGRAM

GO1

MAINDEC-11-DFKAA-B 11/34 CPU TEST
DFKARB.P11

MACY11 27(732) 01-OCT-76 15:03 PAGE 233

235
236
237

DUES NOT PRINT ITS END OF PASS INDICATION WITHIN A REASONABLE
AMOUNT OF TIME. (FIRST MESSAGE SHOULD APPEAR WITHIN 1 SEC.)

238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280

3.3 ERROR DESCRIPTOR INFORMATION

THE DIAGNOSTIC MAILBOX HOLDS THE ERROR INFORMATION NECESSARY TO IDENTIFY THE DETECTED ERROR. THIS INFORMATION HAS BEEN DESIGNED FOR COMPLIANCE WITH THE APT TO DIAGNOSTIC INTERFACE SPECIFICATION. IT IS THE PRIMARY MEDIUM FOR IDENTIFYING ERRORS.

3.2.1 \$MSGTYP

THIS LOCATION IS INCREMENTED FROM ZERO TO ONE BEFORE THE PROGRAM COMES TO A PROGRAMMED HALT. IF THIS LOCATION IS NOT ONE, THEN THE DIAGNOSTIC HAS COME TO AN UNPROGRAMMED HALT. CHECK THE STACK AND PC FOR A CLUE TO THE CAUSE. SUSPECT A TRAP.

3.2.2 \$FATAL

THIS LOCATION IS LOADED WITH A NUMBER BEFORE A HALT IS EXECUTED. EACH PROGRAMMED HALT HAS A UNIQUE NUMBER ASSOCIATED WITH IT WHICH CAN BE USED TO IDENTIFY THE ERROR WHICH HAS BEEN DETECTED.

3.2.3 \$PASS

THIS LOCATION IS INCREMENTED FOR EVERY COMPLETE PASS OF THE DIAGNOSTIC. MONITORING THIS LOCATION WILL INDICATE WHETHER OR NOT THE PROGRAM IS HUNG. IT WILL ALSO INDICATE THE NUMBER OF SUCCESSFUL PASSES COMPLETED BEFORE THE ERROR HALT. A HIGH PASS COUNT MIGHT INDICATE THAT THE ERROR HALT IS ASSOCIATED WITH AN INTERMITTANT FAULT.

3.2.4 \$TESTN

THIS LOCATION IS INCREMENTED IN EACH NEW SUBTEST. THIS SHOULD INDICATE THE TEST BEING EXECUTED WHEN THE ERROR WAS DETECTED. THIS LOCATION IS ALSO USED TO DETECT A SEQUENCE ERROR.

281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318

3.4 ERROR IDENTIFICATION

BECAUSE OF THE OVERHEAD ASSOCIATED WITH EACH HALT IN AN APT COMPATIBLE PROGRAM THE SEQUENCE CHECK CODE WILL SHARE THE ERROR HALT OF FUNCTIONAL ERROR WITHIN EACH SUBTEST. TO DETERMINE WHICH ERROR IS BEING REPORTED, LOCATIONS \$FATAL AND \$TESTN ARE USED TOGETHER. WHEN AN ERROR HALT OCCURS, CHECK \$FATAL TO DETERMINE THE NUMBER OF THE ERROR DETECTED. NOW, CHECK THAT THE TEST NUMBER WHERE THIS ERROR IS DETECTED CORRESPONDS TO THE VALUE IN \$TESTN. IF THESE AGREE THE ERROR WAS A FUNCTIONAL ERROR AS DESCRIBED IN THE LISTINGS. IF THESE NUMBERS DO NOT AGREE, THEN A SEQUENCE ERROR WAS DETECTED. IN THIS CASE \$TESTN WILL CONTAIN ONE MORE THAN THE NUMBER OF THE LAST TEST SUCCESSFULLY COMPLETED. SEQUENCE ERRORS WHICH SHARE THE ERROR HALTS OF FUNCTIONAL ERRORS WILL ALWAYS BE REPORTED BY THE LAST HALT IN THE SUBTEST IN WHICH THEY WERE DISCOVERED.

4.0 PROGRESS REPORT

AT THE END OF EACH SUCCESSFUL PASS (THE EQUIVALENT OF 400 (8) PROGRAM PASSES, EXCEPT THE FIRST PASS WHICH IS ONLY ONE PROGRAM PASS) THE PROGRAM INCREMENTS THE LOCATION \$PASS WHICH IS IN THE APT MAILBOX. THIS LOCATION WILL ALWAYS CONTAIN THE NUMBER OF SUCCESSFUL PASSES COMPLETED. \$PASS IS RESET WITH EVERY RETART FROM LOC. 200.

ADDITIONALLY, THE MESSAGE END OF DFKAA IS PRINTED ON THE CONSOLE TELETYPE AFTER THE FIRST PASS AND FOLLOWING EVERY SUBSEQUENT PASS (400 PROGRAM LOOPS) THEREAFTER.

IF NO TELETYPE IS AVAILABLE, THE CONSOLE OUTPUT MUST BE SUPPRESSED. (SEE SECTION 2.3).

319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366

5.0 TROUBLE SHOOTING

WHEN THE PROGRAM DISCOVERS A FAULT IT WILL HALT. TO DETERMINE THE CAUSE OF THE HALT, THE DIAGNOSTIC PROVIDES ERROR INFORMATION. THIS INFORMATION IS STORED IN THE APT MAILBOX AND IS THE PRIMARY SOURCE OF ERROR IDENTIFICATION.

UPON FINDING AN ERROR, THE FOLLOWING PROCEDURE SHOULD AID IN ISOLATING THE FAULT.

5.1 CHECK THE MAILBOX

1. \$MSGTY THIS LOCATION SHOULD CONTAIN A 1. IF THE PROCESSOR HALTS AND THIS LOCATION IS ZERO, THEN THE PROCESSOR HAS COME TO AN UNEXPECTED HALT. FIRST SUSPECT A TRAP. CHECK THE PC AND IF A TRAP CHECK R6 AND THE STACK FOR THE LOCATION OF THE FAILING INSTRUCTION.
2. \$FATAL THIS LOCATION IS USED TO HOLD THE NUMBER OF THE ERROR WHICH HAS DETECTED. EACH ERROR BEING CHECKED BY THE DIAGNOSTIC IS ASSIGNED A UNIQUE NUMBER WHICH IS STORED IN \$FATAL WHEN THAT ERROR IS DETECTED.

WHEN AN ERROR IS DETECTED, CHECK THE LISTING TO SEE THAT THE ERROR NUMBER STORED IN \$FATAL IS ONE WHICH IS DETECTED IN THE TEST WHOSE NUMBER IS IN \$TESTN. IF THERE IS A DISAGREEMENT THEN THE ERROR BEING REPORTED IS A SEQUENCE ERROR. \$TESTN CONTAINS ONE MORE THAN THE LAST TEST WHICH WAS SUCCESSFULLY COMPLETED.

3. \$TESTN THIS LOCATION IS USED TO INDICATE THE NUMBER OF THE TEST WHICH WAS BEING EXECUTED WHEN THE FAULT WAS DETECTED. \$TESTN IS USED IN CONJUNCTION WITH \$FATAL TO DISTINGUISH BETWEEN SEQUENCE AND FUNCTIONAL ERRORS. (SEE 2. THIS SECTION)
4. \$PASS THIS LOCATION IS USED TO INDICATE THE NUMBER OF SUCCESSFUL PASSES WHICH THE DIAGNOSTIC HAS COMPLETED. THIS WILL GIVE AN INDICATION THAT THE DIAGNOSTIC HAS NOT JUST BEEN HUNG IN A LOOP IF NOT TELETYPE IS AVAILABLE TO REPORT THE PRINTED PROGRESS REPORTS.

IF AN ERROR HAS BEEN DETECTED \$PASS WILL SHOW WHETHER IT WAS A HARD ERROR DISCOVERED DURING THE FIRST TRY OR WHETHER IT WAS INTERMITTANT OR DEVELOPED DURING THE RUNNING OF THE DIAGNOSTIC.

367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422

5.2 SCOPING

WHILE THIS DIAGNOSTIC IS PRIMARILY INTENDED TO BE A FAULT DETECTION PROGRAM, PROVISIONS ARE MADE TO ASSIST A TECHNICIAN WHO MIGHT WANT TO USE THE PROGRAM AS A TROUBLE SHOOTING TEST.

THE PROCEDURE FOR SCOPING A SUBTEST INVOLVES MODIFYING SEVERAL MEMORY LOCATIONS IN THE TEST ITSELF. THE PHILOSOPHY IS TO PROVIDE A SCOPING LOOP WHICH WILL INCLUDE THE CODE WHERE THE ERROR WAS DETECTED. THE LOOP IS SET UP SO THAT THE LOOP WILL NOT BE TERMINATED SHOULD THE ERROR INTERMITTANTLY DISAPPEAR.

THE PROCEDURE IS AS FOLLOWS:

1. DETERMINE WHICH ERROR IS TO BE SCOPED. USE \$FATAL AND \$TESTN FOR THIS (SEE ABOVE)
2. LOCATE THE ERROR ROUTINE IN THE LISTING.
3. CLEAR THE RIGHT BYTE OF THE CONDITIONAL BRANCH INSTRUCTION ASSOCIATED WITH THE ERROR. (THIS IS MARKED WITH <===='S IN THE LISTING.)
4. REPLACE THE INSTRUCTION FOLLOWING <MOV #XXX, -(R2)> WITH THE SCOPING BRANCH PROVIDED IN THE LISTING COMMENTS.
5. RESTART THE PROGRAM. THE PROGRAM MAY BE RESTARTED FROM THE BEGINNING OR FROM THE SUBTEST (SEE 2.0).

6.0 LISTING

000500

000240
000007
000006
177776
177564
177566
140000
030000

000400

```

%
.TITLE MAINDEC-11-DFKAA-B 11/34 CPU TEST
.ENABLE ABS
STBOT=500
.NLIST CND,MC,MD
.LIST ME
SCOPE=NOP
R7=%7
R6=%6
PS=177776
TPS=177564
TPB=177566
USRM=140000
PUSRM=30000
.MCALL .SAPTHDR, .SAPTBL, .SACT11
.SBTTL ACT11 HOOKS
;*****
;HOOKS REQUIRED BY ACT11
SSVPC=. ;SAVE PC

```


423 000046 000046
424 000046 025626
425 000052 000052
426 000052 000000
427 000052 000400
428 000052 000300

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

.SBTTL APT MAILBOX-ETABLE

431
432
433 000300
434 000300 000000
435 000302 000000
436 000304 000000
437 000306 000000
438 000310 000000
439 000312 000000
440 000314 000000
441 000316 000000
442 000320
443 000320 000
444 000321 000
445 000322 000000
446 000324 000000
447 000326 000000

;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

454 000330
455
456
457

\$ETEND:
.MEXIT
.SBTTL APT PARAMETER BLOCK

458
459
460
461 000330
462 000024 000024
463 000024 000200
464 000044 000044
465 000044 000330
466 000044 000330

;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT

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

467
468
469
470

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

471 000330
472 000330 000000
473 000332 000300
474 000334 000010
475 000336 000010
476 000340 000000
477 000342 000014
478

\$APTHD:
\$HIBTS: .WORD 0 ;: TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
\$MBADR: .WORD \$MAIL ;: ADDRESS OF APT MAILBOX (BITS 0-15)
\$STMT: .WORD 10 ;: RUN TIM OF LONGEST TEST
\$PASTM: .WORD 10 ;: RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
\$UNITM: .WORD 0 ;: ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
; .WORD \$ETEND-\$MAIL/2 ;: LENGTH MAILBOX-ETABLE(WORDS)

```

479
480
481      000004
482      000004 026116
483      000006 000000
484      000010 026126
485      000012 000000
486      000014 026136
487
488      000030 026146
489      000032 000000
490      000034 026156
491      000036 000000
492
493      000114 026166
494      000116 000000
495
496      000244 026176
497      000246 000000
498      000250 026206
499      000252 000000
500
501
502
503
504      000370
505      000370 000000 000000 000000
506      000376 000000 000000 000000
507      000404 000001 000001 177777
508
509
510
511      000500
512      000200
513      000200 000167 000274
514
515      000204 012706 000500
516      000210 012702 000304
517      000214 000137
518      000216 000000
519
520      000500
521      000302
522      000304
523      000500 012737 026002 000024
524      000506 012737 000000 000306
525      000514 012737 177777 025652
526      000522 012706 000500
527      000526 012702 000304
528      000532 012737 000000 000304
529      000540 012737 000000 000302
530      000546 012737 000000 000300

```

:SOME POINTERS TO CPU TRAP HANDLERS

```

.=4
T04
0
T010
0
T014
.=30
T030
0
T034
0
.=114
T0114
0
.=244
T0244
0
T0250
0

```

:DATA TABLE FOR USE IN ADDRESSING MODE TESTS

```

.=370
0,0,0,0,0,0
1,1,-1
.=500

```

:SET UP STARTING ADDRESS

```

.$X=
.=200
JMP      START
MOV      #STBOT,R6
MOV      #STSTN,R2
JMP      @PC+
0

```

```

:SET STACK POINTER
:SET MAILBOX POINTER
:JUMP TO SUBTEST
:ADDR. OF SUBTEST GOES HERE

```

```

.=.X
$ERROR=$FATAL
$STSTNM=$STSTN

```

```

START:  MOV      #PWRDN,@#24
        MOV      #0,@#SPASS
        MOV      #-1,@#PASSPT
RESTRT: MOV      #STBOT,R6
        MOV      #STSTN,R2
        MOV      #0,@#STSTNM
        MOV      #0,@#$ERROR
        MOV      #0,@#$MSGTY

```

```

:SET UP FOR POWER FAIL
:CLEAR PASS COUNT
:SET PRINT COUNTER
:INITIALIZE STACK POINTER
:SET UP POINTER TO MESSAGE TYPE
:CLEAR TEST NUMBER
:CLEAR ERROR NUMBER
:CLEAR MESSAGE TYPE(FOR APT)

```



```

531
532
533
534 000554 005212
535 000556 022712 000001
536 000562 001024
537 000564 000257
538 000566 001401
539 000570 000404
540
541
542
543
544 000572
545 000572 012742 000001
546 000576 005242
547 000600 000000
548 000602
549 000602 001004
550
551
552
553
554 000604 012742 000002
555 000610 005242
556 000612 000000
557 000614 000264
558 000616 001001
559 000620 000404
560
561
562
563
564 000622
565 000622 012742 000003
566 000626 005242
567 000630 000000
568 000632
569 000632 001404
570
571
572
573
574 000634 012742 000004
575 000640 005242
576 000642 000000
577

```

```

;*****
;TEST 1 CHECK BRANCHES ON Z BIT
;*****
TST1:  INC      (R2)      ;UPDATE TEST NUMBER
        CMP      #1,(R2)  ;SEQUENCE ERROR?
        BNE     TST2-10  ;BR TO ERROR HALT ON SEQ ERROR
        CCC     ;CLEAR ALL CONDITION CODES
        BEQ     BR1      ;SHOULD BRANCH
        BR      BR2      ;BAD BRANCH OF Z-BIT
                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
                          ; BRANCH INSTRUCTION AND <====
                          ; REPLACE THE MOVE INSTRUCTION <====
                          ; FOLLOWING W/ 774 <====
BR1:   MOV      #1,-(R2)  ;MOVE TO MAILBOX # ***** 1 *****
        INC     -(R2)    ;SET MSGTYP TO FATAL ERROR
        HALT    ;SHOULD HAVE BRANCHED: Z=0
BR2:   BNE     BR3
                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
                          ; CONDITIONAL BRANCH INST. AND <====
                          ; REPLACE THE MOVE INSTRUCTION <====
                          ; WHICH FOLLOWS W/ 770 <====
BR3:   MOV      #2,-(R2)  ;MOVE TO MAILBOX # ***** 2 *****
        INC     -(R2)    ;SET MSGTYP TO FATAL ERROR
        HALT
        SEZ
        BNE     BR4
        BR      BR5
                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
                          ; BRANCH INSTRUCTION AND <====
                          ; REPLACE THE MOVE INSTRUCTION <====
                          ; FOLLOWING W/ 760 <====
BR4:   MOV      #3,-(R2)  ;MOVE TO MAILBOX # ***** 3 *****
        INC     -(R2)    ;SET MSGTYP TO FATAL ERROR
        HALT    ;SHOULD NOT HAVE BRANCHED HERE ON Z=1
BR5:   BEQ     TST2
                          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
                          ; CONDITIONAL BRANCH INST. AND <====
                          ; REPLACE THE MOVE INSTRUCTION <====
                          ; WHICH FOLLOWS W/ 754 <====
        MOV     #4,-(R2)  ;MOVE TO MAILBOX # ***** 4 *****
        INC     -(R2)    ;SET MSGTYP TO FATAL ERROR
        HALT    ;SHOULD HAVE BRANCHED ON Z=1
                          ; OR SEQUENCE ERROR

```


578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629

: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, THE UNIBUS
:DATA TRANSCIEVERS, AND AMUX CONTROL FOR ALU AND UBUS INPUTS.

IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0)
:TO SEE WHICH BITS OF THE DATA PATH ARE FAILING. IF THIS PROVIDES
:INCONCLUSIVE DATA, TRY TO CHECK MODE 3 IR DECODE BY RUNNING
:JUST THE MICROCODE AND IR DECODE TESTS FOR THE MOVE AND COMPARE
:INSTRUCTIONS.

:TEST 2 TEST OF ZEROES IN THE DATA PATH

TST2: INC (R2) ;UPDATE TEST NUMBER
CMP #2,(R2) ;SEQUENCE ERROR?
BNE TST3-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #0,#0 ;MOVE ZEROES THRU ADDRESS LINES, DATA
;LINES AND INTERNAL PATHS
TST #0 ;SUCCESSFUL?
BEQ TST3
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 772 <====
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

TST3: INC (R2) ;UPDATE TEST NUMBER
CMP #3,(R2) ;SEQUENCE ERROR?
BNE TST4-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #125252,#0 ;MOVE ALTERNATING ONES AND ZEROES
;THRU DATA PATHS
CMP #125252,#0 ;SUCCESSFUL
BEQ TST4
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
MOV #6,-(R2) ;MOVE TO MAILBOX # ***** 6 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DATA INCORRECT
; OR SEQUENCE ERROR

000644 005212
000646 022712 000002
000652 001006
000654 012737 000000 000000
000662 005737 000000
000666 001404
000670 012742 000005
000674 005242
000676 000000
000700 005212
000702 022712 000003
000706 001007
000710 012737 125252 000000
000716 022737 125252 000000
000724 001404
000726 012742 000006
000732 005242
000734 000000


```

630
631
632
633
634 000736 005212
635 000740 022712 000004
636 000744 001007
637 000746 012737 052525 000000
638
639 000754 022737 052525 000000
640 000762 001404
641
642
643
644
645 000764 012742 000007
646 000770 005242
647 000772 000000
648
649
650
651
652
653 000774 005212
654 000776 022712 000005
655 001002 001007
656 001004 012737 177777 000000
657 001012 022737 177777 000000
658 001020 001404
659
660
661
662
663 001022 012742 000010
664 001026 005242
665 001030 000000
666

```

```

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

```

```

TST4: INC (R2) ;UPDATE TEST NUMBER
      CMP #4,(R2) ;SEQUENCE ERROR?
      BNE TST5-10 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #052525,a#0 ;MOVE ALTERNATING ZEROES AND ONES
                        ;THRU DATA PATH
      CMP #052525,a#0 ;SUCCESSFUL?
      BEQ TST5
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
      MOV #7,-(R2) ;MOVE TO MAILBOX # ***** 7 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;DATA INCORRECT
           ;OR SEQUENCE ERROR

```

```

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

```

```

TST5: INC (R2) ;UPDATE TEST NUMBER
      CMP #5,(R2) ;SEQUENCE ERROR?
      BNE TST6-10 ;BR TO ERROR HALT ON SEQ ERROR
      MOV #177777,a#0 ;MOVE ONES THRU DATA PATH
      CMP #177777,a#0 ;SUCCESSFUL
      BEQ TST6
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
      MOV #10,-(R2) ;MOVE TO MAILBOX # ***** 10 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;DATA INCORRECT
           ;OR SEQUENCE ERROR

```


667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722

:SBTTL B-REGISTER TEST

: THE B-REGISTER SHIFTING LOGIC TESTS ARE USED TO TEST THAT THE
: B-REGISTER CAN HOLD VARIOUS DATA PATTERNS AND THAT THE ASSOCIATED
: LOGIC SUPPORTS THE SHIFTING FUNCTIONS WITHIN THE B-REGISTER AND C-BIT.
: A ONE IS SHIFTED THROUGH EVERY BIT IN THE B-REGISTER AND C-BIT IN
: BOTH DIRECTIONS.

: THE B-REGISTER ITSELF IS TESTED IN ITS ABILITY AS A BUFFER AND AS
: A SHIFT REGISTER. DATA IS ALSO PASSED THROUGH THE DATA PATH AND ALU.

: IF THESE TESTS FAIL, EXAMINE THE TARGET LOCATION (LOC. 0) TO SEE
: WHICH BITS OF THE B-REGISTER MAY BE FAILING. IF THIS PROVIDES
: INCONCLUSIVE DATA TRY TO CHECK THE MODE 3 IR DECODE BY RUNNING JUST
: THE MICROCODE AND IR DECODE TESTS FOR THE PARTICULAR INSTRUCTIONS.

:TEST 6 SHIFT BIT 0 TO BIT 1

TST6: INC (R2) ;UPDATE TEST NUMBER
CMP #6,(R2) ;SEQUENCE ERROR?
BNE TST7-10 ;BR TO ERROR HALT ON SEQ ERROR
CLC ;CLEAR CARRY BIT
MOV #1,2#0 ;LOAD A 1
ROL 2#0 ;SHIFT LEFT
CMP #2,2#0 ;SUCCESSFUL
BEQ TST7

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 766 <====
: MOVE TO MAILBOX # ***** 11 *****
: SET MSGTYP TO FATAL ERROR
: BIT 1 NOT SET
: OR SEQUENCE ERROR

:TEST 7 SHIFT CARRY INTO BIT 0

TST7: INC (R2) ;UPDATE TEST NUMBER
CMP #7,(R2) ;SEQUENCE ERROR?
BNE TST10-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #0,2#0 ;CLEAR LOCATION
SEC ;SET CARRY
ROL 2#0 ;ROTATE CARRY BIT TO BIT 0
BCC TST10

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 771 <====
: MOVE TO MAILBOX # ***** 12 *****
: SET MSGTYP TO FATAL ERROR
: CARRY CLEAR
: OR SEQUENCE ERROR
: BIT 0 SET

CMP #1,2#0
BEQ TST10


```

723
724
725
726
727 001144 012742 000013      MOV    #13,-(R2)
728 001150 005242              INC    -(R2)
729 001152 000000              HALT
730
731
732
733
734
735 001154 005212
736 001156 022712 000010
737 001162 001014
738 001164 012737 000001 000000
739 001172 012700 177757
740 001176 000241
741 001200 005200
742 001202 001404
743 001204 006137 000000
744 001210 103373
745 001212 001404
746
747
748
749
750 001214
751 001214 012742 000014
752 001220 005242
753 001222 000000
754
755
756
757
758
759 001224 005212
760 001226 022712 000011
761 001232 001012
762 001234 012737 100000 000000
763 001242 000241
764 001244 006037 000000 000000
765 001250 022737 040000 000000
766 001256 001404
767
768
769
770
771 001260 012742 000015
772 001254 005242
773 001266 000000
774
775
776
777
778

```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 761 <====
; MOVE TO MAILBOX # ***** 13 *****
; SET MSGTYP TO FATAL ERROR
; BIT 0 NOT SET
; OR SEQUENCE ERROR

```

```

;*****
;TEST 10 LEFT SHIFT FROM BIT 0 TO C-BIT
;*****

```

```

TST10: INC    (R2)           ;UPDATE TEST NUMBER
        CMP    #10,(R2)    ;SEQUENCE ERROR?
        BNE   TST11-10    ;BR TO ERROR HALT ON SEQ ERROR
        MOV   #1,2#0      ;SET BIT 0
        MOV   #-21,R0     ;SET BIT COUNTER
        CLC                ;CLEAR C-BIT
SHL:   INC    R0           ;INCREMENT BIT COUNTER
        BEQ   SHLE        ;BR TO ERROR HALT IF BIT IS LOST
        ROL   2#0         ;SHIFT LEFT ONE POSITION
        BCC   SHL         ;BRANCH IF C-BIT NOT SET
        BEQ   TST11

```

```

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

```

```

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

```

```

;*****
;TEST 11 SHIFT BIT 15 TO BIT 14
;*****

```

```

TST11: INC    (R2)           ;UPDATE TEST NUMBER
        CMP    #11,(R2)    ;SEQUENCE ERROR?
        BNE   TST12-10    ;BR TO ERROR HALT ON SEQ ERROR
        MOV   #100000,2#0 ;SET BIT 15
        CLC                ;CLEAR CARRY
        ROR   2#0         ;SHIFT BIT 15 TO BIT 14
        CMP   #40000,2#0   ;SUCCESSFUL
        BEQ   TST12

```

```

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

```

```

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

```

```

;*****
;TEST 12 RIGHT SHIFT FROM BIT 15 TO C-BIT
;*****

```


799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854

;SBTTL SCRATCH PAD TESTS

THE SCRATCH PAD TESTS ARE USED TO VERIFY THAT VARIOUS DATA PATTERNS CAN BE SUCCESSFULLY HELD IN THE SCRATCH PAD CIRCUITRY. MOVE AND COMPARE INSTRUCTIONS ARE USED TO TEST THAT RO CAN HOLD VARIOUS DATA PATTERNS. EACH DATA PATTERN IS MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR SCOPING. THE SUCCESSFUL COMPLETION OF THESE TESTS SHOULD VERIFY THE CIRCUITRY EXTERNAL TO THE SCRATCH PAD ITSELF.

THE REMAINDER OF THE GENERAL REGISTERS ARE TESTED BY MOVING A BIT INTO BIT 0 OF THE REGISTER AND SHIFTING IT LEFT ONE BIT AT A TIME INTO THE CARRY BIT. THE RESULT IS THEN CHECKED TO INSURE THAT NO BITS WERE PICKED. THE PROCEDURE IS THEN REPEATED UNDER OPPOSITE CONDITIONS. THE GENERAL REGISTER AND THE CARRY BIT ARE SET TO ALL ONES, AND A ZERO IS SHIFTED LEFT FROM BIT 0 INTO THE CARRY BIT. THE RESULT IS THEN CHECKED TO INSURE THAT NO ZEROES WERE PICKED.

AT THIS POINT ALL OF THE GENERAL REGISTERS HAVE BEEN EXERCISED AS WELL AS REGISTER 11. REGISTERS 10 AND 12 HAVE BEEN ACCESSED BY THE INSTRUCTIONS. REGISTERS 13,14,AND 17 WILL BE TESTED LATER IN THE MICROCODE TESTS.

IF THE PATTERN TESTS WITH REGISTER 0 FAIL CHECK THE RESULTANT DATA FOR A CLUE TO A FAULT IN THE EXTERNAL CIRCUITRY. IF THE PATTERN TESTS WITH RO ARE SUCCESSFUL BUT THE TESTS WITH THE OTHER REGISTERS FAIL, SUSPECT THE REGISTER SELECT LINES AND THEN THE SCRATCH PAD ITSELF.

;TEST 13 TEST IF RO CAN HOLD ALL ZEROES

TST13: INC (R2) ;UPDATE TEST NUMBER
CMP #13,(R2) ;SEQUENCE ERROR?
BNE TST14-10 ;BR TO ERROR HALT ON SEQ ERROR

MOV #0,RO ;MOVE ZEROES TO RO
TST RO ;SUCCESSFUL?
BEQ TST14

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

MOV #17,-(R2) ;MOVE TO MAILBOX # ***** 17 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RO NOT 0
; OR SEQUENCE ERROR

;TEST 14 TEST IF RO CAN HOLD ONES AND ZEROES

TST14: INC (R2) ;UPDATE TEST NUMBER
CMP #14,(R2) ;SEQUENCE ERROR?
BNE TST15-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #125252,RO ;MOVE ALTERNATING ONES AND ZEROES TO RO
CMP RO,#125252 ;SUCCESSFUL?
BEQ TST15


```

855                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
856                                     ; CONDITIONAL BRANCH INST. AND <====
857                                     ; REPLACE THE MOVE INSTRUCTION <====
858                                     ; WHICH FOLLOWS W/ 773 <====
859 001412 012742 000020             MOV #20,-(R2) ; MOVE TO MAILBOX # ***** 20 *****
860 001416 005242                   INC -(R2) ; SET MSGTYP TO FATAL ERROR
861 001420 000000                   HALT ; RD NOT 125252
862                                     ; OR SEQUENCE ERROR

```

```

;*****
;TEST 15 TEST IF RD CAN HOLD ZEROES AND ONES
;*****

```

```

866 001422 005212 000015             TST15: INC (R2) ; UPDATE TEST NUMBER
867 001424 022712                   CMP #15,(R2) ; SEQUENCE ERROR?
868 001430 001005                   BNE TST16-10 ; BR TO ERROR HALT ON SEQ ERROR
869 001432 012700 052525             MOV #052525,RD ; MOVE ALTERNATING ZEROES AND ONES TO RD
870 001436 020027 052525             CMP RD,#052525 ; SUCCESSFUL?
871 001442 001404                   BEQ TST16

```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 773 <====
877 001444 012742 000021             MOV #21,-(R2) ; MOVE TO MAILBOX # ***** 21 *****
878 001450 005242                   INC -(R2) ; SET MSGTYP TO FATAL ERROR
879 001452 000000                   HALT ; RD NOT 52525
880                                     ; OR SEQUENCE ERROR

```

```

;*****
;TEST 16 TEST IF RD CAN HOLD ALL ONES
;*****

```

```

885 001454 005212 000016             TST16: INC (R2) ; UPDATE TEST NUMBER
886 001456 022712                   CMP #16,(R2) ; SEQUENCE ERROR?
887 001462 001005                   BNE TST17-10 ; BR TO ERROR HALT ON SEQ ERROR
888 001464 012700 177777             MOV #177777,RD ; MOVE ALL ONES TO RD
889 001470 020027 177777             CMP RD,#177777 ; SUCCESSFUL?
890 001474 001404                   BEQ TST17

```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 773 <====
895 001476 012742 000022             MOV #22,-(R2) ; MOVE TO MAILBOX # ***** 22 *****
896 001502 005242                   INC -(R2) ; SET MSGTYP TO FATAL ERROR
897 001504 000000                   HALT ; RD NOT 177777
898                                     ; OR SEQUENCE ERROR

```

```

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

```

```

903 001506 005212 000017             TST17: INC (R2) ; UPDATE TEST NUMBER
904 001510 022712                   CMP #17,(R2) ; SEQUENCE ERROR?
905 001514 001012                   BNE TST20-10 ; BR TO ERROR HALT ON SEQ ERROR
906 001516 012701 000001             MOV #1,R1 ; SET BIT 0
907 001522 012700 177757             MOV #-21,R0 ; SET BIT COUNTER
908 001526 000241                   CLC ; CLEAR C-BIT
909 001530 005200                   REG1: INC RD ; INCREMENT BIT COUNTER
910 001532 001403                   BEQ REG1E ; BR TO ERROR HALT IF BIT IS LOST

```



```

911 001534 006101          ROL      R1          ;ROTATE 1 POSITION
912 001536 103374          BCC     REG1        ;ALL DONE
913 001540 001404          BEQ     TST20
914                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
915                                     ;          CONDITIONAL BRANCH INST. AND <====
916                                     ;          REPLACE THE MOVE INSTRUCTION <====
917                                     ;          WHICH FOLLOWS W/ 766 <====
918 001542                                REGIE:
919 001542 012742 000023    MOV     #23,-(R2)    ;MOVE TO MAILBOX # ***** 23 *****
920 001546 005242          INC     -(R2)      ;SET MSGTYP TO FATAL ERROR
921 001550 000000          HALT                    ;FAILURE WITH R1
922                                     ; OR SEQUENCE ERROR

```

```

923
924 ;*****
925 ;TEST 20 TEST IF R1 CAN HOLD A ZERO IN ALL BITS
926 ;*****

```

```

927 001552 005212          TST20: INC     (R2)      ;UPDATE TEST NUMBER
928 001554 022712 000020    CMP     #20,(R2)    ;SEQUENCE ERROR?
929 001560 001014          BNE     TST21-10   ;BR TO ERROR HALT ON SEQ ERROR
930 001562 012701 177776    MOV     #-2,R1     ;SET ALL ONES IN R1 EXCEPT FOR BIT 0
931 001566 012700 177757    MOV     #-21,R0    ;SET BIT COUNTER
932 001572 000261          SEC                    ;SET C-BIT
933 001574 005200          REG1A: INC     R0      ;INCREMENT COUNTER
934 001576 001405          BEQ     RIERR      ;BR TO ERROR HALT IF COUNTER=0
935 001600 006101          ROL     R1          ;ROTATE 1 POSITION
936 001602 103774          BCS    REG1A       ;CONTINUE UNTIL C-BIT IS CLEAR
937 001604 022701 177777    CMP     #-1,R1     ;CHECK DATA IN R1
938 001610 001404          BEQ     TST21

```

```

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

```

```

943 001612                                RIERR:
944 001612 012742 000024    MOV     #24,-(R2)    ;MOVE TO MAILBOX # ***** 24 *****
945 001616 005242          INC     -(R2)      ;SET MSGTYP TO FATAL ERROR
946 001620 000000          HALT                    ;FAILURE WITH R1
947                                     ; OR SEQUENCE ERROR

```

```

948 ;*****
949 ;TEST 21 TEST IF R2 CAN HOLD A ONE IN ALL BITS
950 ;*****

```

```

951 001622 005212          TST21: INC     (R2)      ;UPDATE TEST NUMBER
952 001624 022712 000021    CMP     #21,(R2)    ;SEQUENCE ERROR?
953 001630 001012          BNE     REG2A-14   ;BR TO ERROR HALT ON SEQ ERROR
954 001632 012702 000001    MOV     #1,R2      ;SET BIT 0
955 001636 012700 177757    MOV     #-21,R0    ;SET BIT COUNTER
956 001642 000241          CLC                    ;CLEAR C-BIT
957 001644 005200          REG2: INC     R0      ;INCREMENT BIT COUNTER
958 001646 001403          BEQ     REG2A-14   ;BR TO ERROR HALT IF BIT IS LOST
959 001650 006102          ROL     R2          ;ROTATE 1 POSITION
960 001652 103374          BCC     REG2        ;ALL DONE
961 001654 001406          BEQ     REG2A

```

```

962                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
963                                     ;          BRANCH INSTRUCTION AND <====
964                                     ;          REPLACE THE MOVE INSTRUCTION <====
965                                     ;          FOLLOWING W/ 771 <====

```

```

966 001656 012702 000304    MOV     #STESTN,R2  ;RESTORE POINTER

```



```

967 001662 012742 000025      MOV      #25, -(R2)      ;MOVE TO MAILBOX # ***** 25 *****
968 001666 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
969 001670 000000              HALT                    ;FAILURE WITH R2
970 001672 012702 000304      REG2A:  MOV      #STESTN,R2      ;RESTORE POINTER
971
972 ;*****
973 ;TEST 22      TEST IF R2 CAN HOLD A ZERO IN ALL BITS
974 ;*****
975 001676 005212              †ST22:  INC      (R2)          ;UPDATE TEST NUMBER
976 001700 022712 000022      CMP      #22, (R2)      ;SEQUENCE ERROR?
977 001704 001020              BNE     TST23-10        ;BR TO ERROR HALT ON SEQ ERROR
978 001706 012702 177776      MOV      #-2, R2        ;SET ALL ONES IN R2 EXCEPT FOR BIT 0
979 001712 012700 177757      MOV      #-21, R0       ;SET BIT COUNTER
980 001716 000261              SEC                    ;SET C-BIT
981 001720 005200              REG2B:  INC      R0        ;INCREMENT BIT COUNTER
982 001722 001407              BEQ     R2ERR           ;BR TO ERROR HALT IF COUNTER=0
983 001724 006102              ROL     R2              ;ROTATE 1 POSITION
984 001726 103774              BCS     REG2B           ;CONTINUE UNTIL C-BIT IS CLEAR
985 001730 022702 177777      CMP      #-1, R2        ;CHECK DATA IN R2
986 001734 001406              BEQ     REG2C           ;
987 001736 012702 000304      MOV      #STESTN,R2     ;RESTORE POINTER
988 001742
989 001742 012742 000026      R2ERR:  MOV      #26, -(R2)    ;MOVE TO MAILBOX # ***** 26 *****
990 001746 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
991 001750 000000              HALT                    ;FAILURE WITH R2
992 001752 012702 000304      REG2C:  MOV      #STESTN,R2     ;RESTORE POINTER
993
994 ;*****
995 ;TEST 23      TEST IF R3 CAN HOLD A ONE IN ALL BITS
996 ;*****
997 001756 005212              †ST23:  INC      (R2)          ;UPDATE TEST NUMBER
998 001760 022712 000023      CMP      #23, (R2)      ;SEQUENCE ERROR?
999 001764 001012              BNE     TST24-10        ;BR TO ERROR HALT ON SEQ ERROR
1000 001766 012703 000001      MOV      #1, R3         ;SET BIT 0
1001 001772 012700 177757      MOV      #-21, R0       ;SET BIT COUNTER
1002 001776 000241              CLC                    ;CLEAR C-BIT
1003 002000 005200              REG3:   INC      R0        ;INCREMENT BIT COUNTER
1004 002002 001403              BEQ     REG3E           ;BR TO ERROR HALT IF BIT IS LOST
1005 002004 006103              ROL     R3              ;ROTATE 1 POSITION
1006 002006 103374              BCC     REG3            ;ALL DONE
1007 002010 001404              BEQ     TST24           ;
1008 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1009 ; CONDITIONAL BRANCH INST. AND <====
1010 ; REPLACE THE MOVE INSTRUCTION <====
1011 ; WHICH FOLLOWS W/ 766 <====
1012 002012
1013 002012 012742 000027      REG3E:  MOV      #27, -(R2)    ;MOVE TO MAILBOX # ***** 27 *****
1014 002016 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
1015 002020 000000              HALT                    ;FAILURE WITH R3
1016 ; OR SEQUENCE ERROR
1017
1018 ;*****
1019 ;TEST 24      TEST IF R3 CAN HOLD A ZERO IN ALL BITS
1020 ;*****
1021 002022 005212              †ST24:  INC      (R2)          ;UPDATE TEST NUMBER
1022 002024 022712 000024      CMP      #24, (R2)      ;SEQUENCE ERROR?

```



```

1023 002030 001014          BNE     TST25-10      ;BR TO ERROR HALT ON SEQ ERROR
1024 002032 012703 177776  MOV     #-2,R3        ;SET ALL ONES IN R3 EXCEPT FOR BIT 0
1025 002036 012700 177757  MOV     #-21,R0       ;SET BIT COUNTER
1026 002042 000261          SEC          ;SET C-BIT
1027 002044 005200          REG3A: INC     R0        ;INCREMENT BIT COUNTER
1028 002046 001405          BEQ     R3ERR        ;BR TO ERROR HALT IF COUNTER=0
1029 002050 006103          ROL     R3           ;ROTATE 1 POSITION
1030 002052 103774          BCS     REG3A        ;CONTINUE UNTIL C-BIT IS CLEAR
1031 002054 022703 177777  CMP     #-1,R3       ;CHECK DATA
1032 002060 001404          BEQ     TST25
1033
1034
1035
1036
1037 002062          REG3A:
1038 002062 012742 000030  MOV     #30,-(R2)    ;MOVE TO MAILBOX # ***** 30 *****
1039 002066 005242          INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
1040 002070 000000          HALT              ;FAILURE WITH R3
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074
1075
1076
1077
1078

```

```

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

```

```

;*****
;TEST 25 TEST IF R4 CAN HOLD A ONE IN ALL BITS
;*****
TST25: INC     (R2)      ;UPDATE TEST NUMBER
        CMP     #25,(R2) ;SEQUENCE ERROR?
        BNE     TST26-10 ;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     TST26

```

```

;*****
;TEST 26 TEST IF R4 CAN HOLD A ZERO IN ALL BITS
;*****
TST26: INC     (R2)      ;UPDATE TEST NUMBER
        CMP     #26,(R2) ;SEQUENCE ERROR?
        BNE     TST27-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #-2,R4   ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
        MOV     #-21,R0 ;SET BIT COUNTER
        SEC          ;SET C-BIT
        REG4A: INC     R0 ;INCREMENT BIT COUNTER
        BEQ     R4ERR   ;BR TO ERROR HALT IF COUNTER=0
        ROL     R4      ;ROTATE 1 POSITION

```

```

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

```

```

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
;*****
TST26: INC     (R2)      ;UPDATE TEST NUMBER
        CMP     #26,(R2) ;SEQUENCE ERROR?
        BNE     TST27-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #-2,R4   ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
        MOV     #-21,R0 ;SET BIT COUNTER
        SEC          ;SET C-BIT
        REG4A: INC     R0 ;INCREMENT BIT COUNTER
        BEQ     R4ERR   ;BR TO ERROR HALT IF COUNTER=0
        ROL     R4      ;ROTATE 1 POSITION

```

```

;*****
;TEST 26 TEST IF R4 CAN HOLD A ZERO IN ALL BITS
;*****
TST26: INC     (R2)      ;UPDATE TEST NUMBER
        CMP     #26,(R2) ;SEQUENCE ERROR?
        BNE     TST27-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #-2,R4   ;SET ALL ONES IN R4 EXCEPT FOR BIT 0
        MOV     #-21,R0 ;SET BIT COUNTER
        SEC          ;SET C-BIT
        REG4A: INC     R0 ;INCREMENT BIT COUNTER
        BEQ     R4ERR   ;BR TO ERROR HALT IF COUNTER=0
        ROL     R4      ;ROTATE 1 POSITION

```



```

1079 002166 103774          BCS  REG4A          ;CONTINUE UNTIL C-BIT IS CLEAR
1080 002170 022704 177777    CMP  #-1,R4        ;CHECK DATA
1081 002174 001404          BEQ  TST27         ;
1082                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1083                                     ;          CONDITIONAL BRANCH INST. AND <====
1084                                     ;          REPLACE THE MOVE INSTRUCTION <====
1085                                     ;          WHICH FOLLOWS W/ 764 <====
1086 002176          R4ERR:  MOV  #32, -(R2)      ;MOVE TO MAILBOX # ***** 32 *****
1087 002176 012742 000032    INC  -(R2)        ;SET MSGTYP TO FATAL ERROR
1088 002202 005242          HALT                    ;FAILURE WITH R4
1089 002204 000000          ; OR SEQUENCE ERROR
1090
1091
1092
1093
1094
1095

```

```

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

```

```

1096 002206 005212          TST27: INC  (R2)          ;UPDATE TEST NUMBER
1097 002210 022712 000027    CMP  #27,(R2)      ;SEQUENCE ERROR?
1098 002214 001012          BNE  TST30-10     ;BR TO ERROR HALT ON SEQ ERROR
1099 002216 012705 000001    MOV  #1,R5        ;SET BIT 0
1100 002222 012700 177757    MOV  #-21,R0      ;SET BIT COUNTER
1101 002226 000241          CLC                    ;CLEAR C-BIT
1102 002230 005200          REG5: INC  R0        ;INCREMENT BIT COUNTER
1103 002232 001403          BEQ  REG5E        ;BR TO ERROR HALT IF BIT IS LOST
1104 002234 006105          ROL  R5           ;ROTATE 1 POSITION
1105 002236 103374          BCC  REG5         ;ALL DONE
1106 002240 001404          BEQ  TST30

```

```

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

```

```

1111 002242          REG5E: MOV  #33, -(R2)      ;MOVE TO MAILBOX # ***** 33 *****
1112 002242 012742 000033    INC  -(R2)        ;SET MSGTYP TO FATAL ERROR
1113 002246 005242          HALT                    ;FAILURE WITH R5
1114 002250 000000          ; OR SEQUENCE ERROR
1115
1116

```

```

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

```

```

1120 002252 005212          TST30: INC  (R2)          ;UPDATE TEST NUMBER
1121 002254 022712 000030    CMP  #30,(R2)      ;SEQUENCE ERROR?
1122 002260 001014          BNE  TST31-10     ;BR TO ERROR HALT ON SEQ ERROR
1123 002262 012705 177776    MOV  #-2,R5       ;SET ALL ONES IN R5 EXCEPT FOR BIT 0
1124 002266 012700 177757    MOV  #-21,R0      ;SET BIT COUNTER
1125 002272 000261          SEC                    ;SET C-BIT
1126 002274 005200          REG5A: INC  R0        ;INCREMENT BIT COUNTER
1127 002276 001405          BEQ  R5ERR        ;BR TO ERROR HALT IF COUNTER=0
1128 002300 006105          ROL  R5           ;ROTATE 1 POSITION
1129 002302 103774          BCS  REG5A        ;CONTINUE UNTIL C-BIT IS C;EAR
1130 002304 022705 177777    CMP  #-1,R5       ;CHECK DATA
1131 002310 001404          BEQ  TST31

```

```

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

```

```

1132
1133
1134

```



```

1135
1136 002312
1137 002312 012742 000034
1138 002316 005242
1139 002320 000000
1140
1141
1142
1143
1144
1145 002322 005212
1146 002324 022712 000031
1147 002330 001012
1148 002332 012706 000001
1149 002336 012700 177757
1150 002342 000241
1151 002344 005200
1152 002346 001403
1153 002350 006106
1154 002352 103374
1155 002354 001404
1156
1157
1158
1159
1160 002356
1161 002356 012742 000035
1162 002362 005242
1163 002364 000000
1164
1165
1166
1167
1168
1169 002366 005212
1170 002370 022712 000032
1171 002374 001014
1172 002376 012706 177776
1173 002402 012700 177757
1174 002406 000261
1175 002410 005200
1176 002412 001405
1177 002414 006106
1178 002416 103774
1179 002420 022706 177777
1180 002424 001404
1181
1182
1183
1184
1185 002426
1186 002426 012742 000036
1187 002432 005242
1188 002434 000000
1189

```

```

; WHICH FOLLOWS W/ 764 <====
RSERR: MOV #34, -(R2) ; MOVE TO MAILBOX # ***** 34 *****
        INC -(R2) ; SET MSGTYP TO FATAL ERROR
        HALT ; FAILURE WITH R5
          ; OR SEQUENCE ERROR

;*****
;TEST 31 TEST IF R6 CAN HOLD A ONE IN ALL BITS
;*****
TST31: INC (R2) ; UPDATE TEST NUMBER
        CMP #31, (R2) ; SEQUENCE ERROR?
        BNE TST32-10 ; BR TO ERROR HALT ON SEQ ERROR
        MOV #1, R6 ; SET BIT 0
        MOV #-21, R0 ; SET BIT COUNTER
        CLC ; CLEAR C-BIT
REG6: INC R0 ; INCREMENT BIT COUNTER
        BEQ REG6E ; BR TO ERROR HALT IF BIT IS LOST
        ROL R6 ; ROTATE 1 POSITION
        BCC REG6 ; ALL DONE
        BEQ TST32

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

REG6E: MOV #35, -(R2) ; MOVE TO MAILBOX # ***** 35 *****
        INC -(R2) ; SET MSGTYP TO FATAL ERROR
        HALT ; FAILURE WITH R6
          ; OR SEQUENCE ERROR

;*****
;TEST 32 TEST IF R6 CAN HOLD A ZERO IN ALL BITS
;*****
TST32: INC (R2) ; UPDATE TEST NUMBER
        CMP #32, (R2) ; SEQUENCE ERROR?
        BNE TST33-10 ; BR TO ERROR HALT ON SEQ ERROR
        MOV #-2, R6 ; SET ALL ONES IN R6 EXCEPT FOR BIT 0
        MOV #-21, R0 ; SET BIT COUNTER
        SEC ; SET C-BIT
REG6A: INC R0 ; INCREMENT BIT COUNT
        BEQ R6ERR ; BR TO ERROR HALT IF COUNTER=0
        ROL R6 ; ROTATE 1 POSITION
        BCS REG6A ; CONTINUE UNTIL C-BIT IS CLEAR
        CMP #-1, R6 ; CHECK DATA
        BEQ TST33

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

R6ERR: MOV #36, -(R2) ; MOVE TO MAILBOX # ***** 36 *****
        INC -(R2) ; SET MSGTYP TO FATAL ERROR
        HALT ; FAILURE WITH R6
          ; OR SEQUENCE ERROR

```


1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228
1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241

002436 005212
002440 022712 000033
002444 001010
002446 012706 000500
002452 012737 000000 177776
002460 005737 177776
002464 001404

002466 012742 000037
002472 005242
002474 000000

002476 005212
002500 022712 000034
002504 001007
002506 012737 000252 177776
002514 023727 177776 000252
002522 001404

002524 012742 000040
002530 005242
002532 000000

```
*****  
:SBTTL PSW TESTS  
: THE PSW TESTS ARE USED TO VERIFY THAT VARIOUS DATA  
: PATTERNS CAN BE SUCCESSFULLY HELD IN THE PSW AND THAT THE  
: PSW ADDRESSING LOGIC IS FUNCTIONING. MOVE AND COMPARE INSTRUCTIONS  
: ARE USED TO TEST THAT THE PSW CAN HOLD VARIOUS DATA PATTERNS.  
: EACH DATA PATTERN IS MOVED AND TESTED IN A SMALL LOOP CONVENIENT FOR  
: SCOPING.  
: THE PSW REGISTER ITSELF IS TESTED AS WELL AS THE ADDRESS  
: SELECT CIRCUITRY. THE AMUX INPUTS TO THE PSW MUX ARE TESTED. THE  
: CC INPUTS ARE TESTED LATER IN THE MICROCODE TESTS. SETTING OF  
: THE T-BIT BY THE TEST PATTERNS IS PURPOSELY AVOIDED; TESTING OF THE  
: T-BIT TRAP CIRCUITRY IS LEFT FOR THE TRAP TEST.  
*****  
:TEST 33 TEST IF PSW WILL HOLD ZEROES  
*****  
TST33: INC (R2) ;UPDATE TEST NUMBER  
CMP #33,(R2) ;SEQUENCE ERROR?  
BNE TST34-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #STBOT,R6  
MOV #0,@#PS ;SET PSW TO ZERO  
TST @#PS ;SUCCESSFUL  
BEQ TST34  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
: CONDITIONAL BRANCH INST. AND <=====  
: REPLACE THE MOVE INSTRUCTION <=====  
: WHICH FOLLOWS W/ 770 <=====  
MOV #37,-(R2) ;MOVE TO MAILBOX # ***** 37 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;PSW NOT 0  
: OR SEQUENCE ERROR  
*****  
:TEST 34 TEST IF PSW WILL HOLD ONES AND ZEROES  
*****  
TST34: INC (R2) ;UPDATE TEST NUMBER  
CMP #34,(R2) ;SEQUENCE ERROR?  
BNE TST35-10 ;BR TO ERROR HALT ON SEQ ERROR  
MOV #252,@#PS ;MOVE ALT. ONES AND ZEROES TO PSW  
CMP @#PS,#252 ;SUCCESSFUL?  
BEQ TST35  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <=====  
: CONDITIONAL BRANCH INST. AND <=====  
: REPLACE THE MOVE INSTRUCTION <=====  
: WHICH FOLLOWS W/ 771 <=====  
MOV #40,-(R2) ;MOVE TO MAILBOX # ***** 40 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;PSW NOT 252  
: OR SEQUENCE ERROR
```



```

1243
1244
1245
1246 002534 005212
1247 002536 022712 000035
1248 002542 001007
1249 002544 012737 000105 177776
1250 002552 023727 177776 000105
1251 002560 001404
1252
1253
1254
1255
1256 002562 012742 000041
1257 002566 005242
1258 002570 000000
1259
1260
1261
1262
1263
1264 002572 005212
1265 002574 022712 000036
1266 002600 001007
1267 002602 012737 000357 177776
1268 002610 023727 177776 000357
1269 002616 001404
1270
1271
1272
1273
1274 002620 012742 000042
1275 002624 005242
1276 002626 000000
1277

```

```

:*****
:TEST 35      TEST IF PSW (EXCEPT T-BIT) WILL HOLD ZEROES AND ONES
:*****
TST35:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #35,(R2)     ;SEQUENCE ERROR?
        BNE     TST36-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #105,2#PS    ;MOVE ALT. ONES AND ZEROES TO PSW
        CMP     2#PS,#105    ;SUCCESSFUL?
        BEQ     TST36
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 771 <====
        MOV     #41,-(R2)    ;MOVE TO MAILBOX # ***** 41 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT                    ;PSW NOT 105
; OR SEQUENCE ERROR

:*****
:TEST 36      TEST IF PSW (EXCEPT T-BIT) WILL HOLD ALL ONES
:*****
TST36:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #36,(R2)     ;SEQUENCE ERROR?
        BNE     TST37-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #357,2#PS    ;MOVE ONES TO PSW
        CMP     2#PS,#357    ;SUCCESSFUL
        BEQ     TST37
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;          CONDITIONAL BRANCH INST. AND <====
;          REPLACE THE MOVE INSTRUCTION <====
;          WHICH FOLLOWS W/ 771 <====
        MOV     #42,-(R2)    ;MOVE TO MAILBOX # ***** 42 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT                    ;PSW NOT 357
; OR SEQUENCE ERROR

```


.SBTTL CONDITION CODE TEST

1278
1279
1280
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
1320
1321
1322
1323
1324
1325

002630 005212
002632 022712 000037
002636 001014

002640 000257
002642 000264
002644 001001
002646 001404

002650
002650 012742 000043
002654 005242
002656 000000

002660 000277
002662 000244
002664 001401
002666 001004

002670
002670 012742 000044
002674 005242
002676 000000

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

THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY
USED IN THE TEST ARE VERIFIED HERE.

TEST 37 TEST BRANCHES AROUND Z-BIT

TST37: INC (R2) ;UPDATE TEST NUMBER
CMP #37,(R2) ;SEQUENCE ERROR?
BNE TST40-10 ;BR TO ERROR HALT ON SEQ ERROR
;FIRST WITH Z-BIT ON
CCC ;CC=0100: JUST Z-BIT
SEZ
BNE BRZ1 ;CHECK OPPOSITE CONDITION
BEQ BRZ2

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

BRZ1: MOV #43,-(R2) ;MOVE TO MAILBOX # ***** 43 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ Z=1
;CHECK WITH Z-BIT OFF

BRZ2: SCC ;CC=1011: ALL BUT Z-BIT
CLZ
BEQ BRZ3
BNE TST40

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

BRZ3: MOV #44,-(R2) ;MOVE TO MAILBOX # ***** 44 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ Z=0
; OR SEQUENCE ERROR


```

1326
1327
1328
1329
1330
1331
1332
1333
1334
1335
1336
1337
1338
1339
1340
1341
1342
1343 002700 005212
1344 002702 022712 000040
1345 002706 001014
1346
1347 002710 000257
1348 002712 000270
1349 002714 100001
1350 002716 100404
1351
1352
1353
1354
1355 002720
1356 002720 012742 000045
1357 002724 005242
1358 002726 000000
1359
1360 002730 000277
1361 002732 000250
1362 002734 100401
1363 002736 100004
1364
1365
1366
1367
1368 002740
1369 002740 012742 000046
1370 002744 005242
1371 002746 000000
1372

```

```

*****
THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE N-BIT.
THE N-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
BMI AND BPL ARE TESTED FOR PROPER EXECUTION. THEN THE N-BIT IS
SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
AGAIN FOR PROPER OPERATION.
THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY
USED IN THE TEST ARE VERIFIED HERE.
*****
TEST 40 TEST BRANCHES AROUND N-BIT
*****
TST40: INC (R2) ;UPDATE TEST NUMBER
CMP #40,(R2) ;SEQUENCE ERROR?
BNE TST41-10 ;BR TO ERROR HALT ON SEQ ERROR
;FIRST WITH N-BIT ON
CCC ;CC=1000: JUST N-BIT
SEN
BPL BRN1 ;CHECK OPPOSITE CONDITION
BMI BRN2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 774 <====
BRN1: MOV #45,-(R2) ;MOVE TO MAILBOX # ***** 45 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ N=1
;CHECK WITH N-BIT OFF
BRN2: SCC ;CC=0111
CLN
BMI BRN3 ;CHECK OPPOSITE CONDITION
BPL TST41
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 764 <====
BRN3: MOV #46,-(R2) ;MOVE TO MAILBOX # ***** 46 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ N=0
; OR SEQUENCE ERROR

```


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

002750 005212
002752 022712 000041
002756 001014

002760 000257
002762 000262
002764 102001
002766 102404

002770
002770 012742 000047
002774 005242
002776 000000

003000 000277
003002 000242
003004 102401
003006 102004

003010
003010 012742 000050
003014 005242
003016 000000

```
*****
: 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.
: THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
: CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
: BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
: LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY
: USED IN THE TEST ARE VERIFIED HERE.
*****
: TEST 41 TEST BRANCHES AROUND V-BIT
*****
TST41: INC (R2) ;UPDATE TEST NUMBER
CMP #41,(R2) ;SEQUENCE ERROR?
BNE TST42-10 ;BR TO ERROR HALT ON SEQ ERROR
;FIRST WITH V-BIT ON
CCC ;CC=0010: JUST V-BIT
SEV
BVC BRV1 ;CHECK OPPOSITE CONDITION
BVS BRV2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 774 <====
BRV1: MOV #47,-(R2) ;MOVE TO MAILBOX # ***** 47 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ V=1
;CHECK WITH V-BIT OFF
BRV2: SCC ;CC=1101: ALL BVT V-BIT
CLV
BVS BRV3 ;CHECK OPPOSITE CONDITION
BVC TST42
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 764 <====
BRV3: MOV #50,-(R2) ;MOVE TO MAILBOX # ***** 50 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ V=0
; OR SEQUENCE ERROR
```

725

1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445
1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466

003020 005212
003022 022712 000042
003026 001014

003030 000257
003032 000261
003034 103001
003036 103404

003040
003040 012742 000051
003044 005242
003046 000000

003050 000277
003052 000241
003054 103401
003056 100404

003060
003060 012742 000052
003064 005242
003066 000000

```
*****
THIS TEST CHECKS THE CONDITIONAL BRANCHES INVOLVING THE C-BIT.
THE C-BIT IS SET WITH ALL OTHER CC BITS ZERO AND BOTH CONDITIONS
BCS AND BCC ARE TESTED FOR PROPER EXECUTION. THEN THE C-BIT IS
SET WITH ALL OTHER CC BITS CLEAR AND BOTH CONDITIONS ARE TESTED
AGAIN FOR PROPER OPERATION.
THIS TEST CHECKS THE OPERATION OF THE SET AND CLEAR CONDITION
CODE INSTRUCTIONS AND CHECKS THE CIRCUITRY EXTERNAL TO THE CONDITIONAL
BRANCH ROM. THE BRANCH MICROCODE FOR ALTERING THE PC AND FOR
LEAVING THE PC UNALTERED IS TESTED. ONLY THOSE ROM ADDRESSES SPECIFICALLY
USED IN THE TEST ARE VERIFIED HERE.
*****
TEST 42 TEST BRANCHES AROUND C-BIT
*****
TST42: INC (R2) ;UPDATE TEST NUMBER
CMP #42,(R2) ;SEQUENCE ERROR?
BNE TST43-10 ;BR TO ERROR HALT ON SEQ ERROR
;FIRST WITH C-BIT ON
CCC ;CC=0001: JUST C-BIT
SEC
BCC BRC1 ;CHECK OPPOSITE CONDITION
BCS BRC2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 774 <====
BRC1: MOV #51,-(R2) ;MOVE TO MAILBOX # ***** 51 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ C=1
;CHECK WITH C-BIT OFF
BRC2: SCC ;CC=1110
CLC
BCS BRC3 ;CHECK OPPOSITE CONDITION
BMI TST43
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 764 <====
BRC3: MOV #52,-(R2) ;MOVE TO MAILBOX # ***** 52 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;IMPROPER BR W/ C=0
; OR SEQUENCE ERROR
```


1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522

```

*****
SBTTL MICROCODE TESTS
*****
THE MICROCODE TESTS ARE USED TO VERIFY THE MICROPROGRAMM
FLOW. THE GOAL OF THESE TESTS IS TO EXERCISE EVERY POSSIBLE
BRANCH IN THE MICROPROGRAM FLOW.
THE TEST EXERCISES EVERY BRANCH IN THE MICROCODE BY
TESTING AT LEAST ONE INSTRUCTION FROM EVERY CLASS OF INSTRUCTION IN
ALL POSSIBLE MODES. FOR EXAMPLE, TO TEST THE SINGLE OPERAND INSTRUCTIONS,
AT LEAST ONE SINGLE OPERAND INSTRUCTION IS VERIFIED IN ALL UNIQUE
ADDRESSING MODES. BYTE MODES ARE ALSO TESTED. AS EACH NEW
MODE IS INTRODUCED THE SAME INSTRUCTION IS TRIED AND TESTED IN
A SMALL LOOP CONVENIENT FOR SCOPING. THE TEST IS SET UP USING
ONLY INSTRUCTIONS AND ADDRESSING MODES WHICH HAVE BEEN PREVIOUSLY
VERIFIED.
IF THESE TESTS FAIL, CHECK THE RESULTS FOR A CLUE TO THE
FAULT.
*****

```

```

*****
THE CLR INSTRUCTION IS USED TO INTRODUCE EACH ADDRESSING
MODE WITH THE SINGLE OPERAND INSTRUCTION. FOLLOWING THE SEQUENCE CHECK,
THE CLR INSTRUCTION IS EXECUTED AND A BRANCH TEST IS EXECUTED WHICH
CHECKS THAT THE Z-BIT WAS PROPERLY SET. THIS SMALL TEST IS SELF-SUFFICIENT
AND CAN BE SCOPED TO TROUBLE SHOOT ALL OF THE IR DECODE LOGIC AND
MICROCODE FOR SOP INSTRUCTIONS WITH MODE 0. FOLLOWING THIS TEST
SEVERAL OTHER SOP INSTRUCTIONS ARE INTRODUCED WITH MODE 0. THESE
INSTRUCTIONS MAINPULATE DATA AND SERVE TO CHECK THE DATA RESULTS
OF THE SOP INSTRUCTIONS IN THIS TEST. THE DATA IN THIS TEST IS
OPERATED ON BY EACH INSTRUCTION WITHOUT REINITIALIZING.
*****

```

```

*****
TEST 43 TEST MODE 0 USING SOP INST.
*****

```

```

TST43: INC (R2) ;UPDATE TEST NUMBER
CMP #43,(R2) ;SEQUENCE ERROR?
BNE TST44-10 ;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/ 776 <====
MOV #53,-(R2) ;MOVE TO MAILBOX # ***** 53 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR DID NOT SET Z-BIT
SOPOA: INC R0 ;TRY THE INCREMENT INST.
COM R0 ;TRY COMPLEMENT
INC R0
BMI SOPOB
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====

```

```

003070 005212
003072 022712 000043
003076 001020
003100 005000
003102 001404
003104 012742 000053
003110 005242
003112 000000
003114 005200
003116 005100
003120 005200
003122 100404

```



```

1523                                     :          CONDITIONAL BRANCH INST. AND <====
1524                                     :          REPLACE THE MOVE INSTRUCTION <====
1525                                     :          WHICH FOLLOWS W/ 766 <====
1526 003124 012742 000054             MOV  #54, -(R2)      ; MOVE TO MAILBOX # ***** 54 *****
1527 003130 005242                   INC  -(R2)        ; SET MSGTYP TO FATAL ERROR
1528 003132 000000                   HALT              ; NEGATE DID NOT SET N-BIT
1529 003134 005100                   COM  R0          ; TRY COMPLEMENT INST.
1530 003136 001404                   BEQ  TST44
1531                                     :
1532                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1533                                     :          CONDITIONAL BRANCH INST. AND <====
1534                                     :          REPLACE THE MOVE INSTRUCTION <====
1535                                     :          WHICH FOLLOWS W/ 760 <====
1535 003140 012742 000055             MOV  #55, -(R2)      ; MOVE TO MAILBOX # ***** 55 *****
1536 003144 005242                   INC  -(R2)        ; SET MSGTYP TO FATAL ERROR
1537 003146 000000                   HALT              ; CUMMULATIVE RESULT OF CLR, INC, NEG AND COM INSTS. FAILED
1538                                     :          OR SEQUENCE ERROR
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552

```

```

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

```

```

1551 : TEST 44 TEST REMAINDER OF SOP INSTS IN MODE 0
1552 : *****
1553 003150 005212 000044             TST44: INC  (R2)          ; UPDATE TEST NUMBER
1554 003152 022712                   CMP  #44, (R2)     ; SEQUENCE ERROR?
1555 003156 001021                   BNE  TST45-10     ; BR TO ERROR HALT ON SEQ ERROR
1556 003160 005000                   CLR  R0           ; INITIALIZE
1557 003162 005300                   DEC  R0           ; TRY DECREMENT INST.
1558 003164 100404                   BMI  SOPOC
1559                                     :
1560                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1561                                     :          CONDITIONAL BRANCH INST. AND <====
1562                                     :          REPLACE THE MOVE INSTRUCTION <====
1563                                     :          WHICH FOLLOWS W/ 775 <====
1563 003166 012742 000056             MOV  #56, -(R2)      ; MOVE TO MAILBOX # ***** 56 *****
1564 003172 005242                   INC  -(R2)        ; SET MSGTYP TO FATAL ERROR
1565 003174 000000                   HALT              ; N-BIT NOT SET ON DEC
1566 003176 000261                   SEC              ; INITIALIZE CARRY
1567 003200 005500                   ADC  R0           ; TRY ADD CARRY INST
1568 003202 001007                   BNE  SOPOD
1569 003204 000261                   SEC              ; INITIALIZE CARRY
1570 003206 005600                   SBC  R0           ; TRY SUBTRACT-CARRY INST
1571 003210 100004                   BPL  SOPOD
1572 003212 005100                   COM  R0
1573 003214 005200                   INC  R0
1574 003216 005300                   DEC  R0
1575 003220 001404                   BEQ  TST45
1576                                     :
1577                                     : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
1578                                     :          CONDITIONAL BRANCH INST. AND <====
1579                                     :          REPLACE THE MOVE INSTRUCTION <====

```


1579
1580
1581
1582
1583
1584

003222
003222 012742 000057
003226 005242
003230 000000

SOPD: MOV #57, -(R2)
 INC -(R2)
 HALT

; WHICH FOLLOWS W/ 757 <====
; MOVE TO MAILBOX # ***** 57 *****
; SET MSGTYP TO FATAL ERROR
; CUMMULATIVE RESULT OF ADC, SBC, COM, INC AND DEC INSTS. F
; OR SEQUENCE ERROR

15


```

1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595 003232 005212
1596 003234 022712 000045
1597 003240 001012
1598 003242 105000
1599 003244 001404
1600
1601
1602
1603
1604 003246 012742 000060
1605 003252 005242
1606 003254 000000
1607 003256 105100
1608 003260 100002
1609 003262 105200
1610 003264 001404
1611
1612
1613
1614
1615 003266
1616 003266 012742 000061
1617 003272 005242
1618 003274 000000
1619

```

```

;*****
;
; 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
;*****
TST45: INC (R2) ; UPDATE TEST NUMBER
        CMP #45,(R2) ; SEQUENCE ERROR?
        BNE TST46-10 ; BR TO ERROR HALT ON SEG 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/ 776 <====
        MOV #60,-(R2) ; MOVE TO MAILBOX # ***** 60 *****
        INC -(R2) ; SET MSGTYP TO FATAL ERROR
        HALT ; CLRB DID NOT SET Z-BIT
SOPBOA: COMB R0 ; TRY SETTING EVEN BYTE OF REGISTER
        BPL SOPBOB
        INCB R0 ; TRY INCREMENTING EVEN BYTE OF REGISTER>>
        BEQ TST46
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 766 <====
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

```

555
12.9

2

1620
1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658

003276 005212
003300 022712 000046
003304 001014
003306 005000
003310 005010
003312 001404

003314 012742 000062
003320 005242
003322 000000
003324 005310
003326 100003
003330 000261
003332 005510
003334 001404

003336 012742 000063
003342 005242
003344 000000

```
*****
: 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.
*****
TST46: INC (R2) ;UPDATE TEST NUMBER
      CMP #46,(R2) ;SEQUENCE ERROR?
      BNE TST47-10 ;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/ 775 <====
      MOV #62,-(R2) ;MOVE TO MAILBOX # ***** 62 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;CLR DID NOT SET Z-BIT
SOP1A: DEC (R0) ;TRY DECREMENT INST W/MODE 1
      BPL SOP1B
      SEC ;INITIALIZE CARRY
      ADC (R0) ;TRY ADD-CARRY W/MODE 1
      BEQ TST47
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 764 <====
SOP1B: MOV #63,-(R2) ;MOVE TO MAILBOX # ***** 63 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;TEST CUMMULATIVE RESULT OF ABOVE INST
; OR SEQUENCE ERROR
```



```

1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670 003346 005212
1671 003350 022712 000047
1672 003354 001020
1673 003356 005000
1674 003360 005010
1675 003362 005110
1676 003364 105010
1677 003366 001404
1678
1679
1680
1681
1682 003370 012742 000064
1683 003374 005242
1684 003376 000000
1685 003400 005210
1686 003402 100005
1687 003404 105110
1688 003406 105210
1689 003410 100002
1690 003412 105210
1691 003414 001404
1692
1693
1694
1695
1696 003416
1697 003416 012742 000065
1698 003422 005242
1699 003424 000000
1700
1701

```

```

;*****
;
;   THIS TEST VERIFIES THE BYTE INSTRUCTION MICROCODE FOR MODE 1
; SINGLE OPERAND INSTRUCTIONS.
;   THIS IS THE FIRST PLACE THE SIGN EXTEND LOGIC IS EXERCISED
; AND VERIFIED.
;*****
;TEST 47          TEST MODE 1 EVEN BYTE USING SOP INST
;*****
TST47:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #47,(R2)     ;SEQUENCE ERROR?
        BNE     TST50-10     ;BR TO ERROR HALT ON SEQ ERROR
        CLR     RO           ;INITIALIZE RO
        CLR     (RO)         ;INITIALIZE LOC. 0
        COM     (RO)
        CLRB    (RO)         ;TRY TO CLEAR BYTE 0
        BEQ     SOPB1A
;
;   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;   CONDITIONAL BRANCH INST. AND <====
;   REPLACE THE MOVE INSTRUCTION <====
;   WHICH FOLLOWS W/ 773 <====
;
;   MOVE TO MAILBOX # ***** 64 *****
;   SET MSGTYP TO FATAL ERROR
;   CLRB DID NOT SET Z-BIT
;   INCREMENT TO TEST WORD
;
;   COMPLEMENT:  ODD BYTE = 376
;   INC:  ODD BYTE = 377
;
;   INCREMENT ODD BYTE=0
;
;   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
;   CONDITIONAL BRANCH INST. AND <====
;   REPLACE THE MOVE INSTRUCTION <====
;   WHICH FOLLOWS W/ 760 <====
;
;   MOVE TO MAILBOX # ***** 65 *****
;   SET MSGTYP TO FATAL ERROR
;   CHECK CUMMULATIVE RESULT OF ABOVE INST
;   OR SEQUENCE ERROR

```

```

SOPB1A:  MOV      #64,-(R2)
         INC      -(R2)
         HALT
         INC      (RO)
         BPL     SOPB1B
         COMB    (RO)
         INCB   (RO)
         BPL     SOPB1B
         INCB   (RO)
         BEQ     TST50

```

```

SOPB1B:  MOV      #65,-(R2)
         INC      -(R2)
         HALT

```


1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735
1736
1737
1738
1739
1740
1741
1742
1743
1744
1745
1746
1747

003426 005212
003430 022712 000050
003434 001022
003436 005000
003440 005010
003442 005110
003444 005200
003446 105010
003450 001404

003452 012742 000066
003456 005242
003460 000000
003462 005300
003464 005210
003466 005200
003470 105110
003472 105210
003474 100002
003476 105210
003500 001404

003502 012742 000067
003506 005242
003510 000000

```
*****
: 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
*****
TST50: INC (R2) ;UPDATE TEST NUMBER
      CMP #50,(R2) ;SEQUENCE ERROR?
      BNE TST51-10 ;BR TO ERROR HALT ON SEQ ERROR
      CLR RO ;INITIALIZE RO
      CLR (RO) ;INITIALIZE LOC. 0
      COM (RO)
      INC RO ;RO=ODD BYTE
      CLRB (RO) ;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/ 772 <====
      MOV #66,-(R2) ;MOVE TO MAILBOX # ***** 66 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;CLRB DID NOT SET Z-BIT
SOPB1C: DEC RO ;RO=WORD ADDR.
      INC (RO) ;INCREMENT TO TEST WORD
      INC RO ;RO=ODD BYTE
      COMB (RO) ;TRY TO COMPLEMENT BYTE 1
      INCB (RO)
      BPL SOPB1D
      INCB (RO) ;TRY TO INCREMENT BYTE 1
      BEQ TST51
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 756 <====
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
```


1748
1749
1750
1751
1752
1753
1754
1755
1756
1757
1758
1759
1760
1761
1762
1763
1764
1765
1766
1767
1768
1769
1770
1771
1772
1773
1774
1775
1776
1777
1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795

003512 005212
003514 022712 000051
003520 001023
003522 005000
003524 10^F100
003526 00.00
003530 005010
003532 005110
003534 005020
003536 001404

003540 012742 000070
003544 005242
003546 000000
003550 005300
003552 005300
003554 005120
003556 100004
003560 005300
003562 005300
003564 005220
003566 001404

003570 012742 000071
003574 005242
003576 000000

```
*****  
: THIS TEST VERIFIES MODE 2 SINGLE-OPERAND INSTRUCTIONS. PREVIOUSLY  
: TESTED INSTRUCTIONS ARE USED TO SET A POINTER IN R0 TO LOC. 400.  
: LOC. 400 IS INITIALIZED TO -1 BEFORE A CLR MODE 2 IS EXECUTED.  
: THEN R0 IS DECREMENTED BY TWO TO AGAIN POINT TO 400 BEFORE EACH  
: OF SEVERAL MODE 2 INSTRUCTIONS ARE USED TO VERIFY THE DATA RESULTS OF  
: THE TEST. THIS PROCEDURE ALSO VERIFIES THE PROPER INCREMENTING OF THE  
: REGISTER.  
*****  
: TEST 51 TEST MODE 2 USING SOP INST.  
*****  
TST51: INC (R2) ;UPDATE TEST NUMBER  
CMP #51,(R2) ;SEQUENCE ERROR?  
BNE TST52-10 ;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/ 771 <====  
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 TST52  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 755 <====  
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
```



```

1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810 003600 005212
1811 003602 022712 000052
1812 003606 001023
1813 003610 005000
1814 003612 105100
1815 003614 005200
1816 003616 005010
1817 003620 005110
1818 003622 105020
1819 003624 001404
1820
1821
1822
1823
1824 003626 012742 000072
1825 003632 005242
1826 003634 000000
1827 003636 005300
1828 003640 005210
1829 003642 105110
1830 003644 105220
1831 003646 100003
1832 003650 005300
1833 003652 105220
1834 003654 001404
1835
1836
1837
1838
1839 003656
1840 003656 012742 000073
1841 003662 005242
1842 003664 000000
1843

```

```

*****
: THIS TEST VERIFIES MODE 2 SINGLE OPERAND INSTRUCTIONS WHICH
: ADDRESS EVEN BYTES.  RD IS SET TO 400 AND USED TO INITIALIZE LOCATION
: 400 TO -1.  CLRB INSTRUCTION IS THEN EXECUTED ON BYTE 400 WITH
: MODE 2.
: RD 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.
*****
TST52:  INC      (R2)          ; UPDATE TEST NUMBER
        CMP      #52,(R2)     ; SEQUENCE ERROR?
        BNE     TST53-10     ; BR TO ERROR HALT ON SEQ ERROR
        CLR     RD           ; SET RD=400
        COMB   RD
        INC     RD
        CLR     (RD)         ; CLEAR 400
        COM     (RD)         ; INITIALIZE: 400=-1
        CLRB   (RD)+        ; TRY TO CLEAR 400 W/MODE 2
        BEQ    SOPB2A
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
:          CONDITIONAL BRANCH INST. AND <====
:          REPLACE THE MOVE INSTRUCTION <====
:          WHICH FOLLOWS W/ 771 <====
        MOV     #72,-(R2)    ; MOVE TO MAILBOX # ***** 72 *****
        INC     -(R2)
        HALT
SOPB2A: DEC     RD           ; SET MSGTYP TO FATAL ERROR
        INC     (RD)         ; CLR DID NOT SET Z-BIT
        COMB   (RD)         ; RESULT RD=400
        INCB   (RD)+        ; INC 400 TO TEST WORD
: TRY TO INC EVEN BYTE
        BPL     SOPB2B
        DEC     RD           ; RESET RD=400
        INCB   (RD)+        ; TRY INCREMENT OF EVEN BYTE
        BEQ    TST53
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
:          CONDITIONAL BRANCH INST. AND <====
:          REPLACE THE MOVE INSTRUCTION <====
:          WHICH FOLLOWS W/ 755 <====
SOPB2B: MOV     #73,-(R2)    ; MOVE TO MAILBOX # ***** 73 *****
        INC     -(R2)
        HALT
: SET MSGTYP TO FATAL ERROR
: TEST CUMMULATIVE RESULT OF ABOVE INST.
: OR SEQUENCE ERROR

```



```

1844
1845
1846
1847
1848
1849
1850
1851
1852
1853 003666 005212
1854 003670 022712 000053
1855 003674 001026
1856 003676 005000
1857 003700 105100
1858 003702 005200
1859 003704 005010
1860 003706 005110
1861 003710 005200
1862 003712 105020
1863 003714 001404
1864
1865
1866
1867
1868 003716 012742 000074
1869 003722 005242
1870 003724 000000
1871 003726 005300
1872 003730 005300
1873 003732 005220
1874 003734 005300
1875 003736 105110
1876 003740 105220
1877 003742 100003
1878 003744 005300
1879 003746 105220
1880 003750 001404
1881
1882
1883
1884
1885 003752
1886 003752 012742 000075
1887 003756 005242
1888 003760 000000
1889
1890

```

```

:*****
:
:   THIS TEST FOLLOWS THE SAME PROCEDURE DESCRIBED IN THE PREVIOUS
:TEST.  HERE, THE BYTE INSTRUCTION IS USED TO ADDRESS AN ODD BYTE.
:
:*****
:TEST 53          TEST MODE 2 ODD BYTE USING SOP INST.
:*****
TST53:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #53,(R2)    ;SEQUENCE ERROR?
        BNE     TST54-10    ;BR TO ERROR HALT ON SEQ ERROR
        CLR     RO          ;SET RO=400
        COMB    RO
        INC     RO
        CLR     (RO)        ;CLEAR LOC 400
        COM     (RO)        ;INITIALIZE: 400=-1
        INC     RO          ;RO=ODD BYTE
        CLRB   (RO)+       ;TRY TO CLEAR ODD BYTE
        BEQ    SOPB2C
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS          <====
:               CONDITIONAL BRANCH INST. AND        <====
:               REPLACE THE MOVE INSTRUCTION        <====
:               WHICH FOLLOWS W/ 770                <====
:
:   MOV     #74,-(R2)    ;MOVE TO MAILBOX # ***** 74 *****
:   INC     -(R2)
:   HALT
SOPB2C: DEC     RO
        DEC     RO
        INC     (RO)+    ;INCREMENT WORD
        DEC     RO      ;POINT TO ODD BYTE
        COMB    (RO)    ;COMPLEMENT ODD BYTE
        INCB   (RO)+    ;TRY TO INCREMENT ODD BYTE
        BPL    SOPB2D
        DEC     RO
:   RESET RO TO ODD BYTE
:   TRY TO INCREMENT ODD BYTE
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS          <====
:               CONDITIONAL BRANCH INST. AND        <====
:               REPLACE THE MOVE INSTRUCTION        <====
:               WHICH FOLLOWS W/ 752                <====
:
:   MOV     #75,-(R2)    ;MOVE TO MAILBOX # ***** 75 *****
:   INC     -(R2)
:   HALT
:   TEST CUMMULATIVE RESULT OF ABOVE INST.
:   OR SEQUENCE ERROR

```


1891
1892
1893
1894
1895
1896
1897
1898
1899 003762 005212
1900 003764 022712 000054
1901 003770 001035
1902 003772 005000
1903 003774 005200
1904 003776 005400
1905 004000 100003
1906 004002 001402
1907 004004 102401
1908 004006 103404
1909
1910
1911
1912
1913 004010
1914 004010 012742 000076
1915 004014 005242
1916 004016 000000
1917
1918 004020 005200
1919 004022 001404
1920
1921
1922
1923
1924 004024 012742 000077
1925 004030 005242
1926 004032 000000
1927
1928 004034 105100
1929 004036 105400
1930 004040 100403
1931 004042 001402
1932 004044 102401
1933 004046 103404
1934
1935
1936
1937
1938 004050
1939 004050 012742 000100
1940 004054 005242
1941 004056 000000
1942 004060 005300
1943 004062 001404
1944
1945
1946

```
*****
:
: THESE TESTS CHECK THE NEGATE INSTRUCTION IN ALL MODES. PREVIOUSLY
: TESTED SINGLE-OPERAND INSTRUCTIONS ARE USED TO TEST THE NEGATE INSTRUCTION.
:
: *****
: TEST 54 TEST MODE 0 USING NEGATE INSTRUCTION
: *****
TST54: INC (R2) ;UPDATE TEST NUMBER
: CMP #54,(R2) ;SEQUENCE ERROR?
: BNE TST55-10 ;BR TO ERROR HALT ON SEQ ERROR
: CLR RO ;SET RO=0
: INC RO ;RO=1
: NEG RO ;TRY NEGATE MODE 0: RO=-1
: BPL NEG00 ;CC=1001?
: BEQ NEG00
: BVS NEG00
: BCS NEG01
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 771 <====
:
NEG00: MOV #76,-(R2) ;MOVE TO MAILBOX # ***** 76 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;NEGATE DID NOT SET CC'S CORRECTLY
:
NEG01: INC RO ;TEST DATA RESULT
: BEQ NEG02
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 763 <====
:
: MOVE TO MAILBOX # ***** 77 *****
: SET MSGTYP TO FATAL ERROR
: DATA RESULT OF NEGATE INCORRECT
:
NEG02: COMB RO ;RO=377
: NEGB RO ;RO=1
: BMI NEG03 ;CC=0001?
: BEQ NEG03
: BVS NEG03
: BCS NEG04
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 751 <====
:
NEG03: MOV #100,-(R2) ;MOVE TO MAILBOX # ***** 100 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;NEGB DID NOT SET CC'S CORRECTLY
NEG04: DEC RO ;TEST DATA RESULT
: BEQ TST55
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
```



```

2008 ;*****
2009 ;TEST 56 TEST MODE 2 USING NEGATE INSTRUCTION
2010 ;*****
2011 004214 005212 TST56: INC (R2) ;UPDATE TEST NUMBER
2012 004216 022712 000056 CMP #56,(R2) ;SEQUENCE ERROR?
2013 004222 001032 BNE TST57-10 ;BR TO ERROR HALT ON SEQ ERROR
2014 004224 005000 CLR RO ;POINT TO LOC. 0
2015 004226 005010 CLR (RO) ;CLEAR LOC. 0
2016 004230 005210 INC (RO) ;LOC. 0=1
2017 004232 005420 NEG (RO)+ ;TRY NEG.: LOC. 0=-1
2018 004234 100003 BPL NEG20 ;CC=1001?
2019 004236 001402 BEQ NEG20
2020 004240 102401 BVS NEG20
2021 004242 103404 BCS NEG21
2022 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2023 ; CONDITIONAL BRANCH INST. AND <====
2024 ; REPLACE THE MOVE INSTRUCTION <====
2025 ; WHICH FOLLOWS W/ 770 <====
2026 004244 NEG20:
2027 004244 012742 000106 MOV #106,-(R2) ;MOVE TO MAILBOX # ***** 106 *****
2028 004250 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2029 004252 000000 HALT ;NEGATE DID NOT SET CC'S CORRECTLY
2030 004254 105300 NEG21: DECB RO ;RO=LOC. 0
2031 004256 105300 DECB RO
2032 004260 105420 NEGB (RO)+ ;BYTE 0=1 RO=1
2033 004262 105420 NEGB (RO)+ ;BYTE 1=1 RO=2
2034 004264 105340 DECB -(RO) ;RO=1 LOC. 0=01
2035 004266 005300 DEC RO ;RO=0
2036 004270 001404 BEQ NEG22
2037 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2038 ; CONDITIONAL BRANCH INST. AND <====
2039 ; REPLACE THE MOVE INSTRUCTION <====
2040 ; WHICH FOLLOWS W/ 755 <====
2041 004272 012742 000107 MOV #107,-(R2) ;MOVE TO MAILBOX # ***** 107 *****
2042 004276 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2043 004300 000000 HALT ;REGISTER NOT INCREMENTED CORRECTLY
2044 004302 005337 000000 NEG22: DEC @#0 ;LOC. 0=0
2045 004306 001404 BEQ TST57
2046 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2047 ; CONDITIONAL BRANCH INST. AND <====
2048 ; REPLACE THE MOVE INSTRUCTION <====
2049 ; WHICH FOLLOWS W/ 746 <====
2050 004310 012742 000110 MOV #110,-(R2) ;MOVE TO MAILBOX # ***** 110 *****
2051 004314 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2052 004316 000000 HALT ;NEG BYTE INSTRUCTIONS FAILED
2053 ; OR SEQUENCE ERROR

```


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

004320 005212
004322 022712 000057
004326 001020
004330 005000
004332 105100
004334 005200
004336 005010
004340 005030
004342 001404

004344 012742 000111
004350 005242
004352 000000
004354 005300
004356 005300
004360 005130
004362 100002
004364 005230
004366 001404

004370
004370 012742 000112
004374 005242
004376 000000

```
*****
THIS TEST VERIFIES MODE 3 SINGLE OPERAND INSTRUCTIONS. IT
USES LOCATION 0 AS ITS TARGET DATA. A TABLE LOCATED AT LOC. 400
THRU 402 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
INSTRUCTIONS UNDER TEST.
RO IS SET TO 400, THE START OF THE ADDRESS TABLE, AND A CLR
INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR LOC. 0. THEN RO
IS DECREMENTED BY TWO AND TWO OTHER MODE 3 INSTRUCTIONS OPERATE ON
LOC. 0 TO VERIFY THE DATA RESULTS OF THE TEST. THE PROPER INCREMENTING
OF THE REGISTER IS ALSO VERIFIED IN THIS MANNER.
IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
(LOC. 400-402) HAS THE PROPER VALUES (0).
*****
TEST 57 TEST MODE 3 USING SOP INST.
*****
TST57: INC (R2) ;UPDATE TEST NUMBER
CMP #57,(R2) ;SEQUENCE ERROR?
BNE TST60-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;SET RO=400
COMB RO
INC RO
CLR (RO) ;CLEAR LOC 400
CLR @ (RO)+ ;TRY TO CLEAR LOC 0 USING MODE 3 ;RO=402
BEQ SOP3A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 772 <====
MOV #111,-(R2) ;MOVE TO MAILBOX # ***** 111 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR DID NOT SET Z-BIT
SOP3A: DEC RO ;RESET RO=400
DEC RO
COM @ (RO)+ ;TRY TO COMPLEMENT LOC 0 OF MODE 3 ;RO=402
BPL SOP3B ;TRY TO INCREMENT LOC 0 W/MODE 3 ;RO=404
INC @ (RO)+
BEQ TST60
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 760 <====
SOP3B: MOV #112,-(R2) ;MOVE TO MAILBOX # ***** 112 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CUMMULATIVE RESULT OF ABOVE INST FAILED
; OR SEQUENCE ERROR
```


2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142
2143
2144
2145
2146
2147
2148
2149
2150
2151
2152
2153
2154
2155

004400 005212
004402 022712 000060
004406 001026
004410 005004
004412 105104
004414 005204
004416 005000
004420 005010
004422 005110
004424 105034
004426 001404

012742 000113
005242
000000
005304
005304
005234
100006
105434
100004
005304
005304
105234
001404

004464
004464 012742 000114
004470 005242
004472 000000

```
*****
THIS TEST VERIFIES MODE 3 SINGLE OPERAND BYTE INSTRUCTIONS
WHICH ADDRESS EVEN BYTES. AGAIN, THE TARGET LOCATION 0 IS USED
AND THE SAME TABLE AT 400 IS EMPLOYED.
AFTER POINTING R4 TO THE TABLE (400) AND SETTING LOCATION
0 TO -1, A CLRB INSTRUCTION IS USED TO CLEAR BYTE 0.
SEVERAL OTHER MODE 3 INSTRUCTIONS ARE THEN USED WITH THE TABLE
TO VERIFY THE DATA RESULTS AND THE PROPER INCREMENTING OF THE REGISTER.
IF A FAILURE IS DETECTED, BE SURE THAT THE TABLE (LOCATION 400-402) HAS
THE PROPER VALUES (0).
*****
TEST 60 TEST MODE 3 EVEN BYTE USING SOP INST.
*****
TST60: INC (R2) ;UPDATE TEST NUMBER
CMP #60,(R2) ;SEQUENCE ERROR?
BNE TST61-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R4 ;SET R4=400
COMB R4
INC R4
CLR R0 ;INITIALIZE LOC. 0=-1
CLR (R0)
COM (R0) ;LOC. 0=-1
CLRB @R4)+ ;TRY TO CLEAR EVEN BYTE ;LOC. 0=177400 R4=402
BEQ SOPB3A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 770 <====
;MOVE TO MAILBOX # ***** 113 *****
;SET MSGTYP TO FATAL ERROR
;CLRB DID NOT SET Z-BIT
;RESET POINTER R4=400
SOPB3A: DEC R4
DEC R4
INC @R4)+ ;TRY INCREMENTING WORD LOC.0=177401 R4=402
BPL SOPB3B
NEGB @R4)+ ;TRY TO NEGATE EVEN BYTE ;LOC.0=-1 R4=404
BPL SOPB3B
DEC R4 ;R4=402
DEC R4
INCB @R4)+ ;TRY TO INCREMENT EVEN BYTE ;LOC. 0=17400
BEQ TST61
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 752 <====
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
```


2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200
2201
2202
2203
2204
2205
2206
2207

004474 005212
004476 022712
004502 001024
004504 005000
004506 105100
004510 005200
004512 005030
004514 005130
004516 105030
004520 001404

004522 012742
004526 005242
004530 000000
004532 005300
004534 005300
004536 005300
004540 005300
004542 005230
004544 105430
004546 100002
004550 105230
004552 001404

004554 012742
004554 005242
004560 000000

000061

000115

000116

```
*****  
: 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.  
*****  
TST61: INC (R2) ; UPDATE TEST NUMBER  
CMP #61,(R2) ; SEQUENCE ERROR?  
BNE TST62-10 ; 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/ 771 <====  
: MOVE TO MAILBOX # ***** 115 *****  
: SET MSGTYP TO FATAL ERROR  
: CLRB DID NOT SET Z-BIT  
: RESET R0=402  
  
SOPB3C: DEC R0 ; POINT TO EVEN BYTE ADDR.  
DEC R0  
DEC R0  
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 TST62  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 754 <====  
  
SOPB3D: MOV #116, -(R2) ; MOVE TO MAILBOX # ***** 116 *****  
INC -(R2) ; SET MSGTYP TO FATAL ERROR  
HALT ; CUMMULATIVE RESULT OF ABOVE INSTS FAILED  
: OR SEQUENCE ERROR
```



```

2208
2209
2210
2211 004564 005212
2212 004566 022712 000062
2213 004572 001054
2214 004574 005000
2215 004576 105100
2216 004600 005200
2217 004602 005010
2218 004604 005004
2219 004606 005014
2220 004610 005214
2221 004612 005430
2222 004614 100003
2223 004616 001402
2224 004620 102401
2225 004622 103404
2226
2227
2228
2229
2230 004624
2231 004624 012742 000117
2232 004630 005242
2233 004632 000000
2234 004634 005214
2235 004636 001404
2236
2237
2238
2239
2240 004640 012742 000120
2241 004644 005242
2242 004646 000000
2243 004650 105137 000001
2244 004654 005237 000000
2245 004660 105430
2246 004662 100404
2247
2248
2249
2250
2251 004664 012742 000121
2252 004670 005242
2253 004672 000000
2254 004674 105430
2255 004676 100004
2256
2257
2258
2259
2260 004700 012742 000122
2261 004704 005242
2262 004706 000000
2263 004710 105137 000001

```

```

;*****
;TEST 62 TEST MODE 3 USING NEGATE INSTRUCTION
;*****
TST62: INC (R2) ;UPDATE TEST NUMBER
CMP #62,(R2) ;SEQUENCE ERROR?
BNE TST63-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=400
COMB R0
INC R0
CLR (R0) ;LOC. 400=0
CLR R4 ;R4=0
CLR (R4) ;LOC. 0=0
INC (R4) ;LOC. 0=1
NEG @ (R0)+ ;TRY NEGATE LOC. 0=-1 R0=402
BPL NEG30 ;CC=1001?
BEQ NEG30
BVS NEG30
BCS NEG31

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

NEG30: MOV #117,-(R2) ;MOVE TO MAILBOX # ***** 117 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;NEG DID NOT SET CC'S CORRECTLY
NEG31: INC (R4) ;LOC. 0=0
BEQ NEG32

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

NEG32: MOV #120,-(R2) ;MOVE TO MAILBOX # ***** 120 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DATA RESULT OF NEG INCORRECT
COMB @#1 ;LOC 0=177400
INC @#0 ;LOC. 0=177401
NEGB @ (R0)+ ;TRY NEGB LOC. 0=177777 R0=404
BMI NEG33

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

NEG33: MOV #121,-(R2) ;MOVE TO MAILBOX # ***** 121 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;NEGB FAILED WITH EVEN BYTE
NEGB @ (R0)+ ;TRY NEGB LOC.0=777 R0=406
BPL NEG34

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

NEG34: MOV #122,-(R2) ;MOVE TO MAILBOX # ***** 122 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;NEGB FAILED WITH ODD BYTE
COMB @#1 ;LOC. 0=177377

```


L04

2264 004714 105237 000001
2265 004720 005214
2266 004722 001404
2267
2268
2269
2270
2271 004724 012742 000123
2272 004730 005242
2273 004732 000000
2274

INCB @#1
INC (R4)
BEQ TST63

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

;LOC. 0=177777
;LOC. 0=0

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 724 <====
; MOVE TO MAILBOX # ***** 123 *****
; SET MSGTYP TO FATAL ERROR
; DATA RESULT OF NEGB'S INCORRECT
; OR SEQUENCE ERROR

2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287 004734 005212
2288 004736 022712 000063
2289 004742 001021
2290 004744 005000
2291 004746 105100
2292 004750 005200
2293 004752 005040
2294 004754 001404
2295
2296
2297
2298
2299 004756 012742 000124
2300 004762 005242
2301 004764 000000
2302 004766 005200
2303 004770 005200
2304 004772 005140
2305 004774 100004
2306 004776 005200
2307 005000 005200
2308 005002 005240
2309 005004 001404
2310
2311
2312
2313
2314 005006
2315 005006 012742 000125
2316 005012 005242
2317 005014 000000
2318

```
*****
: THIS TEST VERIFIES MODE 4 SINGLE OPERAND INSTRUCTIONS.
: RD IS SET TO 400. A CLR INSTRUCTION IS EXECUTED IN MODE 4 TO CLEAR
: LOC. 376. RD 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
*****
TST63: INC (R2) ;UPDATE TEST NUMBER
CMP #63,(R2) ;SEQUENCE ERROR?
BNE TST64-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;SET RO=400
COMB RO
INC RO
CLR -(RO) ;TRY TO CLEAR USING MODE 4
BEQ SOP4A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 773 <====
;MOVE TO MAILBOX # ***** 124 *****
;SET MSGTYP TO FATAL ERROR
;CLR DID NOT SET Z-BIT
;RESET RO
SOP4A: INC RO
INC RO
COM -(RO) ;TRY TO COMPLEMENT USING MODE 4
BPL SOP4B
INC RO
INC RO
INC -(RO)
BEQ TST64
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 757 <====
SOP4B: MOV #125,-(R2) ;MOVE TO MAILBOX # ***** 125 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CHECK CUMMULATIVE RESULT OF ABOVE INST.
; OR SEQUENCE ERROR
```


2319
2320
2321
2322
2323
2324
2325
2326
2327
2328
2329
2330
2331
2332
2333
2334
2335
2336
2337
2338
2339
2340
2341
2342
2343
2344
2345
2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367

005016 005212
005020 022712 000064
005024 001017
005026 005000
005030 005020
005032 105400
005034 005050
005036 001404

005040 012742 000126
005044 005242
005046 000000
005050 005200
005052 005200
005054 005150
005056 100002
005060 005250
005062 001404

005064
005064 012742 000127
005070 005242
005072 000000

```
*****
THIS TEST VERIFIES MODE 5 SINGLE OPERAND INSTRUCTIONS. IT
USES LOCATION 0 AS ITS TARGET DATA. A TABLE LOCATED AT LOC. 372
THRU 374 IS USED TO SUPPLY THE ADDRESS OF LOCATION 0 TO THE
INSTRUCTIONS UNDER TEST.
RD IS SET TO 376. (THE START OF THE ADDRESS TABLE) +2,
AND A CLR INSTRUCTION IS EXECUTED WITH MODE 3 TO CLEAR
LOC. 0. THEN RD IS INCREMENTED BY TWO AND TWO OTHER MODE 3
INSTRUCTIONS OPERATE ON LOC. 0 TO VERIFY THE DATA RESULTS OF
THE TEST. THE PROPER DECREMENTING OF THE REGISTER IS ALSO
VERIFIED IN THIS MANNER.
IF A FAILURE IS DETECTED BE SURE TO VERIFY THAT THE TABLE
(LOC. 372 THRU 374) HAS THE PROPER VALUES (0).
*****
TEST 64 TEST MODE 5 USING SOP INSTS
*****
TST64: INC (R2) ;UPDATE TEST NUMBER
CMP #64,(R2) ;SEQUENCE ERROR?
BNE TST65-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RD ;SET RD=376
CLR (RD)+
NEGB RD
CLR @-(RD) ;TRY TO CLEAR LOC 0 W/MODE 5
BEQ SOP5A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 773 <====
;MOVE TO MAILBOX # ***** 126 *****
;SET MSGTYP TO FATAL ERROR
;CLR DID NOT SET Z-BIT
;RESET RD
SOP5A: INC RD
INC RD
COM @-(RD) ;TRY TO COMPLEMENT LOC. 0 W/MODE 5
BPL SOP5B
INC @-(RD) ;TRY TO INCREMENT LOC. 0 W/MODE 5
BEQ TST65
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 761 <====
SOP5B: MOV #127,-(R2) ;MOVE TO MAILBOX # ***** 127 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INSTS
; OR SEQUENCE ERROR
```


2368
2369
2370
2371
2372
2373
2374
2375
2376
2377
2378
2379
2380
2381
2382
2383
2384
2385
2386
2387
2388
2389
2390
2391
2392
2393
2394
2395
2396
2397
2398
2399
2400
2401
2402
2403
2404
2405
2406
2407

005074 005212
005076 022712 000065
005102 001020
005104 005000
005106 105100
005110 005200
005112 005060 177400
005116 001404

005120 012742 000130
005124 005242
005126 000000
005130 005160 177400
005134 100003
005136 005260 177400
005142 001404

005144
005144 012742 000131
005150 005242
005152 000000

```
*****
: 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
*****
TST65: INC (R2) ; UPDATE TEST NUMBER
      CMP #65, (R2) ; SEQUENCE ERROR?
      BNE TST66-10 ; 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/ 772 <====
      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 TST66
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 760 <====
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
```


2408
2409
2410
2411
2412
2413
2414
2415
2416
2417
2418
2419
2420
2421
2422
2423
2424
2425
2426
2427
2428
2429
2430
2431
2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450

005154 005212
005156 022712 000066
005162 001021
005164 005000
005166 105100
005170 005200
005172 005210
005174 005070 000002
005200 001404

005202 012742 000132
005206 005242
005210 000000
005212 005170 000002
005216 100003
005220 005270 000002
005224 001404

005226 012742 000133
005226 005242
005232 000000
005234 000000

```
*****
THIS TEST VERIFIES MODE 7 SINGLE OPERAND INSTRUCTIONS. IT USES
THE POINTER TO LOC. 0 WHICH IS STORED AT LOC. 402.
RO 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.
*****
TST66: INC (R2) ;UPDATE TEST NUMBER
CMP #66,(R2) ;SEQUENCE ERROR?
BNE TST67-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;SET RO=400
COMB RO
INC RO
INC (RO) ;RO=1
CLR @2(RO) ;TRY TO CLEAR LOC. 0 W/MODE 7
BEQ SOP7A
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
;MOVE TO MAILBOX # ***** 132 *****
;SET MSGTYP TO FATAL ERROR
;CLR DID NOT SET Z-BIT
SOP7A: COM @2(RO) ;TRY TO COMPLEMENT LOC. 0 W/MODE 7
BPL SOP7B
INC @2(RO) ;TRY TO INCREMENT LOC. 0 W/MODE 7
BEQ TST67
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 757 <====
SOP7B: MOV #133,-(R2) ;MOVE TO MAILBOX # ***** 133 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST CUMMULATIVE RESULT OF ABOVE INSTS.
; OR SEQUENCE ERROR
```



```

2451 *****
2452 ;TEST 67 TEST MODE 4 WITH NEGATE INSTRUCTION
2453 *****
2454 TST67: INC (R2) ;UPDATE TEST NUMBER
2455 005236 005212 000067 CMP #67,(R2) ;SEQUENCE ERROR?
2456 005240 022712 BNE TST70-10 ;BR TO ERROR HALT ON SEQ ERROR
2457 005244 001024 CLR RO
2458 005246 005000 CLR (RO)
2459 005250 005010 COM (RO)+ ;LOC. 0=177777, RO=2
2460 005252 005120 NEG -(RO) ;TRY NEGATE, LOC. 0=1
2461 005254 005440 BMI NEG40 ;CC=0001?
2462 005256 100403 BEQ NEG40
2463 005260 001402 BEQ NEG40
2464 005262 102401 BVS NEG40
2465 005264 103404 BCS NEG41
2466 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2467 ; CONDITIONAL BRANCH INST. AND <====
2468 ; REPLACE THE MOVE INSTRUCTION <====
2469 ; WHICH FOLLOWS W/ 770 <====
2470 005266 012742 000134 NEG40: MOV #134,-(R2) ;MOVE TO MAILBOX # ***** 134 *****
2471 005266 012742 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2472 005272 005242 HALT ;NEG DID NOT SET CC'S CORRECTLY
2473 005274 000000 NEG41: NEG RO ;TST RO WITH A NEG.
2474 005276 005400 BEQ NEG42
2475 005300 001404
2476 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2477 ; CONDITIONAL BRANCH INST. AND <====
2478 ; REPLACE THE MOVE INSTRUCTION <====
2479 ; WHICH FOLLOWS W/ 762 <====
2480 005302 012742 000135 MOV #135,-(R2) ;MOVE TO MAILBOX # ***** 135 *****
2481 005306 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2482 005310 000000 HALT ;RO NOT DECREMENTED PROPERLY
2483 005312 005310 NEG42: DEC (RO) ;TEST DTA RESULT OF NEG
2484 005314 001404 BEQ TST70
2485 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2486 ; CONDITIONAL BRANCH INST. AND <====
2487 ; REPLACE THE MOVE INSTRUCTION <====
2488 ; WHICH FOLLOWS W/ 754 <====
2489 005316 012742 000136 MOV #136,-(R2) ;MOVE TO MAILBOX # ***** 136 *****
2490 005322 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2491 005324 000000 HALT ;DATA RESULT OF NEG INCORRECT
; OR SEQUENCE ERROR

```



```

2492
2493
2494
2495 005326 005212
2496 005330 022712 000070
2497 005334 001031
2498 005336 005000
2499 005340 005010
2500 005342 105100
2501 005344 005200
2502 005346 005010
2503 005350 005004
2504 005352 005314
2505 005354 005450
2506 005356 100403
2507 005360 001402
2508 005362 102401
2509 005364 103404
2510
2511
2512
2513
2514 005366
2515 005366 012742 000137
2516 005372 005242
2517 005374 000000
2518 005376 005314
2519 005400 001404
2520
2521
2522
2523
2524 005402 012742 000140
2525 005406 005242
2526 005410 000000
2527 005412 105100
2528 005414 005300
2529 005416 001404
2530
2531
2532
2533
2534 005420 012742 000141
2535 005424 005242
2536 005426 000000
2537

```

```

:*****
:TEST 70 TEST MODE 5 WITH NEGATE INSTRUCTION
:*****
TST70: INC (R2) ;UPDATE TEST NUMBER
CMP #70,(R2) ;SEQUENCE ERROR?
BNE TST71-10 ;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/ 764 <====

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/ 756 <====

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 TST71

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

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

```



```

2538 ;*****
2539 ;TEST 71 TEST MODE 6 WITH NEGATE
2540 ;*****
2541 005430 005212 TST71: INC (R2) ;UPDATE TEST NUMBER
2542 005432 022712 000071 CMP #71,(R2) ;SEQUENCE ERROR?
2543 005436 001022 BNE TST72-10 ;BR TO ERROR HALT ON SEQ ERROR
2544 005440 005000 CLR R0 ;R0=0
2545 005442 005004 CLR R4 ;R4=0
2546 005444 105100 COMB R0 ;R0=377
2547 005446 005014 CLR (R4) ;LOC. 0=0
2548 005450 105024 CLRB (R4)+ ;LOC. 0=177777, R4=1
2549 005452 105114 COMB (R4) ;LOC. 0=177400
2550 005454 005460 177401 NEG -377(R0) ;LOC. 0=400
2551 005460 100403 BMI NEG60 ;CC=0001
2552 005462 001402 BEQ NEG60
2553 005464 102401 BVS NEG60
2554 005466 103404 BCS NEG61
2555 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2556 ; CONDITIONAL BRANCH INST. AND <====
2557 ; REPLACE THE MOVE INSTRUCTION <====
2558 ; WHICH FOLLOWS W/ 764 <====
2559 005470 NEG60:
2560 005470 012742 000142 MOV #142,-(R2) ;MOVE TO MAILBOX # ***** 142 *****
2561 005474 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2562 005476 000000 HALT ;NEG DID NOT SET CC'S CORRECTLY
2563 005500 105314 NEG61: DECB (R4)
2564 005502 001404 BEQ TST72
2565 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
2566 ; CONDITIONAL BRANCH INST. AND <====
2567 ; REPLACE THE MOVE INSTRUCTION <====
2568 ; WHICH FOLLOWS W/ 756 <====
2569 005504 012742 000143 MOV #143,-(R2) ;MOVE TO MAILBOX # ***** 143 *****
2570 005510 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
2571 005512 000000 HALT ;DATA RESULT OF NEG INCORRECT
2572 ; OR SEQUENCE ERROR

```



```

2573
2574
2575
2576 005514 005212
2577 005516 022712 000072
2578 005522 001024
2579 005524 005000
2580 005526 005010
2581 005530 005110
2582 005532 105100
2583 005534 105470 000005
2584 005540 100403
2585 005542 001402
2586 005544 102401
2587 005546 103404
2588
2589
2590
2591
2592 005550
2593 005550 012742 000144
2594 005554 005242
2595 005556 000000
2596 005560 105100
2597 005562 105120
2598 005564 105310
2599 005566 005467 172206
2600 005572 001404
2601
2602
2603
2604
2605 005574 012742 000145
2606 005600 005242
2607 005602 000000
2608

```

```

:*****
:TEST 72 TEST MODE 7 W/ NEGATE
:*****
TST72: INC (R2) ;UPDATE TEST NUMBER
CMP #72,(R2) ;SEQUENCE ERROR?
BNE TST73-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;RO=0
CLR (RO) ;LOC. 0=0
COM (RO) ;LOC. 0=177777
COMB RO ;RO=377
NEGB #5(RO) ;RO+5=404, 404=1, LOC. 0=777
BMI NEG70 ;CC=0001?
BEQ NEG70
BVS NEG70
BCS NEG71

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

NEG70: MOV #144,-(R2) ;MOVE TO MAILBOX # ***** 144 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;NEG DID NOT SET CC'S CORRECTLY
NEG71: COMB RO ;RO=0
COMB (RO)+ ;LOC. 0=400, RO=1
DECB (RO) ;LOC. 0=0
NEG 0 ;USE NEG MODE 67 TO TST FOR ZERO
BEQ TST73

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

MOV #145,-(R2) ;MOVE TO MAILBOX # ***** 145 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DATA RESULT OF NEG WAS INCORRECT
; OR SEQUENCE ERROR

```


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

005604 005212
005606 022712 000073
005612 001017
005614 005027
005616 177777
005620 001404

005622 012742 000146
005626 005242
005630 000000
005632 005237 005616
005636 005467 177754
005642 100003
005644 005277 000012
005650 001405

005652
005652 012742 000147
005656 005242
005660 000000
005662 005616

```
*****
:
:   THIS TEST VERIFIES PROGRAM COUNTER ADDRESSING WITH SOP
: INSTRUCTIONS. CLR MODE 77 IS USED TO CLEAR THE LOCATION FOLLOWING THE
: INSTRUCTION (SOPX). THEN SINGLE OPERAND INSTRUCTIONS WITH MODES 37, 67, AND
: 77, USING INDIRECT POINTER SOPXAD ARE USED TO VERIFY THE DATA RESULTS
: OF THESE INSTRUCTIONS.
:
: *****
: TEST 73      TEST SOP INSTRUCTIONS MODES 2,3,6,7 WITH REGISTER 7
: *****
TST73:  INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #73,(R2)    ;SEQUENCE ERROR?
        BNE     SOPB        ;BR TO ERROR HALT ON SEQ ERROR
        CLR     (R7)+       ;CLEAR NEXT LOCATION: (SOPX)
SOPX:   -1
        BEQ     SOPA        ;USE MODE 27
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
:           CONDITIONAL BRANCH INST. AND      <====
:           REPLACE THE MOVE INSTRUCTION      <====
:           WHICH FOLLOWS W/ 775              <====
:
:   MOV      #146, -(R2)    ;MOVE TO MAILBOX # ***** 146 *****
:   INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
:   HALT
SOPA:   INC      @#SOPX    ;CLR DID NOT SET Z-BIT
        NEG     SOPX      ;INC SOPX W/MODE 37
        BPL     SOPB      ;NEGATE SOPX W/MODE 67
        INC     @SOPXAD   ;INC SOPX W/MODE 77
        BEQ     TST74
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
:           CONDITIONAL BRANCH INST. AND      <====
:           REPLACE THE MOVE INSTRUCTION      <====
:           WHICH FOLLOWS W/ 761              <====
:
:   MOV      #147, -(R2)    ;MOVE TO MAILBOX # ***** 147 *****
:   INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
:   HALT
:           OR SEQUENCE ERROR
SOPXAD: SOPX      ;INDIRECT ADDRESS OF SOPX
```


2649
2650
2651
2652
2653
2654
2655
2656
2657
2658
2659
2660
2661
2662
2663
2664
2665
2666
2667
2668
2669
2670
2671
2672
2673
2674
2675
2676
2677
2678
2679
2680

005664 005212
005666 022712 000074
005672 001010
005674 005000
005676 000277
005700 000244
005702 005700
005704 102403
005706 100402
005710 103401
005712 001404

005714
005714 012742 000150
005720 005242
005722 000000

```
*****
:
:   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
: *****
TST74:  INC      (R2)          ; UPDATE TEST NUMBER
        CMP      #74,(R2)    ; SEQUENCE ERROR?
        BNE     TST75-10    ; BR TO ERROR HALT ON SEQ ERROR
        CLR     R0          ; INITIALIZE R0=0
        SCC     ;           ; SET CC=1011
        CLZ     ;
        TST     R0          ; TRY TST W/ MODE 0
        BVS     SNMOA      ; CHECK THAT CC=0100
        BMI     SNMOA
        BCS     SNMOA
        BEQ     TST75

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

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



2681
2682
2683
2684
2685
2686
2687
2688
2689
2690
2691
2692
2693
2694
2695
2696
2697
2698
2699
2700
2701
2702
2703
2704
2705
2706
2707
2708
2709
2710
2711
2712

005724 005212
005726 022712 000075
005732 001010
005734 005000
005736 105100
005740 000277
005742 000250
005744 105700
005746 102402
005750 101401
005752 100404

005754
005754 012742 000151
005760 005242
005762 000000

```
*****
: THIS TEST VERIFIES SINGLE OPERAND NON-MODIFYING BYTE INSTRUCTIONS WITH MODE 0.
: RD 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
*****
TST75: INC (R2) ; UPDATE TEST NUMBER
      CMP #75,(R2) ; SEQUENCE ERROR?
      BNE TST76-10 ; BR TO ERROR HALT ON SEQ ERROR
      CLR RO ; INITIALIZE
      COMB RO ; RO=377
      SCC ; SET CC=0111
      CLN
      TSTB RO ; TRY TST EVEN BYTE
      BVS SNMBOA ; CHECK CC=1000
      BLOS SNMBOA
      BMI TST76
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 770 <====
SNMBOA: MOV #151,-(R2) ; MOVE TO MAILBOX # ***** 151 *****
      INC -(R2) ; SET MSGTYP TO FATAL ERROR
      HALT ; CONDITION CODES NOT SET PROPERLY
; OR SEQUENCE ERROR
```

JF

2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745

005764 005212
005766 022712 000076
005772 001011
005774 005000
005776 005010
006000 000277
006002 000244
006004 005710
006006 102403
006010 103402
006012 100401
006014 001404

006016
006016 012742 000152
006022 005242
006024 000000

```
*****
: 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
*****
TST76: INC (R2) ;UPDATE TEST NUMBER
CMP #76,(R2) ;SEQUENCE ERROR?
BNE TST77-10 ;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 TST77

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

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



```

2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757 006026 005212
2758 006030 022712 000077
2759 006034 001026
2760 006036 005000
2761 006040 005010
2762 006042 105110
2763 006044 000277
2764 006046 000250
2765 006050 105710
2766 006052 102402
2767 006054 101401
2768 006056 100404
2769
2770
2771
2772
2773 006060
2774 006060 012742 000153
2775 006064 005242
2776 006066 000000
2777 006070 005000
2778 006072 005200
2779 006074 000277
2780 006076 000244
2781 006100 105710
2782 006102 102403
2783 006104 103402
2784 006106 100401
2785 006110 001404
2786
2787
2788
2789
2790 006112
2791 006112 012742 000154
2792 006116 005242
2793 006120 000000
2794

```

```

*****
THIS TEST SETS LOCATION 0 TO 377 AND THEN USES R0 TO TEST
THE EVEN BYTE AND THE ODD BYTE USING SOP BYTE INSTRUCTIONS WITH MODE 1.
AGAIN, CONDITIONAL BRANCHES ARE USED TO VERIFY THE SETTING OF THE
PROPER CONDITION CODE BITS.
*****
TEST 77 TEST MODE 1 BYTE INST. NON-MODIFYING
*****
TST77: INC (R2) ;UPDATE TEST NUMBER
CMP #77,(R2) ;SEQUENCE ERROR?
BNE TST100-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;POINT TO LOC 0
CLR (R0) ;CLEAR LOC 0
COMB (R0) ;COMPLEMENT BYTE 0
SCC ;SET CC=0111
CLN
TSTB (R0) ;TRY TST ON EVEN BYTE
BVS SNMB1A
BLOS SNMB1A
BMI SNMB1B

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

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

SNMB1B: CLR R0
INC R0
SCC ;SET CC=1011
CLZ
TSTB (R0) ;TRY TO TST AN ODD BYTE
BVS SNMB1C ;CHECK CC=0100
BCS SNMB1C
BMI SNMB1C
BEQ TST100

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

SNMB1C: MOV #154,-(R2) ;MOVE TO MAILBOX # ***** 154 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S NOT CORRECT
; OR SEQUENCE ERROR

```



```

2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806 006122 005212
2807 006124 022712 000100
2808 006130 001020
2809 006132 005000
2810 006134 005010
2811 006136 000277
2812 006140 000244
2813 006142 005720
2814 006144 102403
2815 006146 103402
2816 006150 100401
2817 006152 001404
2818
2819
2820
2821
2822 006154
2823 006154 012742 000155
2824 006160 005242
2825 006162 000000
2826 006164 005300
2827 006166 005300
2828 006170 001404
2829
2830
2831
2832
2833 006172 012742 000156
2834 006176 005242
2835 006200 000000
2836

```

```

*****
: THIS TEST VERIFIES THE SINGLE-OPERAND NON-MODIFYING INSTRUCTIONS
: USING MODE 2. IT USES THE IDENTICAL PROCEDURE EMPLOYED IN THE
: MODE 1 TESTS. ADDITIONALLY, THE REGISTER IS CHECKED TO ASSURE THAT
: IT IS INCREMENTED PROPERLY.
*****
: TEST 100 TEST MODE 2 WITH SOP NON-MODIFYING
*****
TST100: INC (R2) ;UPDATE TEST NUMBER
CMP #100,(R2) ;SEQUENCE ERROR?
BNE TST101-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;INITIALIZE RO=0
CLR (RO) ;CLEAR LOC 0
SCC ;SET CC=1011
CLZ
TST (RO)+ ;TRY TST W/ MODE 2
BVS SNM2A ;CHECK CC=0100
BCS SNM2A
BMI SNM2A
BEQ SNM2B

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

SNM2A: MOV #155,-(R2) ;MOVE TO MAILBOX # ***** 155 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S NOT CORRECT
SNM2B: DEC RO ;RESET RO
DEC RO
BEQ TST101

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

MOV #156,-(R2) ;MOVE TO MAILBOX # ***** 156 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MODE 2 DID NOT INC REG CORRECTLY
; OR SEQUENCE ERROR

```


2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892

006202 005212
006204 022712 000101
006210 001042
006212 005000
006214 005010
006216 105110
006220 000277
006222 000250
006224 105720
006226 102402
006230 101401
006232 100404

006234
006234 012742 000157
006240 005242
006242 000000
006244 005300
006246 001404

006250 012742 000160
006254 005242
006256 000000
006260 005200
006262 000277
006264 000244
006266 105720
006270 102403
006272 103402
006274 100401
006276 001404

006300
006300 012742 000161
006304 005242

```
*****  
: THIS TEST VERIFIES MODE 2 SINGLE OPERAND NON-MODIFYING BYTE  
: INSTRUCTIONS IT USES R0 TO POINT TO LOC. 0. WITH LOCATION 0  
: SET TO 377, THE EVEN AND ODD BYTE IS TESTED WITH TSTB INSTRUCTIONS  
: TO VERIFY THE CORRECT CC ARE SET. THE REGISTER IS CHECKED FOR  
: PROPER INCREMENTING.  
*****  
: TEST 101 TEST MODE 2 - BYTE W/ SOP NON-MODIFYING  
*****  
TST101: INC (R2) ;UPDATE TEST NUMBER  
CMP #101,(R2) ;SEQUENCE ERROR?  
BNE TST102-10 ;BR TO ERROR HALT ON SEQ ERROR  
CLR R0 ;CLEAR R0  
CLR (R0) ;CLEAR LOC 0  
COMB (R0) ;SET LOC 0=377  
SCC ;SET CC=0111  
CLN  
TSTB (R0)+ ;TRY TST OF EVEN BYTE  
BVS SNMB2A  
BLOS SNMB2A  
BMI SNMB2B  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 767 <====  
  
SNMB2A: MOV #157,-(R2) ;MOVE TO MAILBOX # ***** 157 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;CC'S NOT SET CORRECTLY  
SNMB2B: DEC R0 ;DECREMENT R0  
BEQ SNMB2C  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 761 <====  
  
SNMB2C: MOV #160,-(R2) ;MOVE TO MAILBOX # ***** 160 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR  
HALT ;MODE 2 DID NOT INC REG CORRECTLY  
SNMB2C: INC R0 ;POINT TO ODD BYTE  
SCC ;SET CC=1011  
CLZ  
TSTB (R0)+ ;TRY TST OF ODD BYTE  
BVS SNMB2D ;CHECK CC'S=0100  
BCS SNMB2D  
BMI SNMB2D  
BEQ SNMB2E  
  
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====  
: CONDITIONAL BRANCH INST. AND <====  
: REPLACE THE MOVE INSTRUCTION <====  
: WHICH FOLLOWS W/ 745 <====  
  
SNMB2D: MOV #161,-(R2) ;MOVE TO MAILBOX # ***** 161 *****  
INC -(R2) ;SET MSGTYP TO FATAL ERROR
```


2893 006306 000000
2894 006310 005300
2895 006312 005300
2896 006314 001404

SNMB2E: HALT
DEC RO
DEC RO
BEG TST102

;CC'S NOT CORRECT

2897
2898
2899
2900
2901 006316 012742 000162
2902 006322 005242
2903 006324 000000
2904

MOV #162, -(R2)
INC -(R2)
HALT

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 736 <====
: MOVE TO MAILBOX # ***** 162 *****
: SET MSGTYP TO FATAL ERROR
: RO DID NOT INCREMENT PROPERLY
: OR SEQUENCE ERROR

2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948

006326 005212
006330 022712 000102
006334 001022
006336 005000
006340 005010
006342 105100
006344 005300
006346 000277
006350 000244
006352 005730
006354 102403
006356 103402
006360 100401
006362 001404

006364 012742 000163
006370 005242
006372 000000
006374 005300
006376 105100
006400 001404

006402 012742 000164
006406 005242
006410 000000

```
*****
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
*****
TST102: INC (R2) ;UPDATE TEST NUMBER
CMP #102,(R2) ;SEQUENCE ERROR?
BNE TST103-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR RO ;RO=0
CLR (RO) ;CLEAR LOC 0
COMB RO ;RO=376
DEC RO
SCC ;SET CC=1011
CLZ
TST 2(RO)+ ;TRY TST W/ MODE 3
BVS SNM3A ;CHECK CC=0100
BCS SNM3A
BMI SNM3A
BEQ SNM3B

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

SNM3A: MOV #163,-(R2) ;MOVE TO MAILBOX # ***** 163 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CC'S NOT CORRECT
SNM3B: DEC RO ;RO=377
RO ;RO=0
COMB
BEQ TST103

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

MOV #164,-(R2) ;MOVE TO MAILBOX # ***** 164 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MODE 3 DID NOT INC REG CORRECTLY
; OR SEQUENCE ERROR
```


2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
3000
3001
3002
3003
3004

006412 005212
006414 022712 000103
006420 001036
006422 005000
006424 005010
006426 105110
006430 105100
006432 005200
006434 005720
006436 000277
006440 000250
006442 105730
006444 102402
006446 101401
006450 100404

006452 012742 000165
006456 005242
006460 000000
006462 000277
006464 000244
006466 105730
006470 102403
006472 103402
006474 100401
006476 001404

006500 012742 000166
006504 005242
006506 000000
006510 005720
006512 005710
006514 100404

```
*****
: 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.
*****
TST103: INC (R2) ;UPDATE TEST NUMBER
: CMP #103,(R2) ;SEQUENCE ERROR?
: BNE TST104-10 ;BR TO ERROR HALT ON SEQ ERROR
: CLR RO ;RO=0
: CLR (RO) ;CLEAR LOC 0
: COMB (RO) ;LOC. 0 =377
: COMB RO
: INC RO
: TST (RO)+ ;RO=402
: SCC ;CC=0111
: CLN
: TSTB 2(RO)+ ;TRY TST OF EVEN BYTE
: BVS SNMB3A ;CHECK CC=1000
: BLOS SNMB3A
: BMI SNMB3B

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

SNMB3A: MOV #165,-(R2) ;MOVE TO MAILBOX # ***** 165 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;CC'S NOT CORRECT
SNMB3B: SCC ;SET CC=1011
: CLZ
: TSTB 2(RO)+ ;TRY TST OF ODD BYTE
: BVS SNMB3C ;CHECK CC=0100
: BCS SNMB3C
: BMI SNMB3C
: BEQ SNMB3D

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

SNMB3C: MOV #166,-(R2) ;MOVE TO MAILBOX # ***** 166 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;CC'S NOT CORRECT
SNMB3D: TST (RO)+ ;RO=410
: TST (RO)
: BMI TST104

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



```

3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063 006604 005212
3064 006606 022712 000105
3065 006612 001022
3066 006614 005000
3067 006616 005010
3068 006620 005110
3069 006622 105100
3070 006624 005200
3071 006626 000277
3072 006630 000250
3073 006632 005750
3074 006634 102402
3075 006636 101401
3076 006640 100404
3077
3078
3079
3080
3081 006642
3082 006642 012742 000172
3083 006646 005242
3084 006650 000000
3085 006652 005200
3086 006654 105100
3087 006656 001404
3088
3089
3090
3091
3092 006660 012742 000173
3093 006664 005242
3094 006666 000000
3095

```

```

*****
: THIS TEST VERIFIES MODE 5 SOP NON-MODIFYING INSTRUCTIONS.
: IT USES A POINTER AT LOC. 376 TO TEST LOC. 0.  RD IS SET
: TO 400, A TST MODE 5 INSTRUCTION IS EXECUTED AND THE CC'S CHECKED.
: RD IS CHECKED TO INSURE PROPER DECREMENTING.
*****
: TEST 105      TEST MODE 5 W/ SOP NON-MODIFYING INSTS
*****
TST105: INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #105,(R2)    ;SEQUENCE ERROR?
        BNE     TST106-10    ;BR TO ERROR HALT ON SEQ ERROR
        CLR     RO           ;RO=0
        CLR     (RO)         ;LOC 0=0
        COM     (RO)         ;LOC 0=-1
        COMB    RO           ;RO=377
        INC     RO           ;RO=400
        SCC     ;SET CC=0111
        CLN
        TST     @-(RO)       ;TRY TST W/ MODE 5
        BVS     SNMSA        ;CHECK CC=1000
        BLOS    SNMSA
        BMI     SNMSB

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

SNMSA: MOV      #172,-(R2)    ;MOVE TO MAILBOX # ***** 172 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT
SNMSB: INC     RO           ;CC'S NOT SET PROPERLY
        COMB    RO           ;RO=377
        BEQ     TST106      ;RO=0

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

        MOV     #173,-(R2)   ;MOVE TO MAILBOX # ***** 173 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT                ;MODE 5 DID NOT DEC RO CORRECTLY
; OR SEQUENCE ERROR

```



```

3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107 006670 005212
3108 006672 022712 000106
3109 006676 001021
3110 006700 005000
3111 006702 005010
3112 006704 005110
3113 006706 105100
3114 006710 000277
3115 006712 000250
3116 006714 005760 177401
3117 006720 102402
3118 006722 101401
3119 006724 100404
3120
3121
3122
3123
3124 006726
3125 006726 012742 000174
3126 006732 005242
3127 006734 000000
3128 006736 105100
3129 006740 001404
3130
3131
3132
3133
3134 006742 012742 000175
3135 006746 005242
3136 006750 000000
3137

```

```

*****
: 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
*****
TST106: INC (R2) ;UPDATE TEST NUMBER
: CMP #106,(R2) ;SEQUENCE ERROR?
: BNE TST107-10 ;BR TO ERROR HALT ON SEQ ERROR
: CLR RO ;RO=0
: CLR (RO) ;LOC 0=0
: COM (RO) ;LOC 0=-1
: COMB RO ;RO=377
: SCC ;SET CC=0111
: CLN
: TST -377(RO) ;TRY TST W/ MODE 6
: BVS SNM6A ;CHECK CC=1000
: BLOS SNM6A
: BMI SNM6B
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 765 <====
:
SNM6A: MOV #174,-(R2) ;MOVE TO MAILBOX # ***** 174 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;CC'S INCORRECT
: SNM6B: COMB RO ;RO=0
: BEQ TST107
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 757 <====
:
: MOV #175,-(R2) ;MOVE TO MAILBOX # ***** 175 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;TST MODE 6 INCORRECTLY CHANGED RC
: OR SEQUENCE ERROR

```



```

3138
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149 006752 005212
3150 006754 022712 000107
3151 006760 001021
3152 006762 005000
3153 006764 005010
3154 006766 005110
3155 006770 105100
3156 006772 000277
3157 006774 000250
3158 006776 005770 000001
3159 007002 102402
3160 007004 101401
3161 007006 100404
3162
3163
3164
3165
3166 007010
3167 007010 012742 000176
3168 007014 005242
3169 007016 000000
3170 007020 105100
3171 007022 001404
3172
3173
3174
3175
3176 007024 012742 000177
3177 007030 005242
3178 007032 000000
3179

```

```

:*****
:      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.
:*****
TST107: INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #107,(R2)   ;SEQUENCE ERROR?
        BNE     TST110-10    ;BR TO ERROR HALT ON SEQ ERROR
        CLR     RO          ;RO=0
        CLR     (RO)        ;LOC 0=0
        COM     (RO)        ;LOC 0=-1
        COMB    RO          ;RO=377
        SCC     ;           ;CC=0111
        CLN
        TST     21(RO)      ;TRY TST W/ MODE 7
        BVS     SNM7A       ;CHECK CC=1000
        BLOS    SNM7A
        BMI     SNM7B

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

SNM7A:  MOV     #176,-(R2)   ;MOVE TO MAILBOX # ***** 176 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT
SNM7B:  COMB    RO          ;CC'S NOT CORRECT
        BEQ     TST110      ;RO=0

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

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

```



```

3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190 007034 005212
3191 007036 022712 000110
3192 007042 001006
3193 007044 005000
3194 007046 005100
3195 007050 005004
3196 007052 060004
3197 007054 005204
3198 007056 001404
3199
3200
3201
3202
3203 007060 012742 000200
3204 007064 005242
3205 007066 000700
3206
3207
3208
3209
3210
3211
3212
3213
3214
3215
3216
3217 007070 005212
3218 007072 022712 000111
3219 007076 001006
3220 007100 005000
3221 007102 005004
3222 007104 005100
3223 007106 010004
3224 007110 005204
3225 007112 001404
3226
3227
3228
3229
3230 007114 012742 000201
3231 007120 005242
3232 007122 000000
3233
3234
3235

```

```

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

```

```

TST110: INC (R2) ;UPDATE TEST NUMBER
: CMP #110,(R2) ;SEQUENCE ERROR?
: BNE TST111-10 ;BR TO ERROR HALT ON SEQ ERROR
: CLR R0 ;R0=0
: COM R0 ;R0=-1
: CLR R4 ;R4=0
: ADD R0,R4 ;TRY ADD: R4=-1
: INC R4 ;R4=0
: BEQ TST111

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 772 <====
: MOVE TO MAILBOX # ***** 200 *****
: SET MSGTYP TO FATAL ERROR
: ADD INST. FAILED W/ MODE 0
: OR SEQUENCE ERROR

```

```

*****
: THIS TEST VERIFIES THE MOVE INSTRUCTION WITH MODE 0 TO MODE 0.
: THIS TEST IS NECESSARY BECAUSE THIS PARTICULAR INSTRUCTION UTILIZES UNIQUE
: MICROCODE.
*****

```

```

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

```

```

TST111: INC (R2) ;UPDATE TEST NUMBER
: CMP #111,(R2) ;SEQUENCE ERROR?
: BNE TST112-10 ;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 TST112

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 772 <====
: MOVE TO MAILBOX # ***** 201 *****
: SET MSGTYP TO FATAL ERROR
: MOVE FAILED MODE 0 TO MODE 0
: OR SEQUENCE ERROR

```

```

*****

```


3236
3237
3238
3239
3240
3241
3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272

007124 005212
007126 022712 000112
007132 001016
007134 005000
007136 005004
007140 005204
007142 160400
007144 100003
007146 001402
007150 102401
007152 103404

007154
012742 000202
007160 005242
007162 000000
007164 005200
007166 001404

012742 000203
007174 005242
007176 000000

THIS TEST VERIFIES THE SUBTRACT INSTRUCTION WITH MODE 0,0.
THIS TEST IS NECESSARY BECAUSE THIS PARTICULAR INSTRUCTION UTILIZES SOME
UNIQUE MICROCODE.

TEST 112 TEST SUB MODE 0,0

TST112: INC (R2) ;UPDATE TEST NUMBER
CMP #112,(R2) ;SEQUENCE ERROR?
BNE TST113-10 ;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/ 770 <====

SUB0: MOV #202,-(R2) ;MOVE TO MAILBOX # ***** 202 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
SUB0A: INC R0 ;CONDITION CODE FAILED ON SUB
BEQ TST113

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

MOV #203,-(R2) ;MOVE TO MAILBOX # ***** 203 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;DATA RESULT OF SUB FAILED
OR SEQUENCE ERROR

2/8

3273
3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302
3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328

007200 005212
007202 022712 000113
007206 001051
007210 005000
007212 010004
007214 001404

007216 012742 000204
007222 005242
007224 000000
007226 005200
007230 005100
007232 005104
007234 040004
007236 005304
007240 001404

007242 012742 000205
007246 005242
007250 000000
007252 050004
007254 005204
007256 005204
007260 001404

007262 012742 000206
007266 005242
007270 000000
007272 005000
007274 105100
007276 005004
007300 005104
007302 040004
007304 060004
007306 005204

```
*****
: THIS TEST QUICKLY VERIFIES THE REMAINING DOP MODIFYING INSTRUCTIONS
: WITH MODE 0,0 TO PROVIDE A BASELINE FOR SUBSEQUENT TESTS.
: SINGLE OPERAND INSTRUCTIONS ARE USED TO SET UP DATA IN R0 AND R4
: BEFORE EACH OF THE SEVERAL DOP MODIFYING INSTRUCTIONS ARE USED AND
: VERIFIED.
*****
: TEST 113 TEST ALL THE DOP INSTRUCTIONS W/ SOURCE MODE 0,0
*****
TST113: INC (R2) ;UPDATE TEST NUMBER
        CMP #113,(R2) ;SEQUENCE ERROR?
        BNE TST114-10 ;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/ 775 <====

        MOV #204,-(R2) ;MOVE TO MAILBOX # ***** 204 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;Z-BIT NOT SET
DOP0A: INC R0 ;R0=1
        COM R0 ;R0=177776
        COM R4 ;R4=177777
        BIC R0,R4 ;TRY BIC: R4=1
        DEC R4 ;R4=0
        BEQ DOP0B

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

        MOV #205,-(R2) ;MOVE TO MAILBOX # ***** 205 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;BIC CLEAR RESULT INCORRECT
DOP0B: BIS R0,R4 ;TRY BIS: R4=177777
        INC R4 ;R4=0
        INC R4
        BEQ DOP0C

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

        MOV #206,-(R2) ;MOVE TO MAILBOX # ***** 206 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;RESULT OF BIS INCORRECT
DOP0C: 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
```



```

3350
3351
3352
3353
3354
3355
3356
3357
3358 007342 005212
3359 007344 022712 000114
3360 007350 001024
3361 007352 005000
3362 007354 005010
3363 007356 105110
3364 007360 005220
3365 007362 005400
3366 007364 060037 000000
3367 007370 100403
3368 007372 001402
3369 007374 102401
3370 007376 103404
3371
3372
3373
3374
3375 007400
3376 007400 012742 000211
3377 007404 005242
3378 007406 000000
3379 007410 105137 000000
3380 007414 005337 000000
3381 007420 001404
3382
3383
3384
3385
3386 007422 012742 000212
3387 007426 005242
3388 007430 000000
3389

```

```

:*****
:
:   THIS TEST VERIFIES MODE 0,X DOUBLE OPERAND INSTRUCTIONS.  IT SETS
:   DATA IN R0 AND LOCATION 0 AND OPERATES UPON IT USING DOP INSTRUCTIONS.
:
:*****
:TEST 114      TEST MODE 0,X DOUBLE-OPERAND INSTRUCTIONS
:*****
TST114: INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #114,(R2)   ;SEQUENCE ERROR?
        BNE     TST115-10   ;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/ 765 <====

DOP03A: MOV      #211,-(R2)   ;MOVE TO MAILBOX # ***** 211 *****
        INC     -(R2)       ;SET MSGTYP TO FATAL ERROR
        HALT                    ;CC'S NOT SET CORRECTLY
DOP03B: COMB    #0          ;LOC. 0=1
        DEC     #0          ;LOC. 0=0
        BEQ     TST115

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

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

```


3390
3391
3392
3393
3394
3395
3396
3397
3398
3399 007432 005212
3400 007434 022712 000115
3401 007440 001042
3402 007442 005000
3403 007444 005004
3404 007446 005204
3405 007450 020400
3406 007452 003004
3407
3408
3409
3410
3411 007454 012742 000213
3412 007460 005242
3413 007462 000000
3414 007464 020004
3415 007466 002404
3416
3417
3418
3419
3420 007470 012742 000214
3421 007474 005242
3422 007476 000000
3423 007500 005200
3424 007502 020400
3425 007504 001404
3426
3427
3428
3429
3430 007506 012742 000215
3431 007512 005242
3432 007514 000000
3433 007516 005000
3434 007520 005100
3435 007522 005004
3436 007524 030004
3437 007526 001404
3438
3439
3440
3441
3442 007530 012742 000216
3443 007534 005242
3444 007536 000000
3445 007540 005304

```
*****
: THIS TEST VERIFIES MODE 0,0 DOP NON-MODIFYING INSTRUCTIONS.
: R0 AND R4 ARE PRESET TO 0 AND 1 RESPECTIVELY. COMPARE INSTRUCTIONS ARE
: THEN EXECUTED AND CHECKED. FIRST R4 IS COMPARED TO R0 THEN R0 TO R4.
*****
: TEST 115 TEST DOP NON-MODIFYING INST. W/ SOURCE MODE 0,0
*****
TST115: INC (R2) ;UPDATE TEST NUMBER
: CMP #115,(R2) ;SEQUENCE ERROR?
: BNE TST116-10 ;BR TO ERROR HALT ON SEQ ERROR
: CLR R0 ;R0=0
: CLR R4 ;R4=0
: INC R4 ;R4=1
: CMP R4,R0 ;TRY COMPARE R4 TO R0
: BGT DNM1
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 773 <====
: MOVE TO MAILBOX # ***** 213 *****
: SET MSGTYP TO FATAL ERROR
: CC'S NOT CORRECT FOR CMP
: TRY COMPARE R0 TO R4
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 765 <====
: MOVE TO MAILBOX # ***** 214 *****
: SET MSGTYP TO FATAL ERROR
: CC'S NOT CORRECT FOR CMP
: R0=1
: TRY COMPARE R4=1 TO R0=1
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 756 <====
: MOVE TO MAILBOX # ***** 215 *****
: SET MSGTYP TO FATAL ERROR
: 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/ 745 <====
: MOVE TO MAILBOX # ***** 216 *****
: SET MSGTYP TO FATAL ERROR
: CC'S NOT CORRECT FOR BIT
: R4=177777
DNM1: MOV #213,-(R2)
: INC -(R2)
: HALT
: CMP R0,R4
: BLT DNM2
DNM2: MOV #214,-(R2)
: INC -(R2)
: HALT
: INC R0
: CMP R4,R0
: BEQ DNM3
DNM3: MOV #215,-(R2)
: INC -(R2)
: HALT
: CLR R0
: COM R0
: CLR R4
: BIT R0,R4
: BEQ DNM4
DNM4: MOV #216,-(R2)
: INC -(R2)
: HALT
: DEC R4
```



```

3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507 007642 005212
3508 007644 022712 000117
3509 007650 001007
3510 007652 005000
3511 007654 005100
3512 007656 005004
3513 007660 005014
3514 007662 005214
3515 007664 061400
3516 007666 001404
3517
3518
3519
3520
3521 007670 012742 000222
3522 007674 005242
3523 007676 000000
3524

```

```

*****
THIS TEST VERIFIES MODE 1 DOP INSTRUCTIONS.  RD IS SET TO -1
AND LOC 0 TO 1.  R4 IS THEN CLEARED AND USED TO POINT TO LOC 0.
IN THE ADD MODE 1 INSTRUCTION, LOC 0 IS ADDED TO RD AND THE
RESULTS VERIFIED.
*****
TEST 117      TEST MODE 1 W/ DOP INST.
*****
TST117: INC      (R2)          ;UPDATE TEST NUMBER
          CMP      #117,(R2)   ;SEQUENCE ERROR?
          BNE     TST120-10   ;BR TO ERROR HALT ON SEQ ERROR
          CLR     RD          ;RD=0
          COM     RD          ;RD=177777
          CLR     R4          ;R4=0
          CLR     (R4)        ;LOC 0=0
          INC     (R4)        ;LOC 0=1
          ADD     (R4),RD     ;TRY ADD SOURCE MODE 1
          BEQ     TST120
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ;          CONDITIONAL BRANCH INST. AND <====
          ;          REPLACE THE MOVE INSTRUCTION <====
          ;          WHICH FOLLOWS W/ 771 <====
          ; MOVE TO MAILBOX # ***** 222 *****
          ; SET MSGTYP TO FATAL ERROR
          ; RESULT OF ADD INCORRECT
          ; OR SEQUENCE ERROR

```


3525
3526
3527
3528
3529
3530
3531
3532
3533
3534
3535
3536
3537
3538
3539
3540
3541
3542
3543
3544
3545
3546
3547
3548
3549
3550
3551
3552

007700 005212
007702 022712 000120
007706 001007
007710 005000
007712 005010
007714 005110
007716 005004
007720 151004
007722 105104
007724 001404

007726 012742 000223
007732 005242
007734 000000

```
*****
THIS TEST VERIFIES MODE 1 DOP BYTE INSTRUCTIONS WHICH ADDRESS
EVEN BYTES. LOC. 0 IS SET TO -1 AND R4 IS CLEARED. THEN R4 IS
SET TO -1 USING A BISB THRU R0 WITH MODE 1.
*****
TEST 120 TEST MODE 1 - EVEN BYTE W/ DOP INSTS.
*****
TST120: INC (R2) ;UPDATE TEST NUMBER
CMP #120,(R2) ;SEQUENCE ERROR?
BNE TST121-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC. 0=0
COM (R0) ;LOC. 0=177777
CLR R4 ;R4=0
BISB (R0),R4 ;TRY MODE 1- EVEN BYTE W/ DOP
COMB R4 ;R4=0
BEQ TST121

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <===
; CONDITIONAL BRANCH INST. AND <===
; REPLACE THE MOVE INSTRUCTION <===
; WHICH FOLLOWS W/ 771 <===
; MOVE TO MAILBOX # ***** 223 *****
; SET MSGTYP TO FATAL ERROR
; RESULT OF BISB IS INCORRECT
; OR SEQUENCE ERROR
```



```

3553
3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564 007736 005212
3565 007740 022712 000121
3566 007744 001007
3567 007746 005000
3568 007750 005010
3569 007752 005110
3570 007754 005004
3571 007756 105104
3572 007760 121004
3573 007762 001404
3574
3575
3576
3577
3578 007764 012742 000224
3579 007770 005242
3580 007772 000000
3581

```

```

:*****
:
:      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.
:*****
TST121: INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #121,(R2)    ;SEQUENCE ERROR?
        BNE     TST122-10    ;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     TST122
:
:   TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
:              CONDITIONAL BRANCH INST. AND    <====
:              REPLACE THE MOVE INSTRUCTION    <====
:              WHICH FOLLOWS W/ 771           <====
: MOVE TO MAILBOX # ***** 224 *****
: SET MSGTYP TO FATAL ERROR
: RESULT OF CMPB INCORRECT
: OR SEQUENCE ERROR

```



3582
3583
3584
3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612
3613
3614
3615
3616
3617
3618
3619
3620
3621
3622
3623
3624
3625
3626

007774 005212
007776 022712 000122
010002 001020
010004 005000
010006 005010
010010 105110
010012 005110
010014 005004
010016 005104
010020 111004
010022 005704
010024 001404

010026 012742 000225
010032 005242
010034 000000
010036 005110
010040 111004
010042 100404

010044 012742 000226
010050 005242
010052 000000

```

*****
THIS TEST VERIFIES MODE 1,0 MOV B INSTRUCTIONS
WHICH ADDRESS EVEN BYTES. LOC. 0 IS SET TO 177400, R0 IS CLEARED AND
R4 IS SET TO -1. MOV B ARE USED TO MOVE BYTE 0 TO R4. THIS
VERIFIES THAT THE PROPER BYTE WAS SELECTED AND THAT THE SIGN-X-TEND
FUNCTION WITH MODE 0.
THEN LOC. 0 IS COMPLEMENTED AND THE SAME PROCEDURE EXERCISES
THE LOGIC FOR COMPLEMENTARY DATA.
THIS TEST EXERCISES UNIQUE MICROCODE.
*****
TEST 122 TEST MOV INSTRUCTION MODE 1,0 EVEN BYTE
*****
TST122: INC (R2) ;UPDATE TEST NUMBER
CMP #122,(R2) ;SEQUENCE ERROR?
BNE TST123-10 ;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
BEG DOP1

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
; MOVE TO MAILBOX # ***** 225 *****
; SET MSGTYP TO FATAL ERROR
; MOV B SHOULD SIGN X-TEND
; LOC 0=177777
; DO MOV B W/ EVEN BYTE

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 760 <====
; MOVE TO MAILBOX # ***** 226 *****
; SET MSGTYP TO FATAL ERROR
; MOV B SHOULD SIGN X-TEND
; OR SEQUENCE ERROR

```

DOP1:

```

MOV #225,-(R2)
INC -(R2)
HALT
DOP1: COM (R0)
MOVB (R0),R4
BMI TST123
MOV #226,-(R2)
INC -(R2)
HALT

```



```

3627
3628
3629
3630
3631
3632
3633
3634
3635
3636
3637
3638 010054 005212
3639 010056 022712 000123
3640 010062 001010
3641 010064 005000
3642 010066 005010
3643 010070 005004
3644 010072 005204
3645 010074 105114
3646 010076 151410
3647 010100 005210
3648 010102 001404
3649
3650
3651
3652
3653 010104 012742 000227
3654 010110 005242
3655 010112 000000
3656

```

```

*****
: 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.
*****
TST123: INC (R2) ;UPDATE TEST NUMBER
: CMP #123,(R2) ;SEQUENCE ERROR?
: BNE TST124-10 ;BR TO ERROR HALT ON SEQ ERROR
: CLR R0 ;R0=0
: CLR (R0) ;LOC. 0=0
: CLR R4 ;R4=0
: INC R4 ;R4=1
: COMB (R4) ;LOC. 0=177400
: BISB (R4),(R0) ;TRY TO BIS LOW ORDER BITS W/ MODE 1
: INC (R0) ;CHECK RESULT
: BEQ TST124
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 770 <====
: MOVE TO MAILBOX # ***** 227 *****
: SET MSGTYP TO FATAL ERROR
: RESULT OF BISB INCORRECT
: OR SEQUENCE ERROR

```



```

3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668 010114 005212
3669 010116 022712 000124
3670 010122 001015
3671 010124 005000
3672 010126 005010
3673 010130 005110
3674 010132 012004
3675 010134 005204
3676 010136 001404
3677
3678
3679
3680
3681 010140 012742 000230
3682 010144 005242
3683 010146 000000
3684 010150 005300
3685 010152 005300
3686 010154 001404
3687
3688
3689
3690
3691 010156 012742 000231
3692 010162 005242
3693 010164 000000
3694

```

```

*****
: THIS TEST VERIFIES MODE 2 DOP INSTRUCTIONS. LOC. 0 IS SET TO -1.
: R0 IS CLEARED AND USED AS THE MODE 2 ADDRESSING REGISTER TO MOVE LOC. 0
: TO R7. THE DATA RESULTS ARE VERIFIED AND THE INCREMENTING OF THE REGISTER
: IS CHECKED.
*****
: TEST 124 TEST MODE 2 W/ DOP INSTS.
*****
TST124: INC (R2) ;UPDATE TEST NUMBER
CMP #124,(R2) ;SEQUENCE ERROR?
BNE TST125-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC. 0=0
COM (R0) ;LOC. 0=177777
MOV (R0)+,R4 ;TRY MOVE MODE 2,0
INC R4 ;CHECK R4
BEQ DOP2

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 772 <====
; MOVE TO MAILBOX # ***** 230 *****
; SET MSGTYP TO FATAL ERROR
; RESULT OF MOV INST INCORRECT
; TEST R0 AFTER MODE 2

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 763 <====
; MOVE TO MAILBOX # ***** 231 *****
; SET MSGTYP TO FATAL ERROR
; REGISTER NOT INCREMENTED IN MODE 2
; OR SEQUENCE ERROR

```

DOP2:

```

MOV #230, -(R2)
INC -(R2)
DOP2: DEC R0
DEC R0
BEQ TST125

MOV #231, -(R2)
INC -(R2)
HALT

```


3695
3696
3697
3698
3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733

010166 005212
010170 022712 000125
010174 001016
010176 005000
010200 010010
010202 005110
010204 142010
010206 105737 000001
010212 001404

010214 012742 000232
010220 005242
010222 000000
010224 105137 000000
010230 001404

010232 012742 000233
010236 005242
010240 000000

```
*****
: THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH ADDRESS
: EVEN BYTES. LOC. 0 IS SET TO -1. R0 IS CLEARED AND USED AS THE
: ADDRESSING REGISTER IN A TEST WHICH TRIES TO CLEAR BYTE 1 USING
: BYTE 0 DATA AND A BICB. UNIQUE IN THIS TEST IS USE OF THE
: SAME ADDRESSING REGISTER FOR BOTH SOURCE AND DESTINATION. THE SOURCE AND
: DESTINATION IS CHECKED TO INSURE PROPER FUNCTIONING.
*****
: TEST 125 TEST MODE 2 - EVEN BYTE W/ DOP INST.
*****
TST125: INC (R2) ;UPDATE TEST NUMBER
        CMP #125,(R2) ;SEQUENCE ERROR?
        BNE TST126-10 ;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/ 771 <====
        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 TST126
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 762 <====
        MOV #233,-(R2) ;MOVE TO MAILBOX # ***** 233 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;BICB SOURCE INCORRECTLY CHANGED
; OR SEQUENCE ERROR
```



```

3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744 010242 005212
3745 010244 022712 000126
3746 010250 001017
3747 010252 005000
3748 010254 005004
3749 010256 005010
3750 010260 005110
3751 010262 105120
3752 010264 112004
3753 010266 005204
3754 010270 001404
3755
3756
3757
3758
3759 010272 012742 000234
3760 010276 005242
3761 010300 000000
3762 010302 005740
3763 010304 005700
3764 010306 001404
3765
3766
3767
3768
3769 010310 012742 000235
3770 010314 005242
3771 010316 000000
3772

```

```

*****
THIS TEST VERIFIES MODE 2 DOP BYTE INSTRUCTIONS WHICH REFERENCE
ODD BYTES. R0 IS SET TO 1, LOC. 0 IS SET TO 177400, AND R4 IS CLEARED.
A MODE 2 MOV B USES R0 TO MOVE BYTE 1 TO R4. AN INCREMENT
IS USED TO CHECK THAT THE PROPER BYTE WAS MOVED AND SIGN X-TENDED.
*****
TEST 126 TEST MODE 2 - ODD BYTE W/ DOP INST.
*****
TST126: INC (R2) ;UPDATE TEST NUMBER
CMP #126,(R2) ;SEQUENCE ERROR?
BNE TST127-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR R4 ;R4=0
CLR (R0) ;LOC. 0=0
COM (R0) ;LOC. 0=177777
COMB (R0)+ ;LOC 0=177400; R0=1
MOVB (R0)+,R4 ;TRY DOP MODE 2 W/ ODD BYTE
INC R4 ;CHECK RESULT OF MOV B
BEQ DOPB2B

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 770 <====
; MOVE TO MAILBOX # ***** 234 *****
; SET MSGTYP TO FATAL ERROR
; RESULT OF MOV B INCORRECT
; BUMP R0 DOWN BY 2
; CHECK R0

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 761 <====
; MOVE TO MAILBOX # ***** 235 *****
; SET MSGTYP TO FATAL ERROR
; MODE 2 BYTE DID NOT INCREMENT REG. CORRECTLY
; OR SEQUENCE ERROR

```

Handwritten notes:
I 175

Handwritten mark:

3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784 010320 005212
3785 010322 022712 000127
3786 010326 001011
3787 010330 012737 052525 000000
3788 010336 012700 125252
3789 010342 053700 000000
3790 010346 005200
3791 010350 001404
3792
3793
3794
3795
3796 010352 012742 000236
3797 010356 005242
3798 010360 000000
3799
3800
3801
3802
3803
3804
3805
3806
3807
3808
3809
3810 010362 005212
3811 010364 022712 000130
3812 010370 001011
3813 010372 012737 052652 000000
3814 010400 005000
3815 010402 153700 000000
3816 010406 022700 000252
3817 010412 001404
3818
3819
3820
3821
3822 010414 012742 000237
3823 010420 005242
3824 010422 000000
3825

: 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.

TST127: INC (R2) ; UPDATE TEST NUMBER
CMP #127, (R2) ; SEQUENCE ERROR?
BNE TST130-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #052525, @#0 ; MOVE 52525 TO LOC. 0
MOV #125252, R0 ; SET ALT. ONE AND ZERO IN R0
BIS @#0, R0 ; TRY TO SET ALL OTHER BITS W/ MODE 3
INC R0 ; TEST RESULT
BEQ TST130

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 767 <====
: MOVE TO MAILBOX # ***** 236 *****
: SET MSGTYP TO FATAL ERROR
: BIS W/ MODE 3 INCORRECT RESULT
: OR SEQUENCE ERROR

: THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS WHICH
: ADDRESS EVEN BYTES. BYTE 0 IS SET TO ALTERNATING 1'S AND 0'S; BYTE 1,
: ALTERNATING 0'S AND 1'S. R0 IS CLEARED AND A BISB IS USED TO
: SET THE LOW BYTE OF R0 TO 252.

: TEST 130 TEST MODE 3 - EVEN BYTE W/ DOP INSTS.

TST130: INC (R2) ; UPDATE TEST NUMBER
CMP #130, (R2) ; SEQUENCE ERROR?
BNE TST131-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #52652, @#0 ; MOVE 1'S AND 0' PATTERN TO LOC. 0
CLR R0 ; R0=0
BISB @#0, R0 ; TRY R0=252 W/ MODE 3 - EVEN BYTE
CMP #252, R0 ; BISB W/ EVEN BYTE SUCCESSFUL?
BEQ TST131

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 767 <====
: MOVE TO MAILBOX # ***** 237 *****
: SET MSGTYP TO FATAL ERROR
: BISB W/ MODE 3 - EVEN BYTE FAILED
: OR SEQUENCE ERROR

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

010424 005212
010426 022712 000131
010432 001011
010434 012737 052652 000000
010442 005000
010444 153700 000001
010450 022700 000125
010454 001404

010456 012742 000240
010462 005242
010464 000000

010466 005212
010470 022712 000132
010474 001017
010476 005000
010500 105100
010502 000263
010504 132700 000200
010510 001403
010512 102402
010514 103001
010516 100404

010520
010520 012742 000241
010524 005242
010526 000000
010530 105100
010532 001404

010534 012742 000242

```
*****
: THIS TEST VERIFIES MODE 3 DOUBLE OPERAND BYTE INSTRUCTIONS
: WHICH ADDRESS ODD BYTES. THE SAME PROCEDURE USED IN PREVIOUS
: TEST IS USED HERE. THIS TIME BYTE 1 IS USED AS THE SOURCE BYTE.
: THE EXPECTED RESULT IS: RO = 125.
*****
: TEST 131 TEST MODE 3 - ODD BYTE W/ DOP INSTS.
*****
TST131: INC (R2) ;UPDATE TEST NUMBER
        CMP #131,(R2) ;SEQUENCE ERROR?
        BNE TST132-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #52652,0#0 ;MOVE 1'S AND 0'S PATTERN TO LOC 0
        CLR RO ;RO=0
        BISB 0#1,RO ;TRY RO=152 W/ MODE 3 - ODD BYTE
        CMP #125,RO ;RO=125?
        BEQ TST132
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
        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
*****
TST132: INC (R2) ;UPDATE TEST NUMBER
        CMP #132,(R2) ;SEQUENCE ERROR?
        BNE TST133-10 ;BR TO ERROR HALT ON SEQ ERROR
        CLR RO ;RO=0
        COMB RO ;RO=377
        +SEC!SEV ;SET C AND V BITS
        BITB #200,RO ;TRY DOPNM DEST. MODE 0-BYTE
        BEQ DNMB0A ;BR TO ERROR IF Z BIT SET
        BVS DNMB0A ;BR TO ERROR IF V BIT SET
        BCC DNMB0A ;BR TO ERROR IF C BIT CLEAR.
        BMI DNMB0B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 767 <====
        DNMB0A: MOV #241,-(R2) ;MOVE TO MAILBOX # ***** 241 *****
                INC -(R2) ;SET MSGTYP TO FATAL ERROR
                HALT ;CC'S INCORRECT
        DNMB0B: COMB RO ;CHECK DESTINATION DATA
                BEQ TST133
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 761 <====
        MOV #242,-(R2) ;MOVE TO MAILBOX # ***** 242 *****
```



```

3882 010540 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
3883 010542 000000          HALT          ;DEST. DATA MODIFIED
3884                                     ; OR SEQUENCE ERROR
3885
3886 ;*****
3887 ;TEST 133      TEST DEST. MODE 1 W/ DOP NON-MODIFYING INST
3888 ;*****
3889 TST133: INC      (R2)          ;UPDATE TEST NUMBER
3890 010546 022712 000133  CMP      #133,(R2)      ;SEQUENCE ERROR?
3891 010552 001017          BNE      TST134-10     ;BR TO ERROR HALT ON SEQ ERROR
3892 010554 005000          CLR      RO           ;RO=0
3893 010556 005010          CLR      (RO)         ;LOC. 0=0
3894 010560 000241          CLC          ;CLEAR C BIT
3895 010562 032710 177777  BIT      #177777,(RO)  ;TRY DOPNM DEST. MODE 1
3896 010566 100403          BMI      DNM1A        ;BR TO ERROR IF N BIT SET
3897 010570 102402          BVS      DNM1A        ;BR TO ERROR IF V BIT SET
3898 010572 103401          BCS      DNM1A        ;BR TO ERROR IF C BIT SET
3899 010574 001404          BEQ      DNM1B
3900                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3901                                     ; CONDITIONAL BRANCH INST. AND <====
3902                                     ; REPLACE THE MOVE INSTRUCTION <====
3903                                     ; WHICH FOLLOWS W/ 767 <====
3904 010576          DNM1A:
3905 010576 012742 000243  MOV      #243,-(R2)    ;MOVE TO MAILBOX # ***** 243 *****
3906 010602 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
3907 010604 000000          HALT          ;COND. CODES INCORRECT
3908 010606 005710          DNM1B: TST      (RO)    ;CHECK TEST DATA
3909 010610 001404          BEQ      TST134
3910                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3911                                     ; CONDITIONAL BRANCH INST. AND <====
3912                                     ; REPLACE THE MOVE INSTRUCTION <====
3913                                     ; WHICH FOLLOWS W/ 761 <====
3914 010612 012742 000244  MOV      #244,-(R2)    ;MOVE TO MAILBOX # ***** 244 *****
3915 010616 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
3916 010620 000000          HALT          ;DESTINATION DATA MODIFIED
3917                                     ; OR SEQUENCE ERROR
3918
3919 ;*****
3920 ;TEST 134      TEST DEST, MODE 2 W/ DOP NON-MODIFYING INST.
3921 ;*****
3922 010622 005212 000134  TST134: INC      (R2)          ;UPDATE TEST NUMBER
3923 010624 022712 000134  CMP      #134,(R2)      ;SEQUENCE ERROR?
3924 010630 001027          BNE      TST135-10     ;BR TO ERROR HALT ON SEQ ERROR
3925 010632 005000          CLR      RO           ;RO=0
3926 010634 005010          CLR      (RO)         ;LOC. 0=0
3927 010636 052710 125252  BIS      #125252,(RO)  ;LOC. 0=125252
3928 010642 032720 077777  BIT      #77777,(RO)+ ;TRY DOPNM INST W/ MODE 2
3929 010646 102402          BVS      DNM2A        ;BR TO ERROR IF V BIT SET
3930 010650 001401          BEQ      DNM2A        ;BR TO ERROR IF Z-BIT SET
3931 010652 100004          BPL      DNM2B
3932                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
3933                                     ; CONDITIONAL BRANCH INST. AND <====
3934                                     ; REPLACE THE MOVE INSTRUCTION <====
3935                                     ; WHICH FOLLOWS W/ 767 <====
3936 010654          DNM2A:
3937 010654 012742 000245  MOV      #245,-(R2)    ;MOVE TO MAILBOX # ***** 245 *****

```



```

3994 011002 005200          DNMB2C: INC      R0          ;RO=1
3995 011004 132720 000201  BITB     #201,(R0)+    ;TRY DOPNM INST. W/MODE 2-ODD BYTE
3996 011010 001402          BEQ      DNMB2D ;BR TO ERROR IF Z-BIT SET
3997 011012 102401          BVS     DNMB2D ;BR TO ERROR IF V-BIT SET
3998 011014 100004          BPL     DNMB2E
3999
4000
4001
4002
4003 011016          DNMB2D:
4004 011016 012742 000252  MOV      #252,-(R2) ;MOVE TO MAILBOX # ***** 252 *****
4005 011022 005242          INC      -(R2) ;SET MSGTYP TO FATAL ERROR
4006 011024 000000          HALT
4007 011026 005300          DNMB2E: DEC      R0 ;COND. CODES INCORRECT
4008 011030 005300          DEC      R0 ;DEC R0 TO CHECK IT.
4009 011032 001404          BEQ      DNMB2F
4010
4011
4012
4013
4014 011034 012742 000253  MOV      #253,-(R2) ;MOVE TO MAILBOX # ***** 253 *****
4015 011040 005242          INC      -(R2) ;SET MSGTYP TO FATAL ERROR
4016 011042 000000          HALT ;DEST. REGISTER NOT INCREMENTED BY 1
4017 011044 022710 052652  DNMB2F: CMP      #52652,(R0) ;CHECK DEST. DATA IS UNMODIFIED
4018 011050 001404          BEQ      TST136
4019
4020
4021
4022
4023 011052 012742 000254  MOV      #254,-(R2) ;MOVE TO MAILBOX # ***** 254 *****
4024 011056 005242          INC      -(R2) ;SET MSGTYP TO FATAL ERROR
4025 011060 000000          HALT ;DEST. DATA WAS MODIFIED.
4026
4027
4028
4029
4030
4031
4032 011062 005212          TST136: INC      (R2) ;UPDATE TEST NUMBER
4033 011064 022712 000136  CMP      #136,(R2) ;SEQUENCE ERROR?
4034 011070 001050          BNE     TST137-10 ;BR TO ERROR HALT ON SEQ ERROR
4035 011072 005000          CLR     R0 ;RO=0
4036 011074 005010          CLR     (R0) ;LOC. 0=0
4037 011076 052710 125125  BIS     #125125,(R0) ;LOC. 0=125125
4038 011102 105100          COMB   R0 ;RO=377
4039 011104 005200          INC     R0 ;RO=400
4040 011106 005010          CLR     (R0) ;LOC. 400=0
4041 011110 000263          +SEC!SEV ;C-BIT=V-BIT=1
4042 011112 132730 000201  BITB     #201,(R0)+ ;TRY DOPNM W/MODE 3-EVEN BYTE
4043 011116 001403          BEQ     DNMB3A ;BR TO ERROR IF Z BIT SET
4044 011120 102402          BVS     DNMB3A ;BR TO ERROR IF V BIT SET
4045 011122 103001          BCC     DNMB3A ;BR TO ERROR IF C BIT CLEAR
4046 011124 100004          BPL     DNMB3B
4047
4048
4049

```

```

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

```

```

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

```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS
; CONDITIONAL BRANCH INST. AND
; REPLACE THE MOVE INSTRUCTION
; WHICH FOLLOWS W/ 727
<====
<====
<====
<====

```

```

; OR SEQUENCE ERROR

```

```

*****
;TEST 136 TEST DEST. MODE 3-BYTES W/DOP NON-MODIFYING INST.
*****

```

```

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

```



```

4050
4051 011126          DNMB3A:  MOV      #255, -(R2)      ; MOVE TO MAILBOX # ***** 255 *****
4052 011126 012742 000255      INC      -(R2)      ; SET MSGTYP TO FATAL ERROR
4053 011132 005242          HALT                    ; COND. CODES INCORRECT
4054 011134 000000          DNMB3B:  CMP      #402, R0      ; CHECK DEST. REGISTER INC. BY 2 AND INC BY 2 AGAIN
4055 011136 022700 000402      BEQ      DNMB3C
4056 011142 001404
4057
4058
4059
4060
4061 011144 012742 000256      MOV      #256, -(R2)      ; MOVE TO MAILBOX # ***** 256 *****
4062 011150 005242          INC      -(R2)      ; SET MSGTYP TO FATAL ERROR
4063 011152 000000          HALT                    ; DEST. REGISTER NOT INCREMENTED BY 2
4064 011154 005200          DNMB3C:  INC      R0      ; R0=404
4065 011156 005200          INC      R0
4066 011160 132730 000201      BITB    #201, 2(R0)+      ; TRY DOPNM DEST MODE 3-BYTE(ODD)
4067 011164 001402          BEQ      DNMB3D          ; BR TO ERROR IF Z BIT SET
4068 011166 102401          BVS     DNMB3D          ; BR TO ERROR IF V BIT SET
4069 011170 100404
4070
4071
4072
4073
4074 011172          DNMB3D:
4075 011172 012742 000257      MOV      #257, -(R2)      ; MOVE TO MAILBOX # ***** 257 *****
4076 011176 005242          INC      -(R2)      ; SET MSGTYP TO FATAL ERROR
4077 011200 000000          HALT                    ; COND. CODES INCORRECT
4078 011202 005004          DNMB3E:  CLR      R4      ; R4=0
4079 011204 022714 125125      CMP      #125125, (R4)    ; CHECK DEST. DATA
4080 011210 001404          BEQ      TST137
4081
4082
4083
4084
4085 011212 012742 000260      MOV      #260, -(R2)      ; MOVE TO MAILBOX # ***** 260 *****
4086 011216 005242          INC      -(R2)      ; SET MSGTYP TO FATAL ERROR
4087 011220 000000          HALT                    ; DEST. DATA MODIFIED
4088
4089
4090
4091
4092
4093 011222 005212          TST137: INC      (R2)      ; UPDATE TEST NUMBER
4094 011224 022712 000137      CMP      #137, (R2)      ; SEQUENCE ERROR?
4095 011230 001033          BNE     TST140-10        ; BR TO ERROR HALT ON SEQ ERROR
4096 011232 005000          CLR      R0      ; R0=0
4097 011234 005010          CLR      (R0)      ; LOC. 0=0
4098 011236 052710 125252      BIS     #125252, (R0)    ; LOC. 0=125125
4099 011242 052700 000002      BIS     #2, R0      ; R0=2
4100 011246 000277          SCC
4101 011250 032740 020000      BIT     #20000, -(R0)    ; SET ALL COND. CODE BITS
4102 011254 100403          BMI     DNM4A          ; TRY DOPNM W/ MODE 4
4103 011256 102402          BVS     DNM4A          ; BR TO ERROR IF N-BIT SET
4104 011260 103001          BCC     DNM4A          ; BR TO ERROR IF V-BIT SET
4105 011262 001004          BNE     DNM4B          ; BR TO ERROR IF C-BIT CHAR

```

```

*****
;TEST 137      TEST DEST. MODE 4 W/DOP NON-MODIFYING INST.
*****

```



```

4106                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4107                                     ; CONDITIONAL BRANCH INST. AND <====
4108                                     ; REPLACE THE MOVE INSTRUCTION <====
4109                                     ; WHICH FOLLOWS W/ 763 <====
4110 011264                                DNM4A:
4111 011264 012742 000261                MOV    #261,-(R2)    ;MOVE TO MAILBOX # ***** 261 *****
4112 011270 005242                        INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
4113 011272 000000                        HALT                               ;COND. CODES INCORRECT
4114 011274 005700                                TST    R0           ;CHECK DEST. REGISTER
4115 011276 001404                                BEQ    DNM4C
4116                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4117                                     ; CONDITIONAL BRANCH INST. AND <====
4118                                     ; REPLACE THE MOVE INSTRUCTION <====
4119                                     ; WHICH FOLLOWS W/ 755 <====
4120 011300 012742 000262                MOV    #262,-(R2)    ;MOVE TO MAILBOX # ***** 262 *****
4121 011304 005242                        INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
4122 011306 000000                        HALT                               ;DEST. REGISTER NOT DECREMENTED BY 2
4123 011310 022737 125252 000000 DNM4C:  CMP    #125252,R0    ;CHECK DEST. DATA
4124 011316 001404                                BEQ    TST140
4125                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4126                                     ; CONDITIONAL BRANCH INST. AND <====
4127                                     ; REPLACE THE MOVE INSTRUCTION <====
4128                                     ; WHICH FOLLOWS W/ 745 <====
4129 011320 012742 000263                MOV    #263,-(R2)    ;MOVE TO MAILBOX # ***** 263 *****
4130 011324 005242                        INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
4131 011326 000000                        HALT                               ;DEST. DATA MODIFIED
4132                                     ; OR SEQUENCE ERROR
4133
4134                                     ;*****
4135 ;TEST 140 TEST DEST. MODE 4-BYTE W/ DOP NON-MODIFYING INST.
4136 ;*****
4137 011330 005212                                TST140: INC    (R2)    ;UPDATE TEST NUMBER
4138 011332 022712 000140                CMP    #140,(R2)    ;SEQUENCE ERROR?
4139 011336 001051                        BNE    TST141-10    ;BR TO ERROR HALT ON SEQ ERROR
4140 011340 005000                                CLR    R0           ;R0=0
4141 011342 005010                                CLR    (R0)        ;LOC. 0=0
4142 011344 052710 052652                BIS    #52652,(R0)  ;LOC. 0=52652
4143 011350 052700 000002                BIS    #2,R0       ;R0=2
4144 011354 000257                                CCC                               ;COND. CODES=0
4145 011356 132740 000201                BITB   #201,-(R0)   ;TRY DOPNM INST W/MODE 4 ODD BYTE
4146 011362 102403                        BVS    DNMB4A       ;BR TO ERROR IF V BIT SET
4147 011364 001402                        BEQ    DNMB4A       ;BR TO ERROR IF Z BIT SET
4148 011366 103401                        BCS    DNMB4A       ;BR TO ERROR IF C BIT SET
4149 011370 001004                        BNE    DNMB4B
4150                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4151                                     ; CONDITIONAL BRANCH INST. AND <====
4152                                     ; REPLACE THE MOVE INSTRUCTION <====
4153                                     ; WHICH FOLLOWS W/ 763 <====
4154 011372                                DNMB4A:
4155 011372 012742 000264                MOV    #264,-(R2)    ;MOVE TO MAILBOX # ***** 264 *****
4156 011376 005242                        INC    -(R2)        ;SET MSGTYP TO FATAL ERROR
4157 011400 000000                        HALT                               ;COND. CODES INCORRECT
4158 011402 022700 000001                DNMB4B: CMP    #1,R0  ;CHECK DEST. REGISTER
4159 011406 001404                                BEQ    DNMB4C
4160                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4161                                     ; CONDITIONAL BRANCH INST. AND <====

```



```

4218 011534          DNMSA:
4219 011534 012742 000271      MOV      #271, -(R2)      ;MOVE TO MAILBOX # ***** 271 *****
4220 011540 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4221 011542 000000          HALT                    ;COND. CODES INCORRECT
4222 011544 022700 000400      DNMSB:  CMP      #400, R0  ;CHECK DEST. REGISTER
4223 011550 001404          BEQ      DNMSC          ;
4224          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4225          ;          CONDITIONAL BRANCH INST. AND <====
4226          ;          REPLACE THE MOVE INSTRUCTION <====
4227          ;          WHICH FOLLOWS W/ 754 <====
4228 011552 012742 000272      MOV      #272, -(R2)      ;MOVE TO MAILBOX # ***** 272 *****
4229 011556 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4230 011560 000000          HALT                    ;DEST. REGISTER NOT DECREMENTED BY 2
4231 011562 022737 100000 000000 DNMSC:  CMP      #100000, @#0    ;CHECK DESTINATION DATA
4232 011570 001404          BEQ      TST142         ;
4233          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4234          ;          CONDITIONAL BRANCH INST. AND <====
4235          ;          REPLACE THE MOVE INSTRUCTION <====
4236          ;          WHICH FOLLOWS W/ 744 <====
4237 011572 012742 000273      MOV      #273, -(R2)      ;MOVE TO MAILBOX # ***** 273 *****
4238 011576 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4239 011600 000000          HALT                    ;DEST. DATA INCORRECTLY MODIFIED
4240          ; OR SEQUENCE ERROR
4241
4242          ;*****
4243          ;TEST 142      TEST DEST. MODE 6 W/DOP NON-MODIFYING INST.
4244          ;*****
4245 011602 005212          TST142: INC      (R2)          ;UPDATE TEST NUMBER
4246 011604 022712 000142      CMP      #142, (R2)      ;SEQUENCE ERROR?
4247 011610 001033          BNE     TST143-10       ;BR TO ERROR HALT ON SEQ ERROR
4248 011612 005000          CLR     R0              ;R0=0
4249 011614 005010          CLR     (R0)            ;LOC) 0=0
4250 011616 052710 000001      BIS     #1, (R0)        ;LOC. 0=1
4251 011622 005100          COM     R0              ;R0=-1 C-BIT=1
4252 011624 032760 000001 000001  BIT     #1, 1(R0)       ;TRY DOPNM W/MODE 6
4253 011632 001403          BEQ     DNMSA           ;BR TO ERROR IF Z-BIT SET
4254 011634 102402          BVS     DNMSA           ;BR TO ERROR IF V-BIT SET
4255 011636 103001          BCC     DNMSA           ;BR TO ERROR IF C-BIT CLEAR
4256 011640 100004          BPL     DNMSB          ;
4257          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4258          ;          CONDITIONAL BRANCH INST. AND <====
4259          ;          REPLACE THE MOVE INSTRUCTION <====
4260          ;          WHICH FOLLOWS W/ 764 <====
4261 011642          DNMSA:
4262 011642 012742 000274      MOV      #274, -(R2)      ;MOVE TO MAILBOX # ***** 274 *****
4263 011646 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4264 011650 000000          HALT                    ;COND CODES INCORRECT
4265 011652 022700 177777      DNMSB:  CMP      #-1, R0  ;CHECK DEST. REGISTER
4266 011656 001404          BEQ     DNMSB          ;
4267          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4268          ;          CONDITIONAL BRANCH INST. AND <====
4269          ;          REPLACE THE MOVE INSTRUCTION <====
4270          ;          WHICH FOLLOWS W/ 755 <====
4271 011660 012742 000275      MOV      #275, -(R2)      ;MOVE TO MAILBOX # ***** 275 *****
4272 011664 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4273 011666 000000          HALT                    ;DEST. REGISTER MODIFIED

```



```

4274 011670 022737 000001 000000 DNM6C: CMP #1,0#0 ;CHECK DEST. DATA
4275 011676 001404 BEQ TST143 ;
4276 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4277 ; CONDITIONAL BRANCH INST. AND <====
4278 ; REPLACE THE MOVE INSTRUCTION <====
4279 ; WHICH FOLLOWS W/ 745 <====
4280 011700 012742 000276 MOV #276,-(R2) ;MOVE TO MAILBOX # ***** 276 *****
4281 011704 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4282 011706 000000 HALT ;DEST. DATA MODIFIED
4283 ; OR SEQUENCE ERROR
4284

```

```

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

```

```

4288 011710 005212 TST143: INC (R2) ;UPDATE TEST NUMBER
4289 011712 022712 000143 CMP #143,(R2) ;SEQUENCE ERROR?
4290 011716 001034 BNE TST144-10 ;BR TO ERROR HALT ON SEQ ERROR
4291 011720 005000 CLR RO ;RO=0
4292 011722 005010 CLR (RO) ;LOC. 0=0 C-BIT=0
4293 011724 052710 125125 BIS #125125,(RO) ;LOC. 0=125125
4294 011730 052700 000001 BIS #1,RO ;RO=1
4295 011734 132770 000125 000403 BITB #125,0403(RO) ;TRY DOPNM W/MODE 7
4296 011742 102403 BVS DNM7A ;BR TO ERROR IF V-BIT SET
4297 011744 100402 BMI DNM7A ;BR TO ERROR IF N-BIT SET
4298 011746 103401 BCS DNM7A ;BR TO ERROR IF C-BIT SET
4299 011750 001404 BEQ DNM7B

```

```

4300 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4301 ; CONDITIONAL BRANCH INST. AND <====
4302 ; REPLACE THE MOVE INSTRUCTION <====
4303 ; WHICH FOLLOWS W/ 763 <====

```

```

4304 011752 DNM7A: MOV #277,-(R2) ;MOVE TO MAILBOX # ***** 277 *****
4305 011752 012742 000277 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4306 011756 005242 HALT ;COND. CODES INCORRECT
4307 011760 000000 DNM7B: CMP #1,RO ;CHECK DEST. REGISTER
4308 011762 022700 000001 BEQ DNM7C
4309 011766 001404

```

```

4310 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4311 ; CONDITIONAL BRANCH INST. AND <====
4312 ; REPLACE THE MOVE INSTRUCTION <====
4313 ; WHICH FOLLOWS W/ 754 <====

```

```

4314 011770 012742 000300 MOV #300,-(R2) ;MOVE TO MAILBOX # ***** 300 *****
4315 011774 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4316 011776 000000 HALT ;DESTINATION REGISTER MODIFIED
4317 012000 022737 125125 000000 DNM7C: CMP #125125,0#0 ;CHECK DEST. DATA
4318 012006 001404 BEQ TST144

```

```

4319 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4320 ; CONDITIONAL BRANCH INST. AND <====
4321 ; REPLACE THE MOVE INSTRUCTION <====
4322 ; WHICH FOLLOWS W/ 744 <====

```

```

4323 012010 012742 000301 MOV #301,-(R2) ;MOVE TO MAILBOX # ***** 301 *****
4324 012014 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4325 012016 000000 HALT ;DEST. DATA INCORRECT
4326 ; OR SEQUENCE ERROR
4327

```

```

;*****
;

```


4330
4331
4332
4333
4334
4335
4336
4337 012020 005212
4338 012022 022712 000144
4339 012026 001016
4340 012030 005000
4341 012032 005010
4342 012034 005100
4343 012036 005004
4344 012040 010014
4345 012042 102402
4346 012044 001401
4347 012046 100404
4348
4349
4350
4351
4352 012050
4353 012050 012742 000302
4354 012054 005242
4355 012056 000000
4356 012060 005704
4357 012062 001404
4358
4359
4360
4361
4362 012064 012742 000303
4363 012070 005242
4364 012072 000000
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376 012074 005212
4377 012076 022712 000145
4378 012102 001025
4379 012104 005000
4380 012106 005010
4381 012110 005110
4382 012112 010020
4383 012114 100402
4384 012116 102401
4385 012120 001404

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

TST144: INC (R2) ;UPDATE TEST NUMBER
CMP #144,(R2) ;SEQUENCE ERROR?
BNE TST145-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC. 0=0
COM R0 ;R0=-1
CLR R4 ;R4 POINTS TO LOC. 0
MOV R0,(R4) ;TRY MOVE MODE 0,1
BVS MDM1A ;BR TO ERROR IF V SET
BEQ MDM1A ;BR TO ERROR IF Z SET
BMI MDM1B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 770 <====

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

MDM1B: TST R4
BEQ TST145
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 762 <====

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

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

TEST 145 TEST MOV DESTINATION MODE 2

TST145: INC (R2) ;UPDATE TEST NUMBER
CMP #145,(R2) ;SEQUENCE ERROR?
BNE TST146-10 ;BR TO ERROR HALT ON SEQ ERROR
CLR R0 ;R0=0
CLR (R0) ;LOC. 0=0
COM (R0) ;LOC. 0=+1
MOV R0,(R0)+ ;TRY MOVE MODE 0,2
BMI MDM2A ;BR TO ERROR IF N SET
BVS MDM2A ;BR TO ERROR IF V SET
BEQ MDM2B

4386 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4387 : CONDITIONAL BRANCH INST. AND <====
4388 : REPLACE THE MOVE INSTRUCTION <====
4389 : WHICH FOLLOWS W/ 771 <====

4390 012122 MDM2A: MOV #304, -(R2) ; MOVE TO MAILBOX # ***** 304 *****
4391 012122 012742 000304 INC -(R2) ; SET MSGTYP TO FATAL ERROR
4392 012126 005242 HALT ; CC'S INCORRECT
4393 012130 000000

4394 012132 005300 MDM2B: DEC R0
4395 012134 005300 DEC R0
4396 012136 001404 BEQ MDM2D

4397 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4398 : CONDITIONAL BRANCH INST. AND <====
4399 : REPLACE THE MOVE INSTRUCTION <====
4400 : WHICH FOLLOWS W/ 762 <====

4401 012140 MDM2C: MOV #305, -(R2) ; MOVE TO MAILBOX # ***** 305 *****
4402 012140 012742 000305 INC -(R2) ; SET MSGTYP TO FATAL ERROR
4403 012144 005242 HALT ; DESTINATION REGISTER NOT INCREMENTED PROPERLY
4404 012146 000000

4405 012150 005737 000000 MDM2D: TST #0
4406 012154 001404 BEQ TST146

4407 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4408 : CONDITIONAL BRANCH INST. AND <====
4409 : REPLACE THE MOVE INSTRUCTION <====
4410 : WHICH FOLLOWS W/ 753 <====

4411 012156 012742 000306 MOV #306, -(R2) ; MOVE TO MAILBOX # ***** 306 *****
4412 012162 005242 INC -(R2) ; SET MSGTYP TO FATAL ERROR
4413 012164 000000 HALT ; DESTINATION DATA INCORRECT
4414 : OR SEQUENCE ERROR

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

TEST 146 TEST MOV-BYTE DESTINATION MODE 2

4423 TST146: INC (R2) ; UPDATE TEST NUMBER
4424 012166 005212 000146 CMP #146, (R2) ; SEQUENCE ERROR?
4425 012170 022712 BNE TST147-10 ; BR TO ERROR HALT ON SEQ ERROR
4426 012174 001046 CLR R0 ; R0=0
4427 012176 005000 CLR (R0) ; LOC. 0=0
4428 012200 005010 MOV B #125, (R0)+ ; TRY DESTINATION MODE 2 W/EVEN BYTE
4429 012202 112720 000125 BVS MBDM2A ; BR TO ERROR IF V SET
4430 012206 102402 BEQ MBDM2A ; BR TO ERROR IF Z SET
4431 012210 001401 BPL MBDM2B

4432 : TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4433 : CONDITIONAL BRANCH INST. AND <====
4434 : REPLACE THE MOVE INSTRUCTION <====
4435 : WHICH FOLLOWS W/ 771 <====

4436 MBDM2A: MOV #307, -(R2) ; MOVE TO MAILBOX # ***** 307 *****
4437 012214 012742 000307 INC -(R2) ; SET MSGTYP TO FATAL ERROR
4438 012220 005242 HALT ; CC'S INCORRECT
4439 012222 000000

4440 012224 022700 000001 MBDM2B: CMP #1, R0


```

4442 012230 001404          BEQ      MBDM2C          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4443                                     ;          CONDITIONAL BRANCH INST. AND <====
4444                                     ;          REPLACE THE MOVE INSTRUCTION <====
4445                                     ;          WHICH FOLLOWS W/ 762 <====
4446                                     ;
4447 012232 012742 000310    MOV      #310,-(R2)    ; MOVE TO MAILBOX # ***** 310 *****
4448 012236 005242          INC      -(R2)        ; SET MSGTYP TO FATAL ERROR
4449 012240 000000          HALT                    ; REGISTER NOT INCREMENTED BY ONE
4450 012242 112720 000252    MBDM2C: MOVB   #252,(R0)+ ; TRY DESTINATION MODE 2 W/ODD BYTE
4451 012246 102402          BVS     MBDM2D
4452 012250 001401          BEQ     MBDM2D
4453 012252 100404          BMI     MBDM2E
4454                                     ;
4455                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4456                                     ;          CONDITIONAL BRANCH INST. AND <====
4457                                     ;          REPLACE THE MOVE INSTRUCTION <====
4458                                     ;          WHICH FOLLOWS W/ 751 <====
4459 012254 012742 000311    MBDM2D: MOV      #311,-(R2) ; MOVE TO MAILBOX # ***** 311 *****
4460 012260 005242          INC      -(R2)        ; SET MSGTYP TO FATAL ERROR
4461 012262 000000          HALT                    ; CC'S NOT SET CORRECT
4462 012264 022700 000002    MBDM2E: CMP     #2,R0
4463 012270 001404          BEQ     MBDM2F
4464                                     ;
4465                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4466                                     ;          CONDITIONAL BRANCH INST. AND <====
4467                                     ;          REPLACE THE MOVE INSTRUCTION <====
4468                                     ;          WHICH FOLLOWS W/ 742 <====
4469 012272 012742 000312    MOV      #312,-(R2)    ; MOVE TO MAILBOX # ***** 312 *****
4470 012276 005242          INC      -(R2)        ; SET MSGTYP TO FATAL ERROR
4471 012300 000000          HALT                    ; REGISTER NOT INCREMENTED BY ONE
4472 012302 022737 125125 000000 MBDM2F: CMP     #125125,#0 ; CHECK DATA
4473 012310 001404          BEQ     TST147
4474                                     ;
4475                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4476                                     ;          CONDITIONAL BRANCH INST. AND <====
4477                                     ;          REPLACE THE MOVE INSTRUCTION <====
4478                                     ;          WHICH FOLLOWS W/ 732 <====
4479 012312 012742 000313    MOV      #313,-(R2)    ; MOVE TO MAILBOX # ***** 313 *****
4480 012316 005242          INC      -(R2)        ; SET MSGTYP TO FATAL ERROR
4481 012320 000000          HALT                    ; DESTINATION DATA INCORRECT
4482                                     ; OR SEQUENCE ERROR

```

```

*****
; THIS TEST VERIFIES MOV DESTINATION MODE 3. R0 IS USED TO PICK UP
; AN ADDRESS AT LOC. 400. LOC 400 POINTS TO LOC. 0 THE EFFECTIVE DEST. ADDR.. ALSO, MOV
; 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
; *****

```

```

4489 012322 005212          TST147: INC     (R2)      ; UPDATE TEST NUMBER
4490 012324 022712 000147    CMP      #147,(R2)    ; SEQUENCE ERROR?
4491 012330 001057          BNE     TST150-10    ; BR TO ERROR HALT ON SEQ ERROR
4492 012332 012700 000400    MOV      #400,R0     ; R0=400
4493 012336 005010          CLR     (R0)         ; LOC. 400 POINTS TO LOC. 0
4494 012340 005037 000000    CLR     #0           ; LOC. 0=0
4495 012344 012730 125252    MOV      #125252,#(R0)+ ; TRY MOV DESTINATION MODE 2
4496 012350 102402          BVS     MDM3A        ; BR TO ERROR IF V SET

```



```

4498 012352 001401          BEQ      MDM3A      ;BR TO ERROR IF Z SET
4499 012354 100404          BMI      MDM3B
4500                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4501                                     ;          CONDITIONAL BRANCH INST. AND <====
4502                                     ;          REPLACE THE MOVE INSTRUCTION <====
4503                                     ;          WHICH FOLLOWS W/ 766 <====
4504 012356                                     MDM3A:
4505 012356 012742 000314      MOV      #314,-(R2)      ;MOVE TO MAILBOX # ***** 314 *****
4506 012362 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4507 012364 000000          HALT
4508 012366 022700 000402      MDM3B:  CMP      #402,R0      ;CC'S INCORRECT
4509 012372 001404          BEQ      MDM3C          ;CHECK DEST. MODE REGISTER
4510                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4511                                     ;          CONDITIONAL BRANCH INST. AND <====
4512                                     ;          REPLACE THE MOVE INSTRUCTION <====
4513                                     ;          WHICH FOLLOWS W/ 757 <====
4514 012374 012742 000315      MOV      #315,-(R2)      ;MOVE TO MAILBOX # ***** 315 *****
4515 012400 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4516 012402 000000          HALT
4517 012404 022737 125252, 000000 MDM3C:  CMP      #125252,#0      ;REGISTER NOT INCREMENTED BY 2
4518 012412 001404          BEQ      MDM3D          ;CHECK DESTINATION DATA
4519                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4520                                     ;          CONDITIONAL BRANCH INST. AND <====
4521                                     ;          REPLACE THE MOVE INSTRUCTION <====
4522                                     ;          WHICH FOLLOWS W/ 747 <====
4523 012414 012742 000316      MOV      #316,-(R2)      ;MOVE TO MAILBOX # ***** 316 *****
4524 012420 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4525 012422 000000          HALT
4526 012424 112737 000125 000000 MDM3D:  MOVB     #125,#0          ;DESTINATION DATA INCORRECT
4527 012432 022737 125125 000000      CMP      #125125,#0      ;TRY MOVB DESTINATION MODE Z EVEN BYTE
4528 012440 001404          BEQ      MDM3E          ;CHECK DATA
4529                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4530                                     ;          CONDITIONAL BRANCH INST. AND <====
4531                                     ;          REPLACE THE MOVE INSTRUCTION <====
4532                                     ;          WHICH FOLLOWS W/ 734 <====
4533 012442 012742 000317      MOV      #317,-(R2)      ;MOVE TO MAILBOX # ***** 317 *****
4534 012446 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4535 012450 000000          HALT
4536 012452 112737 000525 000001 MDM3E:  MOVB     #525,#1          ;DESTINATION DATA INCORRECT
4537 012460 022737 052525 000000      CMP      #52525,#0      ;TRY MOVB DESTINATION MODE 2 ODD BYTE
4538 012466 001404          BEQ      TST150        ;CHECK DATA
4539                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4540                                     ;          CONDITIONAL BRANCH INST. AND <====
4541                                     ;          REPLACE THE MOVE INSTRUCTION <====
4542                                     ;          WHICH FOLLOWS W/ 721 <====
4543 012470 012742 000320      MOV      #320,-(R2)      ;MOVE TO MAILBOX # ***** 320 *****
4544 012474 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4545 012476 000000          HALT

```

```

;*****
;
; 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.
;

```

```

4546
4547
4548
4549
4550
4551
4552
4553

```



```

4554 ;*****
4555 ;TEST 150 TEST MOV DESTINATION MODE 4
4556 ;*****
4557 012500 005212          TST150: INC      (R2)          ;UPDATE TEST NUMBER
4558 012502 022712 000150  CMP      #150,(R2)        ;SEQUENCE ERROR?
4559 012506 001026          BNE      TST151-10       ;BR TO ERROR HALT ON SEQ ERROR
4560 012510 005000          CLR      R0              ;R0=0
4561 012512 005010          CLR      (R0)            ;LOC 0=0
4562 012514 012704 000002  MOV      #2,R4          ;R4=2
4563 012520 012744 012345  MOV      #12345,-(R4)   ;TRY MOV DEST. MODE 4
4564 012524 102402          BVS     MDM4A           ;BR TO ERROR IF V-BIT SET
4565 012526 001401          BEQ     MDM4A           ;BR TO ERROR IF Z-BIT SET
4566 012530 100004          BPL     MDM4B
4567 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4568 ; CONDITIONAL BRANCH INST. AND <====
4569 ; REPLACE THE MOVE INSTRUCTION <====
4570 ; WHICH FOLLOWS W/ 767 <====
4571 012532          MDM4A:
4572 012532 012742 000321  MOV      #321,-(R2)     ;MOVE TO MAILBOX # ***** 321 *****
4573 012536 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
4574 012540 000000          HALT
4575 012542 005704          MDM4B: TST      R4      ;CC'S NOT CORRECT
4576 012544 001404          BEQ     MDM4C         ;CHECK DECREMENTING OF MODE 4 REG.
4577 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4578 ; CONDITIONAL BRANCH INST. AND <====
4579 ; REPLACE THE MOVE INSTRUCTION <====
4580 ; WHICH FOLLOWS W/ 761 <====
4581 012546 012742 000322  MOV      #322,-(R2)     ;MOVE TO MAILBOX # ***** 322 *****
4582 012552 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
4583 012554 000000          HALT
4584 012556 022710 012345  MDM4C: CMP      #12345,(R0) ;DESTINATION MODE REGISTER NOT DECREMENTED BY 2
4585 012562 001404          BEQ     TST151        ;CHECK DESTINATION DATA
4586 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4587 ; CONDITIONAL BRANCH INST. AND <====
4588 ; REPLACE THE MOVE INSTRUCTION <====
4589 ; WHICH FOLLOWS W/ 752 <====
4590 012564 012742 000323  MOV      #323,-(R2)     ;MOVE TO MAILBOX # ***** 323 *****
4591 012570 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
4592 012572 000000          HALT
4593 ; DESTINATION DATA INCORRECT
4594 ; OR SEQUENCE ERROR
4595 ;*****
4596 ;
4597 ; THIS TEST VERIFIES THE MOV(B) DESTINATION MODE 4 INSTRUCTION
4598 ; ON BOTH ODD AND EVEN BYTES. SOP INSTRUCTIONS ON R4 ARE
4599 ; USED TO CLEAR TARGET LOCATION 0. R0 IS USED AS THE MODE 4
4600 ; ADDRESSING REGISTER, AND CMP AND CONDITIONAL BRANCH
4601 ; INSTRUCTIONS ARE USED TO VERIFY THE DATA.
4602 ;
4603 ;*****
4604 ;TEST 151 TEST MOV(B) DESTINATION MODE 4
4605 ;*****
4606 012574 005212          TST151: INC      (R2)          ;UPDATE TEST NUMBER
4607 012576 022712 000151  CMP      #151,(R2)        ;SEQUENCE ERROR?
4608 012602 001046          BNE      TST152-10       ;BR TO ERROR HALT ON SEQ ERROR
4609 012604 005004          CLR      R4              ;R4=0

```



```

4610 012606 005014          CLR      (R4)          ;LOC. 0=0
4611 012610 012700 000002  MOV      #2,R0        ;RO = 2
4612 012614 112740 125125  MOV8     #125125,-(R0) ;TRY MOV8 DEST. MODE 4-ODD BYTE
4613 012620 020027 000001  CMP      R0,#1        ;CHECK THAT DEST. REG. WAS DECREMENTED
4614 012624 001404          BEQ      MBDM4A
4615          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4616          ;         CONDITIONAL BRANCH INST. AND <====
4617          ;         REPLACE THE MOVE INSTRUCTION <====
4618          ;         WHICH FOLLOWS W/ 767 <====
4619 012626 012742 000324          MOV      #324,-(R2)   ;MOVE TO MAILBOX # ***** 324 *****
4620 012632 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4621 012634 000000          HALT
4622 012636 021427 052400  MBDM4A: CMP      (R4),#52400 ;DESTINATION REG. NOT DECREMENTED BY 1
4623 012642 001404          BEQ      MBDM4B
4624          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4625          ;         CONDITIONAL BRANCH INST. AND <====
4626          ;         REPLACE THE MOVE INSTRUCTION <====
4627          ;         WHICH FOLLOWS W/ 760 <====
4628 012644 012742 000325          MOV      #325,-(R2)   ;MOVE TO MAILBOX # ***** 325 *****
4629 012650 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4630 012652 000000          HALT
4631 012654 112740 125125  MBDM4B: MOV8     #125125,-(R0) ;DEST. DATA NOT CORRECT
4632 012660 102402          BVS     MBDM4C        ;TRY MOV8 DEST. MODE 4--EVEN BYTE
4633 012662 001401          BEQ     MBDM4C        ;BR. TO ERROR IF V-BIT SET
4634 012664 100004          BPL     MBDM4D        ;BR TO ERROR IF Z-BIT SET
4635          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4636          ;         CONDITIONAL BRANCH INST. AND <====
4637          ;         REPLACE THE MOVE INSTRUCTION <====
4638          ;         WHICH FOLLOWS W/ 747 <====
4639 012666          MBDM4C:
4640 012666 012742 000326          MOV      #326,-(R2)   ;MOVE TO MAILBOX # ***** 326 *****
4641 012672 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4642 012674 000000          HALT
4643 012676 005700          MBDM4D: TST     R0
4644 012700 001404          BEQ     MBDM4E
4645          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4646          ;         CONDITIONAL BRANCH INST. AND <====
4647          ;         REPLACE THE MOVE INSTRUCTION <====
4648          ;         WHICH FOLLOWS W/ 741 <====
4649 012702 012742 000327          MOV      #327,-(R2)   ;MOVE TO MAILBOX # ***** 327 *****
4650 012706 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4651 012710 000000          HALT
4652 012712 021427 052525  MBDM4E: CMP      (R4),#52525 ;DESTINATION REG NOT DECREMENTED BY 1
4653 012716 001404          BEQ     TST152
4654          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4655          ;         CONDITIONAL BRANCH INST. AND <====
4656          ;         REPLACE THE MOVE INSTRUCTION <====
4657          ;         WHICH FOLLOWS W/ 732 <====
4658 012720 012742 000330          MOV      #330,-(R2)   ;MOVE TO MAILBOX # ***** 330 *****
4659 012724 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4660 012726 000000          HALT
4661          ; DESTINATION DATA INCORRECT
4662          ; OR SEQUENCE ERROR

```

```

;*****
;
; THIS TEST VERIFIES THE MOV DESTINATION MODE 5 AND THE MOV8
;

```



```

4666      ;DESTINATION MODE 5 - EVEN BYTE INSTRUCTIONS. R4 IS A
4667      ;POINTER TO TARGET LOCATION 0 AND R0 IS SETUP TO
4668      ;POINT TO LOCATION 376 FOR THE MOV, AND LOCATION 404 FOR
4669      ;THE MOV8 INSTRUCTIONS. CMP INSTRUCTIONS ARE USED TO VERIFY
4670      ;PROPER ADDRESSING AND DATA.
4671
4672      ;*****
4673      ;TEST 152      TEST MOV DESTINATION MODE 5
4674      ;*****
4675      012730 005212      TST152: INC      (R2)      ;UPDATE TEST NUMBER
4676      012732 022712 000152      CMP      #152,(R2)      ;SEQUENCE ERROR?
4677      012736 001051      BNE      TST153-10      ;BR TO ERROR HALT ON SEQ ERROR
4678      012740 005004      CLR      R4      ;R4=0
4679      012742 005014      CLR      (R4)      ;LOC. 0 = 0
4680      012744 012700 000400      MOV      #400,R0      ;R0=400
4681      012750 012750 004321      MOV      #4321,2-(R0) ;TRY MOV DEST. MODE 5
4682      012754 102402      BVS      MDM5A      ;BR TO ERROR IF V-BIT SET
4683      012756 001401      BEQ      MDM5A      ;BR TO ERROR IF Z-BIT SET
4684      012760 100004      BPL      MDM5B
4685      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4686      ; CONDITIONAL BRANCH INST. AND <====
4687      ; REPLACE THE MOVE INSTRUCTION <====
4688      ; WHICH FOLLOWS W/ 767 <====
4689      012762      MDM5A: MOV      #331, -(R2) ;MOVE TO MAILBOX # ***** 331 *****
4690      012762 012742 000331      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4691      012766 005242      HALT      ;COND. CODES INCORRECT
4692      012770 000000      MDM5B: CMP      #376,R0 ;CHECK MODE 5 REG. WAS DECREMENTED
4693      012772 022700 000376      BEQ      MDM5C
4694      012776 001404
4695      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4696      ; CONDITIONAL BRANCH INST. AND <====
4697      ; REPLACE THE MOVE INSTRUCTION <====
4698      ; WHICH FOLLOWS W/ 760 <====
4699      013000 012742 000332      MOV      #332, -(R2) ;MOVE TO MAILBOX # ***** 332 *****
4700      013004 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4701      013006 000000      HALT      ;MODE 5 REGISTER NOT DECREMENTED BY 2
4702      013010 022714 004321      MDM5C: CMP      #4321,(R4) ;CHECK DEST. DATA
4703      013014 001404      BEQ      MDM5D
4704      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4705      ; CONDITIONAL BRANCH INST. AND <====
4706      ; REPLACE THE MOVE INSTRUCTION <====
4707      ; WHICH FOLLOWS W/ 751 <====
4708      013016 012742 000333      MOV      #333, -(R2) ;MOVE TO MAILBOX # ***** 333 *****
4709      013022 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4710      013024 000000      HALT      ;DEST. DATA INCORRECT
4711      013026 012700 000406      MDM5D: MOV      #406,R0 ;R0=406
4712      013032 112750 000377      MOV8     #377,2-(R0) ;TRY MOV DEST. MODE 5 --EVEN BYTE
4713      013036 022700 000404      CMP      #404,R0 ;CHECK MODE 5 REG.
4714      013042 001404      BEQ      MDM5E
4715      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4716      ; CONDITIONAL BRANCH INST. AND <====
4717      ; REPLACE THE MOVE INSTRUCTION <====
4718      ; WHICH FOLLOWS W/ 736 <====
4719      013044 012742 000334      MOV      #334, -(R2) ;MOVE TO MAILBOX # ***** 334 *****
4720      013050 005242      INC      -(R2)      ;SET MSGTYP TO FATAL ERROR
4721      013052 000000      HALT      ;MODE 5 REGISTER NOT DECREMENTED BY 2

```



```

4722 013054 022714 177721 MDM5E: CMP #177721,(R4) ;CHECK DEST. DATA
4723 013060 001404 BEQ TST153
4724
4725 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4726 ; CONDITIONAL BRANCH INST. AND <====
4727 ; REPLACE THE MOVE INSTRUCTION <====
4728 ; WHICH FOLLOWS W/ 727 <====
4728 013062 012742 000335 MOV #335,-(R2) ;MOVE TO MAILBOX # ***** 335 *****
4729 013066 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4730 013070 000000 HALT ;DEST. DATA INCORRECT
4731 ; OR SEQUENCE ERROR
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743

```

```

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

```

TEST 153 TEST MOV DESTINATION MODE 6

```

4744 013072 005212 TST153: INC (R2) ;UPDATE TEST NUMBER
4745 013074 022712 000153 CMP #153,(R2) ;SEQUENCE ERROR?
4746 013100 001054 BNE TST154-10 ;BR TO ERROR HALT ON SEQ ERROR
4747 013102 005000 CLR R0 ;R0=0
4748 013104 005010 CLR (R0) ;LOC. 0=0
4749 013106 005200 INC R0 ;R0=1
4750 013110 012760 052525 177777 MOV #052525,-1(R0) ;TRY MOV DEST. MODE 6
4751 013116 102402 BVS MDM6A ;BR TO ERROR IF V-BIT SET
4752 013120 001401 BEQ MDM6A ;BR TO ERROR IF Z-BIT SET
4753 013122 100004 BPL MDM6B
4754
4755 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4756 ; CONDITIONAL BRANCH INST. AND <====
4757 ; REPLACE THE MOVE INSTRUCTION <====
4758 ; WHICH FOLLOWS W/ 767 <====
4758 013124 MDM6A: MOV #336,-(R2) ;MOVE TO MAILBOX # ***** 336 *****
4759 013124 012742 000336 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4760 013130 005242 HALT ;COND. CODES INCORRECT
4761 013132 000000 MDM6B: CMP #1,R0 ;CHECK DEST. REGISTER UNALTERED
4762 013134 022700 000001 BEQ MDM6C
4763 013140 001404
4764
4765 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4766 ; CONDITIONAL BRANCH INST. AND <====
4767 ; REPLACE THE MOVE INSTRUCTION <====
4768 ; WHICH FOLLOWS W/ 760 <====
4768 013142 012742 000337 MOV #337,-(R2) ;MOVE TO MAILBOX # ***** 337 *****
4769 013146 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
4770 013150 000000 HALT ;DEST. REGISTER INCORRECTLY ALTERED
4771 013152 022737 052525 000000 MDM6C: CMP #52525,R#0 ;CHECK DEST. DATA
4772 013160 001404 BEQ MDM6D
4773
4774 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4775 ; CONDITIONAL BRANCH INST. AND <====
4776 ; REPLACE THE MOVE INSTRUCTION <====
4777 ; WHICH FOLLOWS W/ 750 <====
4777 013162 012742 000340 MOV #340,-(R2) ;MOVE TO MAILBOX # ***** 340 *****

```



```

4778 013166 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
4779 013170 000000          HALT                    ;DEST. DATA INCORRECT
4780 013172 012700 000002   MDM6D: MOV      #2,R0        ;RO=2
4781 013176 112760 000377 177777  MOVB   #377,-1(R0)    ;TRY MOVB DEST. MODE 6
4782 013204 022700 000002   CMP      #2,R0        ;CHECK DEST. REGISTER UNALTERED
4783 013210 001404          BEQ      MDM6E
4784                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4785                                     ;          CONDITIONAL BRANCH INST. AND <====
4786                                     ;          REPLACE THE MOVE INSTRUCTION <====
4787                                     ;          WHICH FOLLOWS W/ 734 <====
4788 013212 012742 000341          MOV      #341,-(R2)    ;MOVE TO MAILBOX # ***** 341 *****
4789 013216 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4790 013220 000000          HALT                    ;DEST. REGISTER INCORRECTLY ALTERED
4791 013222 022737 177525 000000  MDM6E: CMP      #177525,#0 ;CHECK DEST. DATA
4792 013230 001404          BEQ      TST154
4793                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4794                                     ;          CONDITIONAL BRANCH INST. AND <====
4795                                     ;          REPLACE THE MOVE INSTRUCTION <====
4796                                     ;          WHICH FOLLOWS W/ 724 <====
4797 013232 012742 000342          MOV      #342,-(R2)    ;MOVE TO MAILBOX # ***** 342 *****
4798 013236 005242          INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4799 013240 000000          HALT                    ;DEST. DATA INCORRECT
4800                                     ; OR SEQUENCE ERROR

```

```

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

```

```

4809 TEST 154 TEST MOV DESTINATION MODE 7
4810 *****
4811 TST154: INC      (R2)          ;UPDATE TEST NUMBER
4812 013242 005212 000154   CMP      #154,(R2)    ;SEQUENCE ERROR?
4813 013244 022712          BNE     TST155-10    ;BR TO ERROR HALT ON SEQ ERROR
4814 013250 001053          CLR     R4           ;R4=0
4815 013252 005004          CLR     (R4)        ;LOC.0=0
4816 013254 005014          MOV     #403,R0     ;R0=403
4817 013256 012700 000403   MOV     #70707,#-1(R0) ;TRY MOV W/DEST MODE 7
4818 013262 012770 070707 177777  BVS     MDM7A        ;BR. TO ERROR IF V-BIT SET
4819 013270 102402          BEQ     MDM7A        ;BR TO ERROR IF Z-BIT SET
4820 013272 001401          BPL     MDM7B
4821 013274 100004          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4822                                     ;          CONDITIONAL BRANCH INST. AND <====
4823                                     ;          REPLACE THE MOVE INSTRUCTION <====
4824                                     ;          WHICH FOLLOWS W/ 766 <====
4825 MDM7A: MOV      #343,-(R2)    ;MOVE TO MAILBOX # ***** 343 *****
4826 013276 012742 000343   INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
4827 013276 005242          HALT                    ;COND. CODES INCORRECT
4828 013302 000000          MDM7B: CMP     #403,R0 ;CHECK DEST. REGISTER
4829 013304 000000          BEQ     MDM7C
4830 013306 022700 000403   ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4831 013312 001404          ;          CONDITIONAL BRANCH INST. AND <====
4832
4833

```



```

4834                                     ; REPLACE THE MOVE INSTRUCTION <====
4835                                     ; WHICH FOLLOWS W/ 757 <====
4836 013314 012742 000344             MOV    #344,-(R2)      ; MOVE TO MAILBOX # ***** 344 *****
4837 013320 005242                   INC    -(R2)        ; SET MSGTYP TO FATAL ERROR
4838 013322 000000                   HALT                   ; DEST. REGISTER INCORRECTLY ALTERED
4839 013324 022737 070707 000000 MDM7C: CMP    #70707,#0     ; CHECK DEST. DATA
4840 013332 001404                   BEQ    MDM7D
4841                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4842                                     ; CONDITIONAL BRANCH INST. AND <====
4843                                     ; REPLACE THE MOVE INSTRUCTION <====
4844                                     ; WHICH FOLLOWS W/ 747 <====
4845 013334 012742 000345             MOV    #345,-(R2)      ; MOVE TO MAILBOX # ***** 345 *****
4846 013340 005242                   INC    -(R2)        ; SET MSGTYP TO FATAL ERROR
4847 013342 000000                   HALT                   ; DEST. DATA INCORRECT
4848 013344 112770 107070 000001 MDM7D: MOVB  #107070,#1(R0)  ; TRY MOVB W/DEST MODE 7--ODD BYTE
4849 013352 022700 000403                   CMP    #403,R0
4850 013356 001404                   BEQ    MDM7E
4851                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4852                                     ; CONDITIONAL BRANCH INST. AND <====
4853                                     ; REPLACE THE MOVE INSTRUCTION <====
4854                                     ; WHICH FOLLOWS W/ 735 <====
4855 013360 012742 000346             MOV    #346,-(R2)      ; MOVE TO MAILBOX # ***** 346 *****
4856 013364 005242                   INC    -(R2)        ; SET MSGTYP TO FATAL ERROR
4857 013366 000000                   HALT                   ; DEST. DATA INCORRECT
4858 013370 022737 034307 000000 MDM7E: CMP    #34307,#0     ; CHECK DEST. DATA
4859 013376 001404                   BEQ    TST155
4860                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4861                                     ; CONDITIONAL BRANCH INST. AND <====
4862                                     ; REPLACE THE MOVE INSTRUCTION <====
4863                                     ; WHICH FOLLOWS W/ 725 <====
4864 013400 012742 000347             MOV    #347,-(R2)      ; MOVE TO MAILBOX # ***** 347 *****
4865 013404 005242                   INC    -(R2)        ; SET MSGTYP TO FATAL ERROR
4866 013406 000000                   HALT                   ; DESTINATION DATA INCORRECT
4867                                     ; OR SEQUENCE ERROR

```

Back

```

*****
THIS TEST VERIFIES MODE 4 DOUBLE OPERAND INSTRUCTIONS.
THE TEST USES MODE 4 ADDRESSING WITH REGISTER 0 TO MOVE THRU A
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.
*****
TST155: INC    (R2)           ; UPDATE TEST NUMBER
4884 013410 005212 000155             CMP    #155,(R2)      ; SEQUENCE ERROR?
4885 013412 022712                   BNE   DOP4           ; BR TO ERROR HALT ON SEQ ERROR
4886 013416 001015                   MOV   #TBL1,R0       ; INITIALIZE R0
4887 013420 012700 013472             MOV   -(R0),#TBL1    ; TBL1=125252
4888 013424 014037 013472             MOV   -(R0),#TBL1    ; TBL1=000377
4889 013430 064037 013472             ADD   -(R0),#TBL1

```



```

4890 013434 144037 013472      BICB  -(R0),2#TBL1      ;TBL1=000252
4891 013440 154037 013473      BISB  -(R0),2#TBL1+1    ;TBL1=125252
4892 013444 024037 013472      CMP   -(R0),2#TBL1      ;CHECK RESULT
4893 013450 001411                      BEQ   TST156
4894                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4895                                ;          CONDITIONAL BRANCH INST. AND <====
4896                                ;          REPLACE THE MOVE INSTRUCTION <====
4897                                ;          WHICH FOLLOWS W/ 763 <====
4898 013452                                DOP4:
4899 013452 012742 000350      MOV   #350,-(R2)        ;MOVE TO MAILBOX # ***** 350 *****
4900 013456 005242                      INC   -(R2)             ;SET MSGTYP TO FATAL ERROR
4901 013460 000000                      HALT                    ;RESULT OF MODE 4 INSTS. INCORRECT
4902                                ; OR SEQUENCE ERROR

```

```

4904 013462 125252                      125252
4905 013464 052652                      52652
4906 013466 053125                      53125
4907 013470 125252                      125252
4908 013472 000000

```

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.
*****

```

```

4921 TST156: INC (R2) ;UPDATE TEST NUMBER
4922 013474 005212 000156      CMP   #156,(R2) ;SEQUENCE ERROR?
4923 013476 022712                      BNE   DOP5 ;BR TO ERROR HALT ON SEQ ERROR
4924 013502 001015                      MOV   #TBL2+2,R0 ;INITIALIZE R0
4925 013504 012700 013560      MOV   2-(R0),2#TBL1 ;TBL1=125252
4926 013510 015037 013472      ADD   2-(R0),2#TBL1 ;TBL1=000377
4927 013514 065037 013472      BICB 2-(R0),2#TBL1 ;TBL1=000252
4928 013520 145037 013472      BISB 2-(R0),2#TBL1+1 ;TBL1=125252
4929 013524 155037 013473      CMP   2-(R0),2#TBL1 ;CHECK RESULT
4930 013530 025037 013472      BEQ   TST157
4931 013534 001411
4932                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
4933                                ;          CONDITIONAL BRANCH INST. AND <====
4934                                ;          REPLACE THE MOVE INSTRUCTION <====
4935                                ;          WHICH FOLLOWS W/ 763 <====

```

```

4936 013536                                DOP5:
4937 013536 012742 000351      MOV   #351,-(R2)        ;MOVE TO MAILBOX # ***** 351 *****
4938 013542 005242                      INC   -(R2)             ;SET MSGTYP TO FATAL ERROR
4939 013544 000000                      HALT                    ;RESULT OF MODE 5 INSTS. INCORRECT
4940                                ; OR SEQUENCE ERROR

```

```

4941 013546 013462                      TBL1-10
4942 013550 013464                      TBL1-6
4943 013552 013465                      TBL1-5
4944 013554 013466                      TBL1-4
4945 013556 013470                      TBL2: TBL1-2

```



```

4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959 013560 005212
4960 013562 022712 000157
4961 013566 001022
4962 013570 012700 013466
4963 013574 016037 000002 013472
4964 013602 066037 000000 013472
4965 013610 146037 177777 013472
4966 013616 156037 177776 013473
4967 013624 026037 177774 013472
4968 013632 001404
4969
4970
4971
4972
4973 013634 012742 000352
4974 013640 005242
4975 013642 000000
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990 013644 005212
4991 013646 022712 000160
4992 013652 001022
4993 013654 012700 013552
4994 013660 017037 000004 013472
4995 013666 067037 000002 013472
4996 013674 147037 000000 013472
4997 013702 157037 177776 013473
4998 013710 027037 177774 013472
4999 013716 001404
5000
5001

```

```

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

```

```

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

```

```

TST157: INC (R2) ;UPDATE TEST NUMBER
: CMP #157,(R2) ;SEQUENCE ERROR?
: BNE TST160-10 ;BR TO ERROR HALT ON SEQ ERROR
: MOV #TBL1-4,RO ;INITIALIZE RO
: MOV 2(RO),@#TBL1 ;TBL1=125252
: ADD 0(RO),@#TBL1 ;TBL1=000377
: BICB -1(RO),@#TBL1 ;TBL1=000252
: BISB -2(RO),@#TBL1+1 ;TBL1=125252
: CMP -4(RO),@#TBL1 ;CHECK RESULT
: BEQ TST160

```

```

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: ; CONDITIONAL BRANCH INST. AND <====
: ; REPLACE THE MOVE INSTRUCTION <====
: ; WHICH FOLLOWS W/ 756 <====
: ; MOVE TO MAILBOX # ***** 352 *****
: ; SET MSGTYP TO FATAL ERROR
: ; RESULT OF MODE 6 INSTS. INCORRECT
: ; OR SEQUENCE ERROR

```

```

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

```

```

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

```

```

TST160: INC (R2) ;UPDATE TEST NUMBER
: CMP #160,(R2) ;SEQUENCE ERROR?
: BNE TST161-10 ;BR TO ERROR HALT ON SEQ ERROR
: MOV #TBL2-4,RO ;INITIALIZE RO
: MOV @4(RO),@#TBL1 ;TBL1=125252
: ADD @2(RO),@#TBL1 ;TBL1=000377
: BICB @0(RO),@#TBL1 ;TBL1=000252
: BISB @-2(RO),@#TBL1+1 ;TBL1=125252
: CMP @-4(RO),@#TBL1 ;CHECK RESULT
: BEQ TST161

```

```

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

```



```

5002                                     ; REPLACE THE MOVE INSTRUCTION <====
5003                                     ; WHICH FOLLOWS W/ 756 <====
5004 013720 012742 000353             MOV #353, -(R2) ; MOVE TO MAILBOX # ***** 353 *****
5005 013724 005242                     INC -(R2) ; SET MSGTYP TO FATAL ERROR
5006 013726 000000                     HALT ; RESULT OF MODE 7 INSTS INCORRECT
                                           ; OR SEQUENCE ERROR

```

```

5008                                     ;
5009                                     ;
5010                                     ; THIS TEST VERIFIES THE ROTATE MODE 0 INSTRUCTIONS.
5011 ; RO IS LOADED WITH A DATA PATTERN, THE C-BIT IS LOADED, AND
5012 ; AN ROL INSTRUCTION IS EXECUTED WITH MODE 0. THE OPERATION IS CHECKED
5013 ; BY TESTING THE RESULTING DATA AND THE STATE OF THE C AND V BITS.
5014 ; NEXT, THE SAME PROCEDURE IS EXECUTED TO TEST MODE 0 BYTE INSTRUCTIONS.
5015

```

5016 TEST 161 TEST ROTATE INSTRUCTIONS OF MODE 0

```

5017 TST161: INC (R2) ; UPDATE TEST NUMBER
5018 CMP #161, (R2) ; SEQUENCE ERROR?
5019 013730 005212 000161 BNE TST162-10 ; BR TO ERROR HALT ON SEQ ERROR
5020 013732 022712 000161 MOV #125252, RO ; INITIALIZE DATA
5021 013736 001026 125252 SEC ; SET C-BIT
5022 013740 012700 125252 ROL RO ; TRY ROL W/ MODE 0
5023 013744 000261 SEC ; CC=0011
5024 013746 006100 ROL RO
5025 013750 102004 BVC ROTOA
5026 013752 103003 BCC ROTOA
5027 013754 022700 052525 CMP #052525, RO ; CHECK DATA
5028 013760 001404 BEQ ROTOB

```

```

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

```

```

5029 ROTOA:
5030
5031
5032 ROTOB:
5033 013762 012742 000354 MOV #354, -(R2) ; MOVE TO MAILBOX # ***** 354 *****
5034 013762 012742 000354 INC -(R2) ; SET MSGTYP TO FATAL ERROR
5035 013766 005242 000354 HALT ; ROL MODE 0 FAILED
5036 013770 000000 125252 ROTOB: MOV #125252, RO ; INITIALIZE DATA
5037 013772 012700 125252 SEC ; SET C-BIT
5038 013776 000261 125125 ROLB RO ; TRY ROL W/ MODE 0 EVEN BYTE
5039 014000 106100 125125 BVC ROTOC ; CC=0011
5040 014002 102004 125125 BCC ROTOC
5041 014004 103003 125125 CMP #125125, RO ; CHECK DATA
5042 014006 022700 125125 BEQ TST162

```

```

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

```

```

5043 ROTOC:
5044
5045
5046
5047
5048 014014 012742 000355 MOV #355, -(R2) ; MOVE TO MAILBOX # ***** 355 *****
5049 014014 012742 000355 INC -(R2) ; SET MSGTYP TO FATAL ERROR
5050 014020 005242 000355 HALT ; ROLB MODE 0 FAILED
5051 014022 000000 ; OR SEQUENCE ERROR
5052

```



```

5053
5054
5055
5056
5057
5058
5059
5060
5061
5062
5063
5064
5065
5066 014024 005212
5067 014026 022712 000162
5068 014032 001051
5069 014034 005000
5070 014036 012710 052525
5071 014042 000241
5072 014044 006110
5073 014046 102005
5074 014050 103404
5075 014052 023727 000000 125252
5076 014060 001404
5077
5078
5079
5080
5081 014062
5082 014062 012742 000356
5083 014066 005242
5084 014070 000000
5085 014072 000261
5086 014074 012710 125252
5087 014100 106110
5088 014102 102005
5089 014104 103004
5090 014106 022737 125125 000000
5091 014114 001404
5092
5093
5094
5095
5096 014116
5097 014116 012742 000357
5098 014122 005242
5099 014124 000000
5100 014126 012710 125252
5101 014132 005000
5102 014134 005200
5103 014136 000261
5104 014140 106110
5105 014142 102005
5106 014144 103004
5107 014146 022737 052652 000000
5108 014154 001404

```

```

*****
: THIS TEST VERIFIES THE ROTATE MODE 1 INSTRUCTIONS.
: THE DATA TO BE ROTATED IS IN LOC 0. R0 IS USED AS THE
: ADDRESSING REGISTER. THE C-BIT IS LOADED AND AN ROL IS EXECUTED.
: THE RESULTS ARE CHECKED BY COMPARING THE DATA RESULTS AND TESTING
: THE C AND V BITS. THIS PROCEDURE IS THEN REPEATED TWICE MORE
: TO TEST THE BYTE ROTATES. FIRST ON BYTE 0, THEN ON BYTE 1.
*****
: TEST 162 TEST ROTATE INSTRUCTIONS W/ MODE 1
*****
TST162: INC (R2) ;UPDATE TEST NUMBER
: CMP #162,(R2) ;SEQUENCE ERROR?
: BNE TST163-10 ;BR TO ERROR HALT ON SEQ ERROR
: CLR R0 ;POINT TO LOC. 0
: MOV #52525,(R0) ;INITIALIZE DATA
: CLC ;CLEAR C-BIT
: ROL (R0) ;TRY ROL W/ MODE 1
: BVC ROT1A ;CC=1010
: BCS ROT1A
: CMP #0,#125252 ;CHECK RESULT
: BEQ ROT1B
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 765 <====
:
ROT1A: MOV #356,-(R2) ;MOVE TO MAILBOX # ***** 356 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;ROL MODE 1 FAILED
:
ROT1B: SEC
: MOV #125252,(R0) ;INITIALIZE DATA
: ROLB (R0) ;TRY ROLB W/ MODE 1 EVEN BYTE
: BVC ROT1C ;CC=1011
: BCC ROT1C
: CMP #125125,#0 ;TEST RESULT
: BEQ ROT1D
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 747 <====
:
ROT1C: MOV #357,-(R2) ;MOVE TO MAILBOX # ***** 357 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;ROLB W/ MODE 1 EVEN BYTE FAILED
:
ROT1D: MOV #125252,(R0)
: CLR R0 ;POINT TO ODD BYTE
: INC R0
: SEC ;SET C-BIT
: ROLB (R0) ;TRY ROLB W/ MODE 1 ODD BYTE
: BVC ROT1E ;CC=0011
: BCC ROT1E
: CMP #052652,#0 ;CHECK DATA
: BEQ TST163

```


: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 727 <====

5109
5110
5111
5112
5113 014156
5114 014156 012742 000360
5115 014162 005242
5116 014164 000000
5117
5118
5119
5120
5121
5122
5123
5124
5125
5126
5127
5128

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

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

: TEST 163 TEST ROTATE INSTRUCTIONS W/ MODE 2

5129 014166 005212
5130 014170 022712 000163
5131 014174 001057
5132 014176 005000
5133 014200 012710 173737
5134 014204 000241
5135 014206 006120
5136 014210 103007
5137 014212 022737 167676 000000
5138 014220 001003
5139 014222 005300
5140 014224 005300
5141 014226 001404
5142
5143
5144
5145
5146 014230
5147 014230 012742 000361
5148 014234 005242
5149 014236 000000
5150 014240 005000
5151 014242 012710 004040
5152 014246 000241
5153 014250 106120
5154 014252 103406
5155 014254 022737 004100 000000
5156 014262 001002
5157 014264 005300
5158 014266 001404
5159
5160
5161
5162
5163 014270
5164 014270 012742 000362

TST163: INC (R2) ; UPDATE TEST NUMBER
CMP #163, (R2) ; SEQUENCE ERROR?
BNE TST164-10 ; BR TO ERROR HALT ON SEQ ERROR
CLR RO ; POINT TO LOC 0
MOV #173737, (RO) ; INITIALIZE DATA
CLC ; CLEAR C-BIT
ROL (RO)+ ; TRY ROL W/ MODE 2
BCC ROT2A ; CHECK C-BIT
CMP #167676, @#0 ; CHECK DATA
BNE ROT2A ; BRANCH IF RESULT INCORRECT
DEC RO ; TEST RO
DEC RO
BEQ ROT2B

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

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

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

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

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

5187
5188
5189
5190
5191
5192
5193
5194
5195
5196
5197
5198
5199
5200
5201
5202
5203
5204
5205
5206
5207
5208
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220
5221
5222
5223
5224
5225
5226
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242

014344 005212
014346 022712 000164
014352 001051
014354 012737 052525 000000
014362 000261
014364 006137 000000
014370 103404
014372 022737 125253 000000
014400 001404

014402
014402 012742 000364
014406 005242
014410 000000
014412 012737 125252 000000
014420 000241
014422 106137 000000
014426 103004
014430 023727 000000 125124
014436 001404

014440
014440 012742 000365
014444 005242
014446 000000
014450 012737 125252 000000
014456 000261
014460 106137 000001
014464 103004
014466 022737 052652 000000
014474 001404

014476
014476 012742 000366
014502 005242
014504 000000

```
*****
:
: THIS TEST VERIFIES MODE 3 ROTATE INSTRUCTIONS.
: THIS TEST USES THE SAME PROCEDURES AS IN THE OTHER ROTATE
: TESTS. THE DATA IS STORED IN LOC. 0 AND IS ADDRESSED USING
: MODE 37. BYTE ADDRESSING IS ALSO CHECKED FOR EVEN AND ODD BYTES.
:
: *****
: TEST 164 TEST ROTATE INSTRUCTIONS /W MODE 3
: *****
TST164: INC (R2) ;UPDATE TEST NUMBER
        CMP #164,(R2) ;SEQUENCE ERROR?
        BNE TST165-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #52525,a#0 ;INITIALIZE DATA IN LOC 0
        SEC ;SET C-BIT
        ROL a#0 ;TRO ROL W/ MODE 3
        BCS ROT3A ;CHECK C-BIT
        CMP #125253,a#0 ;CHECK DATA
        BEQ ROT3B
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 765 <====
;
; ***** 364 *****
ROT3A: MOV #364,-(R2) ;MOVE TO MAILBOX # ***** 364 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;ROL W/ MODE 3 FAILED
ROT3B: MOV #125252,a#0 ;INITIALIZE DATA
        CLC ;CLEAR C-BIT
        ROLB a#0 ;TRY ROL W/ MODE 3 EVEN BYTE
        BCC ROT3C ;CHECK C-BIT
        CMP a#0,#125124 ;CHECK DATA
        BEQ ROT3D
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 746 <====
;
; ***** 365 *****
ROT3C: MOV #365,-(R2) ;MOVE TO MAILBOX # ***** 365 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;ROL W/ MODE 3 EVEN BYTE FAILED
ROT3D: MOV #125252,a#0 ;INITIALIZE DATA IN LOC. 0
        SEC ;SET C-BIT
        ROLB a#1 ;TRY ROL W/ MODE 3 ODD BYTE
        BCC ROT3E ;CHECK C-BIT
        CMP #052652,a#0 ;CHECK DATA
        BEQ TST165
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 727 <====
;
; ***** 366 *****
ROT3E: MOV #366,-(R2) ;MOVE TO MAILBOX # ***** 366 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;ROL W/ MODE 3 ODD BYTE FAILED
```


; OR SEQUENCE ERROR

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

TEST 165 TEST MODE 4 W/ ROTATE INSTRUCTIONS

5243
5244
5245
5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256 014506 005212
5257 014510 022712 000165
5258 014514 001016
5259 014516 012737 070707 000000
5260 014524 012700 000002
5261 014530 000261
5262 014532 006140
5263 014534 103406
5264 014536 022737 161617 000000
5265 014544 001002
5266 014546 005700
5267 014550 001404

TST165: INC (R2) ; UPDATE TEST NUMBER
CMP #165,(R2) ; SEQUENCE ERROR?
BNE TST166-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #070707,R#0 ; INITIALIZE DATA IN LOC. 0
MOV #2,RO ; INITIALIZE RO AS POINTER
SEC ; SET C-BIT
ROL -(RO) ; TRY ROL W/ MODE 4
BCS ROT4 ; CHECK C-BIT
CMP #161617,R#0 ; CHECK DATA
BNE ROT4 ; BRANCH IF DATA INCORRECT
TST RO ; CHECK MODE 4 REGISTER
BEQ TST166

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

5272 014552
5273 014552 012742 000367
5274 014556 005242
5275 014560 000000
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287
5288
5289
5290

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

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

TEST 166 TEST MODE 5 W/ ROTATE INSTRUCTIONS

5291 014562 005212
5292 014564 022712 000166
5293 014570 001021
5294 014572 012737 014644 000000
5295 014600 012700 000002
5296 014604 012767 107070 000032
5297 014612 000241
5298 014614 006150

TST166: INC (R2) ; UPDATE TEST NUMBER
CMP #166,(R2) ; SEQUENCE ERROR?
BNE ROT5 ; BR TO ERROR HALT ON SEQ ERROR
MOV #ROTX,R#0 ; MOVE POINTER TO LOC. 0
MOV #2,RO ; SET MODE 5 REG. TO LOC. 0
MOV #107070,ROTX ; INITIALIZE DATA
CLC ; CLEAR C-BIT
ROL R#-(RO) ; TRY ROL W/ MODE 5

5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365
5366
5367
5368
5369
5370
5371
5372
5373
5374
5375
5376
5377
5378
5379
5380
5381
5382
5383
5384
5385
5386
5387
5388
5389
5390
5391
5392
5393
5394
5395
5396
5397
5398

014714 005212
014716 022712 000170
014722 001016
014724 012737 052525 014644
014732 012737 014644 014770
014740 000241
014742 006177 000022
014746 103404
014750 023727 014644 125252
014756 001405

014760
014760 012742 000372
014764 005242
014766 000000

014770 000000

014772 005212
014774 022712 000171
015000 001013
015002 012700 177400
015006 000300
015010 100404

015012 012742 000373
015016 005242
015020 000000

```
*****
: THIS TEST VERIFIES MODE 7 ROTATE INSTRUCTIONS.
: THE DATA IS SET IN LOC. ROTX. (SEE PREVIOUS TEST). THE ROL INSTRUCTION
: ADDRESSES IT INDIRECTLY USING MODE 7 AND INDIRECT ADDRESS LOCATION
: (ROTXAD) FOLLOWING THE TEST CODE.
*****
: TEST 170 TEST MODE 7 W/ ROTATE INSTRUCTIONS
*****
TST170: INC (R2) ; UPDATE TEST NUMBER
        CMP #170,(R2) ; SEQUENCE ERROR?
        BNE ROT7 ; BR TO ERROR HALT ON SEQ ERROR
        MOV #52525,@#ROTX ; INITIALIZE DATA
        MOV #ROTX,@#ROTXAD ; INITIALIZE ADDRESS POINTER
        CLC ; CLEAR C-BIT
        ROL @ROTXAD ; TRY ROL W/ MODE 7
        BCS ROT7 ; CHECK C-BIT
        CMP @#ROTX,#125252 ; CHECK DATA
        BEQ TST171
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 762 <====
ROT7: MOV #372,-(R2) ; MOVE TO MAILBOX # ***** 372 *****
      INC -(R2) ; SET MSGTYP TO FATAL ERROR
      HALT ; ROL W/ MODE 7 FAILED
; OR SEQUENCE ERROR
ROTXAD: 0
*****
: THIS TEST VERIFIES MODE 0 SWAB INSTRUCTION. RO 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.
*****
TST171: INC (R2) ; UPDATE TEST NUMBER
        CMP #171,(R2) ; SEQUENCE ERROR?
        BNE TST172-10 ; BR TO ERROR HALT ON SEQ ERROR
        MOV #177400,RO ; MOVE TEST PATTERN TO RO
        SWAB RO ; 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/ 774 <====
        MOV #373,-(R2) ; MOVE TO MAILBOX # ***** 373 *****
        INC -(R2) ; SET MSGTYP TO FATAL ERROR
        HALT ; SWAB DID NOT SET CC'S CORRECT
```



```

5399 015022 022700 000377 SBO: CMP #377,R0 ;CHECK RESULT
5400 015026 001404 BEQ TST172 ;
5401 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5402 ; CONDITIONAL BRANCH INST. AND <====
5403 ; REPLACE THE MOVE INSTRUCTION <====
5404 ; WHICH FOLLOWS W/ 765 <====
5405 015030 012742 000374 MOV #374,-(R2) ;MOVE TO MAILBOX # ***** 374 *****
5406 015034 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5407 015036 000000 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
*****

```

```

5420 015040 005212 TST172: INC (R2) ;UPDATE TEST NUMBER
5421 015042 022712 000172 CMP #172,(R2) ;SEQUENCE ERROR?
5422 015046 001011 BNE TST173-10 ;BR TO ERROR HALT ON SEQ ERROR
5423 015050 012737 125652 000000 MOV #125652,0#0 ;MOVE TEST PATTERN TO LOC. 0
5424 015056 005000 CLR R0 ;R0=0
5425 015060 000310 SWAB (R0) ;TRY SWAB MODE 1
5426 015062 022737 125253 000000 CMP #125253,0#0 ;CHECK RESULT
5427 015070 001404 BEQ TST173 ;
5428 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5429 ; CONDITIONAL BRANCH INST. AND <====
5430 ; REPLACE THE MOVE INSTRUCTION <====
5431 ; WHICH FOLLOWS W/ 767 <====
5432 015072 012742 000375 MOV #375,-(R2) ;MOVE TO MAILBOX # ***** 375 *****
5433 015076 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
5434 015100 000000 HALT ;RESULT OF SWAB MODE 1 FAILED
; OR SEQUENCE ERROR

```


5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469
5470
5471
5472
5473
5474
5475
5476
5477
5478
5479
5480
5481
5482
5483
5484
5485
5486
5487
5488
5489
5490
5491

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

015102 005212
015104 022712 000173
015110 001020
015112 012737 125152 000000
015120 005000
015122 000320
015124 022737 065252 000000
015132 001404

TST173: INC (R2) ;UPDATE TEST NUMBER
CMP #173,(R2) ;SEQUENCE ERROR?
BNE TST174-10 ;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/ 767 <====

015134 012742 000376
015140 005242
015142 000000
015144 162700 000002
015150 001404

MOV #376,-(R2) ;MOVE TO MAILBOX # ***** 376 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULT OF SWAB MODE 0 FAILED
SB2: SUB #2,R0 ;CHECK EFFECT OF REG.
BEQ TST174

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

015152 012742 000377
015156 005242
015160 000000

MOV #377,-(R2) ;MOVE TO MAILBOX # ***** 377 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;REGISTER VALUE INCORRECT
OR SEQUENCE ERROR

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

TEST 174 TEST MODE 3 W/SWAB INST.

015162 005212
015164 022712 000174
015170 001011
015172 012737 000377 000000
015200 000337 000000
015204 022737 177400 000000
015212 001404

TST174: INC (R2) ;UPDATE TEST NUMBER
CMP #174,(R2) ;SEQUENCE ERROR?
BNE TST175-10 ;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 TST175

5493			
5494			
5495			
5496	015214	012742	000400
5497	015220	005242	
5498	015222	000000	
5499			

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

```

: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
:           CONDITIONAL BRANCH INST. AND <====
:           REPLACE THE MOVE INSTRUCTION <====
:           WHICH FOLLOWS W/ 767 <====
: MOVE TO MAILBOX # ***** 400 *****
: SET MSGTYP TO FATAL ERROR
: RESULT OF SWAB INCORRECT
: OR SEQUENCE ERROR

```


5500
5501
5502
5503
5504
5505
5506
5507
5508
5509
5510
5511
5512
5513
5514
5515
5516
5517
5518
5519
5520
5521
5522
5523
5524
5525
5526
5527
5528
5529
5530
5531
5532
5533
5534
5535
5536
5537

015224 005212
015226 022712 000175
015232 001020
015234 012737 125652 000000
015242 012700 000002
015246 000340
015250 022737 125253 000000
015256 001404

015260 012742 000401
015264 005242
015266 000000
015270 005700
015272 001404

015274 012742 000402
015300 005242
015302 000000

```
*****
: THIS TEST VERIFIES MODE 4 SWAB INSTRUCTIONS. THE DATA
: IS MOVED TO LOC 0. R0 IS SET TO 2 AND USED AS THE MODE 4 ADDRESSING
: REGISTER. THE DATA IS CHECKED WITH A COMPARE AND R0 IS CHECKED
: FOR PROPER DECREMENTING.
*****
: TEST 175 TEST MODE 4 W/ SWAB INST
*****
TST175: INC (R2) ;UPDATE TEST NUMBER
: CMP #175,(R2) ;SEQUENCE ERROR?
: BNE TST176-10 ;BR TO ERROR HALT ON SEQ ERROR
: MOV #125652,R#0 ;MOVE TEST PATTERN TO LOC. 0
: MOV #2,R0 ;SET UP REGISTER POINTER
: SWAB -(R0) ;TRY SWAB MODE 4
: CMP #125253,R#0 ;CHECK RESULT
: BEQ SB4
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 766 <====
: 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 TST176
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 760 <====
: MOV #402,-(R2) ;MOVE TO MAILBOX # ***** 402 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;REGISTER VALUE INCORRECT
: OR SEQUENCE ERROR
```



```

5538
5539
5540
5541
5542
5543
5544
5545
5546
5547
5548
5549
5550
5551 015304 005212
5552 015306 022712 000176
5553 015312 001021
5554 015314 012700 015372
5555 015320 012767 125125 000040
5556 015326 000350
5557 015330 022767 052652 000030
5558 015336 001404
5559
5560
5561
5562
5563 015340 012742 000403
5564 015344 005242
5565 015346 000000
5566 015350 020027 015370
5567 015354 001406
5568
5569
5570
5571
5572 015356
5573 015356 012742 000404
5574 015362 005242
5575 015364 000000
5576
5577 015366 000000
5578 015370 015366
5579

```

```

*****
: THIS TEST VERIFIES MODE 5 SWAB INSTRUCTION. THE TEST USES
: TWO LOCATIONS FOLLOWING THE TEST CODE. SBSX HOLDS THE DATA;
: SBSXAD IS A POINTER TO THE DATA LOCATION. THE DATA IS MOVED TO
: SBSX AND RO IS SET TO TWO PLUS THE ADDRESS OF SBSXAD. FOLLOWING
: THE MODE 5 SWAB SBSX IS CHECKED FOR THE PROPER DATA. RO IS
: CHECKED TO SEE THAT IT WAS DECREMENTED PROPERLY.
*****
: TEST 176 TEST MODE 5 W/ SWAB INST.
*****
TST176: INC (R2) ;UPDATE TEST NUMBER
: CMP #176,(R2) ;SEQUENCE ERROR?
: BNE SBS ;BR TO ERROR HALT ON SEQ ERROR
: MOV #SBSXAD+2,RO ;SET UP POINTER TO WORK LOCATION
: MOV #125125,SBSX ;MOVE PATTERN TO WORK LOCATION
: SWAB @-(RO) ;TRY SWAB MODE 5
: CMP #52652,SBSX ;CHECK RESULT
: BEQ SBSA
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 766 <====
: MOV #403,-(R2) ;MOVE TO MAILBOX # ***** 403 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;RESULT OF SWAB INCORRECT
SBSA: CMP RO,#SBSXAD ;CHECK RESULT OF REG.
: BEQ TST177
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 757 <====
:
SBS: MOV #404,-(R2) ;MOVE TO MAILBOX # ***** 404 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;REGISTER VALUE INCORRECT
: OR SEQUENCE ERROR
SBSX: 0 ;WORK LOCATION
SBSXAD: SBSX

```



```

5580
5581
5582
5583
5584
5585
5586
5587
5588
5589
5590
5591
5592
5593 015372 005212
5594 015374 022712 000177
5595 015400 001013
5596 015402 012767 125125 000030
5597 015410 012700 015432
5598 015414 000360 000006
5599 015420 022760 052652 000006
5600 015426 001405
5601
5602
5603
5604
5605 015430
5606 015430 012742 000405
5607 015434 005242
5608 015436 000000
5609
5610 015440 000000
5611

```

```

:*****
:
:      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.
:*****
TST177: INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #177,(R2)    ;SEQUENCE ERROR?
        BNE     SB6          ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #125125,SB6X  ;MOVE PATTERN TO WORK LOCATION
        MOV     #SB6X-6,R0   ;MOVE OFFSET POINTER TO R0
        SWAB   6(R0)        ;TRY SWAB W/ MODE 6
        CMP     #52652,6(R0) ;CHECK RESULT
        BEQ    TST200
:
:      TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
:      CONDITIONAL BRANCH INST. AND <====
:      REPLACE THE MOVE INSTRUCTION <====
:      WHICH FOLLOWS W/ 765 <====
:
SB6:    MOV     #405,-(R2)    ;MOVE TO MAILBOX # ***** 405 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT
:
SB6X:   0
:WORK LOCATION

```


5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626 015442 005212
5627 015444 022712 000200
5628 015450 001013
5629 015452 012767 177400 000030
5630 015460 012700 015420
5631 015464 000370 000072
5632 015470 027027 000072 000377
5633 015476 001406
5634
5635
5636
5637
5638 015500
5639 015500 012742 000406
5640 015504 005242
5641 015506 000000
5642
5643 015510 000000
5644 015512 015510
5645

```

*****
: 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. RD IS LOADED WITH 72 LESS THAN THE ADDRESS
: OF THE ADDRESS POINTER. THE DATA IS SWAB'ED USING A MODE 7
: INSTRUCTION WITH AN OFFSET OF +72. THE DATA IS VERIFIED WITH A
: COMPARE.
*****
: TEST 200 TEST MODE 7 W/ SWAB INST.
*****
TST200: INC (R2) ;UPDATE TEST NUMBER
: CMP #200,(R2) ;SEQUENCE ERROR?
: BNE SB7 ;BR TO ERROR HALT ON SEQ ERROR
: MOV #177400,SB7X ;MOVE PATTERN TO WORK LOCATION
: MOV #SB7XAD-72,RD ;MOVE OFFSET POINTER TO RD
: SWAB @72(RD) ;TRY SWAB MODE 7
: CMP @72(RD),#377 ;CHECK RESULTS
: BEQ TST201
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 765 <====
:
SB7: MOV #406,-(R2) ;MOVE TO MAILBOX # ***** 406 *****
: INC -(R2) ;SET MSGTYP TO FATAL ERROR
: HALT ;RESULT OF SWAB INCORRECT
: OR SEQUENCE ERROR
SB7X: 0 ;WORK LOCATION
SB7XAD: SB7X ;POINTER TO WORK LOCATION

```


5646
5647
5648
5649
5650
5651
5652
5653
5654
5655
5656
5657
5658
5659
5660
5661
5662
5663
5664
5665
5666
5667
5668
5669
5670
5671
5672
5673
5674
5675
5676
5677
5678
5679
5680
5681
5682
5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701

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
BLOCK 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

```

TST201: 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/ 770 <====
        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 JMPS 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/ 760 <====

```

```

015514 005212
015516 022712 000201
015522 001150
015524 005067 000326
015530 012700 015610
015534 000110
015536 022700 015540
015542 001404

015544 012742 000407
015550 005242
015552 000000
015554 026727 000276 000001
015562 001404

```


5702	015564	012742	000410		MOV	#410, -(R2)	; MOVE TO MAILBOX # ***** 410 *****
5703	015570	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
5704	015572	000000			HALT		; SHOULD BE HERE FROM JMP MODE 2 ONLY
5705	015574	012700	015606	JMP3B:	MOV	#IIMP4, R0	; POINT R0 TO INDIRECT JMP ADDR.
5706	015600	005267	000252		INC	JMPSEQ	; UPDATE SEQUENCE CHECKER
5707	015604	000130			JMP	2(R0)+	; TRY JMP MODE 3
5708	015606	015640		IIMP4:	JMP4		; ADDRESS INDIRECT JUMP
5709							
5710	015610	005767	000242	JMP2:	TST	JMPSEQ	; CHECK THAT JMPs ARE IN SEQUENCE: JMPSEQ=0?
5711	015614	001404			BEQ	JMP2A	
5712							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5713							CONDITIONAL BRANCH INST. AND <====
5714							REPLACE THE MOVE INSTRUCTION <====
5715							WHICH FOLLOWS W/ 743 <====
5716	015616	012742	000411		MOV	#411, -(R2)	; MOVE TO MAILBOX # ***** 411 *****
5717	015622	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
5718	015624	000000			HALT		; SHOULD BE HERE FROM JMP MODE 1 ONLY
5719	015626	005267	000224	JMP2A:	INC	JMPSEQ	; UPDATE SEQUENCE CHECKER
5720	015632	012700	015536		MOV	#JMP3, R0	; SET R0=JUMP TARGET
5721	015636	000120			JMP	(R0)+	; TRY A JUMP MODE 2 TO "JMP3"
5722	015640	022700	015610	JMP4:	CMP	#IIMP4+2, R0	; CHECK RESULT OF REGISTER IN MODE 3 JUMP
5723	015644	001404			BEQ	JMP4A	
5724							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5725							CONDITIONAL BRANCH INST. AND <====
5726							REPLACE THE MOVE INSTRUCTION <====
5727							WHICH FOLLOWS W/ 727 <====
5728	015646	012742	000412		MOV	#412, -(R2)	; MOVE TO MAILBOX # ***** 412 *****
5729	015652	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
5730	015654	000000			HALT		; REGISTER VALUE AFTER MODE 3 JUMP INCORRECT
5731	015656	022767	000002 000172	JMP4A:	CMP	#2, JMPSEQ	; CHECK JUMP SEQUENCE: JMPSEQ=2?
5732	015664	001404			BEQ	JMP4B	
5733							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5734							CONDITIONAL BRANCH INST. AND <====
5735							REPLACE THE MOVE INSTRUCTION <====
5736							WHICH FOLLOWS W/ 717 <====
5737	015666	012742	000413		MOV	#413, -(R2)	; MOVE TO MAILBOX # ***** 413 *****
5738	015672	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
5739	015674	000000			HALT		; SHOULD BE ONLY FROM MODE 3 JUMP
5740	015676	012700	015746	JMP4B:	MOV	#JMP5+2, R0	; SET UP POINTER TO JUMP TARGET
5741	015702	005267	000150		INC	JMPSEQ	; UPDATE SEQUENCE CHECKER
5742	015706	000140			JMP	-(R0)	; TRY JUMP MODE 4 TO "JMP4"
5743							
5744	015710	022767	000004 000140	JMP6:	CMP	#4, JMPSEQ	; CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=4?
5745	015716	001404			BEQ	JMP6A	
5746							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
5747							CONDITIONAL BRANCH INST. AND <====
5748							REPLACE THE MOVE INSTRUCTION <====
5749							WHICH FOLLOWS W/ 702 <====
5750	015720	012742	000414		MOV	#414, -(R2)	; MOVE TO MAILBOX # ***** 414 *****
5751	015724	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
5752	015726	000000			HALT		; SHOULD BE HERE ONLY FROM MODE 5 JUMP
5753	015730	012700	016376	JMP6A:	MOV	#JMP7+376, R0	; SET UP OFFSET POINTER TO JUMP TARGET
5754	015734	005267	000116		INC	JMPSEQ	; UPDATE JUMP SEQUENCE
5755	015740	000160	177402		JMP	-376(R0)	; TRY MODE 6 JUMP
5756							
5757	015744	022767	000003 000104	JMP5:	CMP	#3, JMPSEQ	; CHECK THAT JUMPS ARE IN SEQUENCE: JMPSEQ=3?

5796
5797
5798
5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838
5839
5840
5841
5842
5843
5844
5845
5846
5847
5848
5849
5850
5851

016060 005212
016062 022712 000202
016066 001001
016070 000402
016072 000137 016526
016076 012706 000500
016102 012700 016210
016106 005037 016506
016112 005001
016114 005101
016116 004110
016120
016120 012742 000420
016124 005242
016126 000000
016130 022737 000001 016506
016136 001014
016140 020127 016272
016144 001011
016146 022706 000476
016152 001006
016154 022716 125252
016160 001003
016162 022700 016132
016166 001404
016170
016170 012742 000421
016174 005242

```
*****
THIS TEST VERIFIES ALL LEGAL MODES OF THE JSR INSTRUCTION.
THE CONCEPT OF LEAP FROGGING AND SEQUENCE CHECKING (JSRSEQ) IS
IDENTICAL TO THAT USED IN JMP TEST (SEE PREVIOUS TEST). EACH
BLOCK OF CODE VERIFIES THE PREVIOUS JSR BY CHECKING THE SEQUENCE,
CHECKING THAT THE PC WAS SAVED IN THE SPECIFIED REGISTER, CHECKING
THAT THE SP WAS DECREMENTED, CHECKING THAT THE REGISTER WAS
SAVED ON THE STACK, AND FINALLY CHECKING THAT ANY MODE ADDRESS
REGISTER ALTERATIONS (E.G. INCREMENT REGISTER IN MODE 2) WERE
SUCCESSFUL. R1 IS USED AS THE REGISTER IN ALL JSR INSTRUCTIONS.
IF A FAILURE OCCURS, THE SEQUENCE CHECKER WILL ASSIST IN
DETERMINING JUST WHICH MODE FAILED. IF THE SEQUENCE IS CORRECT
THEN THE ERROR DETECTED WAS A FUNCTIONAL FAILURE (E.G., INCORRECT
REGISTER SAVED).
*****
TEST 202 TEST JSR INSTRUCTION W/ ALL MODES
*****
TST202: INC (R2) ;UPDATE TEST NUMBER
        CMP #202,(R2) ;SEQUENCE ERROR?
        BNE JSR0 ;BR TO ERROR HALT ON SEQ ERROR
        BR JSR1
JSR0: JMP @#JSRCK1
JSR1: MOV #STBOT,R6 ;SET STACK POINTER
      MOV #JSR2,R0 ;SET TARGET ADDRESS
      CLR @#JSRSEQ ;INITIALIZE SEQUENCE CHECKER
      CLR R1 ;INITIALIZE R1
      COM R1
      JSR R1,(R0) ;TRY JSR MODE 1
      ; TO SCOPE: REPLACE THE MOVE INSTRUCTION <====
      ; FOLLOWING W/ 774 <====
JSR1A: MOV #420,-(R2) ;MOVE TO MAILBOX # ***** 420 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
      HALT ;JSR MODE 1 FAILED
JSR3: CMP #1,@#JSRSEQ ;CHECK SEQUENCE: JSRSEQ=1?
      BNE JSR3A ;BRANCH IF OUT OF SEQUENCE
      CMP R1,#JSR4 ;PROPER PC SAVED?
      BNE JSR3A ;BRANCH IF PC WRONG
      CMP #STBOT-2,R6 ;STACK POINTER DECREMENTED?
      BNE JSR3A ;BRANCH IF SP WRONG
      CMP #125252,(R6) ;REG SAVED ON STACK?
      BNE JSR3A ;BRANCH IF REG. NOT SAVED
      CMP #JSR3+2,R0 ;MODE 2 INCREMENT CORRECT?
      BEQ JSR3B
      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
      ; CONDITIONAL BRANCH INST. AND <====
      ; REPLACE THE MOVE INSTRUCTION <====
      ; WHICH FOLLOWS W/ 740 <====
JSR3A: MOV #421,-(R2) ;MOVE TO MAILBOX # ***** 421 *****
      INC -(R2) ;SET MSGTYP TO FATAL ERROR
```


9962
9963
9964
9965
9966
9967
9968
9969
9970
9971
9972
9973
9974
9975
9976
9977
9978
9979
9980
9981
9982
9983
9984
9985
9986
9987
9988
9989
9990
9991
9992
9993

016536 005212
016540 022712 000203
016544 001016
016546 012706 000500
016552 012746 052525
016556 012700 016574
016562 000200

016564 012742 000430
016570 005242
016572 000000
016574 022700 052525
016600 001404

016602 012742 000431
016606 005242
016610 000000

```
*****
: THIS TEST VERIFIES THE RTS INSTRUCTION. THE STACK POINTER
: IS INITIALIZED AND A TEST PATTERN STORED ON STACK. RO IS LOADED
: WITH RETURN ADDRESS. AN RTS IS EXECUTED, AND, AT THE TARGET
: ADDRESS, A CHECK IS MADE THAT RO WAS PROPERLY RESTORED FROM THE
: STACK.
*****
: TEST 203 TEST RTS INSTRUCTION
*****
TST203: INC (R2) ; UPDATE TEST NUMBER
: CMP #203,(R2) ; SEQUENCE ERROR?
: BNE TST204-10 ; BR TO ERROR HALT ON SEQ ERROR
: MOV #STBOT,R6 ; INITIALIZE STACK POINTER
: MOV #52525,-(R6) ; INITIALIZE TOP OF STACK
: MOV #RTS1,RO ; INITIALIZE RETURN REGISTER
: RTS RO ; TRY RTS THROUGH RO
: ; TO SCOPE: REPLACE THE MOVE INSTRUCTION <====
: ; FOLLOWING W/ 770 <====
: MOV #430,-(R2) ; MOVE TO MAILBOX # ***** 430 *****
: INC -(R2) ; SET MSGTYP TO FATAL ERROR
: HALT ; RTS FAILED
RTS1: CMP #52525,RO ; CHECK THAT RO RESTORED FROM STACK
: BEQ TST204 ;
: ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: ; CONDITIONAL BRANCH INST. AND <====
: ; REPLACE THE MOVE INSTRUCTION <====
: ; WHICH FOLLOWS W/ 762 <====
: MOV #431,-(R2) ; MOVE TO MAILBOX # ***** 431 *****
: INC -(R2) ; SET MSGTYP TO FATAL ERROR
: HALT ; RTS MALFUNCTIONED
: ; OR SEQUENCE ERROR
```



```

5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011 016612 005212
6012 016614 022712 000204
6013 016620 001022
6014 016622 000277
6015 016624 000251
6016 016626 012700 100000
6017 016632 101402
6018 016634 102401
6019 016636 100404
6020
6021
6022
6023
6024 016640
6025 016640 012742 000432
6026 016644 005242
6027 016646 000000
6028
6029 016650 000277
6030 016652 000244
6031 016654 012700 000000
6032 016660 101002
6033 016662 102401
6034 016664 100004
6035
6036
6037
6038
6039 016666
6040 016666 012742 000433
6041 016672 005242
6042 016674 000000
6043
6044
6045
6046
6047 016676 005212
6048 016700 022712 000205
6049 016704 001024

```

```

*****
THESE NEXT FOUR TESTS VERIFY THE FUNCTIONING OF A GROUP
OF FOUR INSTRUCTIONS. THE GROUP CONSISTS OF THE INSTRUCTIONS:
MOV, BIC, BIT, AND BIS. THESE INSTRUCTIONS ARE SIMILAR IN THE
WAY THEY EFFECT THE C AND V BITS. THEY ALL LEAVE THE V-BIT
CLEAR AND THE C-BIT UNAFFECTED.
THE TEST PROCEDURE IS AS FOLLOWS: THE N, Z, AND V BITS
ARE LOADED WITH THE COMPLEMENT OF THE EXPECTED RESULTS, THE C-BIT
IS LOADED WITH THE DESIRED RESULT. THE INSTRUCTION IS EXECUTED
WITH DIFFERENT DATA PATTERNS AND THE RESULTS ARE VERIFIED WITH
A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS. THE DATA IS CHOSEN
TO PRODUCT ALL POSSIBLE COMBINATIONS OF THE C AND V BITS.
*****
TEST 204 TEST MOV INSTRUCTION
*****
TST204: INC (R2) ;UPDATE TEST NUMBER
CMP #204,(R2) ;SEQUENCE ERROR?
BNE TST205-10 ;BR TO ERROR HALT ON SEQ ERROR
SCC ;CC=0110
+CLN!CLC
MOV #100000,R0 ;CC=1000
BLOS MOV1
BVS MOV1
BMI MOV2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
MOV1: MOV #432,-(R2) ;MOVE TO MAILBOX # ***** 432 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MOV DID NOT SET CC'S CORRECTLY
MOV2: SCC ;CC=1011
CLZ
MOV #0,R0 ;CC=0101
BHI MOV3 ;C OR Z = 0?
BVS MOV3 ;V=1?
BPL TST205
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 756 <====
MOV3: MOV #433,-(R2) ;MOVE TO MAILBOX # ***** 433 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MOV DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR
*****
TEST 205 TEST BIT INSTRUCTION
*****
TST205: INC (R2) ;UPDATE TEST NUMBER
CMP #205,(R2) ;SEQUENCE ERROR?
BNE TST206-10 ;BR TO ERROR HALT ON SEQ ERROR

```



```

6050 016706 012700 100001      MOV      #100001,R0
6051 016712 000277      SCC
6052 016714 000251      +CLN!CLC      ;CC=0110
6053 016716 032700 100000      BIT      #100000,R0      ;CC=1000
6054 016722 101402      BLOS    BIT1
6055 016724 102401      BVS     BIT1
6056 016726 100404      BMI     BIT2
6057
6058
6059
6060
6061 016730      BIT1:
6062 016730 012742 000434      MOV      #434,-(R2)      ;MOVE TO MAILBOX # ***** 434 *****
6063 016734 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6064 016736 000000      HALT
6065
6066 016740 000277      BIT2:      SCC          ;CC=1011
6067 016742 000244      CLZ
6068 016744 032700 077776      BIT      #77776,R0      ;CC=0101
6069 016750 101002      BHI     BIT3
6070 016752 102401      BVS     BIT3
6071 016754 100004      BPL     TST206
6072
6073
6074
6075
6076 016756      BIT3:
6077 016756 012742 000435      MOV      #435,-(R2)      ;MOVE TO MAILBOX # ***** 435 *****
6078 016762 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6079 016764 000000      HALT
6080
6081
6082
6083
6084 016766 005212      TST206: INC      (R2)      ;UPDATE TEST NUMBER
6085 016770 022712 000206      CMP      #206,(R2)      ;SEQUENCE ERROR?
6086 016774 001024      BNE     TST207-10      ;BR TO ERROR HALT ON SEQ ERROR
6087 016776 012700 177777      MOV      #177777,R0
6088 017002 000277      SCC          ;CC=0110
6089 017004 000251      +CLN!CLC
6090 017006 042700 077777      BIC      #77777,R0      ;CC=1000
6091 017012 101402      BLOS    BIC1
6092 017014 102401      BVS     BIC1
6093 017016 100404      BMI     BIC2
6094
6095
6096
6097
6098 017020      BIC1:
6099 017020 012742 000436      MOV      #436,-(R2)      ;MOVE TO MAILBOX # ***** 436 *****
6100 017024 005242      INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6101 017026 000000      HALT
6102 017030 000277      BIC2:      SCC          ;CC=1011
6103 017032 000244      CLZ
6104 017034 042700 100000      BIC      #100000,R0      ;CC=0101
6105 017040 101002      BHI     BIC3

```

```

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

```

```

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

```

```

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

```

```

;*****
;TEST 206 TEST BIC INSTRUCTION
;*****

```


E11

6106	017042	102401		BVS	BIC3		
6107	017044	100004		BPL	TST207		
6108							
6109							
6110							
6111							
6112	017046						
6113	017046	012742	000437	BIC3:	MOV	#437, -(R2)	
6114	017052	005242			INC	-(R2)	
6115	017054	000000			HALT		
6116							
6117							
6118							
6119							

```

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

```

```

; MOVE TO MAILBOX # ***** 437 *****
; SET MSGTYP TO FATAL ERROR
; BIC DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR

```

```

;*****
;TEST 207 TEST BIS INSTRUCTION
;*****

```

6120	017056	005212		TST207:	INC	(R2)	
6121	017060	022712	000207		CMP	#207, (R2)	
6122	017064	001025			BNE	TST210-10	
6123	017066	005000			CLR	RO	
6124	017070	000277			SCC		
6125	017072	000251			+CLN!CLC		
6126	017074	052700	000000		BIS	#0, RO	
6127	017100	103403			BCS	BIS1	
6128	017102	102402			BVS	BIS1	
6129	017104	100401			BMI	BIS1	
6130	017106	001404			BEQ	BIS2	
6131							
6132							
6133							
6134							

```

; UPDATE TEST NUMBER
; SEQUENCE ERROR?
; BR TO ERROR HALT ON SEQ ERROR
; RO=0
; CC=1010
; CC=0100 RO=0

```

```

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

```

6135	017110			BIS1:			
6136	017110	012742	000440		MOV	#440, -(R2)	
6137	017114	005242			INC	-(R2)	
6138	017116	000000			HALT		
6139	017120	000277		BIS2:	SCC		
6140	017122	000250			CLN		
6141	017124	052700	177777		BIS	#177777, RO	
6142	017130	103003			BCC	BIS3	
6143	017132	102402			BVS	BIS3	
6144	017134	001401			BEQ	BIS3	
6145	017136	100404			BMI	TST210	
6146							
6147							
6148							
6149							

```

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

```

6150	017140			BIS3:			
6151	017140	012742	000441		MOV	#441, -(R2)	
6152	017144	005242			INC	-(R2)	
6153	017146	000000			HALT		
6154							

```

; MOVE TO MAILBOX # ***** 441 *****
; SET MSGTYP TO FATAL ERROR
; BIS DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR

```



```

6155
6156
6157
6158
6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170 017150 005212
6171 017152 022712 000210
6172 017156 001037
6173 017160 012700 077777
6174 017164 000257
6175 017166 000264
6176 017170 005200
6177 017172 101402
6178 017174 100001
6179 017176 102404
6180
6181
6182
6183
6184 017200
6185 017200 012742 000442
6186 017204 005242
6187 017206 000000
6188 017210 052700 077777
6189 017214 000261
6190 017216 000244
6191 017220 005200
6192 017222 100403
6193 017224 102402
6194 017226 103001
6195 017230 001404
6196
6197
6198
6199
6200 017232
6201 017232 012742 000443
6202 017236 005242
6203 017240 000000
6204
6205 017242 000277
6206 017244 000241
6207 017246 005200
6208 017250 101402
6209 017252 100401
6210 017254 100004

```

```

*****
THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE INC AND
DEC INSTRUCTIONS. THESE INSTRUCTIONS BOTH EFFECT THE C AND V
BITS THE SAME; THE C-BIT IS LEFT UNCHANGED AND THE V-BIT IS DEPENDENT
UPON THE DATA RESULTS. THE SAME PROCEDURE IS USED. THE CONDITION
CODE BITS ARE INITIALIZED, THE INSTRUCTION IS EXECUTED AND THE
RESULTS ARE VERIFIED WITH A SERIES OF CONDITIONAL BRANCH INSTRUCTIONS.
THIS PROCEDURE IS REPEATED WITH SEVERAL DATA PATTERNS TO PRODUCE
DIFFERENT COMBINATIONS OF THE C AND V BITS.

```

```

*****
TEST 210 TEST INC INSTRUCTION
*****

```

```

TST210: INC (R2) ;UPDATE TEST NUMBER
CMP #210,(R2) ;SEQUENCE ERROR?
BNE TST211-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #077777,R0 ;RO=077777
CCC ;CC=0100
SEZ
INC R0 ;CC=1010 RO=10000
BLOS INC1
BPL INC1
BVS INC2

```

```

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

```

```

INC1: MOV #442,-(R2) ;MOVE TO MAILBOX # ***** 442 *****
INC INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INC DID NOT SET CC'S CORRECTLY
INC2: BIS #77777,R0 ;RO=177777
SEC ;CC=1011
CLZ
INC R0 ;CC=0101 RO=0
BMI INC3
BVS INC3
BCC INC3
BEQ INC4

```

```

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

```

```

INC3: MOV #443,-(R2) ;MOVE TO MAILBOX # ***** 443 *****
INC INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;INC DID NOT SET CC'S CORRECTLY
INC4: SCC ;CC=1110
CLC
INC R0 ;CC=0000 RO=1
BLOS INC5
BMI INC5
BPL TST211

```


6211 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6212 ; CONDITIONAL BRANCH INST. AND <====
6213 ; REPLACE THE MOVE INSTRUCTION <====
6214 ; WHICH FOLLOWS W/ 741 <====

6215 017256 INC5: MOV #444, -(R2) ; MOVE TO MAILBOX # ***** 444 *****
6216 017256 012742 000444 INC -(R2) ; SET MSGTYP TO FATAL ERROR
6217 017262 005242 HALT ; INC DID NOT SET CC'S CORRECTLY
6218 017264 000000 ; OR SEQUENCE ERROR
6219
6220

; TEST 211 TEST DEC INSTRUCTION

6221
6222
6223
6224 017266 005212 TST211: INC (R2) ; UPDATE TEST NUMBER
6225 017270 022712 000211 CMP #211, (R2) ; SEQUENCE ERROR?
6226 017274 001051 BNE TST212-10 ; BR TO ERROR HALT ON SEQ ERROR
6227 017276 012700 000002 MOV #2, R0 ; RO=2
6228 017302 000277 SCC ; CC=1111
6229 017304 005300 DEC R0 ; CC=0001 RO=1
6230 017306 100403 BMI DEC1
6231 017310 001402 BEQ DEC1
6232 017312 102401 BVS DEC1
6233 017314 103404 BCS DEC2

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

6234
6235
6236
6237
6238 017316 DEC1: MOV #445, -(R2) ; MOVE TO MAILBOX # ***** 445 *****
6239 017316 012742 000445 INC -(R2) ; SET MSGTYP TO FATAL ERROR
6240 017322 005242 HALT ; DEC DID NOT SET CC'S CORRECTLY
6241 017324 000000 DEC2: SEC ; CC=1011
6242 017326 000261 CLZ ; CC=0101 RO=0
6243 017330 000244 DEC R0
6244 017332 005300 BHI DEC3
6245 017334 101002 BMI DEC3
6246 017336 100401 BVC DEC4
6247 017340 102004

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

6248
6249
6250
6251
6252 017342 DEC3: MOV #446, -(R2) ; MOVE TO MAILBOX # ***** 446 *****
6253 017342 012742 000446 INC -(R2) ; SET MSGTYP TO FATAL ERROR
6254 017346 005242 HALT ; DEC DID NOT SET CC'S CORRECTLY
6255 017350 000000 DEC4: SCC ; CC=0110
6256 017352 000277 +CLN!CLC ; CC=1000 RO=177777
6257 017354 000251 DEC R0
6258 017356 005300 BLOS DEC5
6259 017360 101402 BVS DEC5
6260 017362 102401 BMI DEC6
6261 017364 100404

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

6262
6263
6264
6265
6266 017366 DEC5:

H11

MAINDEC-11-DFKAA-B 11/34 CPU TEST MACY11 27(732) 01-OCT-76 15:03 PAGE 364
DFKAAB.P11 T211 TEST DEC INSTRUCTION

6267	017366	012742	000447		MOV	#447, -(R2)	; MOVE TO MAILBOX # ***** 447 *****
6268	017372	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
6269	017374	000000			HALT		; DEC DID NOT SET CC'S CORRECTLY
6270	017376	042700	077777	DEC6:	BIC	#77777, R0	; R0=100000
6271	017402	000277			SCC		; CC=0101
6272	017404	000252			+CLN!CLV		
6273	017406	005300			DEC	R0	; CC=1011 R0=77777
6274	017410	100403			BMI	DEC7	; CC=0011
6275	017412	001402			BEQ	DEC7	
6276	017414	102001			BVC	DEC7	
6277	017416	103404			BCS	TST212	
6278							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6279							; CONDITIONAL BRANCH INST. AND <====
6280							; REPLACE THE MOVE INSTRUCTION <====
6281							; WHICH FOLLOWS W/ 727 <====
6282	017420			DEC7:			
6283	017420	012742	000450		MOV	#450, -(R2)	; MOVE TO MAILBOX # ***** 450 *****
6284	017424	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
6285	017426	000000			HALT		; DEC DID NOT SET CC'S CORRECTLY
6286							; OR SEQUENCE ERROR
6287							


```

6288
6289
6290
6291
6292
6293
6294
6295
6296
6297
6298
6299
6300
6301 017430 005212
6302 017432 022712 000212
6303 017436 001007
6304 017440 000277
6305 017442 000244
6306 017444 005000
6307 017446 100403
6308 017450 102402
6309 017452 103401
6310 017454 001404
6311
6312
6313
6314
6315 017456
6316 017456 012742 000451
6317 017462 005242
6318 017464 000000
6319
6320
6321
6322
6323
6324 017466 005212
6325 017470 022712 000213
6326 017474 001022
6327 017476 000277
6328 017500 000244
6329 017502 005700
6330 017504 100403
6331 017506 102402
6332 017510 103401
6333 017512 001404
6334
6335
6336
6337
6338 017514
6339 017514 012742 000452
6340 017520 005242
6341 017522 000000
6342 017524 005300
6343 017526 000277

```

```

*****
THESE NEXT THREE TESTS VERIFY THE FUNCTIONING OF THE CLR,
TST, AND SWAB INSTRUCTIONS. THESE THREE INSTRUCTIONS ALL LEAVE
THE C AND V BITS CLEARED. AGAIN, THE CONDITION CODES ARE PRESET,
THE INSTRUCTION EXECUTED AND THE RESULTS CHECKED WITH CONDITIONAL
BRANCH INSTRUCTIONS. THE PROCEDURE IS REPEATED TO PRODUCE OTHER
COMBINATIONS OF CONDITION CODES.
*****
TEST 212 TEST CLR INSTRUCTION
*****
TST212: INC (R2) ;UPDATE TEST NUMBER
CMP #212,(R2) ;SEQUENCE ERROR?
BNE TST213-10 ;BR TO ERROR HALT ON SEQ ERROR
SCC ;CC=1011
CLZ
CLR RO ;CC=0100 RO=0
BMI CLR1
BVS CLR1
BCS CLR1
BEQ TST213
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
CLR1: MOV #451,-(R2) ;MOVE TO MAILBOX # ***** 451 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;CLR DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR
*****
TEST 213 TEST TST INSTRUCTION
*****
TST213: INC (R2) ;UPDATE TEST NUMBER
CMP #213,(R2) ;SEQUENCE ERROR?
BNE TST214-10 ;BR TO ERROR HALT ON SEQ ERROR
SCC ;CC=1011
CLZ
TST RO ;CC=0100
BMI TEST1
BVS TEST1
BCS TEST1
BEQ TEST2
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
TEST1: MOV #452,-(R2) ;MOVE TO MAILBOX # ***** 452 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;TEST DID NOT SET CC'S CORRECTLY
TEST2: DEC RO ;MAKE RO NEGATIVE
SCC ;CC=0111

```



```

6344 017530 000250          CLN
6345 017532 005700          TST      RO          ;CC=1000
6346 017534 101402          BLOS    TEST3
6347 017536 102401          BVS     TEST3
6348 017540 100404          BMI     TST214
6349                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6350                                     ;          CONDITIONAL BRANCH INST. AND <====
6351                                     ;          REPLACE THE MOVE INSTRUCTION <====
6352                                     ;          WHICH FOLLOWS W/ 756 <====
6353 017542          TEST3:
6354 017542 012742 000453    MOV     #453, -(R2)    ;MOVE TO MAILBOX # ***** 453 *****
6355 017546 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
6356 017550 000000          HALT                    ;TEST DID NOT SET CC'S CORRECTLY
6357                                     ;          OR SEQUENCE ERROR
6358 ;*****
6359 ;TEST 214          TEST SWAB INSTRUCTION
6360 ;*****
6361 017552 005212          TST214: INC     (R2)          ;UPDATE TEST NUMBER
6362 017554 022712 000214    CMP     #214, (R2)     ;SEQUENCE ERROR?
6363 017560 001023          BNE     TST215-10     ;BR TO ERROR HALT ON SEQ ERROR
6364 017562 012700 170000    MOV     #170000, RO    ;RO=170000
6365 017566 000277          SCC                    ;CC=0111
6366 017570 000250          CLN
6367 017572 000300          SWAB    RO          ;CC=1000  RO=360
6368 017574 101402          BLOS    SWB1
6369 017576 102401          BVS     SWB1
6370 017600 100404          BMI     SWB2
6371                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6372                                     ;          CONDITIONAL BRANCH INST. AND <====
6373                                     ;          REPLACE THE MOVE INSTRUCTION <====
6374                                     ;          WHICH FOLLOWS W/ 770 <====
6375 017602          SWB1:
6376 017602 012742 000454    MOV     #454, -(R2)    ;MOVE TO MAILBOX # ***** 454 *****
6377 017606 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
6378 017610 000000          HALT                    ;SWAB DID NOT SET CC'S CORRECTLY
6379 017612 000277          SWB2: SCC                    ;CC=1011
6380 017614 000244          CLZ
6381 017616 000300          SWAB    RO          ;CC=0100  RO=170000
6382 017620 102403          BVS     SWB3
6383 017622 103402          BCS     SWB3
6384 017624 100401          BMI     SWB3
6385 017626 001404          BEQ     TST215
6386                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6387                                     ;          CONDITIONAL BRANCH INST. AND <====
6388                                     ;          REPLACE THE MOVE INSTRUCTION <====
6389                                     ;          WHICH FOLLOWS W/ 755 <====
6390 017630          SWB3:
6391 017630 012742 000455    MOV     #455, -(R2)    ;MOVE TO MAILBOX # ***** 455 *****
6392 017634 005242          INC     -(R2)         ;SET MSGTYP TO FATAL ERROR
6393 017636 000000          HALT

```



```

6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408 017640 005212
6409 017642 022712 000215
6410 017646 001062
6411 017650 012700 040000
6412 017654 000277
6413 017656 062700 030000
6414 017662 101402
6415 017664 102401
6416 017666 100004
6417
6418
6419
6420
6421 017670
6422 017670 012742 000456
6423 017674 005242
6424 017676 000000
6425 017700 000264
6426
6427 017702 062700 010000
6428 017706 101402
6429 017710 102001
6430 017712 100404
6431
6432
6433
6434
6435 017714
6436 017714 012742 000457
6437 017720 005242
6438 017722 000000
6439 017724 000257
6440 017726 000270
6441 017730 062700 100000
6442 017734 101002
6443 017736 102001
6444 017740 100004
6445
6446
6447
6448
6449 017742

```

```

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

```

```

TST215: INC      (R2)          ;UPDATE TEST NUMBER
        CMP      #215,(R2)    ;SEQUENCE ERROR?
        BNE     TST216-10    ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #40000,R0    ;R0=40000
        SCC     ;CC=1111
        ADD     #30000,R0    ;CC=0000  R0=70000
        BLOS   ADD1
        BVS    ADD1
        BPL    ADD2

```

```

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

```

```

ADD1:   MOV     #456,-(R2)    ;MOVE TO MAILBOX # ***** 456 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT                               ;ADD DID NOT SET CC'S CORRECTLY
ADD2:   SEZ     ;CC=0100

```

```

;CC=1010  40=100000

```

```

        ADD     #10000,R0
        BLOS   ADD3
        BVC    ADD3
        BMI    ADD4

```

```

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

```

```

ADD3:   MOV     #457,-(R2)    ;MOVE TO MAILBOX # ***** 457 *****
        INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
        HALT                               ;ADD DID NOT SET CC'S CORRECTLY
ADD4:   CCC     ;CC=1000
        SEN

```

```

        ADD     #100000,R0    ;CC=0111  R0=0
        BHI    ADD5
        BVC    ADD5
        BPL    ADD6

```

```

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

```

```

ADD5:

```



```

6450 017742 012742 000460      MOV    #460,-(R2)      ;MOVE TO MAILBOX # ***** 460 *****
6451 017746 005242      INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6452 017750 000000      HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6453 017752 062700 177777  ADD6:  ADD    #177777,R0  ;CC=1000  RO=177777
6454 017756 101402      BLOS  ADD7
6455 017760 102401      BVS   ADD7
6456 017762 100404      BMI   ADD8
6457                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
6458                                ;          CONDITIONAL BRANCH INST. AND  <====
6459                                ;          REPLACE THE MOVE INSTRUCTION  <====
6460                                ;          WHICH FOLLOWS W/ 732  <====
6461 017764      ADD7:  MOV    #461,-(R2)      ;MOVE TO MAILBOX # ***** 461 *****
6462 017764 012742 000461      INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6463 017770 005242      HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6464 017772 000000      ADD8:  SCC
6465 017774 000277      +CLC!CLZ              ;CC=1010
6466 017776 000245      ADD    #1,R0          ;CC=0101  R=0
6467 020000 062700 000001      BVS   ADD9
6468 020004 102403      BCC   ADD9
6469 020006 103002      BMI   ADD9
6470 020010 100401      BEQ   TST216
6471 020012 001404
6472                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
6473                                ;          CONDITIONAL BRANCH INST. AND  <====
6474                                ;          REPLACE THE MOVE INSTRUCTION  <====
6475                                ;          WHICH FOLLOWS W/ 716  <====
6476 020014      ADD9:  MOV    #462,-(R2)      ;MOVE TO MAILBOX # ***** 462 *****
6477 020014 012742 000462      INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6478 020020 005242      HALT                    ;ADD DID NOT SET CC'S CORRECTLY
6479 020022 000000      ; OR SEQUENCE ERROR
6480
6481 ;*****
6482 ;TEST 216 TEST ADC INSTRUCTION
6483 ;*****
6484 ;*****
6485 020024 005212 000216  TST216: INC    (R2)          ;UPDATE TEST NUMBER
6486 020026 022712      CMP    #216,(R2)      ;SEQUENCE ERROR?
6487 020032 001037      BNE   TST217-10      ;BR TO ERROR HALT ON SEQ ERROR
6488 020034 012700 077777      MOV    #077777,R0
6489 020040 000277      SCC
6490 020042 000252      +CLN!CLV              ;CC=0101
6491 020044 005500      ADC   RO              ;CC=1010
6492 020046 101402      BLOS  ADC1
6493 020050 102001      BVC   ADC1
6494 020052 100404      BMI   ADC2
6495                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
6496                                ;          CONDITIONAL BRANCH INST. AND  <====
6497                                ;          REPLACE THE MOVE INSTRUCTION  <====
6498                                ;          WHICH FOLLOWS W/ 770  <====
6499 020054      ADC1:  MOV    #463,-(R2)      ;MOVE TO MAILBOX # ***** 463 *****
6500 020054 012742 000463      INC    -(R2)          ;SET MSGTYP TO FATAL ERROR
6501 020060 005242      HALT                    ;ADC DID NOT SET CC'S CORRECTLY
6502 020062 000000      ADC2:  BIS    #77777,R0  ;CC=1011
6503 020064 052700 077777      SCC
6504 020070 000277      CLZ
6505 020072 000244

```


M11

6506	020074	005500		ADC	RO		;CC=0101	RO=0	
6507	020076	101002		BHI	ADC3				
6508	020100	102401		BVS	ADC3				
6509	020102	100004		BPL	ADC4				
6510									
6511									
6512									
6513									
6514	020104			ADC3:					
6515	020104	012742	000464		MOV	#464, -(R2)			
6516	020110	005242			INC	-(R2)			
6517	020112	000000			HALT				
6518	020114	000277		ADC4:	SCC				
6519	020116	000245			+CLZ!CLC				
6520	020120	005500			ADC	RO			
6521	020122	102403			BVS	ADC5			
6522	020124	103402			BCS	ADC5			
6523	020126	100401			BMI	ADC5			
6524	020130	001404			BEQ	TST217			
6525									
6526									
6527									
6528									
6529	020132			ADC5:					
6530	020132	012742	000465		MOV	#465, -(R2)			
6531	020136	005242			INC	-(R2)			
6532	020140	000000			HALT				
6533									

; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 754 <====
; MOVE TO MAILBOX # ***** 464 *****
; SET MSGTYP TO FATAL ERROR
; ADC DID NOT SET CC'S CORRECTLY
; CC=0100 ;CC=1010
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 741 <====
; MOVE TO MAILBOX # ***** 465 *****
; SET MSGTYP TO FATAL ERROR
; ADC DID NOT SET CC'S CORRECTLY
; OR SEQUENCE ERROR

6534
6535
6536
6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562
6563
6564
6565
6566
6567
6568
6569
6570
6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589

020142 005212
020144 022712 000217
020150 001042
020152 012700 000001
020156 000277
020160 000251
020162 005400
020164 103003
020166 102402
020170 001401
020172 100404

020174
020174 012742 000466
020200 005242
020202 000000
020204 042700 077777
020210 000257
020212 000264
020214 005400
020216 102003
020220 103002
020222 001401
020224 100404

020226
020226 012742 000467
020232 005242
020234 000000
020236 005000
020240 000277
020242 000244
020244 005400
020246 102403
020250 103402
020252 001001

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

TEST 217 TEST NEG INSTRUCTION

TST217: INC (R2) ;UPDATE TEST NUMBER
CMP #217,(R2) ;SEQUENCE ERROR?
BNE TST220-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #1,R0
SCC ;CC=0110
+CLN!CLC
NEG R0 ;CC=1001 RO=177777
BCC NEG1
BVS NEG1
BEQ NEG1
BMI NEG2

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

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

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

NEG3: MOV #467,-(R2) ;MOVE TO MAILBOX # ***** 467 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;NEG DID NOT SET CC'S CORRECTLY
NEG4: CLR R0
SCC ;CC=1011
CLZ
NEG R0 ;CC=0100 RO=0
BVS NEG5
BCS NEG5
BNE NEG5


```

6590 020254 100004          BPL      TST220
6591
6592
6593
6594
6595 020256          NEG5:
6596 020256 012742 000470      MOV     #470, -(R2)
6597 020262 005242          INC     -(R2)
6598 020264 000000          HALT
6599
6600
6601
6602
6603
6604 020266 005212          *
6605 020270 022712 000220      *TEST 220      TEST CMP INSTRUCTION
6606 020274 001060          *
6607 020276 012700 000005      *
6608 020302 000257          *
6609 020304 000271          *
6610 020306 022700 000005      *
6611 020312 101002          *
6612 020314 102401          *
6613 020316 100004          *
6614
6615
6616
6617
6618 020320          CMP1:
6619 020320 012742 000471      MOV     #471, -(R2)
6620 020324 005242          INC     -(R2)
6621 020326 000000          HALT
6622 020330 012700 100000      CMP2:
6623 020334 000277          MOV     #100000, R0
6624 020336 000242          SCC
6625 020340 020027 077777      CLV
6626 020344 101402          CMP     R0, #77777
6627 020346 102001          BLOS   CMP3
6628 020350 100004          BVC   CMP3
6629
6630
6631
6632
6633 020352          CMP3:
6634 020352 012742 000472      MOV     #472, -(R2)
6635 020356 005242          INC     -(R2)
6636 020360 000000          HALT
6637 020362 052700 040000      CMP4:
6638 020366 000257          BIS     #40000, R0
6639 020370 000264          CCC
6640 020372 022700 040000      SEZ
6641 020376 102003          CMP     #40000, R0
6642 020400 103002          BVC   CMP5
6643 020402 001401          BCC   CMP5
6644 020404 100404          BEQ   CMP5
6645

```

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

: MOVE TO MAILBOX # ***** 470 *****
: SET MSGTYP TO FATAL ERROR
: NEG DID NOT SET CC'S CORRECTLY
: OR SEQUENCE ERROR

: TEST 220 TEST CMP INSTRUCTION

↑ST220: INC (R2) ; UPDATE TEST NUMBER
CMP #220, (R2) ; SEQUENCE ERROR?
BNE TST221-10 ; BR TO ERROR HALT ON SEQ ERROR
MOV #5, R0
CCC ; CC=1010
+SEN!SEC
CMP #5, R0 ; CC=0101
BHI CMP1
BVS CMP1
BPL CMP2

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

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

: CC=1101
: CC=0010

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

: MOVE TO MAILBOX # ***** 472 *****
: SET MSGTYP TO FATAL ERROR
: CMP DID NOT SET CC'S CORRECTLY

: RO=140000
: CC=0100
: CC=1011

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

6692
6693
6694
6695
6696
6697
6698
6699
6700
6701
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747

020506 005212
020510 022712 000222
020514 001055
020516 012700 125252
020522 000257
020524 000271
020526 162700 125252
020532 101002
020534 102401
020536 100004

020540
020540 012742 000476
020544 005242
020546 000000
020550 052700 100000
020554 000277
020556 000242
020560 162700 077777
020564 101402
020566 102001
020570 100004

020572
020572 012742 000477
020576 005242
020600 000000
020602 005100
020604 000277

020606 162700 100000
020612 101402
020614 102401
020616 100004

```

*****
      THESE NEXT TWO TESTS VERIFY THE FUNCTIONING OF THE SUB
AND SBC INSTRUCTIONS. BOTH OF THESE INSTRUCTIONS HANDLE THE
C AND V BITS IDENTICALLY. THE PROCEDURE IS TO PRESET THE CONDITION
CODES, EXECUTE THE INSTRUCTION WITH A PARTICULAR SET OF DATA, AND
THEN CHECK THE RESULTS BY EXECUTING A SERIES OF CONDITIONAL
BRANCHES. THIS PROCEDURE IS REPEATED SEVERAL TIMES WITH DIFFERENT
DATA PATTERNS TO PROVIDE EVERY COMBINATION OF THE C AND V BITS.
*****
TEST 222      TEST SUB INSTRUCTION
*****
TST222: INC      (R2)          ;UPDATE TEST NUMBER
          CMP      #222,(R2)   ;SEQUENCE ERROR?
          BNE     TST223-10    ;BR TO ERROR HALT ON SEQ ERROR
          MOV     #125252,R0
          CCC
          +SEN!SEC             ;CC=1010
          SUB     #125252,R0    ;CC=0101  RO=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/ 767 <====

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
          CLV
          SUB     #77777,R0    ;CC=0010  RO=1
          BLOS   SUB3
          BVC     SUB3
          BPL     SUB4

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

SUB3:    MOV     #477,-(R2)    ;MOVE TO MAILBOX # ***** 477 *****
          INC     -(R2)        ;SET MSGTYP TO FATAL ERROR
          HALT

SUB4:    COM     R0
          SCC
          ;RO=177777
          ;CC=11111

          SUB     #100000,R0    ;CC=0000  RO=77777
          BLOS   SUB5
          BVS     SUB5
          BPL     SUB6

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

```



```

6748                                     ; REPLACE THE MOVE INSTRUCTION <====
6749                                     ; WHICH FOLLOWS W/ 737 <====
6750 020620 SUB5: MOV #500,-(R2) ;MOVE TO MAILBOX # ***** 500 *****
6751 020620 012742 000500 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6752 020624 005242 HALT ;SUB DID NOT SET CC'S CORRECTLY
6753 020626 000000 SUB6: CCC ;CC=0100
6754 020630 000257 SEZ
6755 020632 000264 SUB #140000,R0 ;CC=1011
6756 020634 162700 140000 SUB SUB7
6757 020640 102003 BVC SUB7
6758 020642 103002 BCC SUB7
6759 020644 001401 BEQ SUB7
6760 020646 100404 BMI TST223
6761                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6762                                     ; CONDITIONAL BRANCH INST. AND <====
6763                                     ; REPLACE THE MOVE INSTRUCTION <====
6764                                     ; WHICH FOLLOWS W/ 723 <====
6765 020650 SUB7: MOV #501,-(R2) ;MOVE TO MAILBOX # ***** 501 *****
6766 020650 012742 000501 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6767 020654 005242 HALT
6768 020656 000000
6769
6770 *****
6771 ;TEST 223 TEST SBC INSTRUCTION
6772 *****
6773 020660 005212 TST223: INC (R2) ;UPDATE TEST NUMBER
6774 020662 022712 000223 CMP #223,(R2) ;SEQUENCE ERROR?
6775 020666 001053 BNE TST224-10 ;BR TO ERROR HALT ON SEQ ERROR
6776 020670 012700 000001 MOV #1,R0
6777 020674 000277 SCC ;CC=1011
6778 020676 000244 CLZ
6779 020700 005600 SBC R0 ;CC=0100 R=0
6780 020702 103403 BCS SBC1
6781 020704 102402 BVS SBC1
6782 020706 100401 BMI SBC1
6783 020710 001404 BEQ SBC2
6784                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6785                                     ; CONDITIONAL BRANCH INST. AND <====
6786                                     ; REPLACE THE MOVE INSTRUCTION <====
6787                                     ; WHICH FOLLOWS W/ 767 <====
6788 020712 SBC1: MOV #502,-(R2) ;MOVE TO MAILBOX # ***** 502 *****
6789 020712 012742 000502 INC -(R2) ;SET MSGTYP TO FATAL ERROR
6790 020716 005242 HALT ;SBC DID NOT SET CC'S CORRECTLY
6791 020720 000000 SBC2: SCC ;CC=1010
6792 020722 000277 +CLZ!CLC
6793 020724 000245 SBC R0 ;CC=0100 R=0
6794 020726 005600 BCS SBC3
6795 020730 103403 BVS SBC3
6796 020732 102402 BMI SBC3
6797 020734 100401 BEQ SBC4
6798 020736 001404
6799                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6800                                     ; CONDITIONAL BRANCH INST. AND <====
6801                                     ; REPLACE THE MOVE INSTRUCTION <====
6802                                     ; WHICH FOLLOWS W/ 754 <====
6803 020740 SBC3:

```


6804	020740	012742	000503		MOV	#503, -(R2)	; MOVE TO MAILBOX # ***** 503 *****
6805	020744	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
6806	020746	000000			HALT		; SBC DID NOT SET CC'S CORRECTLY
6807	020750	000277		SBC4:	SCC		; CC=0111
6808	020752	000250			CLN		
6809	020754	005600			SBC	RO	; CC=1001 RO=177777
6810	020756	103003			BCC	SBC5	
6811	020760	102402			BVS	SBC5	
6812	020762	001401			BEQ	SBC5	
6813	020764	100404			BMI	SBC6	
6814							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6815							; CONDITIONAL BRANCH INST. AND <====
6816							; REPLACE THE MOVE INSTRUCTION <====
6817							; WHICH FOLLOWS W/ 741 <====
6818	020766			SBC5:			
6819	020766	012742	000504		MOV	#504, -(R2)	; MOVE TO MAILBOX # ***** 504 *****
6820	020772	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
6821	020774	000000			HALT		; SBC DID NOT SET CC'S CORRECTLY
6822	020776	042700	077777	SBC6:	BIC	#77777, RO	; RO=100000
6823	021002	000277			SCC		; CC=1101
6824	021004	000242			CLV		
6825	021006	005600			SBC	RO	; CC=0010
6826	021010	101402			BLOS	SBC7	
6827	021012	102001			BVC	SBC7	
6828	021014	100004			BPL	TST224	
6829							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6830							; CONDITIONAL BRANCH INST. AND <====
6831							; REPLACE THE MOVE INSTRUCTION <====
6832							; WHICH FOLLOWS W/ 725 <====
6833	021016			SBC7:			
6834	021016	012742	000505		MOV	#505, -(R2)	; MOVE TO MAILBOX # ***** 505 *****
6835	021022	005242			INC	-(R2)	; SET MSGTYP TO FATAL ERROR
6836	021024	000000			HALT		; SBC DID NOT SET CC'S CORRECTLY
6837							; OR SEQUENCE ERROR
6838							

6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852 021026 005212
6853 021030 022712 000224
6854 021034 001053
6855 021036 012700 144000
6856 021042 000257
6857 021044 000266
6858 021046 006100
6859 021050 103003
6860 021052 102402
6861 021054 001401
6862 021056 100404
6863
6864
6865
6866
6867 021060
6868 021060 012742 000506
6869 021064 005242
6870 021066 000000
6871 021070 000277
6872 021072 000243
6873 021074 006100
6874 021076 103003
6875 021100 102002
6876 021102 001401
6877 021104 100004
6878
6879
6880
6881
6882 021106
6883 021106 012742 000507
6884 021112 005242
6885 021114 000000
6886 021116 000277
6887 021120 000250
6888 021122 006100
6889 021124 101402
6890 021126 102401
6891 021130 100004
6892
6893
6894

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

TEST 224 TEST ROL INSTRUCTION

ST224: INC (R2) ;UPDATE TEST NUMBER
CMP #224,(R2) ;SEQUENCE ERROR?
BNE TST225-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #144000,R0 ;RO=144000
CCC ;CC=0110
+SEZ!SEV
ROL R0 ;CC=1001 RO=110000
BCC ROL1
BVS ROL1
BEQ ROL1
BMI ROL2

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

ROL1: MOV #506,-(R2) ;MOVE TO MAILBOX # ***** 506 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT
ROL2: SCC ;CC=1100
+CLV!CLC
ROL R0 ;CC=0011 RO=020000
BCC ROL3
BVC ROL3
BEQ ROL3
BPL ROL4

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

ROL3: MOV #507,-(R2) ;MOVE TO MAILBOX # ***** 507 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT
ROL4: SCC ;ROL DID NOT SET CC'S CORRECTLY
CLN ;CC=0111
ROL R0 ;CC=0000 RO=040001
BLOS ROL5
BVS ROL5
BPL ROL6

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


```

6895                                     ;           WHICH FOLLOWS W/ 742           <====
6896 021132                                ROL5:                                     ;
6897 021132 012742 000510                MOV     #510,-(R2)                       ;MOVE TO MAILBOX # ***** 510 *****
6898 021136 005242                        INC     -(R2)                           ;SET MSGTYP TO FATAL ERROR
6899 021140 000000                        HALT                                     ;ROL DID NOT SET CC'S CORRECTLY
6900 021142 000257                                ROL6:                                     ;CC=0101
6901 021144 000265                        +SEZ!SEC                                ;
6902 021146 006100                        ROL     RO                               ;CC=1010  RO=100003
6903 021150 101405                        BLOS   ROL7
6904 021152 102004                        BVC   ROL7
6905 021154 100003                        BPL   ROL7
6906 021156 022700 100003                CMP     #100003,RO
6907 021162 001404                        BEQ    TST225
6908                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
6909                                     ; CONDITIONAL BRANCH INST. AND           <====
6910                                     ; REPLACE THE MOVE INSTRUCTION         <====
6911                                     ; WHICH FOLLOWS W/ 725                 <====
6912 021164                                ROL7:                                     ;
6913 021164 012742 000511                MOV     #511,-(R2)                       ;MOVE TO MAILBOX # ***** 511 *****
6914 021170 005242                        INC     -(R2)                           ;SET MSGTYP TO FATAL ERROR
6915 021172 000000                        HALT                                     ;ROL MALFUNCTIONED
6916                                     ; OR SEQUENCE ERROR
6917 ;*****
6918 ;TEST 225 TEST ROR INSTRUCTION
6919 ;*****
6920 021174 005212                                TST225: INC (R2)                       ;UPDATE TEST NUMBER
6921 021176 022712 000225                CMP     #225,(R2)                       ;SEQUENCE ERROR?
6922 021202 001051                        BNE    TST226-10                       ;BR TO ERROR HALT ON SEQ ERROR
6923 021204 012700 000023                MOV     #23,RO                          ;RO=23
6924 021210 000277                        SCC                                     ;CC=0111
6925 021212 000250                        CLN
6926 021214 006000                        ROR     RO                               ;CC=1001  RO=100011
6927 021216 102403                        BVS   ROR1
6928 021220 103002                        BCC   ROR1
6929 021222 001401                        BEQ   ROR1
6930 021224 100404                        BMI   ROR2
6931                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
6932                                     ; CONDITIONAL BRANCH INST. AND           <====
6933                                     ; REPLACE THE MOVE INSTRUCTION         <====
6934                                     ; WHICH FOLLOWS W/ 767                 <====
6935 021226                                ROR1:                                     ;
6936 021226 012742 000512                MOV     #512,-(R2)                       ;MOVE TO MAILBOX # ***** 512 *****
6937 021232 005242                        INC     -(R2)                           ;SET MSGTYP TO FATAL ERROR
6938 021234 000000                        HALT                                     ;ROR DID NOT SET CC'S CORRECTLY
6939 021236 000257                                ROR2:                                     ;CC=1100
6940 021240 000274                        +SEN!SEZ                                ;
6941 021242 006000                        ROR     RO                               ;CC=0011  RO=040004
6942 021244 102003                        BVC   ROR3
6943 021246 103002                        BCC   ROR3
6944 021250 001401                        BEQ   ROR3
6945 021252 100004                        BPL   ROR4
6946                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS  <====
6947                                     ; CONDITIONAL BRANCH INST. AND           <====
6948                                     ; REPLACE THE MOVE INSTRUCTION         <====
6949                                     ; WHICH FOLLOWS W/ 754                 <====
6950 021254                                ROR3:

```



```

6951 021254 012742 000513      MOV      #513,-(R2)      ;MOVE TO MAILBOX # ***** 513 *****
6952 021260 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6953 021262 000000              HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6954 021264 000277      ROR4:   SCC              ;CC=1110
6955 021266 000241              CLC
6956 021270 006000              ROR      RO              ;CC=0000  RO=020002
6957 021272 101403              BLOS     ROR5
6958 021274 102402              BVS     ROR5
6959 021276 001401              BEQ     ROR5
6960 021300 100004              BPL     ROR6
6961                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6962                                ;          CONDITIONAL BRANCH INST. AND <====
6963                                ;          REPLACE THE MOVE INSTRUCTION <====
6964                                ;          WHICH FOLLOWS W/ 741 <====
6965 021302 012742 000514      ROR5:   MOV      #514,-(R2) ;MOVE TO MAILBOX # ***** 514 *****
6966 021302 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6967 021306 000000              HALT                    ;ROR DID NOT SET CC'S CORRECTLY
6968 021310 000000              ROR6:   CCC              ;CC=0101
6969 021312 000257              +SEC!SEZ
6970 021314 000265              ROR      RO              ;CC=1010  RO=110001
6971 021316 006000              BLOS     ROR7
6972 021320 101402              BVC     ROR7
6973 021322 102001              BMI     TST226
6974 021324 100404
6975                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6976                                ;          CONDITIONAL BRANCH INST. AND <====
6977                                ;          REPLACE THE MOVE INSTRUCTION <====
6978                                ;          WHICH FOLLOWS W/ 727 <====
6979 021326 012742 000515      ROR7:   MOV      #515,-(R2) ;MOVE TO MAILBOX # ***** 515 *****
6980 021326 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
6981 021332 000000              HALT                    ;ROR DID NOT PRODUCE CORRECT RESULTS
6982 021334 000000
6983                                ; OR SEQUENCE ERROR
6984                                ;*****
6985                                ;TEST 226      TEST ASL INSTRUCTION
6986                                ;*****
6987 021336 005212      TST226: INC      (R2)          ;UPDATE TEST NUMBER
6988 021340 022712 000226      CMP      #226,(R2)      ;SEQUENCE ERROR?
6989 021344 001054              BNE     TST227-10      ;BR TO ERROR HALT ON SEQ ERROR
6990 021346 012700 144000      MOV      #144000,RO     ;RO=14000
6991 021352 000257              CCC              ;CC=0110
6992 021354 000271              +SEN!SEC
6993 021356 006300              ASL     RO              ;CC=1001  RO=110000
6994 021360 103003              BCC     ASL1
6995 021362 102402              BVS     ASL1
6996 021364 001401              BEQ     ASL1
6997 021366 100404              BMI     ASL2
6998                                ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
6999                                ;          CONDITIONAL BRANCH INST. AND <====
7000                                ;          REPLACE THE MOVE INSTRUCTION <====
7001                                ;          WHICH FOLLOWS W/ 767 <====
7002 021370 012742 000516      ASL1:   MOV      #516,-(R2) ;MOVE TO MAILBOX # ***** 516 *****
7003 021370 005242              INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
7004 021374 000000              HALT
7005 021376 000277      ASL2:   SCC              ;CC=1100
7006 021400 000277

```


7007	021402	000243			+CLV!CLC			
7008	021404	006300			ASL	RO	;CC=0011	RO=020000
7009	021406	103003			BCC	ASL3		
7010	021410	102002			BVC	ASL3		
7011	021412	001401			BEQ	ASL3		
7012	021414	100004			BPL	ASL4		
7013							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
7014							CONDITIONAL BRANCH INST. AND	<====
7015							REPLACE THE MOVE INSTRUCTION	<====
7016							WHICH FOLLOWS W/ 754	<====
7017	021416			ASL3:				
7018	021416	012742	000517		MOV	#517, -(R2)	;MOVE TO MAILBOX # ***** 517 *****	
7019	021422	005242			INC	-(R2)	;SET MSGTYP TO FATAL ERROR	
7020	021424	000000			HALT		;ASL DID NOT SET CC'S CORRECTLY	
7021	021426	000277		ASL4:	SCC		;CC=0111	
7022	021430	000250			CLN			
7023	021432	006300			ASL	RO	;CC=0000	RO=040000
7024	021434	101402			BLOS	ASL5		
7025	021436	102401			BVS	ASL5		
7026	021440	100004			BPL	ASL6		
7027							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
7028							CONDITIONAL BRANCH INST. AND	<====
7029							REPLACE THE MOVE INSTRUCTION	<====
7030							WHICH FOLLOWS W/ 742	<====
7031	021442			ASL5:				
7032	021442	012742	000520		MOV	#520, -(R2)	;MOVE TO MAILBOX # ***** 520 *****	
7033	021446	005242			INC	-(R2)	;SET MSGTYP TO FATAL ERROR	
7034	021450	000000			HALT		;ASL DID NOT SET CC'S CORRECTLY	
7035	021452	000257		ASL6:	CCC		;CC=0101	
7036	021454	000265			+SEZ!SEC			
7037	021456	006300			ASL	RO	;CC=1010	RO=100000
7038	021460	103406			BCS	ASL7		
7039	021462	001405			BEQ	ASL7		
7040	021464	102004			BVC	ASL7		
7041	021466	100003			BPL	ASL7		
7042	021470	022700	100000		CMP	#100000, RO		
7043	021474	001404			BEQ	TST227		
7044							; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS	<====
7045							CONDITIONAL BRANCH INST. AND	<====
7046							REPLACE THE MOVE INSTRUCTION	<====
7047							WHICH FOLLOWS W/ 724	<====
7048	021476			ASL7:				
7049	021476	012742	000521		MOV	#521, -(R2)	;MOVE TO MAILBOX # ***** 521 *****	
7050	021502	005242			INC	-(R2)	;SET MSGTYP TO FATAL ERROR	
7051	021504	000000			HALT		;ASL MALFUNCTIONED	
7052							; OR SEQUENCE ERROR	


```

7053 ;*****
7054 ;TEST 227 TEST ASR INSTRUCTION
7055 ;*****
7056 021506 005212 ;ST227: INC (R2) ;UPDATE TEST NUMBER
7057 021510 022712 000227 CMP #227,(R2) ;SEQUENCE ERROR?
7058 021514 001060 BNE TST230-10 ;BR TO ERROR HALT ON SEQ ERROR
7059 021516 012700 100023 MOV #100023,R0 ;RO=100023
7060 021522 000277 SCC ;CC=0110
7061 021524 000250 CLN ;
7062 021526 006200 ASR R0 ;CC=1001 RP=140011
7063 021530 102403 BVS ASR1
7064 021532 103002 BCC ASR1
7065 021534 001401 BEQ ASR1
7066 021536 100404 BMI ASR2
7067 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7068 ; CONDITIONAL BRANCH INST. AND <====
7069 ; REPLACE THE MOVE INSTRUCTION <====
7070 ; WHICH FOLLOWS W/ 767 <====
7071 021540 ASR1:
7072 021540 012742 000522 MOV #522,-(R2) ;MOVE TO MAILBOX # ***** 522 *****
7073 021544 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7074 021546 000000 HALT ;ASR DID NOT SET CC'S CORRECTLY
7075 021550 042700 100000 ASR2: BIC #100000,R0 ;RO=40011
7076 021554 000277 SCC ;CC=1100
7077 021556 000243 +CLV!CLC ;
7078 021560 006200 ASR R0 ;CC=0011 RO=020004
7079 021562 102003 BVC ASR3
7080 021564 103002 BCC ASR3
7081 021566 001401 BEQ ASR3
7082 021570 100004 BPL ASR4
7083 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7084 ; CONDITIONAL BRANCH INST. AND <====
7085 ; REPLACE THE MOVE INSTRUCTION <====
7086 ; WHICH FOLLOWS W/ 752 <====
7087 021572 ASR3:
7088 021572 012742 000523 MOV #523,-(R2) ;MOVE TO MAILBOX # ***** 523 *****
7089 021576 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7090 021600 000000 HALT ;ASR DID NOT SET CC'S CORRECTLY
7091 021602 000277 ASR4: SCC ;CC=1111
7092 ;
7093 021604 006200 ASR R0 ;CC=0000 RO=010002
7094 021606 101403 BLOS ASR5
7095 021610 102402 BVS ASR5
7096 021612 001401 BEQ ASR5
7097 021614 100004 BPL ASR6
7098 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7099 ; CONDITIONAL BRANCH INST. AND <====
7100 ; REPLACE THE MOVE INSTRUCTION <====
7101 ; WHICH FOLLOWS W/ 740 <====
7102 021616 ASR5:
7103 021616 012742 000524 MOV #524,-(R2) ;MOVE TO MAILBOX # ***** 524 *****
7104 021622 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7105 021624 000000 HALT ;ASR DID NOT SET CC'S CORRECTLY
7106 021626 052700 100000 ASR6: BIS #100000,R0 ;RO=110002
7107 021632 000257 CCC ;CC=0101
7108 021634 000265 +SEZ!SEC

```



```

7109 021636 006200          ASR      RO          ;C=1010  RO=144001
7110 021640 101406          BLOS    ASR7
7111 021642 102005          BVC     ASR7
7112 021644 100004          BPL     ASR7
7113 021646 001403          BEQ     ASR7
7114 021650 022700 144001    CMP     #144001,RO    ;CHECK RESULT OF ASR'S
7115 021654 001404          BEQ     TST230
7116                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7117                                     ;          CONDITIONAL BRANCH INST. AND <====
7118                                     ;          REPLACE THE MOVE INSTRUCTION <====
7119                                     ;          WHICH FOLLOWS W/ 720 <====

```

```

7120 021656          ASR7:
7121 021656 012742 000525    MOV     #525, -(R2) ;MOVE TO MAILBOX # ***** 525 *****
7122 021662 005242          INC     -(R2)      ;SET MSGTYP TO FATAL ERROR
7123 021664 000000          HALT
7124
7125
7126
7127
7128
7129
7130
7131
7132
7133
7134
7135
7136
7137
7138

```

```

:*****
: THIS TEST VERIFIES THE SXT INSTRUCTION. CONDITION CODES
: ARE PRESET IN EACH OF THE TWO POSSIBLE CASES. WITH THE N-BIT SET,
: 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.
:*****

```

```

7139 021666 005212          TST230: INC     (R2) ;UPDATE TEST NUMBER
7140 021670 022712 000230    CMP     #230, (R2) ;SEQUENCE ERROR?
7141 021674 001033          BNE     TST231-10 ;BR TO ERROR HALT ON SEQ ERROR
7142 021676 005000          CLR     RO
7143 021700 000277          SCC
7144 021702 000244          CLZ
7145 021704 006700          SXT     RO ;TRY SXT
7146 021706 100006          BPL     SXT0 ;TEST CC=1001
7147 021710 001405          BEQ     SXT0
7148 021712 102404          BVS     SXT0
7149 021714 103003          BCC     SXT0
7150 021716 022700 177777    CMP     #-1, RO ;CHECK DATA RESULT
7151 021722 001404          BEQ     SXT1

```

```

7152                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7153                                     ;          CONDITIONAL BRANCH INST. AND <====
7154                                     ;          REPLACE THE MOVE INSTRUCTION <====
7155                                     ;          WHICH FOLLOWS W/ 765 <====

```

```

7156 021724          SXT0:
7157 021724 012742 000526    MOV     #526, -(R2) ;MOVE TO MAILBOX # ***** 526 *****
7158 021730 005242          INC     -(R2)      ;SET MSGTYP TO FATAL ERROR
7159 021732 000000          HALT ;RESULTS OF SXT INCORRECT
7160 021734 005000          SXT1: CLR     RO ;RO=0
7161 021736 005010          CLR     (RO) ;LOC. 0=0
7162 021740 005110          COM     (RO) ;LOC. 0=177777
7163 021742 000257          CCC ;SET CC=0110
7164 021744 000266          +SEZ!SEV

```


7165 021746 006710
7166 021750 001005
7167 021752 103404
7168 021754 102403
7169 021756 100402
7170 021760 005710
7171 021762 001404
7172
7173
7174
7175
7176 021764
7177 021764 012742 000527
7178 021770 005242
7179 021772 000000
7180

SXT (R0)
BNE SXT2
BCS SXT2
BVS SXT2
BMI SXT2
TST (R0)
BEQ TST231

;TEST CC=0100

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

SXT2:

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

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

7181
7182
7183
7184
7185
7186
7187
7188
7189
7190
7191
7192
7193
7194
7195
7196
7197
7198
7199
7200
7201
7202
7203
7204
7205
7206
7207
7208
7209
7210
7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226
7227
7228
7229
7230
7231
7232

021774 005212
021776 022712 000231
022002 001035
022004 012700 007463
022010 012701 031525
022014 000277
022016 000241
022020 074100
022022 101406
022024 102405
022026 001404
022030 100403
022032 022700 036146
022036 001404

022040
022040 012742 000530
022044 005242
022046 000000
022050 010104
022052 000261
022054 000241
022056 074400
022060 101406
022062 102405
022064 001404
022066 100403
022070 022700 007463
022074 001404

022076
022076 012742 000531
022102 005242
022104 000000

```
*****
:
: THIS TEST VERIFIES THE XOR INSTRUCTION. UNIQUE PATTERNS
: OF ONES AND ZEROES ARE MOVED TO DATA REGISTERS R0 AND R1.
: AFTER THE FIRST XOR INSTRUCTION R0=36146. AN XOR IS THEN
: EXECUTED WITH THIS NEW VALUE AND THE CONTENTS OF R1 TO
: REPRODUCE THE ORIGINAL VALUE IF R0=31525.
:
: *****
: TEST 231 TEST THE XOR INSTRUCTION
: *****
TST231: INC (R2) ;UPDATE TEST NUMBER
        CMP #231,(R2) ;SEQUENCE ERROR?
        BNE TST232-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #7463,R0 ;SET UP R0
        MOV #31525,R1 ;SET UP R1
        SCC ;SET CC=1110
        CLC
        XOR R1,R0 ;TRY XOR
        BLOS XOR1 ;CC=0000?
        BVS XOR1
        BEQ XOR1
        BMI XOR1
        CMP #36146,R0 ;DATA RESULT CORRECT?
        BEQ XOR2
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 762 <====
XOR1: MOV #530,-(R2) ;MOVE TO MAILBOX # ***** 530 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT
XOR2: MOV R1,R4
        SEC ;CC=1110
        CLC
        XOR R4,R0 ;TRY XOR MODE 0,0
        BLOS XOR3 ;CC=0000?
        BVS XOR3
        BEQ XOR3
        BMI XOR3
        CMP #7463,R0
        BEQ TST232
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 743 <====
XOR3: MOV #531,-(R2) ;MOVE TO MAILBOX # ***** 531 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;RESULT OF XOR INCORRECT
: OR SEQUENCE ERROR
```


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

022106 005212
022110 022712 000232
022114 001023
022116 012700 000525
022122 010004
022124 000277
022126 101002
022130 100001
022132 102404

022134
022134 012742 000532
022140 005242
022142 000000
022144 005304
022146 000277
022150 077012
022152 101004
022154 100003
022156 102002
022160 005704
022162 001404

022164
022164 012742 000533
022170 005242
022172 000000

```
*****
: THIS TEST VERIFIES THE SOB INSTRUCTION. R4 IS USED AS A
: COUNTER WHILE R0 IS THE ADDRESS REGISTER. CONDITIONAL
: BRANCHES ARE USED TO VERIFY PROPER TRANSFER OF CONTROL
: WHILE R4 IS CHECKED TO INSURE PROPER DECREMENTING OF R0.
*****
: TEST 232 TEST SOB INSTRUCTION
*****
TST232: INC (R2) ;UPDATE TEST NUMBER
CMP #232,(R2) ;SEQUENCE ERROR?
BNE TST233-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #525,R0
MOV R0,R4
SCC ;SET CC=1111
SOB1: BHI SOB2 ;CC=1111?
BPL SOB2
BVS SOB3
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 771 <====
SOB2: MOV #532,-(R2) ;MOVE TO MAILBOX # ***** 532 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
SOB3: HALT
DEC R4 ;COUNT ITERATIONS
SCC ;CC=1111
SOB RO,SOB1 ;DO SOB W/ R0
BHI SOB4 ;CHECK CC=1111
BPL SOB4
BVC SOB4
TST R4 ;ITERATION COUNT OK?
BEQ TST233
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 755 <====
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
```



```

7277
7278
7279
7280
7281
7282
7283
7284
7285
7286
7287 022174 005212
7288 022176 022712 000233
7289 022202 001062
7290 022204 012706 000500
7291 022210 012746 125252
7292 022214 162706 000074
7293 022220 012705 022246
7294 022224 012746 006436
7295 022230 000277
7296 022232 000137 000400
7297 022236 012742 000534
7298 022242 005242
7299 022244 000000
7300 022246 101010
7301 022250 100007
7302 022252 102006
7303 022254 020527 125252
7304 022260 001003
7305 022262 022706 000500
7306 022266 001404
7307
7308
7309
7310
7311 022270
7312 022270 012742 000535
7313 022274 005242
7314 022276 000000
7315 022300 012746 052525
7316 022304 012746 006400
7317 022310 010605
7318 022312 004737 022322
7319 022316 000137 022334
7320 022322 000205
7321 022324 012742 000536
7322 022330 005242
7323 022332 000000
7324 022334 022706 000500
7325 022340 001003
7326 022342 022705 052525
7327 022346 001404
7328
7329
7330
7331
7332 022350

```

```

*****
: 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
*****
TST233: INC (R2) ;UPDATE TEST NUMBER
CMP #233,(R2) ;SEQUENCE ERROR?
BNE TST234-10 ;BR TO ERROR HALT ON SEQ ERROR
MOV #STBOT,SP
MOV #125252,-(SP) ;PUT R5 VALUE ON STACK
SUB #74,SP ;EFFECTIVELY PUT 36 ARGUMENTS ON STACK
MOV #MRK1,R5 ;SET NEW PC IN R5
MOV #6436,-(SP) ;PUT MARK 36 INST. ON STACK
SCC ;SET CC=1111
JMP @#400 ;XFER CONTL TO MARK 36 INST. ON STACK
MOV #534,-(R2) ;MOVE TO MAILBOX # ***** 534 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;MARK INST. SHOULD HAVE JUMPED TO MRK1
MRK1: BHI MRK2 ;TEST CC UNAFFECTED
BPL MRK2 ;IE. CC=1111
BVC MRK2
CMP R5,#125252 ;CHECK R5 RESTORED FROM STACK
BNE MRK2
CMP #STBOT,R6 ;CHECK STACK POINTER READJUSTED CORRECTLY.
BEQ MRK3
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 746 <====
MRK2: MOV #535,-(R2) ;MOVE TO MAILBOX # ***** 535 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RESULTS OF MARK INCORRECT
MRK3: MOV #52525,-(SP)
MOV #6400,-(SP) ;PUT MARK 0 INST. ON STACK
MOV SP,R5 ;SET ADDR. OF MARK INST. IN R5
JSR PC,@#MRK4 ;DO JSR
JMP @#MRK5
MRK4: RTS R5 ;DO RTS WITH R5 TO MARK INST ON STACK
MOV #536,-(R2) ;MOVE TO MAILBOX # ***** 536 *****
INC -(R2) ;SET MSGTYP TO FATAL ERROR
HALT ;RTS,MARK SEQUENCE FAILED
MRK5: CMP #STBOT,R6 ;STACK ADJUSTED CORRECTLY
BNE MRK6 ;IF NOT: BR
CMP #52525,R5 ;CHECK IF R5 RESTORED FROM STACK
BEQ TST234
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 716 <====
MRK6:

```


7333 022350 012742 000537
7334 022354 005242
7335 022356 000000
7336

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

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


```

7337      177776
7338
7339
7340
7341
7342
7343
7344
7345
7346
7347
7348
7349
7350 022360 005212
7351 022362 022712 000234
7352 022366 001024
7353 022370 012700 000377
7354 022374 000257
7355 022376 106400
7356 022400 022767 000357 155370
7357 022406 001404
7358
7359
7360
7361
7362 022410 012742 000540
7363 022414 005242
7364 022416 000000
7365 022420 005000
7366 022422 005010
7367 022424 000277
7368 022426 106410
7369 022430 100403
7370 022432 102402
7371 022434 103401
7372 022436 001004
7373
7374
7375
7376
7377 022440
7378 022440 012742 000541
7379 022444 005242
7380 022446 000000
7381
7382
7383
7384
7385
7386 022450 005212
7387 022452 022712 000235
7388 022456 001021
7389 022460 005000
7390 022462 012710 177777
7391 022466 005037 177776
7392 022472 106420

```

PS=177776

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

TEST 234 TEST MTPS INSTRUCTION

```

TST234: INC      (R2)           ;UPDATE TEST NUMBER
        CMP      #234,(R2)     ;SEQUENCE ERROR?
        BNE     TST235-10     ;BR TO ERROR HALT ON SEQ ERROR
        MOV     #377,R0
        CCC
        MTPS   R0
        CMP     #357,PS
        BEQ    MTPS1

```

```

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

```

```

        MOV     #540,-(R2)     ;MOVE TO MAILBOX # ***** 540 *****
        INC     -(R2)
        HALT
MTPS1:  CLR     R0
        CLR     (R0)
        SCC
        MTPS   (R0)           ;CC=1111
        BMI    MTPS1A         ;TRY MTPS MODE 1
        BVS    MTPS1A         ;CHECK PS
        BCS    MTPS1A
        BNE    TST235

```

```

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

```

```

MTPS1A: MOV     #541,-(R2)     ;MOVE TO MAILBOX # ***** 541 *****
        INC     -(R2)
        HALT
        OR     OR SEQUENCE ERROR

```

TEST 235 TEST MTPS MODE 2

```

TST235: INC      (R2)           ;UPDATE TEST NUMBER
        CMP     #235,(R2)     ;SEQUENCE ERROR?
        BNE     TST236-10     ;BR TO ERROR HALT ON SEQ ERROR
        CLR     R0
        MOV     #-1,(R0)      ;R0=0
        CLR     @#PS          ;LOC. 0=-1
        MTPS   (R0)+         ;PS=0
        ;TRY MTPS W/MODE 2

```



```

7393 022474 022737 000357 177776      CMP      #357, @#PS      ;CHECK DATA
7394 022502 001404                      BEQ      MTPS2
7395                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7396                                     ;          CONDITIONAL BRANCH INST. AND <====
7397                                     ;          REPLACE THE MOVE INSTRUCTION <====
7398                                     ;          WHICH FOLLOWS W/ 766 <====
7399 022504 012742 000542      MOV      #542, -(R2)    ;MOVE TO MAILBOX # ***** 542 *****
7400 022510 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7401 022512 000000      HALT
7402 022514 022700 000001      MTPS2:  CMP      #1, R0  ;DEST. DATA INCORRECT
7403 022520 001404                      BEQ      TST236        ;CHECK DEST. REGISTER.
7404                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7405                                     ;          CONDITIONAL BRANCH INST. AND <====
7406                                     ;          REPLACE THE MOVE INSTRUCTION <====
7407                                     ;          WHICH FOLLOWS W/ 757 <====
7408 022522 012742 000543      MOV      #543, -(R2)    ;MOVE TO MAILBOX # ***** 543 *****
7409 022526 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7410 022530 000000      HALT        ;DEST REGISTER NOT INCREMENTED BY 1
7411                                     ; OR SEQUENCE ERROR

```

```

;*****
;TEST 236      TEST MTPS MODE 3
;*****

```

```

7415 022532 005212      TST236: INC      (R2)        ;UPDATE TEST NUMBER
7416 022534 022712 000236      CMP      #236, (R2)    ;SEQUENCE ERROR?
7417 022540 001024      BNE      TST237-10    ;BR TO ERROR HALT ON SEQ ERROR
7418 022542 012700 000402      MOV      #402, R0     ;R0=402
7419 022546 005010      CLR      (R0)        ;LOC. 402=0
7420 022550 012737 052652 000000      MOV      #52652, @#0  ;LOC. 0=52652
7421 022556 005037 177776      CLR      @#PS        ;PS=0
7422 022562 106430      MTPS    @ (R0)+      ;TRY MTPS W/MODE 3
7423 022564 022737 000252 177776      CMP      #252, @#PS   ;CHECK DEST. DATA
7424 022572 001404      BEQ      MTPS3
7425                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7426                                     ;          CONDITIONAL BRANCH INST. AND <====
7427                                     ;          REPLACE THE MOVE INSTRUCTION <====
7428                                     ;          WHICH FOLLOWS W/ 763 <====
7429 022574 012742 000544      MOV      #544, -(R2)    ;MOVE TO MAILBOX # ***** 544 *****
7430 022600 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7431 022602 000000      HALT        ;DEST. DATA INCORRECT
7432 022604 022700 000404      MTPS3:  CMP      #404, R0 ;CHECK MODE 3 REGISTER.
7433 022610 001404      BEQ      TST237
7434                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7435                                     ;          CONDITIONAL BRANCH INST. AND <====
7436                                     ;          REPLACE THE MOVE INSTRUCTION <====
7437                                     ;          WHICH FOLLOWS W/ 754 <====
7438 022612 012742 000545      MOV      #545, -(R2)    ;MOVE TO MAILBOX # ***** 545 *****
7439 022616 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7440 022620 000000      HALT        ;MODE 3 REGISTER INCORRECT
7441                                     ; OR SEQUENCE ERROR

```

```

;*****
;TEST 237      TEST MTPS MODE 4
;*****

```

```

7442 022622 005212      TST237: INC      (R2)        ;UPDATE TEST NUMBER
7443 022624 022712 000237      CMP      #237, (R2)    ;SEQUENCE ERROR?

```



```

7449 022630 001022      BNE      TST240-10      ;BR TO ERROR HALT ON SEQ ERROR
7450 022632 012700 000001  MOV      #1,R0          ;RO=1
7451 022636 012737 125125 000000  MOV      #125125,R#0    ;LOC. 0 = 125125
7452 022644 005037 177776      CLR      @#PS           ;PS=0
7453 022650 106440      MTPS     -(R0)          ;TRY MTPS W/MODE 4
7454 022652 022737 000105 177776  CMP      #105,@#PS     ;CHECK DEST. DATA
7455 022660 001404      BEQ      MTPS4
7456      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
7457      ;          CONDITIONAL BRANCH INST. AND      <====
7458      ;          REPLACE THE MOVE INSTRUCTION      <====
7459      ;          WHICH FOLLOWS W/ 764      <====
7460 022662 012742 000546      MOV      #546,-(R2)    ;MOVE TO MAILBOX # ***** 546 *****
7461 022666 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7462 022670 000000      HALT
7463 022672 005700      MTPS4: TST      R0      ;DEST. DATA INCORRECT
7464 022674 001404      BEQ      TST240      ;CHECK MODE 4 REGISTER
7465      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
7466      ;          CONDITIONAL BRANCH INST. AND      <====
7467      ;          REPLACE THE MOVE INSTRUCTION      <====
7468      ;          WHICH FOLLOWS W/ 756      <====
7469 022676 012742 000547      MOV      #547,-(R2)    ;MOVE TO MAILBOX # ***** 547 *****
7470 022702 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7471 022704 000000      HALT
7472      ;MODE 4 REGISTER NOT DECREMENTED BY 1
7473      ;OR SEQUENCE ERROR

```

```

;*****
;TEST 240      TEST MTPS MODE 5
;*****

```

```

7474      ;*****
7475      ;*****
7476      ;*****
7477 022706 005212      TST240: INC      (R2)    ;UPDATE TEST NUMBER
7478 022710 022712 000240  CMP      #240,(R2)    ;SEQUENCE ERROR?
7479 022714 001021      BNE      TST241-10    ;BR TO ERROR HALT ON SEQ ERROR
7480 022716 012700 000404  MOV      #404,R0      ;RO=404
7481 022722 012737 177400 000000  MOV      #177400,R#0  ;LOC. 0=177400
7482 022730 000277      SCC
7483 022732 106450      MTPS     @-(R0)      ;SET ALL COND. CODES
7484 022734 005737 177776  TST      @#PS        ;TRY MTPS W/MODE 5
7485 022740 001404      BEQ      MTPS5      ;CHECK DEST. DATA.
7486      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
7487      ;          CONDITIONAL BRANCH INST. AND      <====
7488      ;          REPLACE THE MOVE INSTRUCTION      <====
7489      ;          WHICH FOLLOWS W/ 766      <====
7490 022742 012742 000550      MOV      #550,-(R2)    ;MOVE TO MAILBOX # ***** 550 *****
7491 022746 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7492 022750 000000      HALT
7493 022752 022700 000402      MTPS5: CMP      #402,R0 ;DESTINATION DATA INCORRECT
7494 022756 001404      BEQ      TST241      ;CHECK MODE 5 REGISTER
7495      ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS      <====
7496      ;          CONDITIONAL BRANCH INST. AND      <====
7497      ;          REPLACE THE MOVE INSTRUCTION      <====
7498      ;          WHICH FOLLOWS W/ 757      <====
7499 022760 012742 000551      MOV      #551,-(R2)    ;MOVE TO MAILBOX # ***** 551 *****
7500 022764 005242      INC      -(R2)        ;SET MSGTYP TO FATAL ERROR
7501 022766 000000      HALT
7502      ;MODE 5 REGISTER NOT DECREMENTED BY 2
7503      ;OR SEQUENCE ERROR

```

```

;*****

```


H13

```
7505 ;TEST 241 TEST MTPS MODE 6
7506 :*****
7507 022770 005212 ;TST241: INC (R2) ;UPDATE TEST NUMBER
7508 022772 022712 000241 CMP #241,(R2) ;SEQUENCE ERROR?
7509 022776 001024 BNE TST242-10 ;BR TO ERROR HALT ON SEQ ERROR
7510 023000 012737 052652 000000 MOV #52652,#0 ;LOC. 0=52652
7511 023006 012700 000406 MOV #406,R0 ;R0=406
7512 023012 005037 177776 CLR #PS ;PS=0
7513 023016 106460 177372 MTPS -406(R0) ;TRY MTPS W/MODE 6
7514 023022 022737 000252 177776 CMP #252,#PS ;CHECK DEST. DATA
7515 023030 001404 BEQ MTPS6
7516 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7517 ; CONDITIONAL BRANCH INST. AND <====
7518 ; REPLACE THE MOVE INSTRUCTION <====
7519 ; WHICH FOLLOWS W/ 763 <====
7520 023032 012742 000552 MOV #552,-(R2) ;MOVE TO MAILBOX # ***** 552 *****
7521 023036 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7522 023040 000000 HALT ;DEST. DATA INCORRECT
7523 023042 022700 000406 MTPS6: CMP #406,R0 ;CHECK MODE 6 REGISTER
7524 023046 001404 BEQ TST242
7525 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7526 ; CONDITIONAL BRANCH INST. AND <====
7527 ; REPLACE THE MOVE INSTRUCTION <====
7528 ; WHICH FOLLOWS W/ 754 <====
7529 023050 012742 000553 MOV #553,-(R2) ;MOVE TO MAILBOX # ***** 553 *****
7530 023054 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7531 023056 000000 HALT ;MODE 6 REGISTER MODIFIED
7532 ; OR SEQUENCE ERROR
7533 :*****
7534 ;TEST 242 TEST MTPS MODE 7
7535 :*****
7536 :*****
7537 023060 005212 ;TST242: INC (R2) ;UPDATE TEST NUMBER
7538 023062 022712 000242 CMP #242,(R2) ;SEQUENCE ERROR?
7539 023066 001024 BNE TST243-10 ;BR TO ERROR HALT ON SEQ ERROR
7540 023070 012737 052652 000000 MOV #52652,#0 ;LOC. 0=52652
7541 023076 012700 000410 MOV #410,R0 ;R0=410
7542 023102 005037 177776 CLR #PS ;PS=0
7543 023106 106470 177776 MTPS #2(R0) ;TRY MTPS W/MODE 7
7544 023112 022737 000105 177776 CMP #105,#PS ;CHECK DEST. DATA
7545 023120 001404 BEQ MTPS7
7546 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7547 ; CONDITIONAL BRANCH INST. AND <====
7548 ; REPLACE THE MOVE INSTRUCTION <====
7549 ; WHICH FOLLOWS W/ 763 <====
7550 023122 012742 000554 MOV #554,-(R2) ;MOVE TO MAILBOX # ***** 554 *****
7551 023126 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7552 023130 000000 HALT ;DESTINATION DATA INCORRECT
7553 023132 022700 000410 MTPS7: CMP #410,R0 ;CHECK MODE 7 REGISTER
7554 023136 001404 BEQ TST243
7555 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7556 ; CONDITIONAL BRANCH INST. AND <====
7557 ; REPLACE THE MOVE INSTRUCTION <====
7558 ; WHICH FOLLOWS W/ 754 <====
7559 023140 012742 000555 MOV #555,-(R2) ;MOVE TO MAILBOX # ***** 555 *****
7560 023144 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
```


7561 023146 000000
7562
7563

HALT

;MODE 7 REGISTER MODIFIED
; OR SEQUENCE ERROR


```

7564
7565
7566
7567
7568
7569
7570
7571
7572
7573
7574
7575 023150 005212
7576 023152 022712 000243
7577 023156 001025
7578 023160 012737 000377 177776
7579 023166 106700
7580 023170 022700 177757
7581 023174 001404
7582
7583
7584
7585
7586 023176 012742 000556
7587 023202 005242
7588 023204 000000
7589
7590 023206 005000
7591 023210 012737 177777 000000
7592 023216 005037 177776
7593 023222 106710
7594 023224 105737 000000
7595 023230 001404
7596
7597
7598
7599
7600 023232 012742 000557
7601 023236 005242
7602 023240 000000
7603
7604
7605
7606
7607
7608 023242 005212
7609 023244 022712 000244
7610 023250 001031
7611 023252 005000
7612 023254 005010
7613 023256 012737 000377 177776
7614 023264 106720
7615 023266 103003
7616 023270 102402
7617 023272 001401
7618 023274 100404
7619

```

```

*****
: 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 RO, EITHER DIRECTLY OR INDIRECTLY. CONDITIONAL BRANCHES ARE
: USED TO CHECK PROPER ADDRESSING AND DATA.
*****
: TEST 243 TEST MFPS INSTRUCTION
*****
TST243: INC (R2) ;UPDATE TEST NUMBER
: CMP #243,(R2) ;SEQUENCE ERROR?
: BNE TST244-10 ;BR TO ERROR HALT ON SEQ ERROR
: MOV #377,@#PS
: MFPS RO
: CMP #177757,RO
: BEQ MFPS1
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 771 <====
: MOVE TO MAILBOX # ***** 556 *****
: SET MSGTYP TO FATAL ERROR
: MFPS FAILED
:
MFPS1: CLR RO
: MOV #-1,@#0
: CLR @#PS
: MFPS (RO)
: TSTB @#0
: BEQ TST244
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 753 <====
: MOVE TO MAILBOX # ***** 557 *****
: SET MSGTYP TO FATAL ERROR
: MFPS FAILED
: OR SEQUENCE ERROR
*****
: TEST 244 TEST MFPS MODE 2
*****
TST244: INC (R2) ;UPDATE TEST NUMBER
: CMP #244,(R2) ;SEQUENCE ERROR?
: BNE TST245-10 ;BR TO ERROR HALT ON SEQ ERROR
: CLR RO ;RO=0
: CLR (RO) ;LOC. 0=0
: MOV #377,@#PS ;SET PS=357
: MFPS (RO)+ ;TRY MFPS W/MODE 2
: BCC MFPS2A ;BR TO ERROR IF C BIT CLEAR
: BVS MFPS2A ;BR TO ERROR IF V BIT SET
: BEQ MFPS2A ;BR TO ERROR IF Z BIT SET
: BMI MFPS2B
:
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====

```



```

7620 ;
7621 ; CONDITIONAL BRANCH INST. AND <====
7622 ; REPLACE THE MOVE INSTRUCTION <====
7623 ; WHICH FOLLOWS W/ 755 <====
7624 023276 012742 000560 MFPS2A: MOV #560,-(R2) ;MOVE TO MAILBOX # ***** 560 *****
7625 023302 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7626 023304 000000 HALT ;COND. CODES INCORRECT
7627 023306 022737 000357 000000 MFPS2B: CMP #357,a#0 ;CHECK DEST. DATA
7628 023314 001404 BEQ MFPS2C
7629 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7630 ; CONDITIONAL BRANCH INST. AND <====
7631 ; REPLACE THE MOVE INSTRUCTION <====
7632 ; WHICH FOLLOWS W/ 756 <====
7633 023316 012742 000561 MOV #561,-(R2) ;MOVE TO MAILBOX # ***** 561 *****
7634 023322 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7635 023324 000000 HALT ;DEST. DATA INCORRECT
7636 023326 022700 000001 MFPS2C: CMP #1,R0 ;CHECK MODE Z REGISTER
7637 023332 001404 BEQ TST245
7638 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7639 ; CONDITIONAL BRANCH INST. AND <====
7640 ; REPLACE THE MOVE INSTRUCTION <====
7641 ; WHICH FOLLOWS W/ 747 <====
7642 023334 012742 000562 MOV #562,-(R2) ;MOVE TO MAILBOX # ***** 562 *****
7643 023340 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7644 023342 000000 HALT ;MODE 2 REGISTER NOT INCREMENTED 1
7645 ; OR SEQUENCE ERROR
7646
7647 ;*****
7648 ;TEST 245 TEST MFPS MODE 3
7649 ;*****
7650 023344 005212 TST245: INC (R2) ;UPDATE TEST NUMBER
7651 023346 022712 000245 CMP #245,(R2) ;SEQUENCE ERROR?
7652 023352 001033 BNE TST246-10 ;BR TO ERROR HALT ON SEQ ERROR
7653 023354 012700 000406 MOV #406,R0 ;R0=406
7654 023360 005037 000000 CLR a#0 ;LOC. 0=0
7655 023364 012737 000252 177776 MOV #252,a#PS ;PS=252
7656 023372 106730 MFPS MFP3 a(R0)+ ;TRY MFPS WITH MODE 3
7657 023374 103403 BCS MFP3A ;BR TO ERROR IF C-BIT SET
7658 023376 102402 BVS MFP3A ;BR TO ERROR IF V-BIT SET
7659 023400 001401 BEQ MFP3A ;BR TO ERROR IF Z-BIT SET
7660 023402 100404 BMI MFP3B
7661 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7662 ; CONDITIONAL BRANCH INST. AND <====
7663 ; REPLACE THE MOVE INSTRUCTION <====
7664 ; WHICH FOLLOWS W/ 764 <====
7665 023404 MFPS3A:
7666 023404 012742 000563 MOV #563,-(R2) ;MOVE TO MAILBOX # ***** 563 *****
7667 023410 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7668 023412 000000 HALT ;CONDITION CODES INCORRECT
7669 023414 022737 125000 000000 MFPS3B: CMP #125000,a#0 ;CHECK DEST. DATA
7670 023422 001404 BEQ MFPS3C
7671 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7672 ; CONDITIONAL BRANCH INST. AND <====
7673 ; REPLACE THE MOVE INSTRUCTION <====
7674 ; WHICH FOLLOWS W/ 754 <====
7675 023424 012742 000564 MOV #564,-(R2) ;MOVE TO MAILBOX # ***** 564 *****

```



```

7676 023430 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
7677 023432 000000          HALT                    ;DEST DATA INCORRECT
7678 023434 020027 000410  MFPS3C: CMP      R0,#410      ;CHECK MODE 3 REGISTER.
7679 023440 001404          BEQ      TST246
7680                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7681                                     ;          CONDITIONAL BRANCH INST. AND <====
7682                                     ;          REPLACE THE MOVE INSTRUCTION <====
7683                                     ;          WHICH FOLLOWS W/ 745 <====
7684 023442 012742 000565          MOV      #565,-(R2)     ;MOVE TO MAILBOX # ***** 565 *****
7685 023446 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
7686 023450 000000          HALT                    ;MODE 3 REGISTER NOT INCREMENTED BY 2
7687                                     ; OR SEQUENCE ERROR

```

```

;*****
;TEST 246      TEST MFPS MODE 4
;*****

```

```

7691
7692 023452 005212          TST246: INC      (R2)          ;UPDATE TEST NUMBER
7693 023454 022712 000246          CMP      #246,(R2)     ;SEQUENCE ERROR?
7694 023460 001033          BNE     TST247-10      ;BR TO ERROR HALT ON SEQ ERROR
7695 023462 012700 000002          MOV      #2,R0         ;R0=2
7696 023466 005037 000000          CLR     @#0           ;LOC. 0=0
7697 023472 012737 000125 177776          MOV      #125,@#PS     ;PS=125
7698 023500 106740          MFPS   -(R0)          ;TRY MFPS W/MODE 4
7699 023502 103003          BCC     MFPS4A         ;BR TO ERROR IF C-BIT CLEAR
7700 023504 102402          BVS     MFPS4A         ;BR TO ERROR IF V-BIT SET
7701 023506 001401          BEQ     MFPS4A         ;BR TO ERROR IF Z-BIT SET
7702 023510 100004          BPL     MFPS4B

```

```

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

```

```

7707 023512          MFPS4A:
7708 023512 012742 000566          MOV      #566,-(R2)     ;MOVE TO MAILBOX # ***** 566 *****
7709 023516 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
7710 023520 000000          HALT                    ;COND. CODES INCORRECT
7711 023522 022737 042400 000000  MFPS4B: CMP      #42400,@#0     ;CHECK DEST. DATA
7712 023530 001404          BEQ     MFPS4C

```

```

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

```

```

7717 023532 012742 000567          MOV      #567,-(R2)     ;MOVE TO MAILBOX # ***** 567 *****
7718 023536 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
7719 023540 000000          HALT                    ;DEST. DATA INCORRECT
7720 023542 020027 000001  MFPS4C: CMP      R0,#1      ;CHECK MODE 4 REGISTER
7721 023546 001404          BEQ     TST247

```

```

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

```

```

7726 023550 012742 000570          MOV      #570,-(R2)     ;MOVE TO MAILBOX # ***** 570 *****
7727 023554 005242          INC      -(R2)         ;SET MSGTYP TO FATAL ERROR
7728 023556 000000          HALT                    ;MODE 4 REGISTER NOT DECREMENTED BY 1

```

```

; OR SEQUENCE ERROR

```

```

;*****

```

```

7729
7730
7731

```



```

7732 ;TEST 247 TEST MFPS MODE 5
7733 ;*****
7734 023560 005212 ;TST247: INC (R2) ;UPDATE TEST NUMBER
7735 023562 022712 000247 CMP #247,(R2) ;SEQUENCE ERROR?
7736 023566 001033 BNE TST250-10 ;BR TO ERROR HALT ON SEQ ERROR
7737 023570 012700 000410 MOV #410,R0 ;R0=410
7738 023574 012737 177777 000000 MOV #-1,@#0 ;LOC. 0=-1
7739 023602 005037 177776 CLR @#PS ;PS=0
7740 023606 106750 MFPS @-(R0) ;TRY MFPS W/MODE 5
7741 023610 103403 BCS MFPS5A ;BR TO ERROR IF C-BIT SET
7742 023612 102402 BVS MFPS5A ;BR TO ERROR IF V-BIT SET
7743 023614 100401 BMI MFPS5A ;BR TO ERROR IF N-BIT SET
7744 023616 001404 BEQ MFPS5B
7745 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7746 ; CONDITIONAL BRANCH INST. AND <====
7747 ; REPLACE THE MOVE INSTRUCTION <====
7748 ; WHICH FOLLOWS W/ 764 <====
7749 023620 MFPS5A:
7750 023620 012742 000571 MOV #571,-(R2) ;MOVE TO MAILBOX # ***** 571 *****
7751 023624 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7752 023626 000000 HALT ;COND. CODES INCORRECT
7753 023630 022737 000377 000000 MFPS5B: CMP #377,@#0 ;CHECK DEST. DATA
7754 023636 001404 BEQ MFPS5C
7755 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7756 ; CONDITIONAL BRANCH INST. AND <====
7757 ; REPLACE THE MOVE INSTRUCTION <====
7758 ; WHICH FOLLOWS W/ 754 <====
7759 023640 012742 000572 MOV #572,-(R2) ;MOVE TO MAILBOX # ***** 572 *****
7760 023644 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7761 023646 000000 HALT ;DEST DATA INCORRECT
7762 023650 020027 000406 MFPS5C: CMP R0,#406 ;CHECK MODE 5 REGISTER
7763 023654 001404 BEQ TST250
7764 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7765 ; CONDITIONAL BRANCH INST. AND <====
7766 ; REPLACE THE MOVE INSTRUCTION <====
7767 ; WHICH FOLLOWS W/ 745 <====
7768 023656 012742 000573 MOV #573,-(R2) ;MOVE TO MAILBOX # ***** 573 *****
7769 023662 005242 INC -(R2) ;SET MSGTYP TO FATAL ERROR
7770 023664 000000 HALT ;MODE 5 REGISTER NOT DECREMENTED BY 2
7771 ; OR SEQUENCE ERROR
7772 ;*****
7773 ;TEST 250 TEST MFPS MODE 6
7774 ;*****
7775 ;*****
7776 023666 005212 ;TST250: INC (R2) ;UPDATE TEST NUMBER
7777 023670 022712 000250 CMP #250,(R2) ;SEQUENCE ERROR?
7778 023674 001034 BNE TST251-10 ;BR TO ERROR HALT ON SEQ ERROR
7779 023676 012700 000401 MOV #401,R0 ;R0=410
7780 023702 005037 000000 CLR @#0 ;LOC. 0=0
7781 023706 012737 000252 177776 MOV #252,@#PS ;PS=252
7782 023714 106760 177377 MFPS -401(R0) ;TRY MFPS W/MODE 6
7783 023720 102403 BVS MFPS6A ;BR TO ERROR IF V-BIT SET
7784 023722 103402 BCS MFPS6A ;BR TO ERROR IF C-BIT SET
7785 023724 001401 BEQ MFPS6A ;BR TO ERROR IF Z-BIT SET
7786 023726 100404 BMI MFPS6B
7787 ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====

```



```

7788                                     ;          CONDITIONAL BRANCH INST. AND <====
7789                                     ;          REPLACE THE MOVE INSTRUCTION <====
7790                                     ;          WHICH FOLLOWS W/ 763 <====
7791 023730                               MFPS6A:
7792 023730 012742 000574                 MOV    #574,-(R2) ;MOVE TO MAILBOX # ***** 574 *****
7793 023734 005242                       INC    -(R2)      ;SET MSGTYP TO FATAL ERROR
7794 023736 000000                       HALT                                     ;COND. CODES INCORRECT
7795 023740 022737 000252 000000 MFPS6B: CMP    #252,@#0 ;CHECK DEST. DATA
7796 023746 001404                       BEQ    MFPS6C
7797                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7798                                     ;          CONDITIONAL BRANCH INST. AND <====
7799                                     ;          REPLACE THE MOVE INSTRUCTION <====
7800                                     ;          WHICH FOLLOWS W/ 753 <====
7801 023750 012742 000575                 MOV    #575,-(R2) ;MOVE TO MAILBOX # ***** 575 *****
7802 023754 005242                       INC    -(R2)      ;SET MSGTYP TO FATAL ERROR
7803 023756 000000                       HALT                                     ;DEST. DATA INCORRECT
7804 023760 022700 000401 MFPS6C: CMP    #401,R0 ;CHECK DEST. REGISTER
7805 023764 001404                       BEQ    TST251
7806                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7807                                     ;          CONDITIONAL BRANCH INST. AND <====
7808                                     ;          REPLACE THE MOVE INSTRUCTION <====
7809                                     ;          WHICH FOLLOWS W/ 744 <====
7810 023766 012742 000576                 MOV    #576,-(R2) ;MOVE TO MAILBOX # ***** 576 *****
7811 023772 005242                       INC    -(R2)      ;SET MSGTYP TO FATAL ERROR
7812 023774 000000                       HALT                                     ;DEST. DATA INCORRECT
7813                                     ; OR SEQUENCE ERROR
7814
7815 ;*****
7816 ;TEST 251 TEST MFPS MODE 7
7817 ;*****
7818 023776 005212 000251 †TST251: INC    (R2) ;UPDATE TEST NUMBER
7819 024000 022712 000251                 CMP    #251,(R2) ;SEQUENCE ERROR?
7820 024004 001034                       BNE    TST252-10 ;BR TO ERROR HALT ON SEQ ERROR
7821 024006 012700 000777                 MOV    #777,R0 ;RO=777
7822 024012 005037 000000                 CLR    @#0 ;LOC. 0=0
7823 024016 012737 000125 177776         MOV    #125,@#PS ;PS=125
7824 024024 106770 177407 MFPS    @-371(R0) ;TRY MFPS W/MODE 7
7825 024030 102403 BVS    MFPS7A ;BR TO ERROR IF V-BIT SET
7826 024032 103002 BCC    MFPS7A ;BR TO ERROR IF C-BIT SET
7827 024034 001401 BEQ    MFPS7A ;BR TO ERROR IF Z-BIT SET
7828 024036 100004 BPL    MFPS7B
7829
7830                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7831                                     ;          CONDITIONAL BRANCH INST. AND <====
7832                                     ;          REPLACE THE MOVE INSTRUCTION <====
7833                                     ;          WHICH FOLLOWS W/ 763 <====
7833 024040                               MFPS7A:
7834 024040 012742 000577                 MOV    #577,-(R2) ;MOVE TO MAILBOX # ***** 577 *****
7835 024044 005242                       INC    -(R2)      ;SET MSGTYP TO FATAL ERROR
7836 024046 000000                       HALT                                     ;CONDITION CODE INCORRECT
7837 024050 022737 042400 000000 MFPS7B: CMP    #42400,@#0 ;CHECK DESTINATION DATA
7838 024056 001404                       BEQ    MFPS7C
7839                                     ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7840                                     ;          CONDITIONAL BRANCH INST. AND <====
7841                                     ;          REPLACE THE MOVE INSTRUCTION <====
7842                                     ;          WHICH FOLLOWS W/ 753 <====
7843 024060 012742 000600                 MOV    #600,-(R2) ;MOVE TO MAILBOX # ***** 600 *****

```



```

7844 024064 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
7845 024066 000000          HALT                    ;DEST. DATA INCORRECT
7846 024070 022700 000777  MFPS70: CMP      #777,R0    ;CHECK MODE 7 REGISTER
7847 024074 001404          BEQ      TST252
7848          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7849          ;          CONDITIONAL BRANCH INST. AND <====
7850          ;          REPLACE THE MOVE INSTRUCTION <====
7851          ;          WHICH FOLLOWS W/ 744 <====
7852 024076 012742 000601          MOV      #601, -(R2)    ;MOVE TO MAILBOX # ***** 601 *****
7853 024102 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
7854 024104 000000          HALT                    ;MODE 7 REGISTER MODIFIED
7855          ; 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 256 (DECIMAL)
: PASSES.

```

```

*****
: TEST 252 TEST THAT RESET DOES NOT CLEAR PSW
*****
7868 024106 005212          TST252: INC      (R2)          ;UPDATE TEST NUMBER
7869 024110 022712 000252          CMP      #252, (R2)    ;SEQUENCE ERROR?
7870 024114 001014          BNE     TST253-10     ;BR TO ERROR HALT ON SEQ ERROR
7871 024116 123727 025652 000377  CMPB    @#PASSPT, #377 ;ONLY DUE RESET EVERY 256. PASSES
7872 024124 001014          BNE     REST          ;BR IF TO SKIP TEST
7873 024126 012737 000357 177776  MOV     #357, @#PS     ;MOV ONES TO PSW
7874 024134 000005          RESET
7875 024136 022737 000357 177776  CMP     #357, @#PS     ;PSW CORRECT?
7876 024144 001404          BEQ     TST253
7877          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
7878          ;          CONDITIONAL BRANCH INST. AND <====
7879          ;          REPLACE THE MOVE INSTRUCTION <====
7880          ;          WHICH FOLLOWS W/ 764 <====
7881 024146 012742 000602          MOV      #602, -(R2)    ;MOVE TO MAILBOX # ***** 602 *****
7882 024152 005242          INC      -(R2)          ;SET MSGTYP TO FATAL ERROR
7883 024154 000000          HALT                    ;RESET ALTERED PSW
7884          ; OR SEQUENCE ERROR

```

```

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

```

```

*****
: TEST 253 TEST USER MODE R6 CAN HOLD A ONE IN EVERY POSITION
*****
7895 024156 005212          TST253: INC      (R2)          ;UPDATE TEST NUMBER
7896 024160 022712 000253          CMP      #253, (R2)    ;SEQUENCE ERROR?
7897 024164 001014          BNE     TST254-10     ;BR TO ERROR HALT ON SEQ ERROR
7898 024166 052767 140000 153602  BIS     #USR, PS       ;SET USER MODE
7899 024174 012706 000001          MOV     #1, R6        ;SET BIT0

```



```

7900 024200 000241          CLC                ;CLEAR C-BIT
7901 024202 006106          USP1: ROL          R6                ;ROTATE 1 POSITION
7902 024204 103376          BCC          USP1                ;BR IF NOT ALL DONE
7903 024206 001407          BEQ          USP1A                ;BR IF NO BITS PICKED
7904 024210 042767 140000 153560 BIC          #USRM,PS          ;CLEAR USER MODE
7905 024216 012742 000603 MOV          #603, -(R2)        ;MOVE TO MAILBOX # ***** 603 *****
7906 024222 005242          INC          -(R2)            ;SET MSGTYP TO FATAL ERROR
7907 024224 000000          HALT                ;USER MODE R6 PICKED A BIT
7908 024226

```

```

7909
7910
7911
7912
7913
7914
7915
7916
7917
7918
7919

```

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.

```

7920 024226 005212          TST254: INC          (R2)          ;UPDATE TEST NUMBER
7921 024230 022712 000254  CMP          #254, (R2)        ;SEQUENCE ERROR?
7922 024234 001036          BNE          USP4-14          ;BR TO ERROR HALT ON SEQ ERROR

```

```

7923 024236 052767 140000 153532 BIS          #USRM,PS          ;SET USER MODE
7924 024244 012706 177777 MOV          #-1,R6            ;SET USER R6 TO ALL ONES
7925 024250 022706 177777 CMP          #-1,R6            ;READ AND CHECK USER R6
7926 024254 001407          BEQ          USP2                ;BR IF NO ERROR
7927 024256 042767 140000 153512 BIC          #USRM,PS          ;CLEAR USER MODE
7928 024264 012742 000604 MOV          #604, -(R2)        ;MOVE TO MAILBOX # ***** 604 *****
7929 024270 005242          INC          -(R2)            ;SET MSGTYP TO FATAL ERROR
7930 024272 000000          HALT                ;USER R6 WILL NOT HOLD ALL ONES
7931 024274 042767 140000 153474 USP2: BIC          #USRM,PS          ;SET KERNEL MODE
7932 024302 022706 177777  CMP          #-1,R6            ;KERNEL MODE R6 ADDR. FROM USER MODE??
7933 024306 001004          BNE          USP3

```

```

7934
7935
7936
7937

```

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

```

7938 024310 012742 000605 MOV          #605, -(R2)        ;MOVE TO MAILBOX # ***** 605 *****
7939 024314 005242          INC          -(R2)            ;SET MSGTYP TO FATAL ERROR
7940 024316 000000          HALT                ;DUAL ADDRESSING ERROR USER/KERNEL R6
7941 024320 005006          USP3: CLR          R6                ;CLEAR KERNEL MODE SP
7942 024322 052767 140000 153446 BIS          #USRM,PS          ;SET USER MODE
7943 024330 022706 177777  CMP          #-1,R6            ;CHECK USER R6 NOT ADDR. FROM KERNEL MODE
7944 024334 001404          BEQ          USP4                ;BR IF NO ERROR
7945 024336 012742 000606 MOV          #606, -(R2)        ;MOVE TO MAILBOX # ***** 606 *****
7946 024342 005242          INC          -(R2)            ;SET MSGTYP TO FATAL ERROR
7947 024344 000000          HALT                ;DUAL ADDRESSING ERROR OR SEQUENCE ERROR
7948 024346 012706 000500  USP4: MOV          #STBOT,R6          ;RESTORE SP USER
7949 024352 042767 140000 153416 BIC          #USRM,PS          ;SET KERNEL MODE
7950 024360 012706 000500  MOV          #STBOT,R6          ;RESTORE SP KERNEL

```

```

7951
7952
7953
7954
7955

```

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

```

7956
7957
7958
7959
7960 024364 005212
7961 024366 022712 000255
7962 024372 001032
7963 024374 012706 000500
7964 024400 012767 140000 153370
7965 024406 012706 026116
7966 024412 006506
7967 024414 022767 140000 153354
7968 024422 001407
7969 024424 042767 140000 153344
7970 024432 012742 000607
7971 024436 005242
7972 024440 000000
7973 024442 022767 000500 001444 MFPIO:
7974 024450 001407
7975 024452 042767 140000 153316
7976 024460 012742 000610
7977 024464 005242
7978 024466 000000
7979 024470
7980
7981
7982
7983
7984 024470 005212
7985 024472 022712 000256
7986 024476 001032
7987 024500 005067 153272
7988 024504 005006
7989 024506 012767 140000 153262
7990 024514 012706 026116
7991 024520 012746 000500
7992 024524 006606
7993 024526 022767 140000 153242
7994 024534 001407
7995 024536 042767 140000 153232
7996 024544 012742 000611
7997 024550 005242
7998 024552 000000
7999 024554 005067 153216
8000 024560 020627 000500
8001 024564 001404
8002
8003
8004
8005
8006 024566 012742 000612
8007 024572 005242
8008 024574 000000
8009
8010

```

```

;*****
;TEST 255 TEST MFPI WITH R6 IN MODE 0
;*****
TST255: INC (R2) ;UPDATE TEST NUMBER
        CMP #255,(R2) ;SEQUENCE ERROR?
        BNE TST256-10 ;BR TO ERROR HALT ON SEQ ERROR
        MOV #STBOT,R6 ;INITIALIZE KERNEL STACK POINTER
        MOV #USRM,PS ;SET USER MODE/PREVIOUS KERNEL
        MOV #USTBOT,R6 ;INITIALIZE USER STACK POINTER
        MFPI R6 ;TRY MFPI WITH MODE 0
        CMP #140000,PS ;CHECK PSW
        BEQ MFPIO ;BR IF NO ERROR
        BIC #USRM,PS ;CLEAR USER MODE
        MOV #607,-(R2) ;MOVE TO MAILBOX # ***** 607 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;INCORRECT PSM FROM MFPI
MFPIO:  CMP #STBOT,USTBOT-2 ;CHECK DATA ON STACK
        BEQ MFPI0A ;BR IF NO ERROR
        BIC #USRM,PS ;CLEAR USER MODE
        MOV #610,-(R2) ;MOVE TO MAILBOX # ***** 610 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;INCORRECT DATA FROM MFPI
MFPI0A:
;*****
;TEST 256 TEST MTPI WITH R6 IN MODE 0
;*****
TST256: INC (R2) ;UPDATE TEST NUMBER
        CMP #256,(R2) ;SEQUENCE ERROR?
        BNE TST257-10 ;BR TO ERROR HALT ON SEQ ERROR
        CLR PS ;SET KERNEL MODE
        CLR R6 ;INITIALIZE KERNEL R6
        MOV #USRM,PS ;SET USER MODE/PREVIOUS KERNEL
        MOV #USTBOT,R6 ;INITIALIZE USER STACK POINTER
        MOV #STBOT,-(R6) ;SET UP TARGET DATA
        MTPI R6 ;TRY MODE 0 MTPI
        CMP #USRM,PS ;CHECK PSW
        BEQ MTPIO ;BR IF NO ERROR
        BIC #USRM,PS ;CLEAR USER MODE
        MOV #611,-(R2) ;MOVE TO MAILBOX # ***** 611 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;PS INCORRECT FOLLOWING MTPI
MTPIO:  CLR PS ;SET KERNEL MODE
        CMP R6,#STBOT ;CHECK TARGET DATA
        BEQ TST257
; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
; CONDITIONAL BRANCH INST. AND <====
; REPLACE THE MOVE INSTRUCTION <====
; WHICH FOLLOWS W/ 745 <====
        MOV #612,-(R2) ;MOVE TO MAILBOX # ***** 612 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;DATA INCORRECT FOLLOWING MTPI
; OR SEQUENCE ERROR

```



```

8011
8012
8013
8014
8015
8016
8017
8018
8019
8020
8021
8022
8023
8024
8025
8026
8027
8028
8029
8030
8031
8032
8033
8034
8035
8036
8037
8038 024576 005212
8039 024600 022712 000257
8040 024604 001062
8041 024606 012700 025706
8042 024612 012704 025744
8043 024616 012767 000017 000142
8044 024624 012067 000110
8045 024630 012401
8046 024632 012767 177777 000074
8047 024640 012703 000020
8048 024644 005267 000064
8049 024650 032701 100000
8050 024654 013705 177776
8051 024660 042705 177773
8052 024664 000165 024670
8053 024670 000167 000020
8054 024674 012767 024770 000042
8055 024702 012767 024752 000040
8056 024710 000167 000014
8057 024714 012767 024752 000022
8058 024722 012767 024770 000020
8059 024730 006101
8060
8061 024732 012737
8062 024734 000000
8063 024736 177776
8064 024740 000000
8065 024742 000137
8066 024744 000000

```

```

*****
: THIS TEST VERIFIES THE CONTENTS OF THE BRANCH ROM. THE 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
*****
TST257: INC (R2) ;UPDATE TEST NUMBER
: CMP #257,(R2) ;SEQUENCE ERROR?
: BNE ER ;BR TO ERROR HALT ON SEQ ERROR
SETUP: MOV #BRTAB,R0 ;INITIALIZE BRANCH TABLE POINTER
: MOV #YNTAB,R4 ;INITIALIZE YES/NO BRANCH MAP POINTER
: MOV #15,BRCT ;INITIALIZE BRANCH TABLE COUNT
SETBR: MOV (R0)+,BRH ;GET NEXT BRANCH INST.
: MOV (R4)+,R1 ;GET NEXT BRANCH MAP
: MOV #-1,CC ;INITIALIZE CONDITION CODE VALUE
: MOV #16,R3 ;INITIALIZE CONDITION CODE COUNT
SETCC: INC CC ;SET FOR NEXT CC VALUE
: BIT #100000,R1 ;SEE IF SHOULD BR W/ THESE CC'S
: MOV @#177776,R5 ;SIMULATE A JNE
: BIC #177773,R5 ; (JUMP NOT EQUAL)
: JMP .+4(R5) ; TO SET2BR
: JMP SET2BR
: MOV #CONT,NBR ;SET TO CONTINUE IF NO BRANCH
: MOV #ER,YBR ;SET TO REPORT ERROR IF BRANCH
: JMP AROUND ;GO AROUND OPPOSITE CONDITION
SET2BR: MOV #ER,NBR ;SET TO REPORT ERROR IF NO BRANCH
: MOV #CONT,YBR ;SET TO CONTINUE IF BRANCH
AROUND: ROL R1 ;UPDATE BIT MAP
: MOV (PC)+,@(PC)+ ;SET CONDITION CODE
CC: 0 ;NEW CC VALUE GOES HERE
: 177776
BRH: 0 ;BRANCH INST. GOES HERE
: JMP @ (PC)+ ;THIS JUMP IF NO BRANCH
NBR: 0 ;WHERE TO GO IF NO BRANCH OCCURS

```


8067	024746	000137		JMP	2(PC)+	; THIS JUMP IF BRANCH OCCURS
8068	024750	000000		0		; WHERE TO GO IF BRANCH OCCURS
8069	024752	012702	000304	YBR: MOV	#\$TESTN,R2	; RESTORE POINTER
8070	024756	012742	000613	ER: MOV	613,-(R2)	; MOVE TO MAILBOX # ***** 613 *****
8071	024762	005242		MOV	-(R2)	; SET MSGTYP TO FATAL ERROR
8072	024764	000000		INC		
8073	024766	000000		HALT		
8074	024770	005303		BRCT: 0		
8075	024772	013705	177776	CONT: DEC	R3	; CC'S DONE?
8076	024776	042705	177773	MOV	2#177776,R5	; SIMULATE A JNE
8077	025002	000165	025006	BIC	177773,R5	(JUMP NOT EQUAL)
8078	025006	000167	177632	JMP	4(R5)	TO SETCC
8079	025012	005367	177750	JMP	SETCC	
8080	025016	013705	177776	DEC	BRCT	; BR'S DONE?
8081	025022	042705	177773	MOV	2#177776,R5	; SIMULATE A JNE
8082	025026	000165	025032	BIC	177773,R5	(JUMP NOT EQUAL)
8083	025032	000167	177566	JMP	4(R5)	TO SETBR
				JMP	SETBR	


```

8084
8085
8086
8087
8088
8089
8090
8091
8092
8093
8094
8095 025036 005212
8096 025040 022712 000260
8097 025044 001052
8098 025046 005000
8099 025050 005001
8100 025052 005002
8101 025054 005003
8102 025056 005004
8103 025060 005005
8104 025062 005006
8105 025064 052700 000001
8106 025070 052701 000002
8107 025074 052702 000004
8108 025100 052703 000010
8109 025104 052704 000020
8110 025110 052705 000040
8111 025114 052706 000100
8112 025120 022706 000100
8113 025124 001022
8114 025126 022705 000040
8115 025132 001017
8116 025134 022704 000020
8117 025140 001014
8118 025142 022703 000010
8119 025146 001011
8120 025150 022702 000004
8121 025154 001006
8122 025156 022701 000002
8123 025162 001003
8124 025164 022700 000001
8125 025170 001404
8126
8127
8128
8129
8130 025172
8131 025172 012742 000614
8132 025176 005242
8133 025200 000000
8134 025202 012702 000304
8135

```

```

*****
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
*****
TST260: INC (R2) ;UPDATE TEST NUMBER
          CMP #260,(R2) ;SEQUENCE ERROR?
          BNE DAERR ;BR TO ERROR HALT ON SEQ ERROR
BITCLR: CLR R0 ;INITIALIZE ALL REGISTERS
         CLR R1
         CLR R2
         CLR R3
         CLR R4
         CLR R5
         CLR R6
BITSET: BIS #1,R0 ;SET R0=1
         BIS #2,R1 ;R1=2
         BIS #4,R2 ;R2=4
         BIS #10,R3 ;R3=10
         BIS #20,R4 ;R4=20
         BIS #40,R5 ;R5=40
         BIS #100,R6 ;R6=100
BITCHK: CMP #100,R6 ;TEST THAT NO DUAL ADDRESSING OCCURRED
         BNE DAERR ;BR TO ERROR HALT IF ANY OTHER BITS ARE SET
         CMP #40,R5
         BNE DAERR
         CMP #20,R4
         BNE DAERR
         CMP #10,R3
         BNE DAERR
         CMP #4,R2
         BNE DAERR
         CMP #2,R1
         BNE DAERR
         CMP #1,R0
         BEQ BITCON
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 726 <====
DAERR: MOV #614,-(R2) ;MOVE TO MAILBOX # ***** 614 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;DUAL ADDRESSING ERROR
BITCON: MOV #STESTN,R2 ;RESTORE POINTER

```



```

8136
8137
8138
8139
8140
8141
8142
8143
8144
8145 025206 005212
8146 025210 022712 000261
8147 025214 001012
8148 025216 052737 170357 177776
8149 025224 105037 177776
8150 025230 013700 177776
8151 025234 032700 170000
8152 025240 001006
8153 025242 005037 177776
8154 025246 012742 000615
8155 025252 005242
8156 025254 000000
8157 025256 005037 177776
8158
8159
8160
8161
8162
8163
8164
8165
8166
8167
8168 025262 005212
8169 025264 022712 000262
8170 025270 001010
8171 025272 000277
8172 025274 000252
8173 025276 000167 000000
8174 025302 100403
8175 025304 001002
8176 025306 102401
8177 025310 103404
8178
8179
8180
8181
8182 025312
8183 025312 012742 000616
8184 025316 005242
8185 025320 000000
8186

```

```

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

```

```

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

```

```

TST261: INC (R2) ;UPDATE TEST NUMBER
          CMP #261,(R2) ;SEQUENCE ERROR?
          BNE BTERR ;BR TO ERROR HALT ON SEQ ERROR
          BIS #170357,a#PS ;SET ALL POSSIBLE BITS IN PSW
          CLRB a#PS ;CLR PR LEVEL AND CC'S
          MOV a#PS,RO ;COPY CONTENTS OF PSW
          BIT #170000,RO ;TEST THAT UPPER BYTE IS UNAFFECTED
          BNE BTCON ;CONTINUE IF OK
BTERR: CLR a#PS ;RETURN TO KERNEL MODE
        MOV #615,-(R2) ;MOVE TO MAILBOX # ***** 615 *****
        INC -(R2) ;SET MSGTYP TO FATAL ERROR
        HALT ;BYTE INSTRUCTION ALTERED PSW
BTCON: CLR a#PS ;RETURN TO KERNEL MODE

```

```

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

```

```

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

```

```

TST262: INC (R2) ;UPDATE TEST NUMBER
          CMP #262,(R2) ;SEQUENCE ERROR?
          BNE TST263-10 ;BR TO ERROR HALT ON SEQ ERROR
          SCC
          +CLN!CLV ;CC=0101
JMPT: JMP JMPT ;JUMP TO TEST PSW
        BMI JMPERR ;BR TO ERROR HALT IF N-BIT IS SET
        BNE JMPERR ;BR TO ERROR HALT IF Z-BIT IS CLEAR
        BVS JMPERR ;BR TO ERROR HALT IF V-BIT IF SET
        BCS TST263
          ; TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
          ; CONDITIONAL BRANCH INST. AND <====
          ; REPLACE THE MOVE INSTRUCTION <====
          ; WHICH FOLLOWS W/ 770 <====
JMPTERR: MOV #616,-(R2) ;MOVE TO MAILBOX # ***** 616 *****
          INC -(R2) ;SET MSGTYP TO FATAL ERROR
          HALT ;JMP INSTRUCTION AFFECTED CC'S
          ; OR SEQUENCE ERROR

```



```

8187
8188
8189
8190
8191
8192
8193
8194
8195
8196
8197
8198
8199
8200
8201
8202
8203
8204 025322 005212
8205 025324 022712 000263
8206 025330 001062
8207 025332 012767 000240 000024
8208 025340 012767 000017 000032
8209 025346 012767 000261 000102
8210 025354 012767 000001 000110
8211 025362 000277
8212 025364 000000
8213 025366 013704 177776
8214 025372 042704 177760
8215 025376 022704
8216 025400 000000
8217 025402 001404
8218
8219
8220
8221
8222 025404 012742 000617
8223 025410 005242
8224 025412 000000
8225 025414 005367 177760
8226 025420 005267 177740
8227 025424 026727 177734 000257
8228 025432 003753
8229 025434 026727 177724 000260
8230 025442 001004
8231 025444 012767 000017 177726
8232 025452 000743
8233 025454 000257
8234 025456 000000
8235 025460 013704 177776
8236 025464 042704 177760
8237 025470 022704
8238 025472 000000
8239 025474 001404
8240
8241
8242

```

```

*****
: 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
*****
TST263: INC (R2) ;UPDATE TEST NUMBER
: CMP #263,(R2) ;SEQUENCE ERROR?
: BNE CCERR ;BR TO ERROR HALT ON SEQ ERROR
: MOV #240,CC1 ;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
CC1: 0 ;CONDITION CODE INSTRUCTION
: MOV @#PS,R4 ;COPY THE PSW
: BIC #177760,R4 ;ISOLATE CONDITION CODES
: CMP (PC)+,R4 ;CHECK THAT PROPER CC'S WERE CLEARED
CC2: 0 ;OCTAL REPRESENTATION OF CC'S
: BEQ CON1
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====
: WHICH FOLLOWS W/ 753 <====
: MOVE TO MAILBOX # ***** 617 *****
: SET MSGTYP TO FATAL ERROR
: CLEAR CC INSTRUCTION FAILED
CON1: DEC CC2
: INC CC1
: CMP CC1,#257 ;GET NEXT CLEAR CC INSTRUCTION
: BLE CLRCD ;TEST FOR CCC INSTRUCTION
: GO TEST NEXT INSTRUCTION IF NOT FOUND
: CMP CC1,#260 ;CHECK FOR NOP=260
: BNE SETCD ;GO TEST SET CC INSTRUCTIONS
: MOV #17,CC2 ;SET OCTAL MAP TO TEST NOP
: BR CLRCD ;GO TEST NOP
SETCD: CCC ;CLEAR ALL CONDITION CODES
SC3: 0 ;CONDITION CODE INSTRUCTION
: MOV @#PS,R4 ;COY PSW
: BIC #177760,R4 ;CLEAR AWAY UNWANTED BITS
: CMP (PC)+,R4 ;CHECK THAT PROPER CC'S WERE SET
SC4: 0 ;OCTAL REPRESENTATION OF CC'S
: BEQ CON2
: TO SCOPE: CLEAR THE RIGHT BYTE OF THIS <====
: CONDITIONAL BRANCH INST. AND <====
: REPLACE THE MOVE INSTRUCTION <====

```



```

8252 ;*****
8253 ;TEST 264      END OF PASS SEQUENCE
8254 ;*****
8255 025526 005212      †ST264: INC      (R2)      ;UPDATE TEST NUMBER
8256 025530 022712 000264  CMP      #264,(R2) ;SEQUENCE ERROR?
8257 025534 001042      BNE      EOP1      ;BR TO ERROR HALT ON SEQ ERROR
8258 025536 105267 000110  INCB     PASSPT    ;SHOULD PRINT THIS PASS?
8259 025542 001035      BNE      GOAGIN    ;NO
8260 025544 005237 000306  INC      @#$PASS
8261 025550 132767 000040 152543  BITB     #40,$ENVM ;WILL APT ALLOW PRINTING?
8262 025556 001017      BNE      ACT      ;NO
8263 025560 023727 000042 025626  CMP      @#42,$SENDAD ;UNDER ACT AUTO ACCEPT?
8264 025566 001413      BEQ      ACT      ;IF SO SKIP PRINTOUT
8265 025570 012700 025654  MOV      #MSG,RO   ;GET MSG ADDR.
8266 025574 105737 177564  WAIT:   TSTB     @#TPS ;TTY READY
8267 025600 100375      BPL     WAIT      ;NO WAIT
8268 025602 121027 000377  CMPB     (RO),#377 ;IS NEXT CHAR. THE TERMINATOR
8269 025606 001403      BEQ     ACT      ;YES THEN BR
8270 025610 112037 177566  MOVB     (RO)+,@#TPB ;PRINT CHARACTER
8271 025614 000767      BR      WAIT      ;NEXT IF NOT DONE.
8272 025616 013700 000042  ACT:    MOV      @#42,RO ;CHECK ACT
8273 025622 001405      BEQ     GOAGIN    ;KEEP GOING
8274 025624 000005      RESET
8275 025626 004710  SENDAD: JSR      PC,(RO) ;ACT HOOKS
8276 025630 000240      NOP
8277 025632 000240      NOP
8278 025634 000240      NOP
8279 025636 000167 152660  GOAGIN: JMP      RESTRT ;DO NEXT PASS
8280 025642      EJP1:
8281 025642 012742 000621  MOV      #621,-(R2) ;MOVE TO MAILBOX # ***** 621 *****
8282 025646 005242      INC     -(R2)      ;SET MSGTYP TO FATAL ERROR
8283 025650 000000      HALT    ;SEQUENCE ERROR
8284 025652 177777      PASSPT: -1
8285 025654 005015 000000 000000  MSG:    .ASCII <15><12><0><0><0><0><0><0>.END OF.
8286 025662 000000 047105 020104
8287 025670 043117
8288 025672 042040 045506 040501  .ASCII . DFKAA-B.<0><0><0><377>
8289 025700 041055 000000 177400

```



```

8290
8291 025706 000402      .EVEN
8292 025710 001002      BRTAB: BR      .+6
8293 025712 001402      BNE      .+6
8294 025714 002002      BEQ      .+6
8295 025716 002402      BGE      .+6
8296 025720 003002      BLT      .+6
8297 025722 003402      BGT      .+6
8298 025724 100002      BLE      .+6
8299 025726 100402      BPL      .+6
8300 025730 101002      BMI      .+6
8301 025732 101402      BHI      .+6
8302 025734 102002      BLOS     .+6
8303 025736 102402      BVC      .+6
8304 025740 103002      BVS      .+6
8305 025742 103402      BCC      .+6      ; SAME AS BHIS
8306
8307
8308 025744 177777      .RADIX 2
8309 025746 170360      YNTAB: 1111111111111111      ; BR
8310 025750 007417      1111000011110000      ; BNE: Z=0
8311 025752 146063      0000111100001111      ; BEQ: Z=1
8312 025754 031714      1100110000110011      ; BGE: N XOR V =0
8313 025756 140060      0011001111001100      ; BLT: N XOR V =1
8314 025760 037717      1100000000110000      ; BGT: Z+(N XOR V) =0
8315
8316 025762 177400      0011111111001111      ; BLE: Z+(N XOR V) =1
8317 025764 000377      1111111110000000      ; BPL: N=0
8318 025766 120240      0000000011111111      ; BMI: N=1
8319 025770 057537      1010000010100000      ; BHI: C+Z=0
8320 025772 146314      0101111101011111      ; BLOS: C+Z=1
8321 025774 031463      1100110011001100      ; BVC: V=0
8322 025776 125252      0011001100110011      ; BVS: V=1
8323 026000 052525      1010101010101010      ; BCC: C=0
8324
8325
8326 026002 012737 026012 000024 PWRDN: MOV      #PWRUP, @#24      ; BCS: C=1
8327 026010 000000      HALT
8328
8329 026012 012737 026002 000024 PWRUP: MOV      #PWRDN, @#24      ; SET UP FOR A POWER UP
8330 026020 012706 000500      MOV      #STBOT, R6      ; SET UP FOR A POWER FAIL
8331 026024 132767 000040 152267      BITB     #40, SENVM      ; SET UP STACK POINTER
8332 026032 001010      BNE      PWR2            ; SHOULD PRINT?
8333 026034 012700 026060      MOV      #PFMES, R0      ; IF NOT: BR
8334 026040 105737 177564      WATE:   TSTB    @#TPS      ; GET POWER FAIL MESSG.
8335 026044 100375      BPL      WATE            ; TTY READY?
8336 026046 112037 177566      MOV      (R0)+, @#TPB    ; IF NOT: BR
8337 026052 001372      BNE      WATE            ; PRINT NEXT CHAR.
8338 026054 000137 000500      PWR2:   JMP      @#START  ; IF NOT DONE: BR
8339
8340 026060 006412 047520 042527 PFMES: .ASCIZ <12><15>.POWER FAILURE.<12><15> ; START PROGRAM AGAIN
8341 026066 020122 040506 046111
8342 026074 051125 005105 000015
8343
8344 026102 000006      .EVEN
8345 026116      .BLKW 6
      USTBOT:

```



```

8346
8347
8348
8349
8350
8351
8352 026116
8353 026116 012742 000622
8354 026122 005242
8355 026124 000000
8356 026126
8357 026126 012742 000623
8358 026132 005242
8359 026134 000000
8360 026136
8361 026136 012742 000624
8362 026142 005242
8363 026144 000000
8364 026146
8365 026146 012742 000625
8366 026152 005242
8367 026154 000000
8368 026156
8369 026156 012742 000626
8370 026162 005242
8371 026164 000000
8372 026166
8373 026166 012742 000627
8374 026172 005242
8375 026174 000000
8376 026176
8377 026176 012742 000630
8378 026202 005242
8379 026204 000000
8380 026206
8381 026206 012742 000631
8382 026212 005242
8383 026214 000000
8384 000001

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

T04:      MOV      #622, -(R2)      ; MOVE TO MAILBOX # ***** 622 *****
          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
          HALT                    ; TRAPPED THRU LOC. 4

T010:     MOV      #623, -(R2)      ; MOVE TO MAILBOX # ***** 623 *****
          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
          HALT                    ; TRAPPED THRU LOC. 10

T014:     MOV      #624, -(R2)      ; MOVE TO MAILBOX # ***** 624 *****
          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
          HALT                    ; TRAPPED THRU LOC. 14

T030:     MOV      #625, -(R2)      ; MOVE TO MAILBOX # ***** 625 *****
          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
          HALT                    ; TRAPPED THRU LOC. 30

T034:     MOV      #626, -(R2)      ; MOVE TO MAILBOX # ***** 626 *****
          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
          HALT                    ; TRAPPED THRU LOC. 34

T0114:    MOV      #627, -(R2)      ; MOVE TO MAILBOX # ***** 627 *****
          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
          HALT                    ; TRAPPED THRU LOC. 114

T0244:    MOV      #630, -(R2)      ; MOVE TO MAILBOX # ***** 630 *****
          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
          HALT                    ; TRAPPED THRU LOC. 244

T0250:    MOV      #631, -(R2)      ; MOVE TO MAILBOX # ***** 631 *****
          INC      -(R2)          ; SET MSGTYP TO FATAL ERROR
          HALT                    ; TRAPPED THRU LOC. 250

.END

```


ABASE =	000000	432			
ACDW1 =	000000	432			
ACDW2 =	000000	432			
ACPUOP =	000000	432	447		
ACT	025E16	8262	8264	8269	8272#
ADC1	020054	6492	6493	6499#	
ADC2	020064	6494	6503#		
ADC3	020104	6507	6508	6514#	
ADC4	020114	6509	6518#		
ADC5	020132	6521	6522	6523	6527#
ADDW0 =	000000	432			
ADDW1 =	000000	432			
ADDW10 =	000000	432			
ADDW11 =	000000	432			
ADDW12 =	000000	432			
ADDW13 =	000000	432			
ADDW14 =	000000	432			
ADDW15 =	000000	432			
ADDW2 =	000000	432			
ADDW3 =	000000	432			
ADDW4 =	000000	432			
ADDW5 =	000000	432			
ADDW6 =	000000	432			
ADDW7 =	000000	432			
ADDW8 =	000000	432			
ADDW9 =	000000	432			
ADD1	017670	6414	5415	6421#	
ADD2	017700	6416	6425#		
ADD3	017714	6428	6429	6435#	
ADD4	017724	6430	6439#		
ADD5	017742	6442	6443	6449#	
ADD6	017752	6444	6453#		
ADD7	017764	6454	6455	6461#	
ADD8	017774	6456	6465#		
ADD9	020014	6468	6469	6470	6476#
ADEVCT =	000000	432	438		
ADEVN =	000000	432			
AENV =	000000	432	443		
ARENVM =	000000	432	444		
AFATAL =	000000	432	435		
AMADR1 =	000000	432			
AMADR2 =	000000	432			
AMADR3 =	000000	432			
AMADR4 =	000000	432			
AMAMS1 =	000000	432			
AMAMS2 =	000000	432			
AMAMS3 =	000000	432			
AMAMS4 =	000000	432			
AMSGAD =	000000	432	440		
AMSGLG =	000000	432	441		
AMSGTY =	000000	432	434		
AMTYP1 =	000000	432			
AMTYP2 =	000000	432			
AMTYP3 =	000000	432			
AMTYP4 =	000000	432			
APASS =	000000	432	437		

APRIOR=	000000	432				
AROUND	024730	8056	8059#			
ASL1	021370	6994	6995	6996	7002#	
ASL2	021400	6997	7006#			
ASL3	021416	7009	7010	7011	7017#	
ASL4	021426	7012	7021#			
ASL5	021442	7024	7025	7031#		
ASL6	021452	7026	7035#			
ASL7	021476	7038	7039	7040	7041	7048#
ASR1	021540	7063	7064	7065	7071#	
ASR2	021550	7066	7075#			
ASR3	021572	7079	7080	7081	7087#	
ASR4	021602	7082	7091#			
ASR5	021616	7094	7095	7096	7102#	
ASR6	021626	7097	7106#			
ASR7	021656	7110	7111	7112	7113	7120#
ASWREG=	000000	432	445			
ATESTN=	000000	432	436			
AUNIT =	000000	432	439			
AUSWR =	000000	432	446			
AVECT1=	000000	432				
AVECT2=	000000	432				
BIC1	017020	6091	6092	6098#		
BIC2	017030	6093	6102#			
BIC3	017046	6105	6106	6112#		
BIS1	017110	6127	6128	6129	6135#	
BIS2	017120	6130	6139#			
BIS3	017140	6142	6143	6144	6150#	
BITCHK	025120	8112#				
BITCLR	025046	8098#				
BITCON	025202	8125	8134#			
BITSET	025064	8105#				
BIT1	016730	6054	6055	6061#		
BIT2	016740	6056	6066#			
BIT3	016756	6069	6070	6076#		
BRC1	024766	8043#	8073#	8079#		
BRC2	003040	1443	1449#			
BRC3	003050	1444	1454#			
BRC4	003067	1456	1462#			
BRH	024740	8044#	8064#			
BRN1	002720	1349	1355#			
BRN2	002730	1350	1360#			
BRN3	002740	1362	1368#			
BRTAB	025706	8041	8291#			
BRV1	002770	1396	1402#			
BRV2	003000	1397	1407#			
BRV3	003010	1409	1415#			
BRZ1	002650	1302	1308#			
BRZ2	002660	1303	1313#			
BRZ3	002670	1315	1321#			
BR1	000572	538	544#			
BR2	000602	539	548#			
BR3	000614	549	557#			
BR4	000622	558	564#			
BR5	000632	559	568#			
BTCON	025256	8152	8157#			

BTERR	025242	8147	8153#						
CC	024734	8046*	8048*	8062#					
CCERR	025476	8206	8244#						
CC1	025364	8207*	8212#	8226*	8227	8229			
CC2	025400	8208*	8216#	8225*	8231*				
CLRCD	025362	8211#	8228	8232					
CLR1	017456	6307	6308	6309	6315#				
CMP1	020320	6611	6612	6618#					
CMP2	020330	6613	6622#						
CMP3	020352	6626	6627	6633#					
CMP4	020362	6628	6637#						
CMP5	020406	6641	6642	6643	6649#				
CMP6	020416	6644	6653#						
CMP7	020436	6656	6657	6663#					
COM1	020476	6679	6680	6686#					
CONT	024770	8054	8058	8074#					
CON1	025414	8217	8225#						
CON2	025506	8239	8248#						
DAERR	025172	8097	8113	8115	8117	8119	8121	8123	8130#
DEC1	017316	6230	6231	6232	6238#				
DEC2	017326	6233	6242#						
DEC3	017342	6245	6246	6252#					
DEC4	017352	6247	6256#						
DECS	017366	6259	6260	6266#					
DEC6	017376	6261	6270#						
DEC7	017420	6274	6275	6276	6282#				
DNMBOA	010520	3863	3864	3865	3871#				
DNMBOB	010530	3866	3875#						
DNMB2A	010756	3973	3974	3975	3981#				
DNMB2B	010766	3976	3985#						
DNMB2C	011002	3986	3994#						
DNMB2D	011016	3996	3997	4003#					
DNMB2E	011026	3998	4007#						
DNMB2F	011044	4009	4017#						
DNMB3A	011126	4043	4044	4045	4051#				
DNMB3B	011136	4046	4055#						
DNMB3C	011154	4056	4064#						
DNMB3D	011172	4067	4068	4074#					
DNMB3E	011202	4069	4078#						
DNMB4A	011372	4145	4147	4148	4154#				
DNMB4B	011402	4149	4158#						
DNMB4C	011420	4159	4167#						
DNMB4D	011430	4168	4174#						
DNMB4E	011440	4169	4178#						
DNMB4F	011454	4179	4187#						
DNM03A	007612	3472	3473	3474	3480#				
DNM03B	007622	3475	3484#						
DNM03C	007632	3485	3492#						
DNM1	007464	3406	3414#						
DNM1A	010576	3896	3897	3898	3904#				
DNM1B	010606	3899	3908#						
DNM2	007500	3415	3423#						
DNM2A	010654	3929	3930	3936#					
DNM2B	010664	3931	3940#						
DNM2C	010672	3947#							
DNM2D	010702	3942	3951#						

MDM5E	013054	4714	4722#		
MDM6A	013124	4751	4752	4758#	
MDM6B	013134	4753	4762#		
MDM6C	013152	4763	4771#		
MDM6D	013172	4772	4780#		
MDM6E	013222	4783	4791#		
MDM7A	013276	4819	4820	4826#	
MDM7B	013306	4821	4830#		
MDM7C	013324	4831	4839#		
MDM7D	013344	4840	4848#		
MDM7E	013370	4850	4859#		
MFP10	024442	7968	7973#		
MFP10A	024470	7974	7979#		
MFPS1	023206	7581	7590#		
MFPS2A	023276	7615	7616	7617	7623#
MFPS2B	023306	7618	7627#		
MFPS2C	023326	7628	7636#		
MFPS3A	023404	7657	7658	7659	7665#
MFPS3B	023414	7660	7669#		
MFPS3C	023434	7670	7678#		
MFPS4A	023512	7699	7700	7701	7707#
MFPS4B	023522	7702	7711#		
MFPS4C	023542	7712	7720#		
MFPS5A	023620	7741	7742	7743	7749#
MFPS5B	023630	7744	7753#		
MFPS5C	023650	7754	7762#		
MFPS6A	023730	7783	7784	7785	7791#
MFPS6B	023740	7786	7795#		
MFPS6C	023760	7796	7804#		
MFPS7A	024040	7825	7826	7827	7833#
MFPS7B	024050	7828	7837#		
MFPS7C	024070	7838	7846#		
MOV1	016640	6017	6018	6024#	
MOV2	016650	6019	6029#		
MOV3	016666	6032	6033	6039#	
MRK1	022246	7293	7300#		
MRK2	022270	7300	7301	7302	7304 7311#
MRK3	022300	7306	7315#		
MRK4	022322	7318	7320#		
MRK5	022334	7319	7324#		
MRK6	022350	7325	7332#		
MSG	025654	8265	8285#		
MTP10	024554	7994	7999#		
MTPS1	022420	7357	7365#		
MTPS1A	022440	7369	7370	7371	7377#
MTPS2	022514	7394	7402#		
MTPS3	022604	7425	7433#		
MTPS4	022672	7455	7463#		
MTPS5	022752	7485	7493#		
MTPS6	023042	7515	7523#		
MTPS7	023132	7545	7553#		
NBR	024744	8054*	8057*	8065#	
NEG00	004010	1905	1906	1907	1913#
NEG01	004020	1908	1918#		
NEG02	004034	1919	1928#		
NEG03	004050	1930	1931	1932	1938#

2090*	2092*	2125*	2126*	2127*	2176*	2177*	2178*	2179*	2180*	2181*	2190*	2191*
2192*	2193*	2194*	2195*	2197*	2214*	2215*	2216*	2217*	2221*	2245*	2254*	2290*
2291*	2292*	2293*	2302*	2303*	2304*	2306*	2307*	2308*	2341*	2342*	2343*	2344*
2353*	2354*	2355*	2357*	2383*	2384*	2385*	2386*	2395*	2397*	2424*	2425*	2426*
2427*	2428*	2437*	2439*	2457*	2458*	2459*	2460*	2473*	2482*	2498*	2499*	2500*
2501*	2502*	2505*	2527*	2528*	2544*	2546*	2550*	2579*	2590*	2581*	2582*	2583*
2596*	2597*	2598*	2664*	2667	2696*	2697*	2700	2728*	2729*	2732	2760*	2761*
2762*	2765	2777*	2778*	2781	2809*	2810*	2813	2826*	2827*	2852*	2853*	2854*
2857	2869*	2878*	2881	2894*	2895*	2919*	2920*	2921*	2922*	2925	2938*	2939*
2965*	2966*	2967*	2968*	2969*	2970	2973	2987	3000	3001	3025*	3026*	3027*
3030	3042	3066*	3067*	3068*	3069*	3070*	3073	3085*	3086*	3110*	3111*	3112*
3113*	3116	3128*	3152*	3153*	3154*	3155*	3158	3170*	3193*	3194*	3196	3220*
3222*	3223	3247*	3250*	3263*	3288*	3289	3298*	3299*	3301	3311	3322*	3323*
3326	3327	3337	3361*	3362*	3363*	3364*	3365*	3366	3402*	3405	3414	3423*
3424	3433*	3434*	3436	3446	3467*	3468*	3469*	3470*	3471	3484*	3486*	3510*
3511*	3515*	3538*	3539*	3540*	3542	3567*	3568*	3569*	3572	3600*	3601*	3602*
3603*	3606	3616*	3617	3641*	3642*	3646*	3647*	3671*	3672*	3673*	3674	3684*
3685*	3711*	3712*	3713*	3714*	3747*	3749*	3750*	3751*	3752	3762	3763	3788*
3789*	3790*	3814*	3815*	3816	3840*	3841*	3842	3859*	3860*	3862	3875*	3892*
3893*	3895	3908	3925*	3926*	3927*	3928	3940*	3941*	3951	3968*	3969*	3970*
3972	3985*	3994*	3995	4007*	4008*	4017	4035*	4036*	4037*	4038*	4039*	4040*
4042	4055	4064*	4065*	4066	4096*	4097*	4098*	4099*	4101	4114	4140*	4141*
4142*	4143*	4145	4158	4167	4178	4187	4204*	4205*	4206*	4207*	4209	4222
4248*	4249*	4250*	4251*	4252	4265	4291*	4292*	4293*	4294*	4295	4308	4340*
4341*	4342*	4344	4379*	4380*	4381*	4382*	4394*	4395*	4427*	4428*	4429*	4441
4450*	4462	4493*	4494*	4496*	4508	4560*	4561*	4584	4611*	4612*	4613	4631*
4643	4680*	4681*	4693	4711*	4712*	4713	4747*	4748*	4749*	4750*	4762	4780*
4781*	4782	4817*	4818*	4830	4848*	4849	4887*	4888	4889	4890	4891	4892
4925*	4926	4927	4928	4929	4930	4962*	4963	4964	4965	4966	4967	4993*
4994	4995	4996	4997	4998	5022*	5024*	5027	5037*	5039*	5042	5069*	5070*
5072*	5086*	5087*	5100*	5101*	5102*	5104*	5132*	5133*	5135*	5139*	5140*	5150*
5151*	5153*	5157*	5167*	5168*	5169*	5171*	5175*	5176*	5260*	5262*	5266	5295*
5298*	5302	5389*	5390*	5399	5424*	5425*	5452*	5453*	5463*	5516*	5517*	5527
5554*	5556*	5566	5597*	5598*	5599	5630*	5631*	5632	5685*	5686	5687	5705*
5707	5720*	5721	5722	5740*	5742	5753*	5755	5766*	5768	5780*	5782	5823*
5827	5843	5875*	5876	5891*	5892	5898	5914	5925*	5926	5977*	5978*	5984
6016*	6031*	6050*	6053	6068	6087*	6090*	6104*	6123*	6126*	6141*	6173*	6176*
6188*	6191*	6207*	6227*	6229*	6244*	6258*	6270*	6273*	6306*	6329	6342*	6345
6364*	6367*	6381*	6411*	6413*	6427*	6441*	6453*	6467*	6488*	6491*	6503*	6506*
6520*	6551*	6554*	6567*	6570*	6583*	6586*	6607*	6610	6622*	6625	6637*	6640
6653*	6655	6675*	6678*	6709*	6712*	6724*	6727*	6739*	6742*	6756*	6776*	6779*
6794*	6809*	6822*	6825*	6855*	6858*	6873*	6888*	6902*	6906	6923*	6926*	6941*
6956*	6971*	6990*	6993*	7008*	7023*	7037*	7042	7059*	7062*	7075*	7078*	7093*
7106*	7109*	7114	7142*	7145*	7150	7160*	7161*	7162*	7165*	7170	7195*	7199*
7204	7217*	7222	7246*	7247	7262*	7353*	7355*	7365*	7366*	7368*	7389*	7390*
7392*	7402	7419*	7420*	7423*	7433	7450*	7453*	7463	7480*	7483*	7493	7511*
7513*	7523	7541*	7543*	7553	7579*	7580	7590*	7593*	7611*	7612*	7614*	7636
7653*	7656*	7678	7695*	7698*	7720	7737*	7740*	7762	7779*	7782*	7804	7821*
7824*	7846	8041*	8044	8098*	8105*	8124	8150*	8151	8265*	8268	8270	8272*
8275	8333*	8336										
906*	911*	930*	935*	937	5825*	5826*	5827*	5837	5854*	5858	5873*	5876*
5880	5892*	5896	5909*	5912	5926*	5930	5941*	5949	7196*	7199	7214	8045*
8049	8059*	8099*	8106*	8122								
934	943*											
516*	527*	534*	535	545*	546*	554*	555*	565*	566*	574*	575*	596*
597	607*	608*	615*	616	626*	627*	634*	635	645*	646*	653*	654

R1 =%000001
R1ERR 001612
R2 =%000002

663*	664*	686*	687	698*	699*	706*	707	717*	718*	727*	728*	735*
736	751*	752*	759*	760	771*	772*	779*	780	795*	796*	830*	831
841*	842*	849*	850	859*	860*	867*	868	877*	878*	885*	886	895*
896*	903*	904	919*	920*	927*	928	944*	945*	951*	952	954*	959*
966*	967*	968*	970*	975*	976	978*	983*	985	987*	989*	990*	992*
997*	998	1013*	1014*	1021*	1022	1038*	1039*	1046*	1047	1062*	1063*	1070*
1071	1087*	1088*	1096*	1097	1112*	1113*	1120*	1121	1137*	1138*	1145*	1146
1161*	1162*	1169*	1170	1186*	1187*	1209*	1210	1220*	1221*	1228*	1229	1238*
1239*	1246*	1247	1256*	1257*	1264*	1265	1274*	1275*	1296*	1297	1309*	1310*
1322*	1323*	1343*	1344	1356*	1357*	1369*	1370*	1390*	1391	1403*	1404*	1416*
1417*	1437*	1438	1450*	1451*	1463*	1464*	1506*	1507	1515*	1516*	1526*	1527*
1535*	1536*	1553*	1554	1563*	1564*	1581*	1582*	1595*	1596	1604*	1605*	1616*
1617*	1632*	1633	1642*	1643*	1655*	1656*	1670*	1671	1682*	1683*	1697*	1698*
1715*	1716	1728*	1729*	1744*	1745*	1762*	1763	1776*	1777*	1792*	1793*	1810*
1811	1824*	1825*	1840*	1841*	1853*	1854	1868*	1869*	1886*	1887*	1899*	1900
1914*	1915*	1924*	1925*	1939*	1940*	1948*	1949*	1955*	1956	1971*	1972*	1981*
1982*	1995*	1996*	2004*	2005*	2011*	2012	2027*	2028*	2041*	2042*	2050*	2051*
2072*	2073	2085*	2086*	2099*	2100*	2119*	2120	2134*	2135*	2152*	2153*	2173*
2174	2187*	2188*	2204*	2205*	2211*	2212	2231*	2232*	2240*	2241*	2251*	2252*
2260*	2261*	2271*	2272*	2287*	2288	2299*	2300*	2315*	2316*	2338*	2339	2350*
2351*	2364*	2365*	2380*	2381	2392*	2393*	2404*	2405*	2421*	2422	2434*	2435*
2446*	2447*	2454*	2455	2470*	2471*	2479*	2480*	2488*	2499*	2495*	2496	2515*
2516*	2524*	2525*	2534*	2535*	2541*	2542	2560*	2561*	2569*	2570*	2576*	2577
2593*	2594*	2605*	2606*	2621*	2622	2631*	2632*	2644*	2645*	2661*	2662	2677*
2678*	2693*	2694	2709*	2710*	2725*	2726	2742*	2743*	2757*	2758	2774*	2775*
2791*	2792*	2806*	2807	2823*	2824*	2833*	2834*	2849*	2850	2866*	2867*	2875*
2876*	2891*	2892*	2901*	2902*	2916*	2917	2935*	2936*	2945*	2946*	2962*	2963
2982*	2983*	2997*	2998*	3007*	3008*	3022*	3023	3039*	3040*	3048*	3049*	3063*
3064	3082*	3083*	3092*	3093*	3107*	3108	3125*	3126*	3134*	3135*	3149*	3150
3167*	3168*	3176*	3177*	3190*	3191	3203*	3204*	3217*	3218	3230*	3231*	3244*
3245	3260*	3261*	3269*	3270*	3285*	3286	3295*	3296*	3308*	3309*	3319*	3320*
3334*	3335*	3345*	3346*	3358*	3359	3376*	3377*	3386*	3387*	3399*	3400	3411*
3412*	3420*	3421*	3430*	3431*	3442*	3443*	3452*	3453*	3464*	3465	3481*	3482*
3493*	3494*	3507*	3508	3521*	3522*	3535*	3536	3549*	3550*	3564*	3565	3578*
3579*	3597*	3598	3613*	3614*	3623*	3624*	3638*	3639	3653*	3654*	3668*	3669
3681*	3682*	3691*	3692*	3708*	3709	3721*	3722*	3730*	3731*	3744*	3745	3759*
3760*	3769*	3770*	3784*	3785	3796*	3797*	3810*	3811	3822*	3823*	3836*	3837
3848*	3849*	3856*	3857	3872*	3873*	3881*	3882*	3889*	3890	3905*	3906*	3914*
3915*	3922*	3923	3937*	3938*	3948*	3949*	3957*	3958*	3965*	3966	3982*	3983*
3991*	3992*	4004*	4005*	4014*	4015*	4023*	4024*	4032*	4033	4052*	4053*	4061*
4062*	4075*	4076*	4085*	4086*	4093*	4094	4111*	4112*	4120*	4121*	4129*	4130*
4137*	4138	4155*	4156*	4164*	4165*	4175*	4176*	4184*	4185*	4193*	4194*	4201*
4202	4219*	4220*	4228*	4229*	4237*	4238*	4245*	4246	4262*	4263*	4271*	4272*
4280*	4281*	4288*	4289	4305*	4306*	4314*	4315*	4323*	4324*	4337*	4338	4353*
4354*	4362*	4363*	4376*	4377	4391*	4392*	4402*	4403*	4411*	4412*	4424*	4425
4438*	4439*	4447*	4448*	4459*	4460*	4468*	4469*	4477*	4478*	4490*	4491	4505*
4506*	4514*	4515*	4523*	4524*	4533*	4534*	4543*	4544*	4557*	4558	4572*	4573*
4581*	4582*	4590*	4591*	4606*	4607	4619*	4620*	4628*	4629*	4640*	4641*	4649*
4650*	4658*	4659*	4675*	4676	4690*	4691*	4699*	4700*	4708*	4709*	4719*	4720*
4728*	4729*	4744*	4745	4759*	4760*	4768*	4769*	4777*	4778*	4788*	4789*	4797*
4798*	4812*	4813	4827*	4828*	4836*	4837*	4845*	4846*	4855*	4856*	4864*	4865*
4884*	4885	4899*	4900*	4922*	4923	4937*	4938*	4959*	4960	4973*	4974*	4990*
4991	5004*	5005*	5019*	5020	5034*	5035*	5049*	5050*	5066*	5067	5082*	5083*
5097*	5098*	5114*	5115*	5129*	5130	5147*	5148*	5164*	5165*	5183*	5184*	5198*
5199	5212*	5213*	5226*	5227*	5240*	5241*	5256*	5257	5273*	5274*	5291*	5292
5309*	5310*	5325*	5326	5339*	5340*	5354*	5355	5369*	5370*	5386*	5387	5396*

K15

5397*	5405*	5406*	5420*	5421	5432*	5433*	5448*	5449	5460*	5461*	5469*	5470*
5495*	5486	5496*	5497*	5512*	5513	5524*	5525*	5533*	5534*	5551*	5552	5563*
5564*	5573*	5574*	5593*	5594	5606*	5607*	5626*	5627	5639*	5640*	5681*	5682
5693*	5694*	5702*	5703*	5716*	5717*	5728*	5729*	5737*	5738*	5750*	5751*	5763*
5764*	5777*	5778*	5791*	5792*	5816*	5817	5831*	5832*	5850*	5851*	5869*	5870*
5887*	5888*	5905*	5906*	5921*	5922*	5937*	5938*	5956*	5957*	5972*	5973	5981*
5982*	5990*	5991*	6011*	6012	6025*	6026*	6040*	6041*	6047*	6048	6062*	6063*
6077*	6078*	6084*	6085	6099*	6100*	6113*	6114*	6120*	6121	6136*	6137*	6151*
6152*	6170*	6171	6185*	6186*	6201*	6202*	6216*	6217*	6224*	6225	6239*	6240*
6253*	6254*	6267*	6268*	6283*	6284*	6301*	6302	6316*	6317*	6324*	6325	6339*
6340*	6354*	6355*	6361*	6362	6376*	6377*	6391*	6392*	6408*	6409	6422*	6423*
6436*	6437*	6450*	6451*	6462*	6463*	6477*	6478*	6485*	6486	6500*	6501*	6515*
6516*	6530*	6531*	6548*	6549	6564*	6565*	6580*	6581*	6596*	6597*	6604*	6605
6619*	6620*	6634*	6635*	6650*	6651*	6664*	6665*	6672*	6673	6687*	6688*	6706*
6707	6721*	6722*	6736*	6737*	6751*	6752*	6766*	6767*	6773*	6774	6789*	6790*
6804*	6805*	6819*	6820*	6834*	6835*	6852*	6853	6868*	6869*	6883*	6884*	6897*
6898*	6913*	6914*	6920*	6921	6936*	6937*	6951*	6952*	6966*	6967*	6980*	6981*
6987*	6988	7003*	7004*	7018*	7019*	7032*	7033*	7049*	7050*	7056*	7057	7072*
7073*	7088*	7089*	7103*	7104*	7121*	7122*	7139*	7140	7157*	7158*	7177*	7178*
7192*	7193	7211*	7212*	7229*	7230*	7243*	7244	7257*	7258*	7273*	7274*	7287*
7288	7297*	7298*	7312*	7313*	7321*	7322*	7333*	7334*	7350*	7351	7362*	7363*
7378*	7379*	7386*	7387	7399*	7400*	7408*	7409*	7416*	7417	7430*	7431*	7439*
7440*	7447*	7448	7460*	7461*	7469*	7470*	7477*	7478	7490*	7491*	7499*	7500*
7507*	7508	7520*	7521*	7529*	7530*	7537*	7538	7550*	7551*	7559*	7560*	7575*
7576	7586*	7587*	7600*	7601*	7608*	7609	7624*	7625*	7633*	7634*	7642*	7643*
7650*	7651	7666*	7667*	7675*	7676*	7684*	7685*	7692*	7693	7708*	7709*	7717*
7718*	7726*	7727*	7734*	7735	7750*	7751*	7759*	7760*	7768*	7769*	7776*	7777
7792*	7793*	7801*	7802*	7810*	7811*	7818*	7819	7834*	7835*	7843*	7844*	7852*
7853*	7868*	7869	7881*	7882*	7895*	7896	7905*	7906*	7920*	7921	7928*	7929*
7938*	7939*	7945*	7946*	7960*	7961	7970*	7971*	7976*	7977*	7984*	7985	7996*
7997*	8006*	8007*	8038*	8039	8069*	8070*	8071*	8095*	8096	8100*	8107*	8120
8131*	8132*	8134*	8145*	8146	8154*	8155*	8168*	8169	8183*	8184*	8204*	8205
8222*	8223*	8245*	8246*	8255*	8256	8281*	8282*	8353*	8354*	8357*	8358*	8361*
8362*	8365*	8366*	8369*	8370*	8373*	8374*	8377*	8378*	8381*	8382*		
982	988*											
1000*	1005*	1024*	1029*	1031	8047*	8074*	8101*	8108*	8118			
1028	1037*											
1049*	1054*	1073*	1078*	1080	2122*	2123*	2124*	2128*	2137*	2138*	2139*	2141*
2143*	2144*	2145*	2218*	2219*	2220*	2234*	2265*	2503*	2504*	2518*	2545*	2547*
2548*	2549*	2563*	3195*	3196*	3197*	3221*	3223*	3224*	3248*	3249*	3250	3289*
3300*	3301*	3302*	3311*	3312*	3313*	3324*	3325*	3326*	3327*	3328*	3337*	3338*
3339*	3403*	3404*	3405	3414	3424	3435*	3436	3445*	3446	3512*	3513*	3514*
3515	3541*	3542*	3543*	3570*	3571*	3572	3604*	3605*	3606*	3607	3617*	3643*
3644*	3645*	3646	3674*	3675*	3748*	3752*	3753*	4078*	4079	4343*	4344*	4356
4562*	4563*	4575	4609*	4610*	4622	4652	4678*	4679*	4702	4722	4815*	4816*
7214*	7217	7247*	7260*	7266	8042*	8045	8102*	8109*	8116	8213*	8214*	8215
8235*	8236*	8237										
1077	1086*											
1099*	1104*	1123*	1128*	1130	7293*	7303	7317*	7320*	7326	8050*	8051*	8052
8075*	8076*	8077	8080*	8081*	8082	8103*	8110*	8114				
1127	1136*											
411*	515*	526*	1148*	1153*	1172*	1177*	1179	1212*	5822*	5839	5841	5860
5862	5872*	5975*	5976*	7305	7324	7899*	7901*	7924*	7925	7932	7941*	7943
7948*	7950*	7963*	7965*	7966	7988*	7990*	7991*	7992*	8000	8104*	8111*	8112
8330*												
1176	1185*											

R2ERR 001742
 R3 =%000003
 R3ERR 002062
 R4 =%000004

R4ERR 002176
 R5 =%000005
 R5ERR 002312
 R6 =%000006

R6ERR 002426

R7	=%000007	410#	2624*		
SBC1	020712	6780	6781	6782	6788#
SBC2	020722	6783	6792#		
SBC3	020740	6795	6796	6797	6803#
SBC4	020750	6798	6807#		
SBC5	020766	6810	6811	6812	6818#
SBC6	020776	6813	6822#		
SBC7	021016	6826	6827	6833#	
SB0	015022	5391	5399#		
SB2	015144	5455	5463#		
SB4	015270	5519	5527#		
SB5	015356	5553	5572#		
SB5A	015350	5558	5566#		
SB5X	015366	5555*	5557	5577#	5578
SB5XAD	015370	5554	5566	5578#	
SB6	015430	5595	5605#		
SB6X	015440	5596*	5597	5610#	
SB7	015500	5628	5638#		
SB7X	015510	5629*	5643#	5644	
SB7XAD	015512	5630	5644#		
SCOPE =	000240	409#			
SC3	025456	8209*	8234#	8249*	8250
SC4	025472	8210*	8238#	8248*	
SETBR	024624	8044#	8083		
SETCC	024644	8048#	8078		
SETCD	025454	8230	8233#	8251	
SETUP	024606	8041#			
SET2BR	024714	8053	8057#		
SHL	001200	741#	744		
SHLE	001214	742	750#		
SHR	001314	785#	788		
SHRE	001330	786	794#		
SNMBOA	005754	2701	2702	2708#	
SNMB1A	006060	2766	2767	2773#	
SNMB1B	006070	2768	2777#		
SNMB1C	006112	2782	2783	2784	2790#
SNMB2A	006234	2858	2859	2865#	
SNMB2B	006244	2860	2869#		
SNMB2C	006260	2870	2878#		
SNMB2D	006300	2882	2883	2884	2890#
SNMB2E	006310	2885	2894#		
SNMB3A	006452	2974	2975	2981#	
SNMB3B	006462	2976	2985#		
SNMB3C	006500	2988	2989	2990	2996#
SNMB3D	006510	2991	3000#		
SNM0A	005714	2668	2669	2670	2676#
SNM1A	006016	2733	2734	2735	2741#
SNM2A	006154	2814	2815	2816	2822#
SNM2B	006164	2817	2826#		
SNM3A	006364	2926	2927	2928	2934#
SNM3B	006374	2929	2938#		
SNM4A	006560	3031	3032	3038#	
SNM4B	006570	3033	3042#		
SNM5A	006642	3074	3075	3081#	
SNM5B	006652	3076	3085#		
SNM6A	006726	3117	3118	3124#	

TST134	010622	3891	3909	3922#
TST135	010720	3924	3952	3965#
TST136	011062	3957	4018	4032#
TST137	011222	4034	4080	4093#
TST14	001370	832	836	849#
TST140	011330	4095	4124	4137#
TST141	011472	4139	4188	4201#
TST142	011602	4203	4232	4245#
TST143	011710	4247	4275	4288#
TST144	012020	4290	4318	4337#
TST145	012074	4339	4357	4376#
TST146	012166	4378	4406	4424#
TST147	012322	4426	4472	4490#
TST15	001422	851	854	867#
TST150	012500	4492	4538	4557#
TST151	012574	4559	4585	4606#
TST152	012730	4608	4653	4675#
TST153	013072	4677	4723	4744#
TST154	013242	4746	4792	4812#
TST155	013410	4814	4859	4884#
TST156	013474	4893	4922#	
TST157	013560	4931	4959#	
TST16	001454	869	872	885#
TST160	013644	4961	4968	4990#
TST161	013730	4992	4999	5019#
TST162	014024	5021	5043	5066#
TST163	014166	5068	5108	5129#
TST164	014344	5131	5177	5198#
TST165	014506	5200	5234	5256#
TST166	014562	5258	5267	5291#
TST167	014646	5303	5325#	
TST17	001506	887	890	903#
TST170	014714	5327	5333	5354#
TST171	014772	5363	5386#	
TST172	015040	5388	5400	5420#
TST173	015102	5422	5427	5448#
TST174	015162	5450	5464	5485#
TST175	015224	5487	5491	5512#
TST176	015304	5514	5528	5551#
TST177	015372	5567	5593#	
TST2	000644	536	569	596#
TST20	001552	905	913	927#
TST200	015442	5600	5626#	
TST201	015514	5633	5681#	
TST202	016060	5786	5816#	
TST203	016536	5950	5972#	
TST204	016612	5974	5985	6011#
TST205	016676	6013	6034	6047#
TST206	016766	6049	6071	6084#
TST207	017056	6086	6107	6120#
TST21	001622	929	938	951#
TST210	017150	6122	6145	6170#
TST211	017266	6172	6210	6224#
TST212	017430	6226	6277	6301#
TST213	017466	6303	6310	6324#
TST214	017552	6326	6348	6361#

TST215	017640	6363	6385	6408#
TST216	020024	6410	6471	6485#
TST217	020142	6487	6524	6548#
TST22	001676	975#		
TST220	020266	6550	6590	6604#
TST221	020446	6606	6658	6672#
TST222	020506	6674	6681	6706#
TST223	020660	6708	6760	6773#
TST224	021026	6775	6828	6852#
TST225	021174	6854	6907	6920#
TST226	021336	6922	6974	6987#
TST227	021506	6989	7043	7056#
TST23	001756	977	997#	
TST230	021666	7058	7115	7139#
TST231	021774	7141	7171	7192#
TST232	022106	7194	7223	7243#
TST233	022174	7245	7267	7287#
TST234	022360	7289	7327	7350#
TST235	022450	7352	7372	7386#
TST236	022532	7388	7403	7416#
TST237	022622	7418	7434	7447#
TST24	002022	999	1007	1021#
TST240	022706	7449	7464	7477#
TST241	022770	7479	7494	7507#
TST242	023060	7509	7524	7537#
TST243	023150	7539	7554	7575#
TST244	023242	7577	7595	7608#
TST245	023344	7610	7637	7650#
TST246	023452	7652	7679	7692#
TST247	023560	7694	7721	7734#
TST25	002072	1023	1032	1046#
TST250	023666	7736	7763	7776#
TST251	023776	7778	7805	7818#
TST252	024106	7820	7847	7868#
TST253	024156	7870	7876	7895#
TST254	024226	7897	7920#	
TST255	024364	7960#		
TST256	024470	7962	7984#	
TST257	024576	7986	8001	8038#
TST26	002136	1048	1056	1070#
TST260	025036	8095#		
TST261	025206	8145#		
TST262	025262	8168#		
TST263	025322	8170	8177	8204#
TST264	025526	8255#		
TST27	002206	1072	1081	1096#
TST3	000700	598	602	615#
TST30	002252	1098	1106	1120#
TST31	002322	1122	1131	1145#
TST32	002366	1147	1155	1169#
TST33	002436	1171	1180	1209#
TST34	002476	1211	1215	1228#
TST35	002534	1230	1233	1246#
TST36	002572	1248	1251	1264#
TST37	002630	1266	1269	1296#
TST4	000736	617	621	634#

E16

SERN = 000632

404#	545	546#	554	555#	565	566#	574	575#	607	608#	626	627#
645	646#	663	664#	698	699#	717	718#	727	728#	751	752#	771
772#	795	796#	841	842#	859	860#	977	878#	895	896#	919	920#
944	945#	967	968#	985	990#	1013	1014#	1038	1039#	1062	1063#	1087
1088#	1112	1113#	1137	1138#	1161	1162#	1186	1187#	1220	1221#	1238	1239#
1256	1257#	1274	1275#	1309	1310#	1322	1323#	1356	1357#	1369	1370#	1403
1404#	1416	1417#	1450	1451#	1463	1464#	1515	1516#	1526	1527#	1535	1536#
1563	1564#	1581	1582#	1604	1605#	1616	1617#	1642	1643#	1655	1656#	1692
1683#	1697	1698#	1728	1729#	1744	1745#	1776	1777#	1792	1793#	1824	1825#
1840	1841#	1868	1869#	1886	1887#	1914	1915#	1924	1925#	1939	1940#	1948
1949#	1971	1972#	1981	1982#	1995	1996#	2004	2005#	2027	2028#	2041	2042#
2050	2051#	2085	2086#	2099	2100#	2134	2135#	2152	2153#	2187	2188#	2204
2205#	2231	2232#	2240	2241#	2251	2252#	2260	2261#	2271	2272#	2299	2300#
2315	2316#	2350	2351#	2364	2365#	2392	2393#	2404	2405#	2434	2435#	2446
2447#	2470	2471#	2479	2480#	2488	2489#	2515	2516#	2524	2525#	2534	2535#
2560	2561#	2569	2570#	2593	2594#	2605	2606#	2631	2632#	2644	2645#	2677
2678#	2709	2710#	2742	2743#	2774	2775#	2791	2792#	2823	2824#	2833	2834#
2866	2867#	2875	2876#	2891	2892#	2901	2902#	2935	2936#	2945	2946#	2982
2983#	2997	2998#	3007	3008#	3039	3040#	3048	3049#	3082	3083#	3092	3093#
3125	3126#	3134	3135#	3167	3168#	3176	3177#	3203	3204#	3230	3231#	3260
3261#	3269	3270#	3295	3296#	3308	3309#	3319	3320#	3334	3335#	3345	3346#
3376	3377#	3386	3387#	3411	3412#	3420	3421#	3430	3431#	3442	3443#	3452
3453#	3481	3482#	3493	3494#	3521	3522#	3549	3550#	3578	3579#	3613	3614#
3623	3624#	3653	3654#	3681	3682#	3691	3692#	3721	3722#	3730	3731#	3759
3760#	3769	3770#	3796	3797#	3822	3823#	3848	3849#	3872	3873#	3881	3882#
3905	3906#	3914	3915#	3937	3938#	3948	3949#	3957	3958#	3982	3983#	3991
3992#	4004	4005#	4014	4015#	4023	4024#	4052	4053#	4061	4062#	4075	4076#
4085	4086#	4111	4112#	4120	4121#	4129	4130#	4155	4156#	4164	4165#	4175
4176#	4184	4185#	4193	4194#	4219	4220#	4228	4229#	4237	4238#	4262	4263#
4271	4272#	4280	4281#	4305	4306#	4314	4315#	4323	4324#	4353	4354#	4362
4363#	4391	4392#	4402	4403#	4411	4412#	4438	4439#	4447	4448#	4459	4460#
4468	4469#	4477	4478#	4505	4506#	4514	4515#	4523	4524#	4533	4534#	4543
4544#	4572	4573#	4581	4582#	4590	4591#	4619	4620#	4628	4629#	4640	4641#
4649	4650#	4658	4659#	4690	4691#	4699	4700#	4708	4709#	4719	4720#	4728
4729#	4759	4760#	4768	4769#	4777	4778#	4788	4789#	4797	4798#	4827	4828#
4836	4837#	4845	4846#	4855	4856#	4864	4865#	4899	4900#	4937	4938#	4973
4974#	5004	5005#	5034	5035#	5049	5050#	5082	5083#	5097	5098#	5114	5115#
5147	5148#	5164	5165#	5183	5184#	5212	5213#	5226	5227#	5240	5241#	5273
5274#	5309	5310#	5339	5340#	5369	5370#	5396	5397#	5405	5406#	5432	5433#
5460	5461#	5469	5470#	5496	5497#	5524	5525#	5533	5534#	5563	5564#	5573
5574#	5606	5607#	5639	5640#	5693	5694#	5702	5703#	5716	5717#	5728	5729#
5737	5738#	5750	5751#	5763	5764#	5777	5778#	5791	5792#	5831	5832#	5850
5851#	5869	5870#	5887	5888#	5905	5906#	5921	5922#	5937	5938#	5956	5957#
5981	5982#	5990	5991#	6025	6026#	6040	6041#	6062	6063#	6077	6078#	6099
6100#	6113	6114#	6136	6137#	6151	6152#	6185	6186#	6201	6202#	6216	6217#
6239	6240#	6253	6254#	6267	6268#	6283	6284#	6316	6317#	6339	6340#	6354
6355#	6376	6377#	6391	6392#	6422	6423#	6436	6437#	6450	6451#	6462	6463#
6477	6478#	6500	6501#	6515	6516#	6530	6531#	6564	6565#	6580	6581#	6596
6597#	6619	6620#	6634	6635#	6650	6651#	6664	6665#	6687	6688#	6721	6722#
6736	6737#	6751	6752#	6766	6767#	6789	6790#	6804	6805#	6819	6820#	6834
6835#	6868	6869#	6883	6884#	6897	6898#	6913	6914#	6936	6937#	6951	6952#
6966	6967#	6980	6981#	7003	7004#	7018	7019#	7032	7033#	7049	7050#	7072
7073#	7088	7089#	7103	7104#	7121	7122#	7157	7158#	7177	7178#	7211	7212#
7229	7230#	7257	7258#	7273	7274#	7297	7298#	7312	7313#	7321	7322#	7333
7334#	7362	7363#	7378	7379#	7399	7400#	7408	7409#	7430	7431#	7439	7440#
7460	7461#	7469	7470#	7490	7491#	7499	7500#	7520	7521#	7529	7530#	7550

	7551#	7559	7560#	7586	7587#	7600	7601#	7624	7625#	7633	7634#	7642	7643#	
	7666	7667#	7675	7676#	7684	7685#	7708	7709#	7717	7718#	7726	7727#	7750	
	7751#	7759	7760#	7768	7769#	7792	7793#	7801	7802#	7810	7811#	7834	7835#	
	7843	7844#	7852	7853#	7881	7882#	7905	7906#	7928	7929#	7938	7939#	7945	
	7946#	7970	7971#	7976	7977#	7996	7997#	8006	8007#	8070	8071#	8131	8132#	
	8154	8155#	8183	8184#	8222	8223#	8245	8246#	8281	8292#	8353	8354#	8357	
	8358#	8361	8362#	8365	8366#	8369	8370#	8373	8374#	8377	8378#	8361	8382#	
\$ERROR=	000302	521#	529*											
\$ETABL	000320	442#												
\$ETEND	000330	454#	477											
\$FATAL	000302	435#	521											
\$HIBTS	000330	472#												
\$MAIL	000300	433#	473	477										
\$MBADR	000332	473#												
\$MSGAD	000314	440#												
\$MSGLG	000316	441#												
\$MSGTY	000300	434#	530*											
\$PASS	000306	437#	524*	8260*										
\$PASTM	000336	475#												
\$SVPC =	000400	422#	427											
\$SWR =	000000	404#												
\$SWREG	000322	445#												
\$TESTN	000304	436#	516	522	527	966	970	987	992	8069	8134			
\$TN =	000265	404#	531	537#	569	593	599#	602	612	618#	621	631	637#	640
		650	656#	658	683	689#	693	703	709#	712	722	732	738#	745
		756	762#	766	776	782#	789	827	833#	836	846	852#	854	864
		870#	872	882	888#	890	900	906#	913	924	930#	938	948	954#
		972	978#	994	1000#	1007	1018	1024#	1032	1043	1049#	1056	1067	1073#
	1081	1093	1099#	1106	1117	1123#	1131	1142	1148#	1155	1166	1172#	1180	
	1206	1212#	1215	1225	1231#	1233	1243	1249#	1251	1261	1267#	1269	1293	
	1299#	1316	1340	1346#	1363	1387	1393#	1410	1434	1440#	1457	1503	1509#	
	1530	1550	1556#	1575	1592	1598#	1610	1629	1635#	1649	1667	1673#	1691	
	1712	1718#	1738	1759	1765#	1786	1807	1813#	1834	1850	1856#	1880	1896	
	1902#	1943	1952	1958#	1999	2008	2014#	2045	2069	2075#	2093	2116	2122#	
	2146	2170	2176#	2198	2208	2214#	2266	2284	2290#	2309	2335	2341#	2358	
	2377	2383#	2398	2418	2424#	2440	2451	2457#	2483	2492	2498#	2529	2538	
	2544#	2564	2573	2579#	2600	2618	2624#	2638	2658	2664#	2671	2690	2696#	
	2703	2722	2728#	2736	2754	2760#	2785	2803	2809#	2828	2846	2852#	2896	
	2913	2919#	2940	2959	2965#	3002	3019	3025#	3043	3060	3066#	3087	3104	
	3110#	3129	3146	3152#	3171	3187	3193#	3198	3214	3220#	3225	3241	3247#	
	3264	3282	3288#	3340	3355	3361#	3381	3396	3402#	3447	3461	3467#	3487	
	3504	3510#	3516	3532	3538#	3544	3561	3567#	3573	3594	3600#	3618	3635	
	3641#	3648	3665	3671#	3686	3705	3711#	3725	3741	3747#	3764	3781	3787#	
	3791	3807	3813#	3817	3833	3839#	3843	3853	3859#	3876	3886	3892#	3909	
	3919	3925#	3952	3962	3968#	4018	4029	4035#	4080	4090	4096#	4124	4134	
	4140#	4188	4198	4204#	4232	4242	4248#	4275	4285	4291#	4318	4334	4340#	
	4357	4373	4379#	4406	4421	4427#	4472	4487	4493#	4538	4554	4560#	4585	
	4603	4609#	4653	4672	4678#	4723	4741	4747#	4792	4809	4815#	4859	4881	
	4887#	4893	4919	4925#	4931	4956	4962#	4968	4987	4993#	4999	5016	5022#	
	5043	5063	5069#	5108	5126	5132#	5177	5195	5201#	5234	5253	5259#	5267	
	5288	5294#	5303	5322	5328#	5333	5351	5357#	5363	5383	5389#	5400	5417	
	5423#	5427	5445	5451#	5464	5482	5488#	5491	5509	5515#	5528	5548	5554#	
	5567	5590	5596#	5600	5623	5629#	5633	5678	5684#	5786	5813	5819#	5950	
	5969	5975#	5985	6008	6014#	6034	6044	6050#	6071	6081	6087#	6107	6117	
	6123#	6145	6167	6173#	6210	6221	6227#	6277	6298	6304#	6310	6321	6327#	
	6348	6358	6364#	6385	6405	6411#	6471	6482	6488#	6524	6545	6551#	6590	

G16

\$TSTM 000334
\$TSTNM= 000304
\$UNIT 000312
\$UNITM 000340
\$USWR 000324
\$X = 025536

6601	6607#	6658	6669	6675#	6681	6703	6709#	6760	6770	6776#	6828	6849
6855#	6907	6917	6923#	6974	6994	6990#	7043	7053	7059#	7115	7136	7142#
7171	7189	7195#	7223	7240	7246#	7267	7284	7290#	7327	7347	7353#	7372
7383	7389#	7403	7413	7419#	7434	7444	7450#	7464	7474	7480#	7494	7504
7510#	7524	7534	7540#	7554	7572	7578#	7595	7605	7611#	7637	7647	7653#
7679	7689	7695#	7721	7731	7737#	7763	7773	7779#	7805	7815	7821#	7847
7865	7871#	7876	7892	7898#	7917	7923#	7957	7963#	7981	7987#	8001	8035
8041#	8092	8098#	8142	8148#	8165	8171#	8177	8201	8207#	8252	8258#	
474#												
522#	528*											
439#												
476#												
446#												
537#	552	572	599#	605	618#	624	637#	643	656#	661	689#	696
709#	715	725	738#	748	762#	769	782#	792	833#	839	852#	857
870#	875	888#	893	906#	916	930#	941	954#	978#	1000#	1010	1024#
1035	1049#	1059	1073#	1084	1099#	1109	1123#	1134	1148#	1158	1172#	1183
1212#	1218	1231#	1236	1249#	1254	1267#	1272	1299#	1306	1319	1346#	1353
1366	1393#	1400	1413	1440#	1447	1460	1509#	1513	1524	1533	1556#	1561
1578	1598#	1602	1613	1635#	1640	1652	1673#	1680	1694	1718#	1726	1741
1765#	1774	1789	1813#	1822	1837	1856#	1866	1883	1902#	1911	1922	1936
1946	1958#	1968	1979	1992	2002	2014#	2024	2039	2048	2075#	2083	2096
2122#	2132	2149	2176#	2185	2201	2214#	2228	2238	2249	2258	2269	2290#
2297	2312	2341#	2348	2361	2383#	2390	2401	2424#	2432	2443	2457#	2467
2477	2486	2498#	2512	2522	2532	2544#	2557	2567	2579#	2590	2603	2624#
2629	2641	2664#	2674	2696#	2706	2728#	2739	2760#	2771	2788	2809#	2820
2831	2852#	2863	2873	2888	2899	2919#	2932	2943	2965#	2979	2994	3005
3025#	3036	3046	3066#	3079	3090	3110#	3122	3132	3152#	3164	3174	3193#
3201	3220#	3228	3247#	3257	3267	3288#	3293	3306	3317	3332	3343	3361#
3373	3384	3402#	3409	3418	3428	3440	3450	3467#	3478	3490	3510#	3519
3538#	3547	3567#	3576	3600#	3611	3621	3641#	3651	3671#	3679	3689	3711#
3719	3728	3747#	3757	3767	3787#	3794	3813#	3820	3839#	3846	3859	3869
3879	3892#	3902	3912	3925#	3934	3945	3955	3968#	3979	3989	4001	4012
4021	4035#	4049	4059	4072	4083	4096#	4108	4118	4127	4140#	4152	4162
4172	4182	4191	4204#	4216	4226	4235	4248#	4259	4269	4278	4291#	4302
4312	4321	4340#	4350	4360	4379#	4388	4399	4409	4427#	4435	4445	4456
4466	4475	4493#	4502	4512	4521	4531	4541	4560#	4569	4579	4588	4609#
4617	4626	4637	4647	4656	4678#	4687	4697	4706	4717	4726	4747#	4756
4766	4775	4786	4795	4815#	4824	4834	4843	4853	4862	4887#	4896	4925#
4934	4962#	4971	4993#	5002	5022#	5031	5046	5069#	5079	5094	5111	5132#
5144	5161	5180	5201#	5209	5223	5237	5259#	5270	5294#	5306	5328#	5336
5357#	5366	5389#	5394	5403	5423#	5430	5451#	5458	5467	5483#	5494	5515#
5522	5531	5554#	5561	5570	5596#	5603	5629#	5636	5684#	5691	5700	5714
5726	5735	5748	5761	5775	5789	5819#	5847	5866	5884	5902	5918	5934
5953	5975#	5988	6014#	6022	6037	6050#	6059	6074	6087#	6096	6110	6123#
6133	6148	6173#	6182	6198	6213	6227#	6236	6250	6264	6280	6304#	6313
6327#	6336	6351	6364#	6373	6388	6411#	6419	6433	6447	6459	6474	6488#
6497	6512	6527	6551#	6561	6577	6593	6607#	6616	6631	6647	6661	6675#
6684	6709#	6718	6733	6748	6763	6776#	6786	6801	6816	6831	6855#	6865
6880	6894	6910	6923#	6933	6948	6963	6977	6990#	7000	7015	7029	7046
7059#	7069	7085	7100	7118	7142#	7154	7174	7195#	7208	7226	7246#	7254
7270	7290#	7309	7330	7353#	7360	7375	7389#	7397	7406	7419#	7428	7437
7450#	7458	7467	7480#	7488	7497	7510#	7518	7527	7540#	7548	7557	7578#
7584	7598	7611#	7621	7631	7640	7653#	7663	7673	7682	7695#	7705	7715
7724	7737#	7747	7757	7766	7779#	7789	7799	7808	7821#	7831	7841	7850
7871#	7879	7898#	7923#	7936	7963#	7987#	8004	8041#	8098#	8128	8148#	8171#

\$XX = 177716

8180	8207#	8220	8242	8258#									
552#	572#	605#	624#	643#	661#	696#	715#	725#	748#	769#	792#	839#	
857#	875#	893#	916#	941#	1010#	1035#	1059#	1084#	1109#	1134#	1158#	1183#	
1218#	1236#	1254#	1272#	1306#	1319#	1353#	1366#	1400#	1413#	1447#	1460#	1513#	
1524#	1533#	1561#	1578#	1602#	1613#	1640#	1652#	1680#	1694#	1726#	1741#	1774#	
1789#	1822#	1837#	1866#	1883#	1911#	1922#	1936#	1946#	1968#	1979#	1992#	2002#	
2024#	2039#	2048#	2083#	2096#	2132#	2149#	2185#	2201#	2228#	2238#	2249#	2258#	
2269#	2297#	2312#	2348#	2361#	2390#	2401#	2432#	2443#	2467#	2477#	2486#	2512#	
2522#	2532#	2557#	2567#	2590#	2603#	2629#	2641#	2674#	2706#	2739#	2771#	2788#	
2820#	2831#	2863#	2873#	2888#	2899#	2932#	2943#	2979#	2994#	3005#	3036#	3046#	
3079#	3090#	3122#	3132#	3164#	3174#	3201#	3228#	3257#	3267#	3293#	3306#	3317#	
3332#	3343#	3373#	3384#	3409#	3418#	3428#	3440#	3450#	3478#	3490#	3519#	3547#	
3576#	3611#	3621#	3651#	3679#	3689#	3719#	3728#	3757#	3767#	3794#	3820#	3846#	
3869#	3879#	3902#	3912#	3934#	3945#	3955#	3979#	3989#	4001#	4012#	4021#	4049#	
4059#	4072#	4083#	4108#	4118#	4127#	4152#	4162#	4172#	4182#	4191#	4216#	4226#	
4235#	4259#	4269#	4278#	4302#	4312#	4321#	4350#	4360#	4388#	4399#	4409#	4435#	
4445#	4456#	4466#	4475#	4502#	4512#	4521#	4531#	4541#	4569#	4579#	4588#	4617#	
4626#	4637#	4647#	4656#	4687#	4697#	4706#	4717#	4726#	4756#	4766#	4775#	4786#	
4795#	4824#	4834#	4843#	4853#	4862#	4896#	4934#	4971#	5002#	5031#	5046#	5079#	
5094#	5111#	5144#	5161#	5180#	5209#	5223#	5237#	5270#	5306#	5336#	5366#	5394#	
5403#	5430#	5458#	5467#	5494#	5522#	5531#	5561#	5570#	5603#	5636#	5691#	5700#	
5714#	5726#	5735#	5748#	5761#	5775#	5789#	5847#	5866#	5884#	5902#	5918#	5934#	
5953#	5988#	6022#	6037#	6059#	6074#	6096#	6110#	6133#	6148#	6182#	6198#	6213#	
6236#	6250#	6264#	6280#	6313#	6336#	6351#	6373#	6388#	6419#	6433#	6447#	6459#	
6474#	6497#	6512#	6527#	6561#	6577#	6593#	6616#	6631#	6647#	6661#	6684#	6718#	
6733#	6748#	6763#	6786#	6801#	6816#	6831#	6865#	6880#	6894#	6910#	6933#	6948#	

\$XXX = 000716

8180	8207#	8220	8242	8258#									
552#	572#	605#	624#	643#	661#	696#	715#	725#	748#	769#	792#	839#	
857#	875#	893#	916#	941#	1010#	1035#	1059#	1084#	1109#	1134#	1158#	1183#	
1218#	1236#	1254#	1272#	1306#	1319#	1353#	1366#	1400#	1413#	1447#	1460#	1513#	
1524#	1533#	1561#	1578#	1602#	1613#	1640#	1652#	1680#	1694#	1726#	1741#	1774#	
1789#	1822#	1837#	1866#	1883#	1911#	1922#	1936#	1946#	1968#	1979#	1992#	2002#	
2024#	2039#	2048#	2083#	2096#	2132#	2149#	2185#	2201#	2228#	2238#	2249#	2258#	
2269#	2297#	2312#	2348#	2361#	2390#	2401#	2432#	2443#	2467#	2477#	2486#	2512#	
2522#	2532#	2557#	2567#	2590#	2603#	2629#	2641#	2674#	2706#	2739#	2771#	2788#	
2820#	2831#	2863#	2873#	2888#	2899#	2932#	2943#	2979#	2994#	3005#	3036#	3046#	
3079#	3090#	3122#	3132#	3164#	3174#	3201#	3228#	3257#	3267#	3293#	3306#	3317#	
3332#	3343#	3373#	3384#	3409#	3418#	3428#	3440#	3450#	3478#	3490#	3519#	3547#	
3576#	3611#	3621#	3651#	3679#	3689#	3719#	3728#	3757#	3767#	3794#	3820#	3846#	
3869#	3879#	3902#	3912#	3934#	3945#	3955#	3979#	3989#	4001#	4012#	4021#	4049#	
4059#	4072#	4083#	4108#	4118#	4127#	4152#	4162#	4172#	4182#	4191#	4216#	4226#	
4235#	4259#	4269#	4278#	4302#	4312#	4321#	4350#	4360#	4388#	4399#	4409#	4435#	
4445#	4456#	4466#	4475#	4502#	4512#	4521#	4531#	4541#	4569#	4579#	4588#	4617#	
4626#	4637#	4647#	4656#	4687#	4697#	4706#	4717#	4726#	4756#	4766#	4775#	4786#	
4795#	4824#	4834#	4843#	4853#	4862#	4896#	4934#	4971#	5002#	5031#	5046#	5079#	
5094#	5111#	5144#	5161#	5180#	5209#	5223#	5237#	5270#	5306#	5336#	5366#	5394#	
5403#	5430#	5458#	5467#	5494#	5522#	5531#	5561#	5570#	5603#	5636#	5691#	5700#	
5714#	5726#	5735#	5748#	5761#	5775#	5789#	5847#	5866#	5884#	5902#	5918#	5934#	
5953#	5988#	6022#	6037#	6059#	6074#	6096#	6110#	6133#	6148#	6182#	6198#	6213#	
6236#	6250#	6264#	6280#	6313#	6336#	6351#	6373#	6388#	6419#	6433#	6447#	6459#	
6474#	6497#	6512#	6527#	6561#	6577#	6593#	6616#	6631#	6647#	6661#	6684#	6718#	
6733#	6748#	6763#	6786#	6801#	6816#	6831#	6865#	6880#	6894#	6910#	6933#	6948#	

= 026216

6963#	6977#	7000#	7015#	7029#	7046#	7069#	7085#	7100#	7118#	7154#	7174#	7208#
7226#	7254#	7270#	7309#	7330#	7360#	7375#	7397#	7406#	7428#	7437#	7458#	7467#
7488#	7497#	7518#	7527#	7548#	7557#	7584#	7598#	7621#	7631#	7640#	7663#	7673#
7682#	7705#	7715#	7724#	7747#	7757#	7766#	7789#	7799#	7808#	7831#	7841#	7850#
7879#	7936#	8004#	8128#	8180#	8220#	8242#						
417#	422	423#	425#	427#	428#	461	462#	464#	456#	481#	487#	492#
495#	504#	508#	511	512#	520#	537	552	572	599	605	618	624
637	643	656	661	689	696	709	715	725	738	748	762	769
782	792	833	839	852	857	870	875	888	893	906	916	930
941	954	978	1000	1010	1024	1035	1049	1059	1073	1084	1099	1109
1123	1134	1148	1158	1172	1183	1212	1218	1231	1236	1249	1254	1267
1272	1299	1306	1319	1346	1353	1366	1393	1400	1413	1440	1447	1460
1509	1513	1524	1533	1556	1561	1578	1598	1602	1613	1635	1640	1652
1673	1680	1694	1718	1726	1741	1765	1774	1789	1813	1822	1837	1856
1866	1883	1902	1911	1922	1936	1946	1958	1968	1979	1992	2002	2014
2024	2039	2048	2075	2083	2096	2122	2132	2149	2176	216	2201	2214
2228	2238	2249	2258	2269	2290	2297	2312	2341	2348	2361	2383	2390
2401	2424	2432	2443	2457	2467	2477	2486	2498	2512	2522	2532	2544
2557	2567	2579	2590	2603	2624	2629	2641	2664	2674	2696	2706	2728
2739	2760	2771	2788	2809	2820	2831	2852	2863	2873	2888	2899	2919
2932	2943	2965	2979	2994	3005	3025	3036	3046	3066	3079	3090	3110
3122	3132	3152	3164	3174	3193	3201	3220	3228	3247	3257	3267	3288
3293	3306	3317	3332	3343	3361	3373	3384	3402	3409	3418	3428	3440
3450	3467	3478	3490	3510	3519	3538	3547	3567	3576	3600	3611	3621
3641	3651	3671	3679	3689	3711	3719	3728	3747	3757	3767	3787	3794
3813	3820	3839	3846	3859	3869	3879	3892	3902	3912	3925	3934	3945
3955	3968	3979	3989	4001	4012	4021	4035	4049	4059	4072	4083	4096
4108	4118	4127	4140	4152	4162	4172	4182	4191	4204	4216	4226	4235
4248	4259	4269	4278	4291	4302	4312	4321	4340	4350	4360	4379	4388
4399	4409	4427	4435	4445	4456	4466	4475	4493	4502	4512	4521	4531
4541	4560	4569	4579	4588	4609	4617	4626	4637	4647	4656	4678	4687
4697	4706	4717	4726	4747	4756	4766	4775	4786	4795	4815	4824	4834
4843	4853	4862	4887	4896	4925	4934	4962	4971	4993	5002	5022	5031
5046	5069	5079	5094	5111	5132	5144	5161	5180	5201	5209	5223	5237
5259	5270	5294	5306	5328	5336	5357	5366	5389	5394	5403	5423	5430
5451	5458	5467	5488	5494	5515	5522	5531	5554	5561	5570	5596	5603
5629	5636	5684	5687	5691	5700	5714	5726	5735	5748	5761	5775	5789
5819	5847	5866	5884	5902	5918	5934	5953	5975	5988	6014	6022	6037
6050	6059	6074	6087	6096	6110	6123	6133	6148	6173	6182	6198	6213
6227	6236	6250	6264	6280	6304	6313	6327	6336	6351	6364	6373	6388
6411	6419	6433	6447	6459	6474	6488	6497	6512	6527	6551	6561	6577
6593	6607	6616	6631	6647	6661	6675	6684	6709	6718	6733	6748	6763
6776	6786	6801	6816	6831	6855	6865	6880	6894	6910	6923	6933	6948
6963	6977	6990	7000	7015	7029	7046	7059	7069	7085	7100	7118	7142
7154	7174	7195	7208	7226	7246	7254	7270	7290	7309	7330	7353	7360
7375	7389	7397	7406	7419	7428	7437	7450	7458	7467	7480	7488	7497
7510	7518	7527	7540	7548	7557	7578	7584	7598	7611	7621	7631	7640
7653	7663	7673	7682	7695	7705	7715	7724	7737	7747	7757	7766	7779
7789	7799	7808	7821	7831	7841	7850	7871	7879	7898	7923	7936	7963
7987	8004	8041	8052	8077	8082	8098	8128	8148	8171	8180	8207	8220
8242	8258	8291	8292	8293	8294	8295	8296	8297	8298	8299	8300	8301
8302	8303	8304	8305	8344#								
461#	466	511#	520									

.\$X = 000500

COMMEN	1#														
ENDCOM	1#														
ERROR	404#	544	548	564	568	602	621	640	658	693	712	722	745	766	789
	836	854	872	890	913	938	967	988	1007	1032	1056	1081	1106	1131	1155
	1120	1215	1233	1251	1269	1303	1316	1350	1363	1397	1410	1444	1457	1510	1521
	1530	1558	1575	1599	1610	1637	1649	1677	1691	1723	1738	1771	1786	1819	1834
	1863	1880	1908	1919	1933	1943	1965	1976	1989	1999	2021	2036	2045	2080	2093
	2129	2146	2182	2198	2225	2235	2246	2255	2266	2294	2309	2345	2358	2387	2398
	2429	2440	2464	2474	2483	2509	2519	2529	2554	2564	2587	2600	2626	2638	2671
	2703	2736	2768	2785	2817	2828	2860	2870	2885	2896	2929	2940	2976	2991	3002
	3033	3043	3076	3087	3119	3129	3161	3171	3198	3225	3254	3264	3290	3303	3314
	3329	3340	3370	3381	3406	3415	3425	3437	3447	3475	3487	3516	3544	3573	3608
	3618	3648	3676	3686	3716	3725	3754	3764	3791	3817	3843	3866	3876	3899	3909
	3931	3942	3952	3976	3986	3998	4009	4018	4046	4056	4069	4080	4105	4115	4124
	4149	4159	4169	4179	4188	4213	4223	4232	4256	4266	4275	4299	4309	4318	4347
	4357	4385	4396	4406	4432	4442	4453	4463	4472	4499	4509	4518	4528	4538	4566
	4576	4585	4614	4623	4634	4644	4653	4684	4694	4703	4714	4723	4753	4763	4772
	4783	4792	4821	4831	4840	4850	4859	4893	4931	4968	4999	5028	5043	5076	5091
	5108	5141	5158	5177	5206	5220	5234	5267	5303	5333	5363	5391	5400	5427	5455
	5464	5491	5519	5528	5558	5567	5600	5633	5688	5697	5711	5723	5732	5745	5758
	5772	5786	5830	5844	5863	5881	5899	5915	5931	5950	5981	5985	6019	6034	6056
	6071	6093	6107	6130	6145	6179	6195	6210	6233	6247	6261	6277	6310	6333	6348
	6370	6385	6416	6430	6444	6456	6471	6494	6509	6524	6558	6574	6590	6613	6628
	6644	6658	6681	6715	6730	6745	6760	6783	6798	6813	6828	6862	6877	6891	6907
	6930	6945	6960	6974	6997	7012	7026	7043	7066	7082	7097	7115	7151	7171	7205
	7223	7251	7267	7297	7306	7321	7327	7357	7372	7394	7403	7425	7434	7455	7464
	7485	7494	7515	7524	7545	7554	7581	7595	7618	7628	7637	7660	7670	7679	7702
	7712	7721	7744	7754	7763	7786	7796	7805	7828	7838	7847	7876	7905	7928	7933
	7945	7970	7976	7996	8001	8070	8125	8154	8177	8217	8239	8280	8352	8356	8360
	8364	8368	8372	8376	8380										
ESCAPE	1#														
GETPRI	1#														
GETSWR	1#														
JNE	7125#	8050	8075	8080											
LOOP	404#	552	572	605	624	643	661	696	715	725	748	769	792	839	857
	875	893	916	941	1010	1035	1059	1084	1109	1134	1158	1183	1218	1236	1254
	1272	1306	1319	1353	1366	1400	1413	1447	1460	1513	1524	1533	1561	1578	1602
	1613	1640	1652	1680	1694	1726	1741	1774	1789	1822	1837	1866	1883	1911	1922
	1936	1946	1968	1979	1992	2002	2024	2039	2048	2083	2096	2132	2149	2185	2201
	2228	2238	2249	2258	2269	2297	2312	2348	2361	2390	2401	2432	2443	2467	2477
	2486	2512	2522	2532	2557	2567	2590	2603	2629	2641	2674	2706	2739	2771	2788
	2820	2831	2863	2873	2888	2899	2932	2943	2979	2994	3005	3036	3046	3079	3090
	3122	3132	3164	3174	3201	3228	3257	3267	3293	3306	3317	3332	3343	3373	3384
	3409	3418	3428	3440	3450	3478	3490	3519	3547	3576	3611	3621	3651	3679	3689
	3719	3728	3757	3767	3794	3820	3846	3869	3879	3902	3912	3934	3945	3955	3979
	3989	4001	4012	4021	4049	4059	4072	4083	4108	4118	4127	4152	4162	4172	4182
	4191	4216	4226	4235	4259	4269	4278	4302	4312	4321	4350	4360	4388	4399	4409
	4435	4445	4456	4466	4475	4502	4512	4521	4531	4541	4569	4579	4588	4617	4626
	4637	4647	4656	4687	4697	4706	4717	4726	4756	4766	4775	4786	4795	4824	4834
	4843	4853	4862	4896	4934	4971	5002	5031	5046	5079	5094	5111	5144	5161	5180
	5209	5223	5237	5270	5306	5336	5366	5394	5403	5430	5458	5467	5494	5522	5531
	5561	5570	5603	5636	5691	5700	5714	5726	5735	5748	5761	5775	5789	5847	5866
	5884	5902	5918	5934	5953	5988	6022	6037	6059	6074	6096	6110	6133	6148	6182
	6198	6213	6236	6250	6264	6280	6313	6336	6351	6373	6388	6419	6433	6447	6459
	6474	6497	6512	6527	6561	6577	6593	6616	6631	6647	6661	6684	6718	6733	6748
	6763	6786	6801	6816	6831	6865	6880	6894	6910	6933	6948	6963	6977	7000	7015

K16

	7029	7046	7069	7085	7100	7118	7154	7174	7208	7226	7254	7270	7309	7330	7360
	7375	7397	7406	7428	7437	7458	7467	7488	7497	7518	7527	7548	7557	7584	7598
	7621	7631	7640	7663	7673	7682	7705	7715	7724	7747	7757	7766	7789	7799	7808
	7831	7841	7850	7879	7936	8004	8128	8180	8220	8242					
MULT	1#														
NEWST	1#	404#	531	593	612	631	650	683	703	732	756	776	827	846	864
	882	900	924	948	972	994	1018	1043	1067	1093	1117	1142	1166	1206	1225
	1243	1261	1293	1340	1387	1434	1503	1550	1592	1629	1667	1712	1759	1807	1850
	1896	1952	2008	2069	2116	2170	2208	2284	2335	2377	2418	2451	2492	2538	2573
	2618	2658	2690	2722	2754	2803	2846	2913	2959	3019	3060	3104	3146	3187	3214
	3241	3282	3355	3396	3461	3504	3532	3561	3594	3635	3665	3705	3741	3781	3807
	3833	3853	3886	3919	3962	4029	4090	4134	4198	4242	4285	4334	4373	4421	4487
	4554	4603	4672	4741	4809	4881	4919	4956	4987	5016	5063	5126	5195	5253	5288
	5322	5351	5383	5417	5445	5482	5509	5548	5590	5623	5678	5813	5969	6008	6044
	6081	6117	6167	6221	6298	6321	6358	6405	6482	6545	6601	6669	6703	6770	6849
	6917	6984	7053	7136	7189	7240	7284	7347	7383	7413	7444	7474	7504	7534	7572
	7605	7647	7689	7731	7773	7815	7865	7892	7917	7957	7981	8035	8092	8142	8165
	8201	8252													
POP	1#														
PUSH	1#														
REPORT	1#														
SETPRI	1#														
SETUP	1#														
SKIP	1#														
SLASH	1#														
STARS	1#	404#	420	431	458	460	467	478	480	501	503	509	531	533	579
	593	595	612	614	631	633	650	652	668	683	685	703	705	732	734
	756	758	776	778	800	827	829	846	848	864	866	882	884	900	902
	924	926	948	950	972	974	994	996	1018	1020	1043	1045	1067	1069	1093
	1095	1117	1119	1142	1144	1166	1168	1191	1206	1208	1225	1227	1243	1245	1261
	1263	1280	1293	1295	1327	1340	1342	1374	1387	1389	1421	1434	1436	1468	1486
	1490	1503	1505	1541	1550	1552	1586	1592	1594	1621	1629	1631	1660	1667	1669
	1703	1712	1714	1749	1759	1761	1797	1807	1809	1845	1850	1852	1891	1896	1898
	1952	1954	2008	2010	2055	2069	2071	2104	2116	2118	2157	2170	2172	2208	2210
	2276	2284	2286	2320	2335	2337	2369	2377	2379	2409	2418	2420	2451	2453	2492
	2494	2538	2540	2573	2575	2610	2618	2620	2650	2658	2660	2682	2690	2692	2714
	2722	2724	2747	2754	2756	2796	2803	2805	2838	2846	2848	2906	2913	2915	2950
	2959	2961	3011	3019	3021	3053	3060	3062	3097	3104	3106	3139	3146	3148	3181
	3187	3189	3208	3214	3216	3235	3241	3243	3274	3282	3284	3350	3355	3357	3390
	3396	3398	3456	3461	3463	3497	3504	3506	3526	3532	3534	3554	3561	3563	3583
	3594	3596	3628	3635	3637	3658	3665	3667	3696	3705	3707	3734	3741	3743	3773
	3781	3783	3800	3807	3809	3826	3833	3835	3853	3855	3886	3888	3919	3921	3962
	3964	4029	4031	4090	4092	4134	4136	4198	4200	4242	4244	4285	4287	4328	4334
	4336	4367	4373	4375	4416	4421	4423	4482	4487	4489	4547	4554	4556	4595	4603
	4605	4663	4672	4674	4733	4741	4743	4802	4809	4811	4869	4881	4883	4910	4919
	4921	4946	4956	4958	4977	4987	4989	5008	5016	5018	5054	5063	5065	5119	5126
	5128	5188	5195	5197	5245	5253	5255	5278	5288	5290	5315	5322	5324	5344	5351
	5353	5376	5383	5385	5410	5417	5419	5438	5445	5447	5475	5482	5484	5502	5509
	5511	5539	5548	5550	5581	5590	5592	5613	5623	5625	5647	5678	5680	5797	5813
	5815	5962	5969	5971	5994	6008	6010	6044	6046	6081	6083	6117	6119	6156	6167
	6169	6221	6223	6289	6298	6300	6321	6323	6358	6360	6395	6405	6407	6482	6484
	6535	6545	6547	6601	6603	6669	6671	6693	6703	6705	6770	6772	6839	6849	6851
	6917	6919	6984	6986	7053	7055	7128	7136	7138	7181	7189	7191	7233	7240	7242
	7277	7284	7286	7338	7347	7349	7383	7385	7413	7415	7444	7446	7474	7476	7504
	7506	7534	7536	7564	7572	7574	7605	7607	7647	7649	7689	7691	7731	7733	7773
	7775	7815	7817	7857	7865	7867	7887	7892	7894	7910	7917	7919	7952	7957	7959

	7981	7983	8012	8035	8037	8085	8092	8094	8136	8142	8144	8159	8165	8167	8187
SWRSU	1#														
TYPBIN	1#														
TYPDEC	1#														
TYPNAM	1#														
TYPNUM	1#														
TYPOCS	1#														
TYPOCT	1#														
TYPTXT	1#														
\$\$ERCD	404#	545	554	565	574	607	626	645	663	698	717	727	751	771	795
	841	859	877	895	919	944	967	989	1013	1038	1062	1087	1112	1137	1161
	1186	1220	1238	1256	1274	1309	1322	1356	1369	1403	1416	1450	1463	1515	1526
	1535	1563	1581	1604	1616	1642	1655	1682	1697	1728	1744	1776	1792	1824	1840
	1868	1886	1914	1924	1939	1948	1971	1981	1995	2004	2027	2041	2050	2085	2099
	2134	2152	2187	2204	2231	2240	2251	2260	2271	2299	2315	2350	2364	2392	2404
	2434	2446	2470	2479	2488	2515	2524	2534	2560	2569	2593	2605	2631	2644	2677
	2709	2742	2774	2791	2823	2833	2866	2875	2891	2901	2935	2945	2982	2997	3007
	3039	3048	3082	3092	3125	3134	3167	3176	3203	3230	3260	3269	3295	3308	3319
	3334	3345	3376	3386	3411	3420	3430	3442	3452	3481	3493	3521	3549	3578	3613
	3623	3653	3681	3691	3721	3730	3759	3769	3796	3822	3848	3872	3881	3905	3914
	3937	3948	3957	3982	3991	4004	4014	4023	4052	4061	4075	4085	4111	4120	4129
	4155	4164	4175	4184	4193	4219	4228	4237	4262	4271	4280	4305	4314	4323	4353
	4362	4391	4402	4411	4438	4447	4459	4468	4477	4505	4514	4523	4533	4543	4572
	4581	4590	4619	4628	4640	4649	4658	4690	4699	4708	4719	4728	4759	4768	4777
	4788	4797	4827	4836	4845	4855	4864	4899	4937	4973	5004	5034	5049	5082	5097
	5114	5147	5164	5183	5212	5226	5240	5273	5309	5339	5369	5396	5405	5432	5460
	5469	5496	5524	5533	5563	5573	5606	5639	5693	5702	5716	5728	5737	5750	5763
	5777	5791	5831	5850	5869	5887	5905	5921	5937	5956	5981	5990	6025	6040	6062
	6077	6099	6113	6136	6151	6185	6201	6216	6239	6253	6267	6283	6316	6339	6354
	6376	6391	6422	6436	6450	6462	6477	6500	6515	6530	6564	6580	6596	6619	6634
	6650	6664	6687	6721	6736	6751	6766	6789	6804	6819	6834	6868	6883	6897	6913
	6936	6951	6966	6980	7003	7018	7032	7049	7072	7088	7103	7121	7157	7177	7211
	7229	7257	7273	7297	7312	7321	7333	7362	7378	7399	7408	7430	7439	7460	7469
	7490	7499	7520	7529	7550	7559	7586	7600	7624	7633	7642	7666	7675	7684	7708
	7717	7726	7750	7759	7768	7792	7801	7810	7834	7843	7852	7881	7905	7928	7938
	7945	7970	7976	7996	8006	8070	8131	8154	8183	8222	8245	8281	8353	8357	8361
\$\$ERNU	404#	545	554	565	574	607	626	645	663	699	717	727	751	771	795
	841	859	877	895	919	944	967	989	1013	1038	1062	1087	1112	1137	1161
	1186	1220	1238	1256	1274	1309	1322	1356	1369	1403	1416	1450	1463	1515	1526
	1535	1563	1581	1604	1616	1642	1655	1682	1697	1728	1744	1776	1792	1824	1840
	1868	1886	1914	1924	1939	1948	1971	1981	1995	2004	2027	2041	2050	2085	2099
	2134	2152	2187	2204	2231	2240	2251	2260	2271	2299	2315	2350	2364	2392	2404
	2434	2446	2470	2479	2488	2515	2524	2534	2560	2569	2593	2605	2631	2644	2677
	2709	2742	2774	2791	2823	2833	2866	2875	2891	2901	2935	2945	2982	2997	3007
	3039	3048	3082	3092	3125	3134	3167	3176	3203	3230	3260	3269	3295	3308	3319
	3334	3345	3376	3386	3411	3420	3430	3442	3452	3481	3493	3521	3549	3578	3613
	3623	3653	3681	3691	3721	3730	3759	3769	3796	3822	3848	3872	3881	3905	3914
	3937	3948	3957	3982	3991	4004	4014	4023	4052	4061	4075	4085	4111	4120	4129
	4155	4164	4175	4184	4193	4219	4228	4237	4262	4271	4280	4305	4314	4323	4353
	4362	4391	4402	4411	4438	4447	4459	4468	4477	4505	4514	4523	4533	4543	4572
	4581	4590	4619	4628	4640	4649	4658	4690	4699	4708	4719	4728	4759	4768	4777
	4788	4797	4827	4836	4845	4855	4864	4899	4937	4973	5004	5034	5049	5082	5097
	5114	5147	5164	5183	5212	5226	5240	5273	5309	5339	5369	5396	5405	5432	5460
	5469	5496	5524	5533	5563	5573	5606	5639	5693	5702	5716	5728	5737	5750	5763

M16

	5777	5791	5831	5850	5869	5887	5905	5921	5937	5956	5981	5990	6025	6040	6062
	6077	6099	6113	6136	6151	6185	6201	6216	6239	6253	6267	6283	6316	6339	6354
	6376	6391	6422	6436	6450	6462	6477	6500	6515	6530	6564	6580	6596	6619	6634
	6650	6664	6687	6721	6736	6751	6766	6789	6804	6819	6834	6868	6883	6897	6913
	6936	6951	6966	6980	7003	7018	7032	7049	7072	7088	7103	7121	7157	7177	7211
	7229	7257	7273	7297	7312	7321	7333	7362	7378	7399	7408	7430	7439	7460	7469
	7490	7499	7520	7529	7550	7559	7586	7600	7624	7633	7642	7666	7675	7684	7708
	7717	7726	7750	7759	7768	7792	7801	7810	7834	7843	7852	7881	7905	7928	7938
	7945	7970	7976	7996	8006	8070	8131	8154	8183	8222	8245	8281	8353	8357	8361
	8365	8369	8373	8377	8381										
\$\$ERRO	404#	569	602	621	640	658	693	712	722	745	766	789	836	854	872
	890	913	938	1007	1032	1056	1081	1106	1131	1155	1180	1215	1233	1251	1269
	1316	1363	1410	1457	1530	1575	1610	1649	1691	1738	1786	1834	1880	1943	1999
	2045	2093	2146	2198	2266	2309	2358	2398	2440	2483	2529	2564	2600	2638	2671
	2703	2736	2785	2828	2896	2940	3002	3043	3087	3129	3171	3198	3225	3264	3340
	3381	3447	3487	3516	3544	3573	3618	3648	3686	3725	3764	3791	3817	3843	3876
	3909	3952	4018	4080	4124	4188	4232	4275	4318	4357	4406	4472	4538	4585	4653
	4723	4792	4859	4893	4931	4968	4999	5043	5108	5177	5234	5267	5303	5333	5363
	5400	5427	5464	5491	5528	5567	5600	5633	5786	5950	5985	6034	6071	6107	6145
	6210	6277	6310	6348	6385	6471	6524	6590	6658	6681	6760	6828	6907	6974	7043
	7115	7171	7223	7267	7327	7372	7403	7434	7464	7494	7524	7554	7595	7637	7679
	7721	7763	7805	7847	7876	8001	8177								
\$\$ESCA	1#														
\$\$LOOP	404#	552	572	605	624	643	661	696	715	725	748	769	792	839	857
	875	893	916	941	1010	1035	1059	1084	1109	1134	1158	1183	1218	1236	1254
	1272	1306	1319	1353	1366	1400	1413	1447	1460	1513	1524	1533	1561	1578	1602
	1613	1640	1652	1680	1694	1726	1741	1774	1789	1822	1837	1866	1883	1911	1922
	1936	1946	1968	1979	1992	2002	2024	2039	2048	2083	2096	2132	2149	2185	2201
	2228	2238	2249	2258	2269	2297	2312	2348	2361	2390	2401	2432	2443	2467	2477
	2486	2512	2522	2532	2557	2567	2590	2603	2629	2641	2674	2706	2739	2771	2788
	2820	2831	2863	2873	2888	2899	2932	2943	2979	2994	3005	3036	3046	3079	3090
	3122	3132	3164	3174	3201	3228	3257	3267	3293	3306	3317	3332	3343	3373	3384
	3409	3418	3428	3440	3450	3478	3490	3519	3547	3576	3611	3621	3651	3679	3699
	3719	3728	3757	3767	3794	3820	3846	3869	3879	3902	3912	3934	3945	3955	3979
	3989	4001	4012	4021	4049	4059	4072	4083	4108	4118	4127	4152	4162	4172	4182
	4191	4216	4226	4235	4259	4269	4278	4302	4312	4321	4350	4360	4388	4399	4409
	4435	4445	4456	4466	4475	4502	4512	4521	4531	4541	4569	4579	4588	4617	4626
	4637	4647	4656	4687	4697	4706	4717	4726	4756	4766	4775	4786	4795	4824	4834
	4843	4853	4862	4896	4934	4971	5002	5031	5046	5079	5094	5111	5144	5161	5180
	5209	5223	5237	5270	5306	5336	5366	5394	5403	5430	5458	5467	5494	5522	5531
	5561	5570	5603	5636	5691	5700	5714	5726	5735	5748	5761	5775	5789	5847	5866
	5884	5902	5918	5934	5953	5988	6022	6037	6059	6074	6096	6110	6133	6148	6182
	6198	6213	6236	6250	6264	6280	6313	6336	6351	6373	6388	6419	6433	6447	6459
	6474	6497	6512	6527	6561	6577	6593	6616	6631	6647	6661	6684	6718	6733	6748
	6763	6786	6801	6816	6831	6865	6880	6894	6910	6933	6948	6963	6977	7000	7015
	7029	7046	7059	7085	7100	7118	7154	7174	7208	7226	7254	7270	7309	7330	7360
	7375	7397	7406	7428	7437	7458	7467	7488	7497	7518	7527	7548	7557	7584	7598
	7621	7631	7640	7663	7673	7682	7705	7715	7724	7747	7757	7766	7789	7799	7808
	7831	7841	7850	7879	7936	8004	8128	8180	8220	8242					
\$\$NEWT	1#	404#	531	593	612	631	650	683	703	732	756	776	827	846	864
	882	900	924	948	972	994	1018	1043	1067	1093	1117	1142	1166	1206	1225
	1243	1261	1293	1340	1387	1434	1503	1550	1592	1629	1667	1712	1759	1807	1850
	1896	1952	2008	2069	2116	2170	2208	2284	2335	2377	2418	2451	2492	2538	2573
	2618	2658	2690	2722	2754	2803	2846	2913	2959	3019	3060	3104	3146	3187	3214
	3241	3282	3355	3396	3461	3504	3532	3561	3594	3635	3665	3705	3741	3781	3807
	3833	3853	3886	3919	3962	4029	4090	4134	4198	4242	4285	4334	4373	4421	4487

ADC	1567	1648	6491	6506	6520										
ADD	3196	3327	3366	3515	4889	4927	4964	4995	6413	6427	6441	6453	6467		
ASL	6993	7008	7023	7037											
ASR	7062	7078	7093	7109											
BCC	712	744	788	912	960	1006	1055	1105	1154	1443	3865	3974	4045	4104	4211
	4255	5026	5041	5089	5106	5136	5218	5232	5299	5331	6142	6194	6469	6555	6572
	6642	6758	6810	6859	6874	6928	6943	6994	7009	7064	7080	7149	7615	7699	7826
	7902	8304													
BDS	936	984	1030	1079	1129	1178	1444	1456	1908	1933	1965	1989	2021	2225	2464
	2509	2554	2587	2670	2734	2783	2815	2883	2927	2989	3254	3370	3475	3898	4148
	4298	5074	5154	5172	5204	5263	5361	6127	6233	6277	6309	6332	6383	6522	6588
	6780	6795	7038	7167	7371	7657	7741	7784	8177	8305					
BEQ	538	569	602	621	640	658	693	722	742	745	766	786	789	836	854
	872	890	910	913	934	938	958	961	982	986	1004	1007	1028	1032	1053
	1056	1077	1081	1103	1106	1127	1131	1152	1155	1176	1180	1215	1233	1251	1269
	1303	1315	1510	1530	1575	1599	1610	1637	1649	1677	1691	1723	1738	1771	1786
	1819	1834	1863	1880	1906	1919	1931	1943	1963	1976	1987	1999	2019	2036	2045
	2080	2093	2129	2146	2182	2198	2223	2235	2266	2294	2309	2345	2358	2387	2398
	2429	2440	2462	2474	2483	2507	2519	2529	2552	2564	2585	2600	2626	2638	2671
	2736	2785	2817	2828	2870	2885	2896	2929	2940	2991	3043	3087	3129	3171	3198
	3225	3252	3264	3290	3303	3314	3329	3340	3368	3381	3425	3437	3473	3487	3516
	3544	3573	3608	3648	3676	3686	3716	3725	3754	3764	3791	3817	3843	3863	3876
	3899	3909	3930	3942	3952	3973	3986	3996	4009	4018	4043	4056	4067	4080	4115
	4124	4147	4159	4168	4179	4188	4212	4223	4232	4253	4266	4275	4299	4309	4318
	4346	4357	4385	4396	4406	4431	4442	4452	4463	4472	4498	4509	4518	4528	4538
	4565	4576	4585	4614	4623	4633	4644	4653	4683	4694	4703	4714	4723	4752	4763
	4772	4783	4792	4820	4831	4840	4850	4859	4893	4931	4968	4999	5028	5043	5076
	5091	5108	5141	5158	5177	5206	5220	5234	5267	5303	5333	5363	5400	5427	5455
	5464	5491	5519	5528	5558	5567	5600	5633	5688	5697	5711	5723	5732	5745	5758
	5772	5786	5844	5863	5881	5899	5915	5931	5950	5985	6130	6144	6195	6231	6275
	6310	6333	6385	6471	6524	6557	6573	6643	6759	6783	6798	6812	6861	6876	6907
	6929	6944	6959	6996	7011	7039	7043	7065	7081	7096	7113	7115	7147	7151	7171
	7202	7205	7220	7223	7267	7306	7327	7357	7394	7403	7425	7434	7455	7464	7485
	7494	7515	7524	7545	7554	7581	7595	7617	7628	7637	7659	7670	7679	7701	7712
	7721	7744	7754	7763	7785	7796	7805	7827	7838	7847	7876	7903	7926	7944	7968
	7974	7994	8001	8125	8217	8239	8264	8269	8273	8293					
BGE	8294														
BGT	3406	8296													
BHI	6032	6069	6105	6245	6442	6507	6611	6679	6713	7249	7263	7300	8300		
BIC	3301	3326	6090	6104	6270	6567	6653	6822	7075	7904	7927	7931	7949	7969	7975
	7995	8051	8076	8081	8214	8236									
BICB	3714	4890	4928	4965	4996										
BIS	3311	3789	3927	3970	4037	4098	4099	4142	4143	4206	4207	4250	4293	4294	6126
	6141	6188	6503	6637	6724	7106	7898	7923	7942	8105	8106	8107	8108	8109	8110
	8111	8148													
BISB	3542	3646	3815	3841	4891	4929	4966	4997							
BIT	3436	3446	3895	3928	4101	4209	4252	6053	6068	8049	8151				
BITB	3862	3972	3995	4042	4066	4145	4167	4295	8261	8331					
BLE	8228	8251	8297												
BLOS	2702	2767	2859	2975	3032	3075	3118	3160	6017	6054	6091	6177	6208	6259	6346
	6368	6414	6428	6454	6492	6626	6656	6728	6743	6826	6889	6903	6957	6972	7024
	7094	7110	7200	7218	8301										
BLT	3415	8295													
BMI	1350	1362	1457	1521	1558	1930	1986	2246	2461	2506	2551	2584	2669	2703	2735
	2768	2784	2816	2860	2884	2928	2976	2990	3002	3033	3076	3119	3161	3367	3447
	3472	3618	3866	3896	3976	4069	4102	4169	4213	4297	4347	4383	4453	4499	5391

	6019	6056	6093	6129	6145	6192	6209	6230	6246	6261	6274	6307	6330	6348	6370
	6384	6430	6456	6470	6494	6523	6558	6574	6644	6760	6782	6797	6813	6862	6930
BNE	6974	6997	7066	7169	7203	7221	7369	7618	7660	7743	7786	8174	8299		
	536	549	558	598	617	636	655	688	708	737	761	781	832	851	869
	827	905	929	953	977	999	1023	1048	1072	1098	1122	1147	1171	1211	1230
	1248	1266	1298	1302	1316	1345	1392	1439	1508	1555	1568	1597	1634	1672	1717
	1764	1812	1855	1901	1957	2013	2074	2121	2175	2213	2289	2340	2382	2423	2456
	2497	2543	2578	2623	2663	2695	2727	2759	2808	2851	2918	2964	3024	3065	3109
	3151	3192	3219	3246	3287	3360	3401	3466	3485	3509	3537	3566	3599	3640	3670
	3710	3746	3786	3812	3838	3858	3891	3924	3967	4034	4095	4105	4139	4149	4203
	4247	4290	4339	4378	4426	4492	4559	4608	4677	4746	4814	4886	4924	4961	4992
	5021	5068	5131	5138	5156	5174	5200	5258	5265	5293	5301	5327	5356	5388	5422
	5450	5487	5514	5553	5595	5628	5683	5818	5836	5838	5840	5842	5857	5859	5861
	5879	5895	5897	5911	5913	5929	5948	5974	6013	6049	6086	6122	6172	6226	6303
	6326	6363	6410	6487	6550	6589	6606	6674	6708	6775	6854	6922	6989	7058	7141
	7166	7194	7245	7289	7304	7325	7352	7372	7388	7418	7449	7479	7509	7539	7577
	7610	7652	7694	7736	7778	7820	7870	7872	7897	7922	7933	7962	7986	8040	8097
	8113	8115	8117	8119	8121	8123	8147	8152	8170	8175	8206	8230	8257	8259	8262
BPL	8292	8332	8337												
	1349	1363	1571	1608	1646	1686	1689	1736	1782	1831	1877	1905	1962	2018	2091
	2140	2142	2196	2222	2255	2305	2356	2396	2438	2636	3251	3931	3998	4046	4256
	4432	4566	4634	4684	4753	4821	6034	6071	6107	6178	6210	6416	6444	6509	6590
	6613	6628	6658	6681	6715	6730	6745	6828	6877	6891	6905	6945	6960	7012	7026
	7041	7082	7097	7112	7146	7250	7264	7301	7702	7828	8267	8298	8335		
BR	539	559	5819	8232	8271	8291									
BVC	1396	1410	5025	5040	5073	5088	5105	6247	6276	6429	6443	6493	6571	6627	6641
	6729	6757	6827	6875	6904	6942	6973	7010	7040	7079	7111	7265	7302	8302	
BVS	1397	1409	1907	1932	1964	1988	2020	2224	2463	2508	2553	2586	2668	2701	2733
	2766	2782	2814	2858	2882	2926	2974	2988	3031	3074	3117	3159	3253	3369	3474
	3864	3897	3929	3975	3997	4044	4068	4103	4146	4210	4254	4296	4345	4384	4430
	4451	4497	4564	4632	4682	4751	4819	6018	6033	6055	6070	6092	6106	6128	6143
	6179	6193	6232	6260	6308	6331	6347	6369	6382	6415	6455	6468	6508	6521	6556
	6587	6612	6657	6680	6714	6744	6781	6796	6811	6860	6890	6927	6958	6995	7025
	7063	7095	7148	7168	7201	7219	7251	7370	7616	7658	7700	7742	7783	7825	8176
	8303														
CCC	537	1300	1347	1394	1441	4144	6174	6439	6568	6608	6638	6676	6710	6754	6856
	6900	6939	6969	6991	7035	7107	7163	7354	8233						
CLC	689	740	763	784	908	956	1002	1051	1101	1150	1455	3894	5071	5134	5152
	5216	5297	5359	6015	6052	6089	6125	6206	6257	6466	6519	6553	6793	6872	6955
	7007	7077	7198	7216	7900										
CLN	1361	2699	2764	2856	2972	3072	3115	3157	6015	6052	6089	6125	6140	6257	6272
	6344	6366	6490	6553	6808	6887	6925	7022	7061	8172					
CLR	1509	1556	1635	1636	1673	1674	1718	1719	1765	1768	1770	1813	1816	1856	1859
	1902	1958	1959	2014	2015	2075	2078	2079	2122	2125	2126	2176	2179	2214	2217
	2218	2219	2290	2293	2341	2342	2344	2383	2386	2424	2428	2457	2458	2498	2499
	2502	2503	2544	2545	2547	2579	2580	2624	2664	2696	2728	2729	2760	2761	2777
	2809	2810	2852	2853	2919	2920	2965	2966	3025	3026	3066	3067	3110	3111	3152
	3153	3193	3195	3220	3221	3247	3248	3288	3322	3324	3361	3362	3402	3403	3433
	3435	3467	3468	3510	3512	3513	3538	3539	3541	3567	3568	3570	3600	3601	3604
	3641	3642	3643	3671	3672	3711	3747	3748	3749	3814	3840	3859	3892	3893	3925
	3926	3968	3969	4035	4036	4040	4078	4096	4097	4140	4141	4204	4205	4248	4249
	4291	4292	4340	4341	4343	4379	4380	4427	4428	4494	4495	4560	4561	4609	4610
	4678	4679	4747	4748	4815	4816	5069	5101	5132	5150	5167	5424	5452	5684	5824
	5825	6123	6306	6583	7142	7160	7161	7365	7366	7389	7391	7420	7422	7452	7512
	7542	7590	7592	7611	7612	7654	7696	7739	7780	7822	7941	7987	7988	7999	8098
	8099	8100	8101	8102	8103	8104	8153	8157							

CLRB	1598	1676	1722	1818	1862	2128	2181	2548	8149						
CLV	1408	6272	6490	6624	6726	6824	6872	7007	7077	8172					
CLZ	1314	2666	2731	2780	2812	2880	2924	2986	3029	6030	6067	6103	6190	6243	6305
CMP	6328	6380	6466	6505	6519	6585	6778	6793	7144						
	535	597	616	620	635	639	654	657	687	692	707	721	736	760	765
	780	831	850	853	868	871	886	889	904	928	937	552	976	985	998
	1022	1031	1047	1071	1080	1097	1121	1130	1146	1170	1179	1210	1229	1232	1247
	1250	1265	1268	1297	1344	1391	1438	1507	1554	1596	1633	1671	1716	1763	1811
	1854	1900	1956	2012	2073	2120	2174	2212	2288	2339	2381	2422	2455	2496	2542
	2577	2622	2662	2694	2726	2758	2807	2850	2917	2963	3023	3064	3108	3150	3191
	3218	3245	3286	3359	3400	3405	3414	3424	3465	3471	3508	3536	3565	3598	3639
	3669	3709	3745	3785	3811	3816	3837	3842	3857	3890	3923	3951	3966	4017	4033
	4055	4079	4094	4123	4138	4158	4187	4202	4222	4231	4246	4265	4274	4289	4308
	4317	4338	4377	4425	4441	4462	4471	4491	4508	4517	4527	4537	4558	4584	4607
	4613	4622	4652	4676	4693	4702	4713	4722	4745	4762	4771	4782	4791	4813	4830
	4839	4849	4858	4885	4892	4923	4930	4960	4967	4991	4998	5020	5027	5042	5067
	5075	5090	5107	5130	5137	5155	5173	5199	5205	5219	5233	5257	5264	5292	5300
	5326	5332	5355	5362	5387	5399	5421	5426	5449	5454	5486	5490	5513	5518	5552
	5557	5566	5594	5599	5627	5632	5682	5687	5696	5722	5731	5744	5757	5771	5785
	5817	5835	5837	5839	5841	5843	5858	5860	5862	5878	5880	5894	5896	5898	5910
	5912	5914	5928	5930	5947	5949	5973	5984	6012	6048	6085	6121	6171	6225	6302
	6325	6362	6409	6486	6549	6605	6610	6625	6640	6655	6673	6707	6774	6853	6906
	6921	6988	7042	7057	7114	7140	7150	7193	7204	7222	7244	7288	7303	7305	7324
	7326	7351	7356	7387	7393	7402	7417	7424	7433	7448	7454	7478	7493	7508	7514
	7523	7538	7544	7553	7576	7580	7609	7627	7636	7651	7669	7678	7693	7711	7720
	7735	7753	7762	7777	7795	7804	7819	7837	7846	7869	7875	7896	7921	7925	7932
	7943	7961	7967	7973	7985	7993	8000	8039	8096	8112	8114	8116	8118	8120	8122
	8124	8146	8169	8205	8215	8227	8229	8237	8250	8256	8263				
CMPB	3572	7871	8268												
COM	1519	1529	1572	1675	1720	1769	1781	1817	1860	2090	2127	2180	2304	2355	2395
	2437	2459	2581	3027	3068	3112	3154	3194	3222	3299	3300	3325	3434	3469	3511
COMB	3540	3569	3603	3605	3616	3673	3713	3750	4251	4342	4381	5826	6678	6739	7162
	1607	1687	1734	1766	1814	1829	1857	1875	1928	1984	2076	2123	2177	2215	2243
	2263	2291	2384	2425	2500	2527	2546	2549	2582	2596	2597	2697	2762	2854	2921
	2939	2967	2968	3069	3086	3113	3128	3155	3170	3323	3363	3379	3543	3571	3602
	3645	3724	3751	3860	3875	4038									
DEC	1557	1574	1645	1731	1779	1780	1783	1784	1827	1832	1871	1872	1874	1878	1942
	1998	2035	2044	2088	2089	2137	2138	2143	2144	2190	2191	2192	2193	2482	2504
	2518	2528	2826	2827	2869	2894	2895	2922	2938	3302	3380	3445	3484	3684	3685
	3940	3941	3985	4007	4008	4394	4395	5139	5140	5157	5175	5176	6229	6244	6258
	6273	6342	7260	8074	8079	8225									
DECB	2030	2031	2034	2563	2598										
HALT	417	547	556	567	576	609	628	647	665	700	719	729	753	773	797
	843	861	879	897	921	946	969	991	1015	1040	1064	1089	1114	1139	1163
	1188	1222	1240	1258	1276	1311	1324	1358	1371	1405	1418	1452	1465	1517	1528
	1537	1565	1583	1606	1618	1644	1657	1684	1699	1730	1746	1778	1794	1826	1842
	1870	1888	1916	1926	1941	1950	1973	1983	1997	2006	2029	2043	2052	2087	2101
	2136	2154	2189	2206	2233	2242	2253	2262	2273	2301	2317	2352	2366	2394	2406
	2436	2448	2472	2481	2490	2517	2526	2536	2562	2571	2595	2607	2633	2646	2679
	2711	2744	2776	2793	2825	2835	2868	2877	2893	2903	2937	2947	2984	2999	3009
	3041	3050	3084	3094	3127	3136	3169	3178	3205	3232	3262	3271	3297	3310	3321
	3336	3347	3378	3388	3413	3422	3432	3444	3454	3483	3495	3523	3551	3580	3615
	3625	3655	3683	3693	3723	3732	3761	3771	3798	3824	3850	3874	3883	3907	3916
	3939	3950	3959	3984	3993	4006	4016	4025	4054	4063	4077	4087	4113	4122	4131
	4157	4166	4177	4186	4195	4221	4230	4239	4264	4273	4282	4307	4316	4325	4355
	4364	4393	4404	4413	4440	4449	4461	4470	4479	4507	4516	4525	4535	4545	4574

INC

4583	4592	4621	4630	4642	4651	4660	4692	4701	4710	4721	4730	4761	4770	4779
4790	4799	4829	4838	4847	4857	4866	4901	4939	4975	5006	5036	5051	5084	5099
5116	5149	5166	5185	5214	5228	5242	5275	5311	5341	5371	5398	5407	5434	5462
5471	5498	5526	5535	5565	5575	5608	5641	5695	5704	5718	5730	5739	5752	5765
5779	5793	5833	5852	5871	5889	5907	5923	5939	5958	5983	5992	6027	6042	6064
6079	6101	6115	6138	6153	6187	6203	6218	6241	6255	6269	6285	6318	6341	6356
6378	6393	6424	6438	6452	6464	6479	6502	6517	6532	6566	6582	6598	6621	6636
6652	6666	6689	6723	6738	6753	6768	6791	6806	6821	6836	6870	6885	6899	6915
6938	6953	6968	6982	7005	7020	7034	7051	7074	7090	7105	7123	7159	7179	7213
7231	7259	7275	7299	7314	7323	7335	7364	7380	7401	7410	7432	7441	7462	7471
7492	7501	7522	7531	7552	7561	7588	7602	7626	7635	7644	7668	7677	7686	7710
7719	7728	7752	7761	7770	7794	7803	7812	7836	7845	7854	7883	7907	7930	7940
7947	7972	7978	7998	8008	8072	8133	8156	8185	8224	8247	8283	8327	8355	8359
8363	8367	8371	8375	8379	8383									
534	546	555	566	575	596	608	615	627	634	646	653	664	686	699
706	718	728	735	741	752	759	772	779	785	796	830	842	849	860
867	878	885	896	903	909	920	927	933	945	951	957	968	975	981
990	997	1003	1014	1021	1027	1039	1046	1052	1063	1070	1076	1088	1096	1102
1113	1120	1126	1138	1145	1151	1162	1169	1175	1187	1209	1221	1228	1239	1246
1257	1264	1275	1296	1310	1323	1343	1357	1370	1390	1404	1417	1437	1451	1464
1506	1516	1518	1520	1527	1536	1553	1564	1573	1582	1595	1605	1617	1632	1643
1656	1670	1683	1685	1698	1715	1721	1729	1732	1733	1745	1762	1767	1777	1785
1793	1810	1815	1825	1828	1841	1853	1858	1861	1869	1873	1887	1899	1903	1915
1918	1925	1940	1949	1955	1960	1972	1975	1982	1996	2005	2011	2016	2028	2042
2051	2072	2077	2086	2092	2100	2119	2124	2135	2139	2153	2173	2178	2188	2194
2205	2211	2216	2220	2232	2234	2241	2244	2252	2261	2265	2272	2287	2292	2300
2302	2303	2306	2307	2308	2316	2338	2351	2353	2354	2357	2365	2380	2385	2393
2397	2405	2421	2426	2427	2435	2439	2447	2454	2471	2480	2489	2495	2501	2516
2525	2535	2541	2561	2570	2576	2594	2606	2621	2632	2634	2637	2645	2661	2678
2693	2710	2725	2743	2757	2775	2778	2792	2806	2824	2834	2849	2867	2876	2878
2892	2902	2916	2936	2946	2962	2969	2983	2998	3008	3022	3040	3049	3063	3070
3083	3085	3093	3107	3126	3135	3149	3168	3177	3190	3197	3204	3217	3224	3231
3244	3249	3261	3263	3270	3285	3296	3298	3309	3312	3313	3320	3328	3335	3339
3346	3358	3364	3377	3387	3399	3404	3412	3421	3423	3431	3443	3453	3464	3470
3482	3486	3494	3507	3514	3522	3535	3550	3564	3579	3597	3614	3624	3638	3644
3647	3654	3668	3675	3682	3692	3708	3722	3731	3744	3753	3760	3770	3784	3790
3797	3810	3823	3836	3849	3856	3873	3882	3889	3906	3915	3922	3938	3949	3958
3965	3983	3992	3994	4005	4015	4024	4032	4039	4053	4062	4064	4065	4076	4086
4093	4112	4121	4130	4137	4156	4165	4176	4185	4194	4201	4220	4229	4238	4245
4263	4272	4281	4288	4306	4315	4324	4337	4354	4363	4376	4392	4403	4412	4424
4439	4448	4460	4469	4478	4490	4506	4515	4524	4534	4544	4557	4573	4582	4591
4606	4620	4629	4641	4650	4659	4675	4691	4700	4709	4720	4729	4744	4749	4760
4769	4778	4789	4798	4812	4828	4837	4846	4856	4865	4884	4900	4922	4938	4959
4974	4990	5005	5019	5035	5050	5066	5083	5098	5102	5115	5129	5148	5165	5169
5184	5198	5213	5227	5241	5256	5274	5291	5310	5325	5340	5354	5370	5386	5397
5406	5420	5433	5448	5461	5470	5485	5497	5512	5525	5534	5551	5564	5574	5593
5607	5626	5640	5681	5694	5703	5706	5717	5719	5729	5738	5741	5751	5754	5764
5767	5778	5781	5792	5816	5832	5851	5853	5870	5874	5888	5890	5906	5908	5922
5924	5938	5940	5957	5972	5982	5991	6011	6026	6041	6047	6063	6078	6084	6100
6114	6120	6137	6152	6170	6176	6186	6191	6202	6207	6217	6224	6240	6254	6268
6284	6301	6317	6324	6340	6355	6361	6377	6392	6408	6423	6437	6451	6463	6478
6485	6501	6516	6531	6548	6565	6581	6597	6604	6620	6635	6651	6665	6672	6688
6706	6722	6737	6752	6767	6773	6790	6805	6820	6835	6852	6869	6884	6898	6914
6920	6937	6952	6967	6981	6987	7004	7019	7033	7050	7056	7073	7089	7104	7122
7139	7158	7178	7192	7212	7230	7243	7258	7274	7287	7298	7313	7322	7334	7350
7363	7379	7386	7400	7409	7416	7431	7440	7447	7461	7470	7477	7491	7500	7507

MTPS	7355	7368	7392	7423	7453	7483	7513	7543							
NEG	1904	1961	2017	2221	2460	2473	2505	2550	2599	2635	3365	6554	6570	6586	
NEGB	1929	1985	2032	2033	2141	2195	2245	2254	2343	2583	3338				
NOP	409	8276	8277	8278											
RESET	7874	8274													
ROL	691	711	743	911	935	959	983	1005	1029	1054	1078	1104	1128	1153	1177
	5024	5072	5135	5203	5262	5298	5330	5360	6858	6873	6888	6902	7901	8059	
ROLB	5039	5087	5104	5153	5171	5217	5231								
ROR	764	787	6926	6941	6956	6971									
RTS	5978	7320													
SBC	1570	6779	6794	6809	6825										
SCC	1313	1360	1407	1454	2665	2698	2730	2763	2779	2811	2855	2879	2923	2971	2985
	3028	3071	3114	3156	4100	4208	6014	6029	6051	6066	6088	6102	6124	6139	6205
	6228	6256	6271	6304	6327	6343	6365	6379	6412	6465	6489	6504	6518	6552	6584
	6623	6654	6725	6740	6777	6792	6807	6823	6871	6886	6924	6954	7006	7021	7060
	7076	7091	7143	7197	7248	7261	7295	7367	7482	8171	8211				
SEC	710	932	980	1026	1075	1125	1174	1442	1566	1569	1647	3861	3971	4041	5023
	5038	5085	5103	5170	5202	5230	5261	5329	6189	6242	6609	6677	6711	6901	6970
	6992	7036	7108	7215											
SEN	1348	6440	6609	6711	6940	6992									
SEV	1395	3861	3971	4041	6857	7164									
SEZ	557	1301	6175	6425	6569	6639	6677	6755	6857	6901	6940	6970	7036	7108	7164
SOB	7262														
SUB	3250	3337	5463	6712	6727	6742	6756	7292							
SWAB	5390	5425	5453	5489	5517	5556	5598	5631	6367	6381					
SXT	7145	7165													
TST	601	835	1214	2667	2732	2813	2925	2970	3000	3001	3030	3042	3073	3116	3158
	3607	3762	3763	3908	4114	4178	4356	4405	4575	4643	5266	5302	5527	5710	5856
	6329	6345	7170	7266	7463	7484									
TSTB	2700	2765	2781	2857	2881	2973	2987	3715	7594	8266	8334				
XOR	7199	7217													
.ASCII	8285	8288													
.ASCIZ	8340														
.BLKW	8344														
.BYTE	443	444													
.ENABL	1	405													
.END	8384														
.ENDC	425	427	456	548	557	568	578	611	630	649	667	702	721	731	755
	775	799	845	863	881	899	923	948	970	992	1017	1042	1066	1091	1116
	1141	1165	1190	1224	1242	1260	1278	1312	1326	1359	1373	1406	1420	1453	1467
	1518	1529	1539	1566	1585	1607	1620	1645	1659	1685	1701	1731	1748	1779	1796
	1827	1844	1871	1890	1917	1927	1942	1952	1974	1984	1998	2008	2030	2044	2054
	2088	2103	2137	2156	2190	2208	2234	2243	2254	2263	2275	2302	2319	2353	2368
	2395	2408	2437	2450	2473	2482	2492	2518	2527	2538	2563	2573	2596	2609	2634
	2648	2681	2713	2746	2777	2795	2826	2837	2869	2878	2894	2905	2938	2949	2985
	3000	3011	3042	3052	3085	3096	3128	3138	3170	3180	3207	3234	3263	3273	3298
	3311	3322	3337	3349	3379	3390	3414	3423	3433	3445	3456	3484	3497	3525	3553
	3582	3616	3627	3657	3684	3695	3724	3734	3762	3773	3800	3826	3852	3875	3885
	3908	3918	3940	3951	3961	3985	3994	4007	4017	4027	4055	4064	4078	4089	4114
	4123	4133	4158	4167	4178	4187	4197	4222	4231	4241	4265	4274	4284	4308	4317
	4327	4356	4366	4394	4405	4415	4441	4450	4462	4471	4481	4508	4517	4526	4536
	4546	4575	4584	4594	4622	4631	4643	4652	4662	4693	4702	4711	4722	4732	4762
	4771	4780	4791	4801	4830	4839	4848	4858	4868	4903	4941	4977	5008	5037	5053
	5085	5100	5118	5150	5167	5187	5215	5229	5244	5277	5313	5343	5373	5399	5409
	5436	5463	5473	5500	5527	5537	5566	5577	5610	5643	5696	5705	5719	5731	5740
	5753	5766	5780	5795	5834	5853	5872	5890	5908	5924	5940	5960	5984	5994	6028

	6044	6065	6081	6102	6117	6139	6155	6188	6204	6220	6242	6256	6270	6287	6320
	6342	6358	6379	6394	6425	6439	6453	6465	6481	6503	6518	6534	6567	6583	6600
	6622	6637	6653	6668	6691	6724	6739	6754	6769	6792	6807	6822	6838	6871	6886
	6900	6917	6939	6954	6969	6984	7006	7021	7035	7053	7075	7091	7106	7125	7160
	7121	7214	7233	7260	7277	7300	7315	7324	7337	7365	7382	7402	7412	7433	7443
	7463	7473	7493	7503	7523	7533	7553	7563	7589	7604	7627	7636	7646	7669	7678
	7688	7711	7720	7730	7753	7762	7772	7795	7804	7814	7837	7846	7856	7885	7908
	7931	7941	7948	7973	7979	7999	8010	8073	8134	8157	8187	8225	8248	8284	8356
	8360	8364	8368	8372	8376	8380	8384								
.EVEN	432	8290	8343												
.IF	423	425	454	456	545	549	557	565	569	602	621	640	658	693	712
	722	745	766	789	836	854	872	890	913	938	967	988	1007	1032	1056
	1081	1106	1131	1155	1180	1215	1233	1251	1269	1303	1312	1316	1350	1359	1363
	1397	1406	1410	1444	1453	1457	1510	1518	1521	1529	1530	1558	1566	1575	1599
	1607	1610	1637	1645	1649	1677	1685	1691	1723	1731	1738	1771	1779	1786	1819
	1827	1834	1863	1871	1880	1908	1917	1919	1927	1933	1942	1943	1965	1974	1976
	1984	1989	1998	1999	2021	2030	2036	2044	2045	2080	2088	2093	2129	2137	2146
	2182	2190	2198	2225	2234	2235	2243	2246	2254	2255	2263	2266	2294	2302	2309
	2345	2353	2358	2387	2395	2398	2429	2437	2440	2464	2473	2474	2482	2483	2509
	2518	2519	2527	2529	2554	2563	2564	2587	2596	2600	2626	2634	2638	2671	2703
	2736	2768	2777	2785	2817	2826	2828	2860	2869	2870	2878	2885	2894	2896	2929
	2938	2940	2976	2985	2991	3000	3002	3033	3042	3043	3076	3085	3087	3119	3128
	3129	3161	3170	3171	3198	3225	3254	3263	3264	3290	3298	3303	3311	3314	3322
	3329	3337	3340	3370	3379	3381	3406	3414	3415	3423	3425	3433	3437	3445	3447
	3475	3484	3487	3516	3544	3573	3608	3616	3618	3648	3676	3684	3686	3716	3724
	3725	3754	3762	3764	3791	3817	3843	3866	3875	3876	3899	3908	3909	3931	3940
	3942	3951	3952	3976	3985	3986	3994	3998	4007	4009	4017	4018	4046	4055	4056
	4064	4069	4078	4080	4105	4114	4115	4123	4124	4149	4158	4159	4167	4169	4178
	4179	4187	4188	4213	4222	4223	4231	4232	4256	4265	4266	4274	4275	4299	4308
	4309	4317	4318	4347	4356	4357	4385	4394	4396	4405	4406	4432	4441	4442	4450
	4453	4462	4463	4471	4472	4499	4508	4509	4517	4518	4526	4528	4536	4538	4566
	4575	4576	4584	4585	4614	4622	4623	4631	4634	4643	4644	4652	4653	4684	4693
	4694	4702	4703	4711	4714	4722	4723	4753	4762	4763	4771	4772	4780	4783	4791
	4792	4821	4830	4831	4839	4840	4848	4850	4858	4859	4893	4931	4968	4999	5028
	5037	5043	5076	5085	5091	5100	5108	5141	5150	5158	5167	5177	5206	5215	5220
	5229	5234	5267	5303	5333	5363	5391	5399	5400	5427	5455	5463	5464	5491	5519
	5527	5528	5558	5566	5567	5600	5633	5688	5696	5697	5705	5711	5719	5723	5731
	5732	5740	5745	5753	5758	5766	5772	5780	5786	5830	5844	5853	5863	5872	5881
	5890	5899	5908	5915	5924	5931	5940	5950	5981	5985	6019	6028	6034	6056	6065
	6071	6093	6102	6107	6130	6139	6145	6179	6188	6195	6204	6210	6233	6242	6247
	6256	6261	6270	6277	6310	6333	6342	6348	6370	6379	6385	6416	6425	6430	6439
	6444	6453	6456	6465	6471	6494	6503	6509	6518	6524	6558	6567	6574	6583	6590
	6613	6622	6628	6637	6644	6653	6658	6681	6715	6724	6730	6739	6745	6754	6760
	6783	6792	6798	6807	6813	6822	6828	6862	6871	6877	6886	6891	6900	6907	6930
	6939	6945	6954	6960	6969	6974	6997	7006	7012	7021	7026	7035	7043	7066	7075
	7082	7091	7097	7106	7115	7151	7160	7171	7205	7214	7223	7251	7260	7267	7297
	7306	7315	7321	7327	7357	7365	7372	7394	7402	7403	7425	7433	7434	7455	7463
	7464	7485	7493	7494	7515	7523	7524	7545	7553	7554	7581	7589	7595	7618	7627
	7628	7636	7637	7660	7669	7670	7678	7679	7702	7711	7712	7720	7721	7744	7753
	7754	7762	7763	7786	7795	7796	7804	7805	7828	7837	7838	7846	7847	7876	7905
	7928	7933	7941	7945	7970	7976	7996	8001	8070	8125	8134	8154	8177	8217	8225
	8239	8248	8280	8352	8356	8360	8364	8368	8372	8376	8380				
.IFF	425	427	545	557	565	569	578	602	611	621	630	640	649	658	667
	693	702	712	721	722	731	745	755	766	775	789	799	836	845	854
	863	872	881	890	899	913	923	938	948	967	988	1007	1017	1032	1042
	1056	1066	1081	1091	1106	1116	1131	1141	1155	1165	1180	1190	1215	1224	1233

1242	1251	1260	1269	1278	1312	1316	1326	1359	1363	1373	1406	1410	1420	1453
1457	1467	1518	1529	1530	1539	1566	1575	1585	1607	1610	1620	1645	1649	1659
1685	1691	1701	1731	1738	1748	1779	1786	1796	1827	1834	1844	1871	1880	1890
1917	1927	1942	1943	1952	1974	1984	1998	1999	2008	2030	2044	2045	2054	2088
2093	2103	2137	2146	2156	2190	2198	2208	2234	2243	2254	2263	2266	2275	2302
2309	2319	2353	2358	2368	2395	2398	2408	2437	2440	2450	2473	2482	2483	2492
2518	2527	2529	2538	2563	2564	2573	2596	2600	2609	2634	2638	2648	2671	2681
2703	2713	2736	2746	2777	2785	2795	2826	2828	2837	2869	2878	2894	2896	2905
2938	2940	2949	2985	3000	3002	3011	3042	3043	3052	3085	3087	3096	3128	3129
3138	3170	3171	3180	3198	3207	3225	3234	3263	3264	3273	3298	3311	3322	3337
3340	3349	3379	3381	3390	3414	3423	3433	3445	3447	3456	3484	3487	3497	3516
3525	3544	3553	3573	3582	3616	3618	3627	3648	3657	3684	3686	3695	3724	3725
3734	3762	3764	3773	3791	3800	3817	3826	3843	3852	3875	3876	3885	3908	3909
3918	3940	3951	3952	3961	3985	3994	4007	4017	4018	4027	4055	4064	4078	4080
4089	4114	4123	4124	4133	4158	4167	4178	4187	4188	4197	4222	4231	4232	4241
4265	4274	4275	4284	4308	4317	4318	4327	4356	4357	4366	4394	4405	4406	4415
4441	4450	4462	4471	4472	4481	4508	4517	4526	4536	4538	4546	4575	4584	4585
4594	4622	4631	4643	4652	4653	4662	4693	4702	4711	4722	4723	4732	4762	4771
4780	4791	4792	4801	4830	4839	4848	4858	4859	4868	4893	4903	4931	4941	4968
4977	4999	5008	5037	5043	5053	5085	5100	5108	5118	5150	5167	5177	5187	5215
5229	5234	5244	5267	5277	5303	5313	5333	5343	5363	5373	5399	5400	5409	5427
5436	5463	5464	5473	5491	5500	5527	5528	5537	5566	5567	5577	5600	5610	5633
5643	5696	5705	5719	5731	5740	5753	5766	5780	5786	5795	5830	5853	5872	5890
5908	5924	5940	5950	5960	5981	5985	5994	6028	6034	6044	6065	6071	6081	6102
6107	6117	6139	6145	6155	6188	6204	6210	6220	6242	6256	6270	6277	6287	6310
6320	6342	6348	6358	6379	6385	6394	6425	6439	6453	6465	6471	6481	6503	6518
6524	6534	6567	6583	6590	6600	6622	6637	6653	6658	6668	6681	6691	6724	6739
6754	6760	6769	6792	6807	6822	6828	6838	6871	6886	6900	6907	6917	6939	6954
6969	6974	6984	7006	7021	7035	7043	7053	7075	7091	7106	7115	7125	7160	7171
7181	7214	7223	7233	7260	7267	7277	7297	7315	7321	7327	7337	7365	7372	7382
7402	7403	7412	7433	7434	7443	7463	7464	7473	7493	7494	7503	7523	7524	7533
7553	7554	7563	7589	7595	7604	7627	7636	7637	7646	7669	7678	7679	7688	7711
7720	7721	7730	7753	7762	7763	7772	7795	7804	7805	7814	7837	7846	7847	7856
7876	7885	7905	7928	7941	7945	7970	7976	7996	8001	8010	8070	8134	8154	8177
8187	8225	8248	8280	8352	8356	8360	8364	8368	8372	8376	8380			
.IFT	545	549	557	565	569	602	640	658	693	712	722	745	766	789
836	854	872	890	913	938	967	988	1007	1032	1056	1081	1106	1131	1155
1180	1215	1233	1251	1269	1303	1312	1316	1350	1359	1363	1397	1406	1410	1444
1453	1457	1510	1518	1521	1529	1530	1558	1566	1575	1599	1607	1610	1637	1645
1649	1677	1685	1691	1723	1731	1738	1771	1779	1786	1819	1827	1834	1863	1871
1880	1908	1917	1919	1927	1933	1942	1943	1965	1974	1976	1984	1989	1998	1999
2021	2030	2036	2044	2045	2080	2088	2093	2129	2137	2146	2182	2190	2198	2225
2234	2235	2243	2246	2254	2255	2263	2266	2294	2302	2309	2345	2353	2358	2387
2395	2398	2429	2437	2440	2464	2473	2474	2482	2483	2509	2518	2519	2527	2529
2554	2563	2564	2587	2596	2600	2626	2634	2638	2671	2703	2736	2768	2777	2785
2817	2826	2828	2860	2869	2870	2878	2885	2894	2896	2929	2938	2940	2976	2985
2991	3000	3002	3033	3042	3043	3076	3085	3087	3119	3128	3129	3161	3170	3171
3198	3225	3254	3263	3264	3290	3298	3303	3311	3314	3322	3329	3337	3340	3370
3379	3381	3406	3414	3415	3423	3425	3433	3437	3445	3447	3475	3484	3487	3516
3544	3573	3608	3616	3618	3648	3676	3684	3686	3716	3724	3725	3754	3762	3764
3791	3817	3843	3866	3875	3876	3899	3908	3909	3931	3940	3942	3951	3952	3976
3985	3986	3994	3998	4007	4009	4017	4018	4046	4055	4056	4064	4069	4078	4080
4105	4114	4115	4123	4124	4149	4158	4159	4167	4169	4178	4179	4187	4188	4213
4222	4223	4231	4232	4256	4265	4266	4274	4275	4299	4308	4309	4317	4318	4347
4356	4357	4385	4394	4396	4405	4406	4432	4441	4442	4450	4453	4462	4463	4471
4472	4499	4508	4509	4517	4518	4526	4528	4536	4538	4566	4575	4576	4584	4585

4614	4622	4623	4631	4634	4643	4644	4652	4653	4684	4693	4694	4702	4703	4711
4714	4722	4723	4753	4762	4763	4771	4772	4780	4783	4791	4792	4821	4830	4831
4839	4840	4848	4850	4858	4859	4893	4931	4968	4999	5028	5037	5043	5076	5085
5091	5100	5108	5141	5150	5158	5167	5177	5206	5215	5220	5229	5234	5267	5303
5333	5363	5391	5399	5400	5427	5455	5463	5464	5491	5519	5527	5528	5558	5566
5567	5600	5633	5688	5696	5697	5705	5711	5719	5723	5731	5732	5740	5745	5753
5758	5766	5772	5780	5786	5830	5844	5853	5863	5872	5881	5890	5899	5908	5915
5924	5931	5940	5950	5981	5985	6019	6028	6034	6056	6065	6071	6093	6102	6107
6130	6139	6145	6179	6188	6195	6204	6210	6233	6242	6247	6256	6261	6270	6277
6310	6333	6342	6348	6370	6379	6385	6416	6425	6430	6439	6444	6453	6456	6465
6471	6494	6503	6509	6518	6524	6558	6567	6574	6583	6590	6613	6622	6628	6637
6644	6653	6658	6681	6715	6724	6730	6739	6745	6754	6760	6783	6792	6798	6807
6813	6822	6828	6862	6871	6877	6886	6891	6900	6907	6930	6939	6945	6954	6960
6969	6974	6997	7006	7012	7021	7026	7035	7043	7066	7075	7082	7091	7097	7106
7115	7151	7160	7171	7205	7214	7223	7251	7260	7267	7297	7306	7315	7321	7327
7357	7365	7372	7394	7402	7403	7425	7433	7434	7455	7463	7464	7485	7493	7494
7515	7523	7524	7545	7553	7554	7581	7589	7595	7618	7627	7628	7636	7637	7660
7669	7670	7678	7679	7702	7711	7712	7720	7721	7744	7753	7754	7762	7763	7786
7795	7796	7804	7805	7828	7837	7838	7846	7847	7876	7905	7928	7933	7941	7945
7970	7976	7996	8001	8070	8125	8134	8154	8177	8217	8225	8239	8248	8280	8352
8356	8360	8364	8368	8372	8376	8380								
432	531	536	545	554	565	574	577	593	598	607	610	612	617	626
629	631	636	645	648	650	655	663	666	683	688	698	701	703	708
717	720	727	730	732	737	750	754	756	761	771	774	776	781	794
798	827	832	841	844	846	851	859	862	864	869	877	880	882	887
895	898	900	905	918	922	924	929	943	947	948	953	954	967	972
977	988	994	999	1012	1016	1018	1023	1037	1041	1043	1048	1061	1065	1067
1072	1086	1090	1093	1098	1111	1115	1117	1122	1136	1140	1142	1147	1160	1164
1166	1171	1185	1189	1206	1211	1220	1223	1225	1230	1238	1241	1243	1248	1256
1259	1261	1266	1277	1277	1293	1298	1308	1321	1325	1340	1345	1355	1368	1372
1387	1392	1402	1415	1419	1434	1439	1449	1462	1466	1503	1508	1515	1526	1535
1538	1550	1555	1563	1580	1584	1592	1597	1604	1615	1619	1629	1634	1642	1654
1658	1667	1672	1682	1696	1700	1712	1717	1728	1743	1747	1759	1764	1776	1791
1795	1807	1812	1824	1839	1843	1850	1855	1868	1885	1889	1896	1901	1913	1924
1938	1948	1951	1952	1957	1970	1981	1994	2004	2007	2008	2013	2026	2041	2050
2053	2069	2074	2085	2098	2102	2116	2121	2134	2151	2155	2170	2175	2187	2203
2207	2208	2213	2230	2240	2251	2260	2271	2274	2284	2289	2299	2314	2318	2335
2340	2350	2363	2367	2377	2382	2392	2403	2407	2418	2423	2434	2445	2449	2451
2456	2469	2479	2488	2491	2492	2497	2514	2524	2534	2537	2538	2543	2559	2569
2572	2573	2578	2592	2605	2608	2618	2623	2624	2631	2643	2647	2658	2663	2676
2680	2690	2695	2708	2712	2722	2727	2741	2745	2754	2759	2773	2790	2794	2803
2808	2822	2833	2836	2846	2851	2865	2875	2890	2901	2904	2913	2918	2934	2945
2948	2959	2964	2981	2996	3007	3010	3019	3024	3038	3048	3051	3060	3065	3081
3092	3095	3104	3109	3124	3134	3137	3146	3151	3166	3176	3179	3187	3192	3203
3206	3214	3219	3230	3233	3241	3246	3259	3269	3272	3282	3287	3295	3308	3319
3334	3345	3348	3355	3360	3375	3386	3389	3396	3401	3411	3420	3430	3442	3452
3455	3461	3466	3480	3492	3496	3504	3509	3521	3524	3532	3537	3549	3552	3561
3566	3578	3581	3594	3599	3613	3623	3626	3635	3640	3653	3656	3665	3670	3681
3691	3694	3705	3710	3721	3730	3733	3741	3746	3759	3769	3772	3781	3786	3796
3799	3807	3812	3822	3825	3833	3838	3848	3851	3853	3858	3871	3881	3884	3886
3891	3904	3914	3917	3919	3924	3936	3947	3957	3960	3962	3967	3981	3991	4003
4014	4023	4026	4029	4034	4051	4061	4074	4085	4088	4090	4095	4110	4120	4129
4132	4134	4139	4154	4164	4174	4184	4193	4196	4198	4203	4218	4228	4237	4240
4242	4247	4261	4271	4280	4283	4285	4290	4304	4314	4323	4326	4334	4339	4352
4362	4365	4373	4378	4390	4401	4411	4414	4421	4426	4437	4447	4458	4468	4477
4480	4487	4492	4504	4514	4523	4533	4543	4546	4554	4559	4571	4591	4590	4593

. IIF

4603	4608	4619	4628	4639	4649	4658	4661	4672	4677	4689	4699	4708	4719	4728
4731	4741	4746	4758	4768	4777	4788	4797	4800	4809	4814	4826	4836	4845	4855
4864	4867	4881	4886	4887	4898	4902	4919	4924	4925	4936	4940	4956	4961	4973
4976	4987	4992	5004	5007	5016	5021	5033	5048	5052	5063	5068	5081	5096	5113
5117	5126	5131	5146	5163	5182	5186	5195	5200	5211	5225	5239	5243	5253	5258
5272	5276	5288	5293	5294	5308	5312	5322	5327	5338	5342	5351	5356	5357	5368
5372	5383	5388	5396	5405	5408	5417	5422	5432	5435	5445	5450	5460	5469	5472
5482	5487	5496	5499	5509	5514	5524	5533	5536	5548	5553	5554	5563	5572	5576
5590	5595	5596	5605	5609	5623	5628	5629	5638	5642	5678	5683	5684	5693	5702
5716	5728	5737	5750	5763	5777	5791	5794	5813	5818	5819	5830	5849	5868	5886
5904	5920	5936	5955	5959	5969	5974	5981	5990	5993	6008	6013	6024	6039	6043
6044	6049	6061	6076	6080	6081	6086	6098	6112	6116	6117	6122	6135	6150	6154
6167	6172	6184	6200	6215	6219	6221	6225	6238	6252	6266	6282	6286	6298	6303
6315	6319	6321	6326	6338	6353	6357	6358	6363	6375	6390	6394	6405	6410	6421
6435	6449	6461	6476	6480	6482	6487	6499	6514	6529	6533	6545	6550	6563	6579
6595	6599	6601	6606	6618	6633	6649	6663	6667	6669	6674	6686	6690	6703	6708
6720	6735	6750	6765	6769	6770	6775	6788	6803	6818	6833	6837	6849	6854	6867
6882	6896	6912	6916	6917	6922	6935	6950	6965	6979	6983	6984	6989	7002	7017
7031	7048	7052	7053	7058	7071	7087	7102	7120	7124	7136	7141	7156	7176	7180
7189	7194	7210	7228	7232	7240	7245	7256	7272	7276	7284	7289	7297	7311	7321
7332	7336	7347	7352	7362	7377	7381	7383	7388	7399	7408	7411	7413	7418	7430
7439	7442	7444	7449	7460	7469	7472	7474	7479	7490	7499	7502	7504	7509	7520
7529	7532	7534	7539	7550	7559	7562	7572	7577	7586	7600	7603	7605	7610	7623
7633	7642	7645	7647	7652	7665	7675	7684	7687	7689	7694	7707	7717	7726	7729
7731	7736	7749	7759	7768	7771	7773	7778	7791	7801	7810	7813	7815	7820	7833
7843	7852	7855	7865	7870	7881	7884	7892	7897	7905	7917	7922	7923	7928	7938
7945	7957	7962	7970	7976	7981	7986	7996	8006	8009	8035	8040	8041	8070	8092
8097	8098	8130	8142	8147	8148	8154	8165	8170	8182	8186	8201	8206	8207	8222
8244	8252	8257	8258	8280	8352	8356	8360	8364	8368	8372	8376	8380		
.LIST	1	404	408	417	432	534	537	546	552	555	566	575	596	599
605	608	615	618	624	627	634	637	643	646	653	656	661	664	686
689	696	699	706	709	715	718	725	728	735	738	748	752	759	762
769	772	779	782	792	796	830	833	839	842	849	852	857	860	867
870	875	878	885	888	893	896	903	906	916	920	927	930	941	945
951	954	968	975	978	990	997	1000	1010	1014	1021	1024	1035	1039	1046
1049	1059	1063	1070	1073	1084	1088	1096	1099	1109	1113	1120	1123	1134	1138
1145	1148	1158	1162	1169	1172	1183	1187	1209	1212	1218	1221	1228	1231	1236
1239	1246	1249	1254	1257	1264	1267	1272	1275	1296	1299	1306	1310	1319	1323
1343	1346	1353	1357	1366	1370	1390	1393	1400	1404	1413	1417	1437	1440	1447
1451	1460	1464	1506	1509	1513	1516	1524	1527	1533	1536	1553	1556	1561	1564
1578	1582	1595	1598	1602	1605	1613	1617	1632	1635	1640	1643	1652	1656	1670
1673	1680	1683	1694	1698	1715	1718	1726	1729	1741	1745	1762	1765	1774	1777
1789	1793	1810	1813	1822	1825	1837	1841	1853	1856	1866	1869	1883	1887	1899
1902	1911	1915	1922	1925	1936	1940	1946	1949	1955	1958	1968	1972	1979	1982
1992	1996	2002	2005	2011	2014	2024	2028	2039	2042	2048	2051	2072	2075	2083
2086	2096	2100	2119	2122	2132	2135	2149	2153	2173	2176	2185	2188	2201	2205
2211	2214	2228	2232	2238	2241	2249	2252	2258	2261	2269	2272	2287	2290	2297
2300	2312	2316	2338	2341	2348	2351	2361	2365	2380	2383	2390	2393	2401	2405
2421	2424	2432	2435	2443	2447	2454	2457	2467	2471	2477	2480	2486	2489	2495
2498	2512	2516	2522	2525	2532	2535	2541	2544	2557	2561	2567	2570	2576	2579
2590	2594	2603	2606	2621	2624	2629	2632	2641	2645	2661	2664	2674	2678	2693
2596	2706	2710	2725	2728	2739	2743	2757	2760	2771	2775	2788	2792	2806	2809
2820	2824	2831	2834	2849	2852	2863	2867	2873	2876	2888	2892	2899	2902	2916
2919	2932	2936	2943	2946	2962	2965	2979	2983	2994	2998	3005	3008	3022	3025
3036	3040	3046	3049	3063	3066	3079	3083	3090	3093	3107	3110	3122	3126	3132
3135	3149	3152	3164	3168	3174	3177	3190	3193	3201	3204	3217	3220	3228	3231

689	696	699	706	709	715	718	725	728	735	738	748	752	759	762
769	772	779	782	792	796	830	833	839	842	849	852	857	860	867
870	875	878	885	888	893	896	903	906	916	920	927	930	941	945
951	954	968	975	978	990	997	1000	1010	1014	1021	1024	1035	1039	1046
1049	1059	1063	1070	1073	1084	1088	1096	1099	1109	1113	1120	1123	1134	1138
1145	1148	1158	1162	1169	1172	1183	1187	1209	1212	1218	1221	1228	1231	1236
1239	1246	1249	1254	1257	1264	1267	1272	1275	1296	1299	1306	1310	1319	1323
1343	1346	1353	1357	1366	1370	1390	1393	1400	1404	1413	1417	1437	1440	1447
1451	1460	1464	1506	1509	1513	1516	1524	1527	1533	1536	1553	1556	1561	1564
1578	1582	1595	1598	1602	1605	1613	1617	1632	1635	1640	1643	1652	1656	1670
1673	1680	1683	1694	1698	1715	1718	1726	1729	1741	1745	1762	1765	1774	1777
1789	1793	1810	1813	1822	1825	1837	1841	1853	1856	1866	1869	1883	1887	1899
1902	1911	1915	1922	1925	1936	1940	1946	1949	1955	1958	1968	1972	1979	1982
1992	1996	2002	2005	2011	2014	2024	2028	2039	2042	2048	2051	2072	2075	2083
2086	2096	2100	2119	2122	2132	2135	2149	2153	2173	2176	2185	2188	2201	2205
2211	2214	2228	2232	2238	2241	2249	2252	2258	2261	2269	2272	2287	2290	2297
2300	2312	2316	2338	2341	2348	2351	2361	2365	2380	2383	2390	2393	2401	2405
2421	2424	2432	2435	2443	2447	2454	2457	2467	2471	2477	2480	2486	2489	2495
2498	2512	2516	2522	2525	2532	2535	2541	2544	2557	2561	2567	2570	2576	2579
2590	2594	2603	2606	2621	2624	2629	2632	2641	2645	2661	2664	2674	2678	2693
2696	2706	2710	2725	2728	2739	2743	2757	2760	2771	2775	2788	2792	2806	2809
2820	2824	2831	2834	2849	2852	2863	2867	2873	2876	2888	2892	2899	2902	2916
2919	2932	2936	2943	2946	2962	2965	2979	2983	2994	2998	3005	3008	3022	3025
3036	3040	3046	3049	3063	3066	3079	3083	3090	3093	3107	3110	3122	3126	3132
3135	3149	3152	3164	3168	3174	3177	3190	3193	3201	3204	3217	3220	3228	3231
3244	3247	3257	3261	3267	3270	3285	3288	3293	3296	3306	3309	3317	3320	3332
3335	3343	3346	3358	3361	3373	3377	3384	3387	3399	3402	3409	3412	3418	3421
3428	3431	3440	3443	3450	3453	3464	3467	3478	3482	3490	3494	3507	3510	3519
3522	3535	3538	3547	3550	3564	3567	3576	3579	3597	3600	3611	3614	3621	3624
3638	3641	3651	3654	3668	3671	3679	3682	3689	3692	3708	3711	3719	3722	3728
3731	3744	3747	3757	3760	3767	3770	3784	3787	3794	3797	3810	3813	3820	3823
3836	3839	3846	3849	3856	3859	3869	3873	3879	3882	3889	3892	3902	3906	3912
3915	3922	3925	3934	3938	3945	3949	3955	3958	3965	3968	3979	3983	3989	3992
4001	4005	4012	4015	4021	4024	4032	4035	4049	4053	4059	4062	4072	4076	4083
4086	4093	4096	4108	4112	4118	4121	4127	4130	4137	4140	4152	4156	4162	4165
4172	4176	4182	4185	4191	4194	4201	4204	4216	4220	4226	4229	4235	4238	4245
4248	4259	4263	4269	4272	4278	4281	4288	4291	4302	4306	4312	4315	4321	4324
4337	4340	4350	4354	4360	4363	4376	4379	4388	4392	4399	4403	4409	4412	4424
4427	4435	4439	4445	4448	4456	4460	4466	4469	4475	4478	4490	4493	4502	4506
4512	4515	4521	4524	4531	4534	4541	4544	4557	4560	4569	4573	4579	4582	4588
4591	4606	4609	4617	4620	4626	4629	4637	4641	4647	4650	4656	4659	4675	4678
4687	4691	4697	4700	4706	4709	4717	4720	4726	4729	4744	4747	4756	4760	4766
4769	4775	4778	4786	4789	4795	4798	4812	4815	4824	4828	4834	4837	4843	4849
4853	4856	4862	4865	4884	4887	4896	4900	4922	4925	4934	4938	4959	4962	4973
4974	4990	4993	5002	5005	5019	5022	5031	5035	5046	5050	5066	5069	5079	5083
5094	5098	5111	5115	5129	5132	5144	5148	5161	5165	5180	5184	5198	5201	5209
5213	5223	5227	5237	5241	5256	5259	5270	5274	5291	5294	5306	5310	5325	5329
5336	5340	5354	5357	5366	5370	5386	5389	5394	5397	5403	5406	5420	5423	5430
5433	5448	5451	5458	5461	5467	5470	5485	5488	5494	5497	5512	5515	5522	5525
5531	5534	5551	5554	5561	5564	5570	5574	5593	5596	5603	5607	5626	5629	5636
5640	5681	5684	5691	5694	5700	5703	5714	5717	5726	5729	5735	5738	5748	5751
5761	5764	5775	5778	5789	5792	5816	5819	5832	5847	5851	5866	5870	5884	5888
5902	5906	5918	5922	5934	5938	5953	5957	5972	5975	5982	5988	5991	6011	6014
6022	6026	6037	6041	6047	6050	6059	6063	6074	6078	6084	6087	6096	6100	6110
6114	6120	6123	6133	6137	6148	6152	6170	6173	6182	6186	6198	6202	6213	6217
6224	6227	6236	6240	6250	6254	6264	6268	6280	6284	6301	6304	6313	6317	6324

	6327	6336	6340	6351	6355	6361	6364	6373	6377	6386	6392	6408	6411	6419	6423
	6433	6437	6447	6451	6459	6463	6474	6478	6485	6488	6497	6501	6512	6516	6527
	6531	6548	6551	6561	6565	6577	6581	6592	6597	6604	6607	6616	6620	6631	6635
	6647	6651	6661	6665	6672	6675	6684	6688	6706	6709	6718	6722	6733	6737	6748
	6752	6763	6767	6773	6776	6786	6790	6801	6805	6816	6820	6831	6835	6852	6855
	6865	6869	6880	6884	6894	6898	6910	6914	6920	6923	6933	6937	6948	6952	6963
	6957	6977	6981	6987	6990	7000	7004	7015	7019	7029	7033	7046	7050	7056	7059
	7069	7073	7085	7089	7100	7104	7118	7122	7139	7142	7154	7158	7174	7178	7192
	7195	7208	7212	7226	7230	7243	7246	7254	7258	7270	7274	7287	7290	7298	7309
	7313	7322	7330	7334	7350	7353	7360	7363	7375	7379	7386	7389	7397	7400	7406
	7409	7416	7419	7428	7431	7437	7440	7447	7450	7458	7461	7467	7470	7477	7480
	7488	7491	7497	7500	7507	7510	7518	7521	7527	7530	7537	7540	7548	7551	7557
	7560	7575	7578	7584	7587	7598	7601	7608	7611	7621	7625	7631	7634	7640	7643
	7650	7653	7663	7667	7673	7676	7682	7685	7692	7695	7705	7709	7715	7718	7724
	7727	7734	7737	7747	7751	7757	7760	7766	7769	7776	7779	7789	7793	7799	7802
	7808	7811	7818	7821	7831	7835	7841	7844	7850	7853	7868	7871	7879	7882	7895
	7898	7906	7920	7923	7929	7936	7939	7946	7960	7963	7971	7977	7984	7987	7997
	8004	8007	8038	8041	8071	8095	8098	8128	8132	8145	8148	8155	8168	8171	8180
	8184	8204	8207	8220	8223	8242	8246	8255	8258	8282	8354	8358	8362	8366	8370
	8374	8378	8382												
	8307	8324													
.RADIX															
.REM	1														
.REPT	417														
.SBTTL	418	429	456	534	580	596	615	634	653	669	686	706	735	759	779
	801	830	849	867	885	903	927	951	975	997	1021	1046	1070	1096	1120
	1145	1169	1192	1209	1228	1246	1264	1278	1296	1343	1390	1437	1469	1506	1553
	1595	1632	1670	1715	1762	1810	1853	1899	1955	2011	2072	2119	2173	2211	2287
	2338	2380	2421	2454	2495	2541	2576	2621	2661	2693	2725	2757	2806	2849	2916
	2962	3022	3063	3107	3149	3190	3217	3244	3285	3358	3399	3464	3507	3535	3564
	3597	3638	3668	3708	3744	3784	3810	3836	3856	3889	3922	3965	4032	4093	4137
	4201	4245	4288	4337	4376	4424	4490	4557	4606	4675	4744	4812	4884	4922	4959
	4990	5019	5066	5129	5198	5256	5291	5325	5354	5386	5420	5448	5485	5512	5551
	5593	5626	5681	5816	5972	6011	6047	6084	6120	6170	6224	6301	6324	6361	6408
	6485	6548	6604	6672	6706	6773	6852	6920	6987	7056	7139	7192	7243	7287	7350
	7386	7416	7447	7477	7507	7537	7575	7608	7650	7692	7734	7776	7818	7868	7895
	7920	7960	7984	8038	8095	8145	8168	8204	8255						
.TITLE	404														
.WORD	426	434	435	436	437	438	439	440	441	445	446	447	472	473	474
	475	476	477												

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DFKAA8.SEG/SOL/CRF/PAGNUM/NL:TOC/DS:ERFZ=SYSMAC.CO,DFKAA8.P11
RUN-TIME: 88 120 21 SECONDS
RUN-TIME RATIO: 445/230=1.9
CORE USED: 34K (67 PAGES)

