

CQKDA-C KD11-K BASIC LOGIC TESTS
CQKDAC.P11 07-NOV-78 14:09

MACY11 30A(1052) 15-NOV-78 15:26 PAGE 2
DOCUMENT LISTING

SEQ 0001

.SBTTL DOCUMENT LISTING
.TITLE CQKDACO, KD11-K BLT
.REM *

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56

PRODUCT CODE: AC-8090C-MC
PRODUCT NAME: CQKDACO KD11-K BLT
PRODUCT DATE: 15 NOVEMBER 1978
MAINTAINER: DIAGNOSTIC ENGINEERING
AUTHOR: JOHN CARMODY

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE
AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF SOFTWARE ON
EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL OR ITS AFFILIATED COMPANIES.

COPYRIGHT (C) 1977, 1978, BY DIGITAL EQUIPMENT CORPORATION.

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95

TABLE OF CONTENTS

1.0	GENERAL PROGRAM INFORMATION
1.1	PROGRAM PURPOSE
1.2	SYSTEM REQUIREMENTS
1.3	RELATED DOCUMENTS AND STANDARDS
1.4	DIAGNOSTIC HIERARCHY PREREQUISITES
1.5	FAILURE ASSUMPTIONS
2.0	OPERATING INSTRUCTIONS
2.1	LOADING AND STARTING PROCEDURES
2.2	SPECIAL ENVIRONMENTS
2.3	PROGRAM OPTIONS
2.4	EXECUTION TIMES
3.0	ERROR INFORMATION
3.1	ERROR REPORTING PROCEDURES
3.2	ERROR HALTS
4.0	PERFORMANCE AND PROGRESS REPORTS
4.1	PERFORMANCE REPORTS
4.2	PROGRESS REPORTS
4.3	MAINTENANCE BREAKPOINT FEATURE
5.0	MAINTENANCE PROCEDURES
5.1	THE KD11-K PROCESSOR
5.2	CONDITION CODE SCOPE SYNC FEATURE

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
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151

1.0 GENERAL PROGRAM INFORMATION

1.1 PROGRAM PURPOSE

"CQKDA" IS A DIAGNOSTIC PROGRAM DESIGNED TO DETECT, REPORT, AND IDENTIFY LOGIC FAULTS IN THE KD11-K CENTRAL PROCESSING UNIT OF THE PDP11/6X SYSTEM. IT CONSISTS OF 504(10) INDIVIDUAL TESTS CAREFULLY DESIGNED AND SEQUENCED TO DETECT AND ATTEMPT TO IDENTIFY LOGIC FAULTS AT A MINIMUM HARDWARE/SOFTWARE LEVEL. THESE TESTS ARE PARTITIONED INTO FOUR MAJOR SECTIONS AS DESCRIBED BELOW:

A. BASIC CPU TESTS (BCPT)

THIS IS THE BASIC CPU TEST TO VERIFY THE "HARDCORE". ANY FAULT DETECTED IN THIS SECTION CAUSES THE PROGRAM TO HALT WITH THE PC+2 OF THE HALT INSTRUCTION DISPLAYED ON THE CONSOLE.

B. BASIC INSTRUCTION TESTS (BIT)

THIS SECTION CONSISTS OF A LOGICALLY SEQUENCED SET OF BASIC INSTRUCTION TESTS DESIGNED TO VERIFY THE INTEGRITY OF THOSE INSTRUCTIONS AND LOGIC OPERATIONS USED BY THE UTILITY ROUTINES THAT PROVIDE ERROR LOGGING AND SCOPE LOOPING FACILITIES FOR THE SUBSEQUENT TWO MAJOR SECTIONS. NO UTILITY IS CALLED UNTIL ITS INSTRUCTION COMPLEMENT HAS BEEN VERIFIED. THIS SCHEME ACCOMPLISHES TWO IMPORTANT MAINTENANCE OBJECTIVES: 1)IT MINIMIZES THE POSSIBILITY OF THE ERROR REPORTING ROUTINES CONVEYING AMBIGUOUS ERROR INFORMATION TO THE USER, AND 2)IT MAXIMIZES THE POSSIBILITY THAT THE ERROR WILL BE DETECTED BY A ROUTINE DESIGNED TO IDENTIFY FAILING OPERATIONS RATHER THAN HAVE THE ERROR MANIFEST ITSELF IN A MORE COMPLEX UTILITY ROUTINE THAT IS NOT STRUCTURED TO DIAGNOSE FAULTS.

ANY FAULT DETECTED IN THIS SECTION CAUSES THE PROGRAM TO HALT WITH THE CONSOLE ADDRESS INDICATING THE PC+2 OF THE HALT INSTRUCTION IN THE FAILING TEST. ADDITIONAL FAULT IDENTIFICATION INFORMATION IS AVAILABLE IN THE PROCESSOR'S GENERAL REGISTERS, PSW, STACK, AND PROGRAM ANNOTATION FOR THE FAILING TEST. A LOCK ON HARD ERROR FEATURE IS EMPLOYED TO PREVENT THE PROGRAM FROM CONTINUING ON ONCE A SOLID ERROR IS DETECTED. DEPRESSING CONTINUE AFTER THE ERROR HALT CAUSES A RETPY OF THE FAILING TEST.

C. COMPREHENSIVE INSTRUCTION TESTS (CIT)

THIS SECTION, COMPRISED OF THE BULK OF THE TESTS, CONSISTS OF A LOGICALLY SEQUENCED AND PARTITIONED SET OF INSTRUCTION TESTS DESIGNED TO TEST AND VERIFY ALL THE BASIC INSTRUCTIONS OF THE KD11-K PROCESSOR. THIS EXCLUDES TESTING THOSE LOGIC FUNCTIONS THAT SUPPORT THE CONSOLE FUNCTIONS (LOAD ADDRESS, DEPOSIT, ETC.). EACH TEST IN THIS SECTION CALLS A "SCOPE LOOP"

152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207

UTILITY THAT FACILITATES USER CONTROL OF TEST SELECTION AND EXECUTION VIA THE CONSOLE SWITCH REGISTER.

UPON DETECTION OF A LOGIC FAULT, EACH TEST IN THIS SECTION CALLS AN "ERROR SERVICE" ROUTINE THAT LOGS THE ERROR AND REPORTS IT AS HARD COPY ON THE CONSOLE TERMINAL DEVICE. THE ERROR SERVICE ROUTINE ALSO FACILITATES USER CONTROL OF THE PROGRAM SEQUENCE VIA CONSOLE SWITCH REGISTER OPTIONS. AFTER REPORTING THE ERROR THE PROGRAM CONTINUES ON IN ITS NORMAL SEQUENCE UNLESS MODIFIED BY THE USER ACTIVATING THE "LOCK ON HARD ERROR" SWITCH OPTION.

D. COMBINED INSTRUCTION EXERCISER (IEX)

THIS SECTION CONSISTS OF A MORE COMPLEX SET OF INSTRUCTION TESTS DESIGNED TO TEST THE INSTRUCTIONS WHEN USED IN VARIOUS COMBINATIONS MANIPULATING VARIABLE DATA PATTERNS. IT ALSO TESTS THE MED AND ERROR LOGGING FEATURES OF THE CPU. LIKE THE PREVIOUS SECTION, IT CALLS THE "ERROR SERVICE" AND "SCOPE LOOP" UTILITIES TO REPORT ERRORS AND ALLOW USER CONTROL OF TEST EXECUTION.

1.2 SYSTEM REQUIREMENTS

A. HARDWARE REQUIREMENTS

1. PDP11/6X CPU WITH OPERATOR'S CONSOLE
2. 16K OF CORE STORAGE - MF11/U OR EQUIVALENT
3. DL11-W ASYNCHRONOUS LINE INTERFACE WITH LINE CLOCK

B. SOFTWARE REQUIREMENTS

1. PDP11 ABSOLUTE LOADER PROGRAM FOR PAPER TAPE SYSTEMS
2. XXDP MONITOR FOR DECTAPE, MAGTAPE, CASSETTE, OR DISK SYSTEMS.

1.3 RELATED DOCUMENTS AND STANDARDS

"CQKDA" USES THE STANDARD APT SOFTWARE INTERFACES FOUND IN THE MACY11 SYSMAC PACKAGES.

1.4 DIAGNOSTIC HIERARCHY REQUIREMENTS

"CQKDA" WILL NORMALLY BE THE FIRST DIAGNOSTIC TO BE RUN AS PART OF PDP 11/6X CPU CHECKOUT.

1.5 FAILURE ASSUMPTIONS

"CQKDA" ASSUMES THAT THE STORAGE MEDIUM USED TO STORE THE PROGRAM IS INTACT AND THAT IT CAN BE LOADED INTO CORE.

2.0 OPERATING INSTRUCTIONS

208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263

2.1 LOADING AND STARTING PROCEDURES

A. LOADING PROCEDURES

- 1) STANDARD PDP11 ABSOLUTE LOADER PROCEDURES FOR PAPER TAPE.
- 2) STANDARD XXDP MONITOR LOADING PROCEDURES.
- 3) STANDARD APT OR ACT LOADING

B. MANUAL STARTING PROCEDURES

- 1) LOAD SWITCH REG WITH 000000 (NO SWITCH OPTIONS)
- 2) SET DISPLAY TO 000200
- 3) DEPRESS LOAD ADDRESS
- 4) PRESS CNTRL AND START BUTTONS SIMULTANEOUSLY

2.2 SPECIAL ENVIRONMENTS

16K PDP11/6X SERIES SYSTEMS

FOR 16K SYSTEMS USING THE "XXDP" PACKAGE YOU WILL BE UNABLE TO USE THE "UPDATE" PROGRAMS TO LOAD, SAVE, UPDATE ETC. SINCE THE SIZE OF "CQKDA" WILL NOT PERMIT SIMULTANEOUS RESIDENCY OF THE UPDATE PROGRAMS. SUFFICIENT FREE CORE IS AVAILABLE FOR THE "XXDP" MONITOR SO THAT "CQKDA" CAN BE LOADED BY THE MONITOR.

2.3 PROGRAM OPTIONS

A. SWITCH REGISTER OPTIONS

THE FOLLOWING CONSOLE SWITCH REGISTER OPTIONS ARE ACTIVE UPON ENTERING THE COMPREHENSIVE INSTRUCTION TESTS (CIT) SECTION: (SWITCH OPTION IS ACTIVE WHEN SW IS SET TO A "1")

- SW15 HALT ON ERROR. IF ERROR PRINTING IS ENABLED THE HALT OCCURS AFTER THE PRINTOUT. DEPRESSING "CONTINUE" CAUSES THE PROGRAM TO PROCEED ON IN NORMAL SEQUENCE FROM THE POINT OF ERROR.
- SW14 CONTINUOUSLY LOOP ON THE CURRENT TEST
- SW13 INHIBIT NORMAL ERROR PRINTOUTS - THIS DOES NOT INCLUDE POWER FAIL, BUS ERROR, OR RSVD INSTR TRAPS.
- SW12 INHIBIT ALL PRINTOUTS NOT COVERED UNDER SW13. THIS INCLUDES I.D., BUS ERROR, AND RSVD INSTR TRAPS. NOTE THAT IT IS NOT POSSIBLE TO INHIBIT END PASS OR POWER FAIL PRINTOUTS.

264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319

SW11 INHIBIT SUB-TEST ITERATIONS. TEST ITERATIONS ARE
AUTOMATICALLY INHIBITED ON THE FIRST PASS.

SW10 SEARCH FOR AND CONTINUOUSLY LOOP ON THE TEST NUMBER
SELECTED BY THE CONTENTS OF SW<08:00>. ONLY USE THIS
OPTION FOR TESTS TST176 THRU TST767 SINCE THE "SCOPE"
UTILITY IS NOT ACTIVE UNTIL TEST TST176. LOOPING ON
TST176 WILL CAUSE A LOOP ON THE ENTIRE "BIT" SECTION
(TESTS 0-176).

SW09 LOCK ON HARD ERROR

SW<8:0> USED TO SELECT A PARTICULAR TEST FOR LOOPING IF SW10=1.
TEST NUMBER MUST BE BETWEEN 176 AND 767.

B. MEMORY LOCATIONS

4. BPTLOC: THERE IS A LOCATION TAGGED "BPTLOC" THAT PROVIDES THE
USER THE MECHANISM FOR SETTING SIXTEEN "BREAKPOINT
HALTS" THROUGHOUT THE PROGRAM. THIS ENABLES RAPIDLY
"HOMING IN" ON THE FAILING TEST IN THOSE CASES WHERE
THE FAULT CAUSES A RUNAWAY OR HUNG PROGRAM. REFER TO
PARA. 4.2 FOR A DETAILED DESCRIPTION OF THE USE OF
THIS FEATURE.

2.4 EXECUTION TIMES

ONE COMPLETE ERROR FREE PASS OF "CQKDA" WITH NO TEST ITERATIONS
SHOULD TAKE LESS THAN 7 SECONDS. A SUCCESSFUL PASS WILL BE IN-
DICATED BY THE FOLLOWING PRINTOUT ON THE CONSOLE DEVICE:

END PASS # 000001 ERROR COUNT = 000000

THIS ERROR COUNT IS NOT CLEARED AT THE BEGINNING OF A NEW PASS.
WITH ITERATIONS ENABLED A COMPLETE ERROR FREE PASS SHOULD TAKE
LESS THAN 2.5 MINUTES.

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

A. ERROR MESSAGE FORMATS

THERE ARE SEVERAL DIFFERENT ERROR FORMATS. EACH IS DESCRIBED BELOW.

1.) ERROR 1 IS OF THE FORM

S/B	DST	WAS	DST	DEST	(IR)	TEST	(PC)	(SP)	(PSW)
XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX

WHERE:

3 20
 3 21
 3 22
 3 23
 3 24
 3 25
 3 26
 3 27
 3 28
 3 29
 3 30
 3 31
 3 32
 3 33
 3 34
 3 35
 3 36
 3 37
 3 38
 3 39
 3 40
 3 41
 3 42
 3 43
 3 44
 3 45
 3 46
 3 47
 3 48
 3 49
 3 50
 3 51
 3 52
 3 53
 3 54
 3 55
 3 56
 3 57
 3 58
 3 59
 3 60
 3 61
 3 62
 3 63
 3 64
 3 65
 3 66
 3 67
 3 68
 3 69
 3 70
 3 71
 3 72
 3 73
 3 74
 3 75

S/B DST FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS THIS COLUMN CONTAINS WHAT THE RESULT (DEST. OPERAND) SHOULD HAVE BEEN (S/B).
 WAS DST FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS THIS COLUMN CONTAINS WHAT THE RESULT (DEST. OPERAND) ACTUALLY WAS AFTER THE TEST.
 DEST FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS THIS COLUMN CONTAINS THE DESTINATION ADDRESS.
 (IR) THIS IS A COPY OF THE TEST INSTRUCTION. THIS WILL BE THE FIRST WORD IN THE CASE OF TWO OR THREE WORD INSTRUCTIONS.
 TEST INDICATES THE TEST NO. (IN OCTAL) THAT FAILED
 (PC) INDICATES THE CONTENTS OF THE PROGRAM COUNTER AT THE TIME OF THE ERROR CALL. THIS IS AN ADDRESS NORMALLY USED TO LOCATE THE ERROR CALL STATEMENT IN THE FAILING TEST.
 (SP) INDICATES THE CONTENTS OF THE STACK POINTER (R6) AT THE TIME OF THE ERROR. NOTE THAT THE ERROR CALL WILL PUSH THE STACK TWICE. IN SP TESTS WHERE THE SP MUST BE RESTORED PRIOR TO CALLING THE ERROR ROUTINE, THEN THE ORIGINAL (UNRESTORED) SP IS TYPED, WITHOUT ADDITIONAL PUSHES FROM THE ERROR CALL.
 (PSW) INDICATES THE CONTENTS OF THE PROCESSOR STATUS WORD AT THE TIME OF THE ERROR CALL
 XXXXXX IS AN OCTAL NUMBER.

2.) ERROR 2 AND ERROR 4 ARE THE SAME AS FOR ERROR 1 ABOVE EXCEPT THAT IN THIS CASE THE DESTINATION IS A GENERAL REGISTER (WHICH DOES NOT HAVE A UNIBUS ADDRESS). THE OCTAL NUMBER TYPED OUT IN THE "DEST" COLUMN SHOULD BE IGNORED. THE TYPED OUT WOULD LOOK AS FOLLOWS:

S/B DST	WAS DST	DEST	(IR)	TEST	(PC)	(SP)	(PSW)
XXXXXX	XXXXXX	XXXXXX IS R3	XXXXXX	XXXXXX	XXXXXX	XXXXXX	XXXXXX

3.) ERROR 5, ERROR 6, AND ERROR 7 ARE IDENTICAL TO ERROR 1 EXCEPT THAT ONLY THE LAST 5,6, OR 7 COLUMNS (RESPECTIVELY) ARE PRINTED.

4.) ERROR 3 IS USED IN CASES WHERE THE STACK POINTER IS SPECIFICALLY IN ERROR. THE COLUMNS HAVE THE SAME MEANING AS DESCRIBED FOR ERROR 1 EXCEPT:

S/B SP IS WHAT THE STACK POINTER SHOULD HAVE BEEN (S/B)
 WAS SP IS WHAT THE STACK POINTER ACTUALLY WAS

376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431

5.) OTHER ERRORS TYPE OUT THEIR SPECIFIC ERROR MESSAGE, FOLLOWED BY SELF EXPLANATORY DATA HEADERS, DEPENDING ON THE ERROR. AN EXAMPLE FOLLOWS:

BAD DATA READ BY A MED
PC MEDCODE EXPECTD RECEIVD
XXXXXX XXXXXX XXXXXX XXXXXX

6.) WHEN THE SCOPE ROUTINE BECOMES ACTIVE, IT CHECKS THAT THE TEST NUMBER (IN RO) IS EXACTLY ONE GREATER THAN THE TEST NUMBER ON THE PREVIOUS SCOPE CALL. IF A MACHINE ERROR CAUSES TESTS TO BE SKIPPED, OR THE PROGRAM TO JUMP BACKWARDS, ERROR 11 WILL REPORT THIS AS FOLLOWS:

TESTS SKIPPED
PC EXPCTD ACTUAL (TEST #'S)
XXXXXX XXXXXX XXXXXX

EXPCTD THIS IS THE TEST NUMBER THE SCOPE WAS EXPECTING TO BE CALLED FROM.

ACTUAL THIS IS THE TEST NUMBER THAT IT FOUND IN RO

7.) RESERVED INSTRUCTION TRAP ERROR MESSAGE

ANY RESERVED INSTRUCTION TRAP DETECTED AFTER THE BASIC TESTS RESULTS IN THE FOLLOWING PRINTOUT:

TRAPPED TO 10 PC = XXXXXX

WHERE: XXXXXX IS THE VALUE OF THE PROGRAM COUNTER PUSHED ON THE STACK WHEN THE TRAP WAS SPRUNG.

AFTER REPORTING THE ERROR, THE PROGRAM IS RESTARTED FROM THE BEGINNING.

IF A RSVD INSTRUCTION TRAP OCCURS WHILE IN THE PROCESS OF TRYING TO SERVICE A PREVIOUS RSVD INSTRUCTION TRAP OR A BUS ERROR TRAP THE PROGRAM HALTS. A DESCRIPTION OF THIS HALT IS CONTAINED IN PARA. 3.2.3 BELOW.

IF A RSVD INSTRUCTION TRAP OCCURS PRIOR TO COMPLETION OF THE BASIC INSTRUCTION TEST SECTION THE PROGRAM WILL HALT VIA A TRAPCATCHER IN THE VECTOR. A DESCRIPTION OF THIS HALT IS DESCRIBED IN PARA. 3.2.2 BELOW.

4. BUS ERROR TRAP ERROR MESSAGE

ANY UNEXPECTED BUS ERROR TRAPS (BUS TIMEOUT, ODD ADDRESS ERROR, ILLEGAL INSTRUCTION, OR STACK OVERFLOW) RESULTS IN THE FOLLOWING PRINTOUT:

TRAPPED TO 4 PC = XXXXXX

432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487

WHERE: XXXXXX IS THE VALUE OF THE PC PUSHED ONTO
THE STACK WHEN THE TRAP WAS SPRUNG.

AFTER REPORTING THE ERROR THE PROGRAM IS RESTARTED
FROM THE BEGINNING.

IF A BUS ERROR TRAP OCCURS WHILE A PREVIOUS BUS ERROR
OR RSVD INSTRUCTION IS STILL PENDING THE PROGRAM WILL
HALT. A DESCRIPTION OF THE HALT INTERPRETATION IS GIVEN
IN PARA. 3.2.3 BELOW.

IF A BUS ERROR OCCURS PRIOR TO THE COMPLETION OF THE
BASIC INSTRUCTION TESTS, THE PROGRAM WILL HALT VIA A
TRAPCATCHER IN THE VECTOR. A DESCRIPTION OF THIS HALT
IS INCLUDED IN PARA. 3.2.2 BELOW.

5. POWER FAIL

IF A POWER FAIL CONDITION IS DETECTED, THE FOLLOWING
MESSAGE IS PRINTED:

POWER

AFTER PRINTING AN ATTEMPT IS MADE TO RESTART THE PROGRAM AT
THE BEGINNING.

3.2 ERROR HALTS

1. BASIC INSTRUCTION TESTS (BIT)

ANY ERROR DETECTED IN THE BASIC TESTS CAUSES THE
PROGRAM TO HALT WITH THE PC+2 OF THE LOCATION CONTAINING
THE HALT INSTRUCTION DISPLAYED.

EXAMINING THE CONTENTS OF THE CPU'S GENERAL REGISTERS,
THE PSW, AND THE STACK WILL PROVIDE ADDITIONAL FAULT
IDENTIFICATION INFORMATION.

DEPRESSING "CONTINUE" AFTER THE HALT WILL CAUSE AN
AUTOMATIC RETRY OF THE FAILING TEST. IF THE ERROR IS
SOLID THE PROGRAM WILL LOCK ON THIS TEST, BUT IF IT
IS INTERMITTENT THE PROGRAM WILL CONTINUE ON IN NORMAL
SEQUENCE ONCE THE TEST IS SUCCESSFULLY EXECUTED.

TO ESTABLISH A TIGHT SCOPE LOOP ON THE FAILING TEST,
REPLACE THE "HALT" WITH A 400(8). AND DEPRESS "CONTINUE"
THE "400" IS A "BR .+2" WHICH FUNCTIONS AS A NOP. THIS
IS NECESSARY TO PRESERVE THE INTEGRITY OF THE CONDITION
CODE OPERATE INSTRUCTION THAT IS USED AS A SCOPE SYNC. THIS
BUILT IN SYNC FEATURE IS DESCRIBED IN PARA. 5.0.

2. TRAPCATCHER HALTS

488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543

THE VECTOR AREA (LOC 000 - 776) IS PROGRAM LOADED WITH
A STANDARD TRAPCATCHER AS SHOWN BELOW:

V / V+2
V+2/ HALT

AFTER THE BASIC INSTRUCTION TESTS THE FOLLOWING VECTORS
ARE SET UP TO POINT TO APPROPRIATE SERVICE ROUTINES:

4/6 BUS ERROR SERVICE
10/12 RSVD INSTRUCTION TRAP SERVICE
20/22 SCOPE LOOP SERVICE
24/26 POWER FAIL SERVICE
30/32 ERROR SERVICE
34/36 PRINT SERVICE

AT THE APPROPRIATE POINTS IN THE COMPREHENSIVE INSTR-
UCTION TESTS THE LINE CLOCK VECTOR (100/102) AND THE DL11
VECTORS (60/62 - 64/66) ARE SET UP TO CHECK INTERRUPTS
FROM THESE DEVICES. ALL OTHER VECTORS REMAIN SET UP TO
"CATCH" UNEXPECTED TRAPS OR INTERRUPTS BY HALTING.

WHEN AN UNEXPECTED TRAP OR INTERRUPT NOT SUPPORTED BY
AN APPROPRIATE SERVICE ROUTINE OCCURS THE CPU HALTS.
WITH THE PC+4 OF THE VECTOR DISPLAYED IN THE CONSOLE.
THIS IS USED TO IDENTIFY THE CAUSE OF THE UNEXPECTED
TRAP OR INTERRUPT.

THE LAST ENTRY PUSHED ON THE STACK CAN BE EXAMINED
TO DETERMINE WHERE THE PROGRAM WAS WHEN THE TRAP OR
INTERRUPT WAS SPRUNG. REMEMBER THAT THE "OLD PC" GETS
SAVED ON THE STACK WHEN A TRAP OR INTERRUPT OCCURS.

3. CATASTROPHIC ERROR HALTS

THERE ARE TWO HALTS, ONE IN THE BUS ERROR SERVICE ROU-
TINE AND THE OTHER IN THE RSVD INSTRUCTION TRAP SERVICE
ROUTINE THAT HALT THE PROGRAM IF ONE OF THESE ERRORS
OCCURS WHILE STILL SERVICING A PREVIOUS BUS ERROR
OR RSVD INSTRUCTION TRAP. AFTER THE HALT THE CONSOLE
DISPLAYS THE PC+2 OF THE ERROR HALT. THIS IS USED
TO IDENTIFY WHICH OF THE TWO TYPES OF ERRORS - RSVD
OR BUS ERROR - OCCURRED LAST.

THERE IS A SOFTWARE FLAG TAGGED "CATERR" THAT MAY BE
EXAMINED TO OBTAIN THE FOLLOWING INFORMATION:

[CATERR] = 000002 TWO SUCCESSIVE BUS ERRORS
[CATERR] = 001000 TWO SUCCESSIVE RSVD INSTR. TRAPS
[CATERR] = 000401 A COMBINATION OF THE TWO. THE
CONTENTS OF THE ADDRESS DISPLAY
IDENTIFIES WHICH TYPE OCCURRED LAST.

THE STACK PROVIDES THE FOLLOWING ADDITIONAL INFCRMATION:

544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599

[SP] / PC OF THE 2ND TRAP
[SP+2] / PSW OF THE 2ND TRAP
[SP+4] / PC OF THE 1ST TRAP
[SP+6] / PSW OF THE 1ST TRAP

4.0 PERFORMANCE AND PROGRESS REPORTS

4.1 PERFORMANCE REPORTS

THERE IS ONLY ONE PERFORMANCE REPORT SUPPLIED BY THE PROGRAM AND CONSISTS OF A SIMPLE END OF PASS MESSAGE OF THE FORMAT SHOWN BELOW:

PASCNT = XXXXXX ERRCNT = YYYYYY

WHERE: XXXXXX IS THE TOTAL NUMBER OF COMPLETE PASSES OF THE ENTIRE PROGRAM (OCTAL)

YYYYYY IS THE TOTAL ERROR COUNT IN OCTAL

4.2 PROGRESS REPORTS

THERE ARE TWO PROGRESS REPORTS PRINTED THAT REPORT NORMAL ERROR FREE EXECUTION OF THE PROGRAM.

A. END OF PASS PRINTOUT AS DESCRIBED IN 4.1 ABOVE.

B. PROGRAM IDENTIFICATION MESSAGE AS DESCRIBED BELOW:

CQKDACC KD11-K BASIC LOGIC TESTS

THIS MESSAGE GETS PRINTED THE FIRST TIME THE PROGRAM ENTERS THE COMPREHENSIVE INSTRUCTION TEST SECTION UNLESS INHIBITED BY SW12=1. AFTER THE FIRST PASS THIS PRINTOUT IS AUTOMATICALLY INHIBITED UNLESS THE PROGRAM IS RESTARTED AT 200(8).

4.3 MAINTENANCE BREAKPOINT FEATURE

THERE IS A MANUAL PROGRESS REPORT FEATURE THAT ALLOWS THE USER TO STEP THROUGH THE PROGRAM, HALTING AFTER EVERY NTH TEST WITH PROGRESS INFORMATION DISPLAYED IN THE CONSOLE ADDRESS DISPLAYS. TO ACTIVATE THIS FEATURE THE USER MUST SET THE DESIRED "BREAKPOINT HALT" BITS IN THE MEMORY LOCATION TAGGED "BPTLOC". THIS LOCATION PROVIDES SIXTEEN POSSIBLE HALTS DISPERSED EVENLY THROUGHOUT THE PROGRAM (APPROX. EVERY 20 TESTS). AT EACH CHECKPOINT THE PROGRAM EXAMINES A PARTICULAR BIT IN "BPTLOC" AND HALTS IF THE BIT IS SET TO A "1" OTHERWISE IT CONTINUES IN NORMAL SEQUENCE. AFTER THE HALT DEPRESSING "CONTINUE" WILL CAUSE RESUMPTION OF NORMAL PROGRAM EXECUTION. SETTING LOCATION "BPTLOC" TO ALL 1'S (177777) WILL RESULT IN THE FOLLOWING SIXTEEN HALTS WITH THE INFORMATION SHOWN DISPLAYED IN THE CONSOLE:

656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711

*
*
***>DATA OUT

THE DATA PATHS CONSIST OF A LOGICALLY INTERCONNECTED GROUP OF STATIC DATA FACILITIES (REGISTERS, MULTIPLEXORS, ALU'S ETC.) REQUIRED TO TEMPORARILY STORE, MODIFY, AND TRANSFER DATA ITEMS (16 BIT WORDS OR 8 BIT BYTES) ACCORDING TO THE DESIGN SPECIFICATIONS FOR THE PDP11.

THE CONTROL SECTION SUPPLIES PREDEFINED SEQUENCES OF CONTROL SIGNAL SETS TO ACTIVATE THE REQUIRED DATA FACILITIES WITHIN THE DATA PATHS. IN THE KD11-K THESE CONTROL SIGNAL SETS ARE STORED IN A READ ONLY MEMORY (ROM) AND GENERATED BY READING OUT A UNIQUE SEQUENCE OF ROM WORDS FOR EACH OPERATION TO BE PERFORMED.

THE SEQUENCE GENERATED BY THE CONTROL SECTION IS VARIABLE AND DEPENDENT UPON THE INSTRUCTION OR LOGIC OPERATION BEING EXECUTED. THERE ARE HUNDREDS OF THESE SEQUENCES POSSIBLE DEPENDENT UPON OF THE PROGRAM CODING.

"CQKDA" IS DESIGNED TO GENERATE ALL POSSIBLE MICROINSTRUCTION SEQUENCES AND COMBINATIONS OF DATA AND CONTROL SIGNALS. THE INDIVIDUAL TESTS ARE LOGICALLY SEQUENCED AND STRUCTURED TO DETECT AND ISOLATE PARTICULAR MICROPROGRAM SEQUENCES THAT ARE FAULTY.

5.2 CONDITION CODE SCOPE SYNC FEATURE

FROM THE BIT SECTION TO THE MED TESTS IN THE CIT SECTION, ALL TEST INSTRUCTIONS ARE PRECEDED BY A CONDITION CODE OPERATE INSTRUCTION. THE UBREAK REGISTER IS PROGRAM LOADED TO GENERATE A SYNC PULSE NEAR THE END OF THIS INSTRUCTION. DURING THE MED TESTS, THE PULSE IS GENERATED NEAR THE BEGINNING OF THE MED EXECUTION. THIS PULSE IS GENERATED ON BACKPLANE PIN B03M2 AND MAY BE USED IN CONJUNCTION WITH THE PROGRAM LOOPING FEATURES TO PROBE THE KD11-K DURING THE FAILING TEST.

%

```
.TITLE CQKDA-C KD11-K BASIC LOGIC TESTS
;*COPYRIGHT (C) 1977,1978
;*DIGITAL EQUIPMENT CORP.
;*MAYNARD, MASS. 01754
;
;
; *
; *
; *THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
; *PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
; *
; *
; *SBTTL OPERATIONAL SWITCH SETTINGS
; *
; *      SWITCH           USE
; *      -----          -
```

```

712      *          15          HALT ON ERROR
713      *          14          LOOP ON TEST
714      *          13          INHIBIT ERROR TYPEOUTS
715      *          12          INHIBIT TO MESSAGE & UNEXPECTED TRAP MESSAGES
716      *          11          INHIBIT ITERATIONS
717      *          10          LOOP ON TEST IN SWR<8:0>
718      *          9          LOOP ON ERROR
719      *
720      *ENABLE ABS
721      *SBTTL BASIC DEFINITIONS
722
723      001000      *INITIAL ADDRESS OF THE STACK POINTER *** 1000 ***
724      *STACK= 1000
725      *EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
726      *EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL
727
728      *MISCELLANEOUS DEFINITIONS
729      *HT= 11      ;;CODE FOR HORIZONTAL TAB
730      *LF= 12      ;;CODE FOR LINE FEED
731      *CR= 13      ;;CODE FOR CARRIAGE RETURN
732      *CRLF= 120    ;;CODE FOR CARRIAGE RETURN-LINE FEED
733      *PS= 177776   ;;PROCESSOR STATUS WORD
734      *EQUIV PS,PSW
735      *STKLMT= 177774 ;;STACK LIMIT REGISTER
736      *PIRQ= 177772  ;;PROGRAM INTERRUPT REQUEST REGISTER
737      *DSWR= 177570  ;;HARDWARE SWITCH REGISTER
738      *DDISP= 177570 ;;HARDWARE DISPLAY REGISTER
739
740      *GENERAL PURPOSE REGISTER DEFINITIONS
741      *R0= %0      ;;GENERAL REGISTER
742      *R1= %1      ;;GENERAL REGISTER
743      *R2= %2      ;;GENERAL REGISTER
744      *R3= %3      ;;GENERAL REGISTER
745      *R4= %4      ;;GENERAL REGISTER
746      *R5= %5      ;;GENERAL REGISTER
747      *R6= %6      ;;GENERAL REGISTER
748      *R7= %7      ;;GENERAL REGISTER
749      *SP= %8      ;;STACK POINTER
750      *PC= %9      ;;PROGRAM COUNTER
751
752      *PRIORITY LEVEL DEFINITIONS
753      *PR0= 0      ;;PRIORITY LEVEL 0
754      *PR1= 40     ;;PRIORITY LEVEL 1
755      *PR2= 100    ;;PRIORITY LEVEL 2
756      *PR3= 140    ;;PRIORITY LEVEL 3
757      *PR4= 200    ;;PRIORITY LEVEL 4
758      *PR5= 240    ;;PRIORITY LEVEL 5
759      *PR6= 300    ;;PRIORITY LEVEL 6
760      *PR7= 340    ;;PRIORITY LEVEL 7
761
762      *"SWITCH REGISTER" SWITCH DEFINITIONS
763      *SW15= 100000
764      *SW14= 40000
765      *SW13= 20000
766      *SW12= 10000
767      *SW11= 4000
768      *SW10= 2000

```

```

768      001000      *SW09= 1000
769      000400      *SW08= 400
770      000200      *SW07= 200
771      000100      *SW06= 100
772      000040      *SW05= 40
773      000020      *SW04= 20
774      000010      *SW03= 10
775      000004      *SW02= 4
776      000002      *SW01= 2
777      000001      *SW00= 1
778      *EQUIV SW09,SW9
779      *EQUIV SW08,SW8
780      *EQUIV SW07,SW7
781      *EQUIV SW06,SW6
782      *EQUIV SW05,SW5
783      *EQUIV SW04,SW4
784      *EQUIV SW03,SW3
785      *EQUIV SW02,SW2
786      *EQUIV SW01,SW1
787      *EQUIV SW00,SW0
788
789      *DATA BIT DEFINITIONS (BIT00 TO BIT15)
790      *BIT15= 100000
791      *BIT14= 40000
792      *BIT13= 20000
793      *BIT12= 10000
794      *BIT11= 4000
795      *BIT10= 2000
796      *BIT09= 1000
797      *BIT08= 400
798      *BIT07= 200
799      *BIT06= 100
800      *BIT05= 40
801      *BIT04= 20
802      *BIT03= 10
803      *BIT02= 4
804      *BIT01= 2
805      *BIT00= 1
806      *EQUIV BIT09,BIT9
807      *EQUIV BIT08,BIT8
808      *EQUIV BIT07,BIT7
809      *EQUIV BIT06,BIT6
810      *EQUIV BIT05,BIT5
811      *EQUIV BIT04,BIT4
812      *EQUIV BIT03,BIT3
813      *EQUIV BIT02,BIT2
814      *EQUIV BIT01,BIT1
815      *EQUIV BIT00,BIT0
816
817      *BASIC "CPU" TRAP VECTOR ADDRESSES
818      *SRRVEC= 4      ;;TIME OUT AND OTHER ERRORS
819      *RESVEC= 10     ;;RESERVED AND ILLEGAL INSTRUCTIONS
820      *TRITVEC= 14    ;;TRAP BIT
821      *TRITVEC= 14    ;;TRACE TRAP
822      *BPTVEC= 14     ;;BREAKPOINT TRAP (BPT)
823      *IOTVEC= 20     ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**

```

```

824      000024
825      000030
826      000034
827      000060
828      000064
829      000240
831      000000
832      000000
833      000174
834      000000
835      000176 000000
837      000174 000000
838      000176 000000
839      000200 000137 001630
840      000700
841      000700
842      000700
843      000700
844      000700
845      000700
846      000700
847      000024 000024
848      000024 000024
849      000024 000200
850      000044 000044
851      000044 000700
852      000700
853      000700
854      000700
855      000700
856      000700
857      000700 000000
858      000700 001120
859      000700 060644
860      000704 000000
861      000706 000000
862      000710 000000
863      000712 000014
864      000714
865      000046 000046
866      000052 000052
867      000052 000000
868      000052 000714
869      000052 000000
870      000052 000052
871      000052 000000
872      000052 000000
873      000052 000714

```

```

PWRVEC= 24          ;;POWER FAIL
EMTVEC= 30          ;;EMULATOR TRAP (EMT) **ERROR**
TRAPVEC=34         ;;"TRAP" TRAP
TKVEC= 60          ;;TTY KEYBOARD VECTOR
TPVEC= 64          ;;TTY PRINTER VECTOR
PIRQVEC=240        ;;PROGRAM INTERRUPT REQUEST VECTOR
.SBTTL TRAP CATCHER

.=0
; *ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A "+2,HALT"
; *SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
; *LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
DISPREG: =174
SWREG: =WORD 0      ;;SOFTWARE DISPLAY REGISTER
.SBTTL STARTING ADDRESS(ES)
JMP @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
.=700
.SBTTL APT PARAMETER BLOCK

;*****
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
;*****
.SX=          ;;SAVE CURRENT LOCATION
.=24         ;;SET POWER FAIL TO POINT TO START OF PROGRAM
200         ;;FOR APT START UP
.=44         ;;POINT TO APT INDIRECT ADDRESS PNTR.
$APTHDR     ;;POINT TO APT HEADER BLOCK
.=52         ;;RESET LOCATION COUNTER
;*****
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
;INTERFACE SPEC.

$APTHD:
$BITS: =WORD 0     ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
$BADR: =WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
$STMT: =WORD 0     ;;RUN TIME OF LONGEST TEST
$PASTM: =WORD 0    ;;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
.SBTTL ACT11 HOOKS

;*****
;HOOKS REQUIRED BY ACT11
$SVPC=.          ;;SAVE PC
.=46           ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
$END        ;;
.=52         ;;2)SET LOC.52 TO ZERO
$WORD 0       ;;
.=$$VPC      ;; RESTORE PC

```

```

874      001000
875      001000
876      001002 000
877      001003 000
878      001004 0000000
879      001006 0000000
880      001010 0000000
881      001012 0000000
882      001014 000
883      001015 001
884      001016 0000000
885      001020 0000000
886      001024 0000000
887      001026 0000000
888      001030 0000000
889      001032 0000000
890      001034 000
891      001036 000
892      001040 0000000
893      001042 177570
894      001044 177570
895      001046 177560
896      001050 177564
897      001052 177568
898      001054 000
899      001055 002
900      001056 012
901      001057 000
902      001060 0000000
903      001062 0000000
904      001064 0000000
905      001066 0000000
906      001070 0000000
907      001072 0000000
908      001074 0000000
909      001076 0000000
910      001100 0000000
911      001102 0000000
912      001104 0000000
913      001106 0000000
914      001110 0000000
915      001112 0000000
916      001114 077
917      001115 015
918      001116 000012

```

```

.SBTTL COMMON TAGS

;*****
;THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
;USED IN THE PROGRAM.

.=1000
$CMTAG: =WORD 0      ;;START OF COMMON TAGS
$STNM: =BYTE 0      ;;CONTAINS THE TEST NUMBER
$ERFLG: =BYTE 0     ;;CONTAINS ERROR FLAG
$ICNT: =WORD 0      ;;CONTAINS SUBTEST ITERATION COUNT
$LPADR: =WORD 0     ;;CONTAINS SCOPE LOOP ADDRESS
$LPERR: =WORD 0     ;;CONTAINS SCOPE RETURN FOR ERRORS
$ERTTL: =WORD 0     ;;CONTAINS TOTAL ERRORS DETECTED
$TEMB: =BYTE 0      ;;CONTAINS ITEM CONTROL BYTE
$ERRMAX: =BYTE 1    ;;CONTAINS MAX. ERRORS PER TEST
$ERRPC: =WORD 0     ;;CONTAINS PC OF LAST ERROR INSTRUCTION
$GDADR: =WORD 0     ;;CONTAINS ADDRESS OF "GOOD" DATA
$BDADR: =WORD 0     ;;CONTAINS ADDRESS OF "BAD" DATA
$GDAT: =WORD 0      ;;CONTAINS "GOOD" DATA
$BDAT: =WORD 0      ;;CONTAINS "BAD" DATA
$RESV: =WORD 0      ;;RESERVED--NOT TO BE USED
$AUTOB: =BYTE 0     ;;AUTOMATIC MODE INDICATOR
$INTAG: =BYTE 0     ;;INTERRUPT MODE INDICATOR
$SWR: =WORD 0       ;;ADDRESS OF SWITCH REGISTER
$DISP: =WORD 0      ;;ADDRESS OF DISPLAY REGISTER
$TKS: =WORD 177560  ;;TTY KBD STATUS
$TKB: =WORD 177562  ;;TTY KBD BUFFER
$TPB: =WORD 177564  ;;TTY PRINTER BUFFER REG. ADDRESS
$TPS: =WORD 177568  ;;TTY PRINTER STATUS REG. ADDRESS
$NULL: =BYTE 0      ;;CONTAINS NULL CHARACTER FOR FILLS
$FILLS: =BYTE 2     ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
$FILLC: =BYTE 12    ;;INSERT FILL CHARS. AFTER A "LINE FEED"
$STPLG: =BYTE 0     ;;TERMINAL AVAILABLE FLAG (BIT<07>=0=YES)
$REGAD: =WORD 0     ;;CONTAINS THE ADDRESS FROM WHICH ($REGO) WAS OBTAINED
$REGO: =WORD 0      ;;CONTAINS ((SREGAD)+0)
$REG1: =WORD 0      ;;CONTAINS ((SREGAD)+2)
$REG2: =WORD 0      ;;CONTAINS ((SREGAD)+4)
$REG3: =WORD 0      ;;CONTAINS ((SREGAD)+6)
$REG4: =WORD 0      ;;CONTAINS ((SREGAD)+8)
$REG5: =WORD 0      ;;CONTAINS ((SREGAD)+10)
$REG6: =WORD 0      ;;CONTAINS ((SREGAD)+12)
$TMP0: =WORD 0      ;;USER DEFINED
$TMP1: =WORD 0      ;;USER DEFINED
$TMP2: =WORD 0      ;;USER DEFINED
$TMP3: =WORD 0      ;;USER DEFINED
$TMP4: =WORD 0      ;;USER DEFINED
$TIMES: 0           ;;MAX. NUMBER OF ITERATIONS
$ESCAP: 0           ;;ESCAPE ON ERROR ADDRESS
$QUES: =ASCII ?/?  ;;QUESTION MARK
$CRLF: =ASCII <15> ;;CARRIAGE RETURN
$LF: =ASCII <12>  ;;LINE FEED
;*****

```

930
931
932
933
934 001120
935 001120 000000
936 001122 000000
937 001124 000000
938 001126 000000
939 001130 000000
940 001132 000000
941 001134 000000
942 001136 000000
943 001140
944 001140 000
945 001141 000
946 001142 000000
947 001144 000000
948 001146 000000
949
950
951
952
953
954 001150
955
956

.SBTTL APT MAILBOX-ETABLE

;EVEN
;MAIL: ;APT MAILBOX
;MSGTY: .WORD AMSGTY ;MESSAGE TYPE CODE
;FATAL: .WORD AFATAL ;FATAL ERROR NUMBER
;TESTN: .WORD ATESTN ;TEST NUMBER
;PASS: .WORD APASS ;PASS COUNT
;DEVCT: .WORD ADEVCT ;DEVICE COUNT
;UNIT: .WORD AUNIT ;I/O UNIT NUMBER
;MSGAD: .WORD AMSGAD ;MESSAGE ADDRESS
;MSGLG: .WORD AMSGLG ;MESSAGE LENGTH
;ETABLE: ;APT ENVIRONMENT TABLE
;ENV: .BYTE AENV ;ENVIRONMENT BYTE
;ENVM: .BYTE AENVM ;ENVIRONMENT MODE BITS
;SREG: .WORD ASWREG ;APT SWITCH REGISTER
;USWR: .WORD AUSWR ;USER SWITCHES
;CPUOP: .WORD ACPUDP ;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,P00=07,Q=10
; ;BIT 10=REAL TIME CLOCK
; ;BIT 9=FLOATING POINT PROCESSOR
; ;BIT 8=MEMORY MANAGEMENT
;ETEND:
;MEXIT

957
958
959
960
961
962
963
964
965
966
967
968
969
970
971 001150
972
973
974 001150 064640
975 001152 000000
976 001154 067764
977 001156 000000
978
979 001160 064640
980 001162 065061
981 001164 067764
982 001166 000000
983
984 001170 065013
985 001172 000000
986 001174 070006
987 001176 000000
988
989 001200 064640
990 001202 065072
991 001204 067764
992 001206 000000
993
994 001210 064666
995 001212 000000
996 001214 067772
997 001216 000000
998
999 001220 064660
1000 001222 000000
1001 001224 067770
1002 001226 000000
1003
1004 001230 064650
1005 001232 000000
1006 001234 067766
1007 001236 000000
1008
1009 001240 064724
1010 001242 000000
1011 001244 067764
1012 001246 000000

.SBTTL ERROR POINTER TABLE
;THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
;THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
;LOCATION SITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;NOTE1: IF SITEMB IS 0 THE ONLY PERTINENT DATA IS (\$ERRPC).
;NOTE2: EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:
;* EM ;POINTS TO THE ERROR MESSAGE
;* DH ;POINTS TO THE DATA HEADER
;* DT ;POINTS TO THE DATA
;* DF ;POINTS TO THE DATA FORMAT
\$ERRTB:
;ITEM 1
EM1 ;S/B DST WAS DST DEST (IR) TEST (PC) (SP) (PSW)
DT1 ;\$REG4, \$REG3, \$REG2, \$REG1,\$REG0,\$ERRPC,\$REG5,\$REG6
;ITEM 2
EM2 ;S/B DST WAS DST DEST (IR) TEST (PC) (SP) (PSW)
DT2 ;\$REG4, \$REG3, \$REG2, \$REG1, \$REG0,\$ERRPC,\$REG5,\$REG6
;ITEM 3
EM3 ;S/B SP WAS SP (IR) TEST (PC) (PSW)
DT3 ;\$REG4, \$REG3, \$REG1,\$REG0,\$ERRPC,\$REG6
;ITEM 4
EM4 ;S/B DST WAS DST DEST (IR) TEST (PC) (SP) (PSW)
DT4 ;\$REG4, \$REG3, \$REG2, \$REG1, \$REG0, \$ERRPC, \$REG5, \$REG6
;ITEM 5
EM5 ;(IR) TEST (PC) (SP) (PSW)
DT5 ;\$REG1, \$REG0, \$ERRPC, \$REG5, \$REG6
;ITEM 6
EM6 ; DEST (IR) TEST (PC) (SP) (PSW)
DT6 ;\$REG2, \$REG1, \$REG0, \$ERRPC, \$REG5, \$REG6
;ITEM 7
EM7 ;WAS DST DEST (IR) TEST (PC) (SP) (PSW)
DT7 ;\$REG3, \$REG2, \$REG1, \$REG0, \$ERRPC, \$REG5, \$REG6
;ITEM 10
EM10 ;S/B RES WAS RES DST OP STC OP TEST (PC) (SP) (PSW)
DT10 ;\$REG4, \$REG3, \$REG2, \$REG1, \$REG0, \$ERRPC, \$REG5, \$REG6

1013
1014
1015
1016
1017
1018
1019
1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
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

001250 065260
001251 065276
001254 070024
001256 000000
001260 065334
001262 067372
001264 067664
001266 000000
001270 065373
001272 067372
001274 067664
001276 000000
001300 065425
001302 067372
001304 067664
001306 000000
001310 065445
001312 067303
001314 067670
001316 067756
001320 067034
001322 067606
001324 067670
001326 000000
001330 067005
001332 067606
001334 067670
001336 000000
001340 065704
001342 067606
001344 067670
001346 000000
001350 065534
001352 067335
001354 067640
001356 067760

```

;ITEM 11
EM11 ;TESTS SKIPPED
DH11 ;PC EXPECTD ACTUAL (TEST #'S)
DT11 ;$ERRPC,$TESTN,$REGO
0

;ITEM 12
EM12 ;MED DID NOT ABORT IN USER MODE
DH22 ;PC
DT23 ;$ERRPC
0

;ITEM 13
EM13 ;MED EXECUTED IN USER MODE
DH23 ;PC
DT23 ;$ERRPC
0

;ITEM 14
EM14 ;MED CHANGED PSW
DH23 ;PC
DT23 ;$ERRPC
0

;ITEM 15
EM15 ;MICROBREAK TRAP-TO-4 DID NOT OCCUR
DH15 ;ERRPC MEDCODE MICROBK REG.
DT15 ;$ERRPC,$TMP0,$TMP1,0
DF15 ;0,0

;ITEM 16
EM16 ;CACHE DATA LOGGED INCORRECTLY
DH44 ;PC EXPCT RECVD
DT24 ;$ERRPC,$REG1,$REG0,0
0

;ITEM 17
EM45 ;CACHE TAG LOGGED WRONG
DH44 ;PC EXPCT RECVD
DT24 ;$ERRPC,$REG0,$REG1,0
0

;ITEM 20
EM26 ;PHYS. BA LOGGED WRONG
DH44 ;PC EXPCT RECVD
DT24 ;$ERRPC,$REG1,$REG0,0
0

;ITEM 21
EM21 ;CSP CONSTANT WRONG
DH17 ;PC MEDCODE EXPECTD RECEIVD
DT17 ;$ERRPC,$TMP1,$TMP2,$REG0,0
DF17 ;0,0,0
    
```

1069
1070
1071
1072
1073
1074
1075
1076
1077
1078
1079
1080
1081
1082
1083
1084
1085
1086
1087
1088
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
1108
1109
1110
1111
1112
1113
1114
1115
1116
1117
1118
1119
1120
1121
1122
1123
1124

001360 065557
001362 067335
001364 067664
001366 067760
001370 065606
001372 067372
001374 067664
001376 000000
001400 065625
001402 067372
001404 067670
001406 000000
001410 065510
001412 067606
001414 067670
001416 000000
001420 065704
001422 067436
001424 067706
001426 000000
001430 065731
001432 067520
001434 067732
001436 000000
001440 066001
001442 067372
001444 067664
001446 000000
001450 066604
001452 067372
001454 067664
001456 000000

```

;ITEM 22
EM22 ;BAD DATA READ BY A MED
DH22 ;PC MEDCODE EXPECTD RECEIVD
DT22 ;$ERRPC,$TMP1,$TMP2,$TMP3,0
DF17 ;0,0,0,0

;ITEM 23
EM23 ;NO ODD PC TRAP
DH23 ;PC
DT23 ;$ERRPC
0

;ITEM 24
EM24 ;ODD ADR. BIT NOT SET IN CPU ERROR REGISTER OR LOC JAM
DH24 ;PC CPUERR LOGJAM
DT24 ;$ERRPC,$REG1,$REG0
0

;ITEM 25
EM17 ;LOG CUA LOGGED INCORRECT U-ADDR
DH44 ;PC EXPCTD RECVD
DT24 ;$ERRPC,$REG1,$REG0
0

;ITEM 26
EM26 ;PHYS. BA LOGGED WRONG
DH26 ;PC PA<17:16>-EXPCT-PA<15:0> PA<17:16>-RECVD-PA<15:0>
DT26 ;$ERRPC,$REG1,$REG2,$REG0,$REG3,0
0

;ITEM 27
EM27 ;CACHE PARITY ERROR LOGGED IN BACK UP MODE
DH27 ;PC LOGPBA LOGDATA LOGTAG
DT27 ;$ERRPC,$REG3,$REG1,$REG2
0

;ITEM 30
EM30 ;CACHE PARITY TRAPPED WHEN DISABLED
DH23 ;PC
DT23 ;$ERRPC
0

;ITEM 31
EM31 ;NO CACHE PARITY TRAP
DH23 ;PC
DT23 ;$ERRPC
0

;ITEM 32
    
```

1125			EM32	;MEMORY ERROR REGISTERS INCORRECT
1126	001460	066113	DH32	;PC MEMERR
1127	001462	067553	DT25	;SERRPC,\$REGO
1128	001464	067700	0	
1129	001466	000000		
1130				
1131				;ITEM 33
1132			EM33	;TIMEOUT BIT NOT SET IN CPU ERROR REGISTER OR LOG JAM
1133	001470	066144	DH24	;PC CPUERR LOGJAM
1134	001472	067377	DT24	;SERRPC,\$REG1,\$REGO
1135	001474	067670	0	
1136	001476	000000		
1137				
1138				;ITEM 34
1139			EM34	;NO ILLEGAL INTERNAL ADDRESS TRAP
1140	001500	066222	DH23	;PC
1141	001502	069244	DT23	;SERRPC
1142	001504	067864	0	
1143	001506	000000		
1144				
1145				;ITEM 35
1146			EM35	;INTERNAL ADDRESS ERROR BIT NOT SET IN CPU ERROR REGISTER OR LOG JAM
1147	001510	066257	DH24	;PC CPUERR LOGJAM
1148	001512	067377	DT24	;SERRPC,\$REG1,\$REGO
1149	001514	067670	0	
1150	001516	000000		
1151				
1152				;ITEM 36
1153			EM36	;LAST INTERRUPT/TRAP VECTOR NOT LOGGED IN FLAG REGISTER
1154	001520	066345	DH25	;PC FLGREG
1155	001522	067422	DT25	;SERRPC,\$REGO
1156	001524	067700	0	
1157	001526	000000		
1158				
1159				;ITEM 37
1160			EM37	;LOG FIRST MODE DID NOT INHIBIT ERROR LOG AFTER FIRST ERROR
1161	001530	066422	DH24	;PC CPUERR LOGJAM
1162	001532	067377	DT24	;SERRPC,\$REG1,\$REGO
1163	001534	067670	0	
1164	001536	000000		
1165				
1166				;ITEM 40
1167			EM40	;ERROR LOG WAS NOT RE-ENABLED, ODD ADR BIT CLR IN CPUERR
1168	001540	066515	DH24	;PC CPUERR LOGJAM
1169	001542	067377	DT24	;SERRPC,\$REG1,\$REGO
1170	001544	067670	0	
1171	001546	000000		
1172				
1173				;ITEM 41
1174			EM41	;INSTRUCTION NOT ABORTED IN CACHE ABORT MODE.
1175	001550	066044	DH23	;PC
1176	001552	067372	DT23	;SERRPC
1177	001554	067864	0	
1178	001556	000000		
1179				
1180				;ITEM 42

1181			EM42	;LO BYTE & TAG PARITY BITS NOT SET IN LOG SERVICE
1182	001560	066631	DH42	;PC LOGSERVCE
1183	001562	067567	DT25	;SERRPC,\$REGO,0
1184	001564	067700	0	
1185	001566	000000		
1186				
1187				;ITEM 43
1188			EM43	;LO BYTE & TAG PARITY BITS NOT SET IN MEM ERR REGISTER
1189	001570	066717	DH32	;PC MEMERR
1190	001572	067553	DT25	;SERRPC,\$REGO
1191	001574	067700	0	
1192	001576	000000		
1193				
1194				;ITEM 44
1195			EMEIS1	;EIS SET COND CODES WRONG
1196	001600	067064	DHEIS1	;PSW REG-WAS-REG+1 REG-S/B-REG+1 PC TEST (IR)
1197	001602	067421	DTEIS1	;SREGAD \$REG2 \$REG3 \$REG1 \$REG4 \$SERRPC \$REGO \$TMP0
1198	001604	067734	0	
1199	001606	000000		
1200				
1201				;ITEM 45
1202			EMEIS2	;EIS GAVE WRONG RESULT
1203	001610	067115	DHEIS1	;PSW REG-WAS-REG+1 REG-S/B-REG+1 PC TEST (IR)
1204	001612	067231	DTEIS1	;SREGAD \$REG2 \$REG3 \$REG1 \$REG4 \$SERRPC \$REGO \$TMP0
1205	001614	067734	0	
1206	001616	000000		
1207				
1208				;ITEM 46
1209			EM46	;AUTO-INCREMENT (DECREMENT) DID NOT OCCUR
1210	001620	067143	DH46	;PC (IR) TEST
1211	001622	067762	DT46	;SERRPC \$TMP0 \$REGO
1212	001624	067746	0	
1213	001626	000000		
1214				
1215		076600		MED = 076600
1216		140000		UM = 140000
1217		177770		UBREAK = 177770
1218		177744		MEMERR = 177744
1219		177766		CPUERR = 177766
1220		177746		CCR = 177746
1221		000100		WWP = BIT6
1222		000000		DPRP = BIT0
1223		000200		PABORT = BIT7
1224		000100		LO = BIT6
1225		000200		HI = BIT7
1226		000040		TAG = BITS
1227				.EQUIV SP,KSP
1228				
1229				
1230				
1231				;*
1232				MED OPERATION CODE DEFINITIONS
1233		000226		HCNSSW=226
1234		000022		RDNHAMI=22
1235		000022		RRWHAMI=222
1236		000144		RDFLAG=144

1237 000344
1238 000100
1239 000300
1240 000101
1241 000301
1242 000102
1243 000302
1244 000103
1245 000303
1246 000104
1247 000304
1248 000105
1249 000305
1250 000106
1251 000306
1252 000107
1253 000307
1254 000071
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265

WRFLAG=344
RDLJAM=100
RDLJAM=300
RDLSERVICE=101
RDLSERVICE=301
RDLBBA=102
RDLBBA=302
RDLCUA=103
RDLCUA=303
RDLFGINT=104
RDLFGINT=304
RDLWHAMI=105
RDLWHAMI=305
RDLDATA=106
RDLDATA=306
RDLTAG=107
RDLTAG=307
SWB01=71

;MICRO ADDR. IN SWAB INST.

;ADDRESS ASSIGNMENTS FOR DL11 CONSOLE TERMINAL INTERFACE

RCSR=177560 ;RCVR. CONTROL / STATUS REG. ADDRESS
RDBR = 177562 ;RECEIVER DATA BUFFER REG. ADDR.
YCSR = 177564 ;TRANSMITTER CONTROL / STATUS REG. ADDR
XDBR = 177566 ;TRANSMIT DATA BUFFER REG. ADDR.
LKCSR= 177546 ;LINE CLOCK ADDRESS

1266
1267
1268
1269
1270
1271
1272
1273
1274 001630
1275 001630 000401
1276
1277 001632 000000
1278
1279
1280
1281
1282
1283 001634 000402
1284
1285 001636 000403
1286
1287 001640 000000
1288
1289 001642 000775
1290
1291 001644 000000
1292
1293
1294
1295
1296
1297 001646 100403
1298 001650 001402
1299 001652 102401
1300 001654 103002
1301
1302 001656 000000
1303
1304 001660 000772
1305
1306
1307
1308
1309
1310 001662 000277
1311
1312 001664 100003
1313 001666 001002
1314 001670 102001
1315 001672 103402
1316
1317 001674 000000
1318
1319 001676 000771
1320
1321

```
                ;//////////////////////  
                ;"BCPT" TESTS  
                ;//////////////////////  
; *****  
; .SRTTL BT001 "BR" TEST - POSITIVE OFFSET  
; *****  
START:  
BT001: BR BT002 ;TEST THE BR FORWARD  
E001: HALT ;BR FAILED TO LOAD PC PROPERLY  
; *****  
; .SRTTL BT002 "BR" TEST - NEGATIVE OFFSET  
; *****  
BT002: BR I002 ;GO TO TEST INSTRUCTION  
A002: BR BT003 ;GO TO NEXT TEST  
EX002: HALT ;JUST IN CASE  
I002: BR A002 ;TEST THE BR - NEG. OFFSET  
E2002: HALT ;BR FAILED WITH NEG. OFFSET  
; *****  
; .SRTTL BT003 "BASIC COND. BR" TEST - FLAGS CLEARED  
; *****  
BT003: BMI E003 ;BR IF "N" SET  
 BEQ E004 ;BR IF "Z" SET  
 RVS E003 ;BR IF "V" SET  
 RCC FT004 ;BR IF "C" CLEAR  
E003: HALT ;ERROR - ONE OF THE ABOVE BR'S FAILED  
 ;OR THE FLAGS FAILED TO CLEAR ON "START"  
 BR FT003 ;LOCK ON HARD ERROR  
; *****  
; .SRTTL BT004 "SCC AND COND. BR'S" TEST - FLAGS SET  
; *****  
BT004: SCC ;MAKE N:C=1111  
I004: PPL E004 ;BR IF "N" FAILED TO SET  
 BNE E004 ;BR IF "Z" FAILED TO SET  
 BVC E004 ;BR IF "V" FAILED TO SET  
 BCS BT005 ;BR IF "C" SET OK  
E004: HALT ;ERROR - ONE OF THE ABOVE BR'S FAILED  
 ;OR THE SCC FAILED TO SET ALL THE FLAGS  
 BR BT004 ;LOCK ON HARD ERROR  
; *****
```

1322
1323
1324

; .SBTTL BT005 "CCC AND COND. BR'S" TEST - FLAGS CLEARED
; *****

1325 001700 000257
1326
1327 001702 100403
1328 001704 001402
1329 001706 102401
1330 001710 103002
1331
1332 001712 000000
1333
1334 001714 000771
1335
1336
1337
1338
1339
1340 001716 000257
1341
1342 001720 005000
1343
1344 001722 001402
1345
1346 001724 000000
1347 001726 000773
1348
1349
1350
1351
1352
1353 001730 005000
1354 001732 000257
1355
1356 001734 005700
1357
1358 001736 001402
1359
1360 001740 000000
1361
1362 001742 000772
1363
1364
1365
1366
1367
1368 001744 005000
1369 001746 000257
1370
1371 001750 005100
1372
1373 001752 100001
1374 001754 103402
1375
1376 001756 000000
1377 001760 000771
1378
1379
1380

BT005: CCC ;MAKE N:C=0000
I005: BMI E005 ;BR IF "N" STILL SET
BEQ E005 ;BR IF "Z" STILL SET
BVS E005 ;BR IF "V" STILL SET
BCC BT006 ;BR IF "C" GOT CLEARED
E005: HALT ;ERROR - ONE OF THE ABOVE BR'S FAILED
BR BT005 ;OR THE CCC FAILED TO CLEAR ALL FLAGS
;LOCK ON HARD ERROR
; *****
; .SBTTL BT006 "CLR %R" TEST - SETS THE "Z" BIT
; *****
BT006: CCC ;MAKE N:C=0000
I006: CLR R0 ;TEST THE CLR - IT SHOULD SET "Z"
BEQ BT007 ;BR IF CLR SET "Z"
E006: HALT ;ERROR - CLR FAILED TO SET "Z"
BR BT006 ;LOCK ON HARD ERROR
; *****
; .SBTTL BT007 "TST %R" TEST - USING THE CLR
; *****
BT007: CLR R0 ;MAKE [R0] = 000000
CCC ;MAKE N:C=0000
I007: TST R0 ;TEST THE TST - IT SHOULD SET "Z"
BEQ BT010 ;BR IF "Z" SET OK
E007: HALT ;ERROR - CLR FAILED TO LOAD R0 WITH
BR BT007 ;ALL ZEROES OR TST FAILED
;LOCK ON HARD ERROR
; *****
; .SBTTL BT010 "COM %R" TEST - SHOULD SET "N" AND "C"
; *****
BT010: CLR R0 ;MAKE [R0] = 000000
CCC ;MAKE N:C=0000
I010: COM R0 ;TEST THE COM - [R0] S/B = 177777
BPL E010 ;BR IF "N" FAILED TO SET
BCS BT011 ;BR IF "C" SET OK
E010: HALT ;ERROR - COM FAILED
BR BT010 ;LOCK ON HARD ERROR
; *****
; .SBTTL BT011 "COM %R AND ADC %R" TEST

1381
1382
1383 001762 005000
1384 001764 000257
1385
1386 001766 005100
1387 001770 005500
1388
1389 001772 001001
1390 001774 103402
1391
1392 001776 000000
1393 002000 000770
1394
1395
1396
1397
1398
1399 002002 005000
1400 002004 000257
1401
1402 002006 012700 177777
1403
1404 002012 005100
1405 002014 001402
1406
1407 002016 000000
1408 002020 000770
1409
1410
1411
1412
1413
1414 002022 005000
1415 002024 005100
1416 002026 000257
1417
1418 002030 012700 000000
1419
1420 002034 005100
1421 002036 005500
1422 002040 001402
1423
1424 002042 000000
1425 002044 000766
1426
1427
1428
1429
1430
1431 002046 012706 001000
1432 002052 012700 177776
1433 002056 000277
1434
1435 002060 005010
1436

```
; *****  
BT011: CLR R0 ;MAKE [R0] = 000000  
CCC ;MAKE N:C=0000  
I011: COM R0 ;TEST THE COM - [R0] S/B = 177777  
ADC R0 ;TEST THE ADC - [R0] S/R = 000000  
RNE E011 ;BR IF "Z" DID NOT SET  
BCS BT012 ;BR IF "C" SET OK  
E011: HALT ;ERROR - COM OR ADC FAILED  
BR BT011 ;LOCK ON HARD ERROR  
; *****  
; .SBTTL BT012 "MOV #N,R" TEST WITH N=177777,[R0]=000000  
; *****  
BT012: CLP R0 ;MAKE [R0] = 000000  
CCC ;MAKE N:C=0000  
I012: MOV #-1,R0 ;TEST THE MOV - [R0] S/B = 177777  
COM R0 ;MAKE [R0] = 000000  
BEQ BT013 ;BR IF "Z" SET  
E012: HALT ;ERROR - MOV FAILED TO LOAD R0 WITH ALL 1'S  
BR BT012 ;LOCK ON HARD ERROR  
; *****  
; .SBTTL BT013 "MOV #N,R" TEST WITH N=000000,[R0]=177777  
; *****  
BT013: CLR R0 ;MAKE [R0] = 000000  
COM R0 ;MAKE [R0] = 177777  
CCC ;SCOPE SYNC  
I013: MOV #0,R0 ;TEST THE MOV - [R0] S/R = 000000  
COM R0 ;MAKE [R0] = 177777, SET "C"  
ADC R0 ;MAKE [R0] = 000000  
BEQ BT014 ;BR IF "Z" GCT SET  
E013: HALT ;ERROR - MOV FAILED TO CLEAR R0  
BR BT013 ;LOCK ON HARD ERROR  
; *****  
; .SBTTL BT014 "CLR (R)" TEST - [R] = 177776  
; *****  
BT014: MOV #STACK,SP ;SET UP STACK POINTER  
MOV #PSW,R0 ;R0 POINTS TO PSW  
SCC ;MAKE [PSW] = 017  
I014: CLP (R0) ;TEST THE CLP - IT SHOULD CLEAR PSW
```

1437 002062 001002
1438
1439 002064 000000
1440 002066 000767
1441
1442
1443
1444
1445
1446 002070 012700 177776
1447 002074 000277
1448
1449 002076 005020
1450 002100 001002
1451
1452
1453 002102 000000
1454 002104 000771
1455
1456 002106 005700
1457
1458 002110 001402
1459
1460 002112 000000
1461 002114 000765
1462
1463
1464
1465
1466 002116 012700 177776
1467 002122 000257
1468
1469
1470 002124 005110
1471
1472 002126 100003
1473 002130 001002
1474 002132 102001
1475 002134 103403
1476
1477 002136 005010
1478 002140 000000
1479 002142 000765
1480
1481
1482
1483
1484
1485 002144 012700 177776
1486 002150 005010
1487 002152 000257
1488
1489 002154 005120
1490
1491 002156 100003
1492 002160 001002

```
RNE BT015 ;BR IF CLR MADE "Z" = 0 - IT SHOULD  
E014: HALT ;ERROR- CLR FAILED TO CLEAR PSW  
BR BT014 ;LOCK ON HARD ERROR  
; *****  
; .SBTTL BT015 "CLR (R)+" TEST - [R] = 177776  
; *****  
BT015: MOV #PSW,R0 ;R0 POINTS TO PSW  
SCC ;MAKE [PSW] = 017  
I015: CLR (R0)+ ;TEST THE CLR - IT SHOULD CLEAR PSW  
RNE A015 ;BR IF CLR MADE "Z" = 0 - IT SHOULD  
E1015A: HALT ;ERROR- CLR FAILED TO CLEAR PSW  
BR BT015 ;LOCK ON HARD ERROR  
A015: TST R0 ;AUTO INC SHOULD ZERO R0  
REQ BT016 ;BR IF IT DID  
E2015: HALT ;ERROR - AUTOINC. FAILED  
BR BT015 ;LOCK ON HARD ERROR  
; *****  
; .SBTTL BT016 "COM (R)" TEST - [R] = 177776  
; *****  
BT016: MOV #PSW,R0 ;R0 POINTS TO PSW  
CCC ;MAKE [PSW] = 000  
I016: COM (R0) ;TEST THE COM - [PSW] S/R = 357  
BPL E016 ;N:C=1111 ?  
RNE E016  
BCS BT017  
E016: CLR (R0) ;GO TO KERNEL MODE  
HALT ;ERROR - COM FAILED TO MAKE [PSW] = 357  
BR BT016 ;LOCK ON HARD ERROR  
; *****  
; .SBTTL BT017 "COM (R0)+" TEST - [R0] = 177776  
; *****  
BT017: MOV #PSW,R0 ;R0 POINTS TO PSW  
CLP (R0) ;MAKE [PSW] = 000  
CCC ;SCOPE SYNC  
I017: COM (R0)+ ;TEST THE COM - [PSW] S/R = 357  
BPL EA017 ;N:C = 1111 ?  
BNR EA017
```

1493 002162 102001
 1494 002164 103405
 1495
 1496 002166 012701 177776
 1497 002172 005011
 1498 002174 000000
 1499 002176 000762
 1500
 1501 002200 005100
 1502 002202 005500
 1503 002204 001405
 1504
 1505 002206 012701 177776
 1506 002212 005011
 1507 002214 000000
 1508 002216 000752
 1509
 1510
 1511
 1512
 1513
 1514 002220 012700 177776
 1515 002224 005010
 1516 002226 005000
 1517 002230 005001
 1518 002232 005101
 1519 002234 000257
 1520
 1521 002236 010100
 1522
 1523 002240 100402
 1524
 1525 002242 000000
 1526 002244 000765
 1527
 1528 002246 005100
 1529 002250 001402
 1530
 1531 002252 000000
 1532 002254 000761
 1533
 1534
 1535
 1536
 1537
 1538 002256 005000
 1539 002260 005100
 1540 002262 005001
 1541 002264 000257
 1542
 1543 002266 010100
 1544
 1545 002270 001402
 1546
 1547 002272 000000
 1548 002274 000770

BVC EA017
 BCS A017
 EA017: MOV #PSW,R1
 CLR (R1)
 HALT ;COM FAILED TO SET ALL FLAGS
 BR BT017 ;LOCK ON HARD ERROR
 A017: COM R0 ;SHOULD MAKE [R0] = 177777
 ADC R0 ;SHOULD MAKE [R0] = 000000
 PEQ RT020
 E2017: MOV #PSW,R1
 CLR (R1)
 HALT ;ERROR - COM FAILED TO AUTO INC. R0
 BR BT017 ;LOCK ON HARD ERROR
 ; *****
 ; SBTTL BT020 "MOV RA,RB" TEST - WITH [RA]=177777,[RB]=000000
 ; *****
 BT020: MOV #PSW,R0
 CLR (R0)
 CLR R0 ;MAKE [R0]=000000
 CLR R1 ;MAKE [R1]=000000
 CLM R1 ;MAKE [R1]=0207777
 CCC ;SCOPE SYNC
 I020: MOV R1,R0 ;TEST THE MOV
 RMI A020 ;BR IF "N" GCT SET
 EA020: HALT ;ERROR-MOV FAILED TO SET "N"
 BR BT020 ;LOCK ON HARD ERROR
 A020: COM R0 ;[R0] SHOULD GO TO 000000
 BEQ BT021 ;BR IF IT DID
 E2020: HALT ;ERROR-MOV FAILED TO LOAD R0 WITH 1'S
 BR RT020 ;LOCK ON HARD ERROR
 ; *****
 ; SBTTL BT021 "MOV RA,RB" TEST WITH [RA]=000000,[RB]=177777
 ; *****
 BT021: CLR R0 ;MAKE [R0]=000000
 COM R0 ;MAKE [R0]=177777
 CLR R1 ;MAKE [R1]=000000
 CCC ;SCOPE SYNC
 I021: MOV R1,R0 ;TEST THE MOV
 BEQ A021 ;BR IF "Z" GOT SET
 EA021: HALT ;MOV FAILED TO SET "Z"
 BR RT021 ;LOCK ON HARD ERROR

1549
 1550 002276 005100
 1551 002300 005500
 1552 002302 001402
 1553
 1554 002304 000000
 1555 002306 000763
 1556
 1557
 1558
 1559
 1560
 1561 002310 000257
 1562
 1563 002312 012737 000017 177776
 1564
 1565 002320 100003
 1566 002322 001002
 1567 002324 102001
 1568 002326 103402
 1569
 1570 002330 000000
 1571 002332 000766
 1572
 1573
 1574
 1575
 1576
 1577 002334 012700 177776
 1578 002340 012701 000017
 1579 002344 000257
 1580
 1581 002346 010120
 1582
 1583 002350 100003
 1584 002352 001002
 1585 002354 102001
 1586 002356 103402
 1587
 1588 002360 000000
 1589 002362 000764
 1590
 1591 002364 005700
 1592 002366 001402
 1593
 1594 002370 000000
 1595 002372 000760
 1596
 1597
 1598
 1599
 1600
 1601 002374 012700 177776
 1602 002400 005010
 1603 002402 000273
 1604

A021: COM R0 ;SHOULD MAKE [R0]=177777 AND SET "C"
 ADC R0 ;SHOULD MAKE [R0]=000000
 BEQ RT022 ;BR IF "Z" SET
 E2021: HALT ;MOV FAILED TO ZERO R0
 BR RT021 ;LOCK ON HARD ERROR
 ; *****
 ; SBTTL BT022 "MOV #N,@#A" TEST WITH N=17,A=177776
 ; *****
 BT022: CCC ;MAKE [PSW]=000
 I022: MOV #17,@#PSW ;TEST THE MOV
 RPL E022 ;N:C=1111
 RNE E022
 RVC E022
 BCS RT023
 E022: HALT ;MOV FAILED TO LOAD PSW
 BR BT022 ;LOCK ON HARD ERROR
 ; *****
 ; SBTTL BT023 "MOV RA,(RB)+ TEST WITH [RA]=17,[RB]=177776
 ; *****
 BT023: MOV #PSW,R0 ;R0 POINTS TO PSW
 MOV #17,R1 ;[SOURCE]=017
 CCC ;SCOPE SYNC - MAKE <N:C> = 0000
 I023: MOV R1,(R0)+ ;TEST THE MOV
 RPL EA023 ;N:C = 1111 ?
 RNE EA023
 RVC EA023
 BCS A023
 EA023: HALT ;MOV FAILED TO LOAD PSW
 BR RT023 ;LOCK ON HARD ERROR
 A023: TST R0 ;DID AUTO INC MAKE R0 GO TO 0?
 PEQ RT024 ;BR IF IT DID
 E2023: HALT ;MOV FAILED TO AUTO INC. R0
 BR RT023 ;LOCK ON HARD ERROR
 ; *****
 ; SBTTL BT024 "CMP #N,@#A" TEST WITH N=(A)
 ; *****
 BT024: MOV #PSW,R0 ;R0 POINTS TO PSW
 CLP (R0) ;MAKE [PSW]=000
 273 ;MAKE N:C=1011

```

1605 002404 022737 000013 177776 I024:  CMP      #13,@#PSW      ;TEST THE CMP
1606                                BEQ      BT025          ;BR IF "Z" GOT SET
1607 002412 001402                                E024:  HALT      BT024          ;CMP FAILED TO SET "Z"
1608 002414 000000                                BR      BT024          ;LOCK ON HARD ERROR
1609 002416 000766                                ; *****
1610                                ;.SBTTL BT025 "CMP #N,@#A" WITH N > (A)
1611                                ; *****
1612 002420 000257 BT025:  CCC          ;MAKE [PSW]=000
1613 002422 022737 000017 177776 I025:  CMP      #17,@#PSW      ;TEST THE CMP
1614                                BEQ      E025          ;BR IF "Z" GOT SET
1615 002430 001401                                PR      FT026          ;GO TO NEXT TEST
1616 002432 000402                                E025:  HALT      BT025          ;CMP FAILED TO CLEAR "Z"
1617 002434 000000                                BR      BT025          ;LOCK ON HARD ERROR
1618 002436 000770                                ; *****
1619                                ;.SBTTL BT026 "CMP #N,@#A" WITH N < (A)
1620                                ; *****
1621 002440 000277 RT026:  SCC          ;MAKE [PSW]=017
1622 002442 022737 000000 177776 I026:  CMP      #0,@#PSW      ;TEST THE CMP
1623 002450 001401                                BEQ      E026          ;BR IF "Z" GOT SET
1624 002452 000402                                BR      BT027          ;GO TO NEXT TEST
1625 002454 000000                                E026:  HALT      BT026          ;CMP FAILED TO CLEAR "Z"
1626 002456 000770                                BR      BT026          ;LOCK ON HARD ERROR
1627                                ; *****
1628                                ;.SBTTL BT027 "CMP R,#N" TEST WITH [R]=N
1629                                ; *****
1630 002460 012700 177777 BT027:  MOV      R-1,R0      ;MAKE [R0]=177777
1631 002462 000257                                CCC          ;N:C=0000
1632 002466 020027 177777 I027:  CMP      R0,#-1        ;TEST THE CMP
1633 002472 001402                                BEQ      RT030          ;BR IF CMP SET "Z"
1634 002474 000000                                E027:  HALT      BT027          ;CMP FAILED
1635 002476 000770                                BR      BT027          ;LOCK ON HARD ERROR
1636                                ; *****
1637                                ;.SBTTL BT030 "CMP R,#N" TEST WITH [R] > N
1638                                ; *****
1639 002500 012700 000001 RT030:  MOV      #1,R0      ;MAKE [R0]=000001
1640 002504 000264                                SEZ          ;SET THE "Z" BIT
1641 002506 020027 177777 I030:  CMP      R0,#-1        ;TEST THE CMP

```

```

1661 002512 001002                                BNE      BT031          ;BR IF CMP CLEARED "Z"
1662 002514 000000                                E030:  HALT      BT030          ;CMP FAILED
1663 002516 000770                                BR      BT030          ;LOCK ON HARD ERROR
1664                                ; *****
1665                                ;.SBTTL BT031 "CMP R,#N" TEST WITH [R] < N
1666                                ; *****
1667 002520 012700 000001 RT031:  MOV      #1,R0      ;MAKE [R0] = 000001
1668 002524 000264                                SEZ          ;SET THE "Z" BIT
1669 002526 020027 000017 I031:  CMP      R0,#17        ;TEST THE CMP
1670 002532 001002                                BNE      BT032          ;BR IF CMP CLEARED "Z"
1671 002534 000000                                E031:  HALT      BT031          ;CMP FAILED TO SET "Z"
1672 002536 000770                                BR      BT031          ;LOCK ON HARD ERROR
1673                                ; *****
1674                                ;.SBTTL BT032 "CMP (RA)+,RB" TEST WITH [SOURCE]=[RB]
1675                                ; *****
1676 002540 012700 177776 BT032:  MOV      #PSW,R0      ;R0 POINTS TO PSW
1677 002544 012737 000340 177776 MOV      #340,@#PSW      ;MAKE [PSW]=340
1678 002552 012701 000340 MOV      #340,R1        ;MAKE [DEST]=340
1679 002556 000257                                CCC          ;N:C=0000
1680 002560 022001 I032:  CMP      (R0)+,R1      ;TEST THE CMP
1681 002562 001402                                BEQ      A032          ;BR IF "Z" GOT SET
1682 002564 000000                                EA032: HALT      BT032          ;CMP FAILED TO ACCESS PSW
1683 002566 000764                                BR      BT032          ;LOCK ON HARD ERROR
1684 002570 005700 A032:  TST      R0          ;"Z" SHOULD SET
1685 002572 001402                                BEQ      BT033          ;BR IF "Z" SET
1686 002574 000000                                E2032: HALT      BT032          ;CMP FAILED TO AUTO INC. R0
1687 002576 000760                                BR      BT032          ;LOCK ON HARD ERROR
1688                                ; *****
1689                                ;.SBTTL BT033 "CMP (RA)+,RB" TEST WITH [SOURCE]>[RB]
1690                                ; *****
1691 002600 012700 177776 BT033:  MOV      #PSW,R0      ;R0 POINTS TO PSW
1692 002604 012737 000340 177776 MOV      #340,@#PSW      ;MAKE [PSW]=340
1693 002612 012701 000330 MOV      #330,R1        ;MAKE [DEST]=330
1694 002616 000264                                SEZ          ;SET THE "Z" BIT
1695 002620 022001 I033:  CMP      (R0)+,R1      ;TEST THE CMP
1696 002622 001002                                BNE      A033          ;BR IF "Z" GOT CLEARED
1697 002624 000000                                EA033: HALT      BT033          ;CMP FAILED TO ACCESS PSW
1698 002626 000764                                BR      BT033          ;LOCK ON HARD ERROR

```

```

1717
1718 002630 005700 A033: TST R0 ;"Z" SHOULD SET
1719 002632 001402 BEQ BT034 ;BR IF "Z" SET
1720
1721 002634 000000 E2033: HALT ;CMP FAILED TO AUTO INC. R0
1722 002636 000760 BR BT033 ;LOCK ON HARD ERROR
; *****
; .SRTTL BT034 "CMP (RA)+,RB" TEST WITH [SOURCE]<[RB]
; *****
1723
1724
1725
1726 002640 012700 177776 BT034: MOV #PSW,R0 ;R0 POINTS TO PSW
1727 002644 012737 000330 MOV #330,#PSW ;MAKE [PSW]=330
1728 002652 012701 000340 MOV #340,R1 ;MAKE [DEST]=340
1729 002656 000264 SEZ ;SET THE "Z" BIT
1730
1731
1732 002660 022001 I034: CMP (R0)+,R1 ;TEST THE CMP
1733
1734 002662 001002 BNE A034 ;BR IF "Z" GOT CLEARED
1735
1736 002664 000000 EA034: HALT ;CMP FAILED TO ACCESS PSW
1737 002666 000764 BR BT034 ;LOCK ON HARD ERROR
1738
1739 002670 005700 A034: TST R0 ;"Z" SHOULD SET
1740 002672 001402 BEQ BT035 ;BR IF "Z" SET
1741
1742 002674 000000 E2034: HALT ;CMP FAILED TO AUTO INC. R0
1743 002676 000760 BR BT034 ;LOCK ON HARD ERROR
; *****
; .SRTTL BT035 "CMP RA, RB" TEST WITH [RA] = [RB]
; *****
1744
1745
1746
1747
1748 002700 012700 125252 BT035: MOV #125252,R0 ;MAKE [R0] = 125252
1749 002704 010001 MOV R0,R1 ;MAKE [R1] = 125252
1750 002706 000257 CCC ;SCOPE SYNC
1751
1752 002710 020100 I035: CMP R1,R0 ;TEST THE CMP
1753
1754 002712 001402 BEQ BT036 ;BR IF "Z" GOT SET
1755
1756 002714 000000 E035: HALT ;ERROR - CMP FAILED TO SET "Z"
1757 002716 000770 BR BT035 ;LOCK ON HARD ERROR
; *****
; .SRTTL BT036 "CMP RA, RB" TEST WITH [RA] < [RB]
; *****
1758
1759
1760
1761 002720 012700 025252 BT036: MOV #25252,R0 ;MAKE [R0] = 25252
1762 002724 005001 CLR R1 ;MAKE [R1] = 000000
1763 002726 000264 SEZ ;SCOPE SYNC - SET "Z"
1764
1765
1766 002730 020100 I036: CMP R1,R0 ;TEST THE CMP
1767
1768 002732 001002 BNE BT037 ;BR IF "Z" GOT CLEARED
1769
1770 002734 000000 E036: HALT ;ERROR - CMP FAILED TO SET "Z"
1771 002736 000770 BR BT036 ;LOCK ON HARD ERROR
; *****
1772

```

```

1773
1774
1775
1776 002740 005000 BT037: CLR R0 ;MAKE [R0] = 000000
1777 002742 012701 000017 MOV #17,R1 ;MAKE [R1] = 000017
1778 002746 000264 SEZ ;SCOPE SYNC - SET "Z"
1779
1780 002750 020100 I037: CMP R1,R0 ;TEST THE CMP
1781
1782 002752 001002 BNE BT040 ;BR IF "Z" GOT CLEARED
1783
1784 002754 000000 E037: HALT ;ERROR - CMP FAILED TO SET "Z"
1785 002756 000770 BR BT037 ;LOCK ON HARD ERROR
; *****
; .SRTTL BT040 "MOV (RA),RB" TEST WITH [SOURCE]=[RB]=17
; *****
1786
1787
1788
1789
1790 002760 012700 177776 BT040: MOV #PSW,R0 ;R0 POINTS TO PSW
1791 002764 005010 CLP (R0) ;MAKE [PSW]=000
1792 002766 005001 CLR R1 ;MAKE [R1]=000000
1793 002770 000277 SCC ;MAKE N:C=1111
1794
1795
1796 002772 011001 I040: MOV (R0)+,R1 ;TEST THE MOV
1797
1798 002774 020127 000017 CMP R1,#17 ;DID R1 GET LOADED WITH 000017 ?
1799 003000 001402 BEQ BT041 ;BR IF YES
1800
1801 003002 000000 E040: HALT ;MOV FAILED TO LOAD R1
1802 003004 000765 BR BT040 ;LOCK ON HARD ERROR
; *****
; .SRTTL BT041 "MOV (RA)+,RB" TEST WITH [SOURCE]=[RB]=17
; *****
1803
1804
1805
1806 003006 012700 177776 BT041: MOV #PSW,R0 ;R0 POINTS TO PSW
1807 003012 005010 CLP (R0) ;MAKE [PSW]=000
1808 003014 005001 CLR R1 ;MAKE [R1]=000000
1809 003016 000277 SCC ;MAKE N:C=1111
1810
1811
1812 003020 012001 I041: MOV (R0)+,R1 ;TEST THE MOV
1813
1814 003022 020127 000017 CMP R1,#17 ;DID R1 GET LOADED WITH 000017 ?
1815 003026 001402 BEQ A041 ;BR IF YES
1816
1817 003030 000000 EA041: HALT ;MOV FAILED TO LOAD R1
1818 003032 000765 BR BT041 ;LOCK ON HARD ERROR
1819
1820 003034 005700 A041: TST R0 ;"Z" SHOULD SET
1821 003036 001402 BEQ BT042 ;BR IF "Z" GOT SET
1822
1823 003040 000000 E2041: HALT ;MOV FAILED TO AUTO INC. R0
1824 003042 000761 BR BT041 ;LOCK ON HARD ERROR
; *****
; .SRTTL BT042 "XOR RA, RB" TEST WITH [RA] = [RB] = 000000
; *****
1825
1826
1827
1828

```


1829
1830 003044 005000
1831 003046 005001
1832 003050 000257
1833
1834 003052 074100
1835
1836 003054 005700
1837 003056 001402
1838
1839 003060 000000
1840 003062 000770
1841
1842
1843
1844
1845
1846 003064 005000
1847 003066 005100
1848 003070 010001
1849 003072 000257
1850
1851 003074 074100
1852
1853 003076 005700
1854 003100 001402
1855
1856 003102 000000
1857 003104 000767
1858
1859
1860
1861
1862
1863 003106 012701 125252
1864 003112 012700 052525
1865 003116 000257
1866
1867 003120 074100
1868
1869 003122 020027 177777
1870 003126 001402
1871
1872 003130 000000
1873 003132 000400
1874
1875
1876
1877
1878 003134 012700 125252
1879 003140 012701 052525
1880 003144 000257
1881
1882 003146 074100
1883
1884 003150 020027 177777

```

BT042: CLR R0 ;MAKE [R0] = 000000
        CLR R1 ;MAKE [R1] = 000000
        CCC ;SCOPE SYNC

I042: XOR R1,R0 ;TEST THE XOR

TST R0 ;RESULT = 000000 ?
BR BT043 ;BR IF YES

E042: HALT BT042 ;XOR FAILED
BR

; *****
; SBTTL BT043 "XOR RA,RB" TEST WITH [RA] = [RB] = 177777
; *****

BT043: CLR R0 ;MAKE [R0] = 177777
        COM R0 ;MAKE [R1] = 177777
        MOV R0,R1 ;SCOPE SYNC
        CCC

I043: XOR R1,R0 ;TEST THE XOR

TST R0 ;RESULT = 000000 ?
BEQ BT044 ;BR IF YES

E043: HALT BT043 ;XOR FAILED
BR ;LOCK ON HARD ERROR

; *****
; SBTTL BT044 "XOR RA,RB" TEST WITH [RB]=052525,[RA]=125252
; *****

BT044: MOV #125252,R1 ;MAKE [R1]=125252
        MOV #052525,R0 ;MAKE [R0]=052525
        CCC ;SCOPE SYNC

I044: XOR R1,R0 ;TEST THE XOR

CMP R0,#-1 ;RESULT = 177777 ?
BEQ BT045 ;BR IF YES

E044: HALT BT045 ;XOR FAILED
BR ;LOCK ON HARD ERROR

; *****
; SBTTL BT045 "XOR RA,RB" TEST WITH [RA]=052525,[RB]=125252
; *****

BT045: MOV #125252,R0 ;MAKE [R0]=125252
        MOV #052525,R1 ;MAKE [R1]=052525
        CCC ;SCOPE SYNC

I045: XOR R1,R0 ;TEST THE XOR

CMP R0,#-1 ;RESULT = 177777 ?

```

1885 003154 001402
1886
1887 003156 000000
1888 003160 000765
1889
1890
1891
1892
1893
1894 003162 012700 125252
1895 003166 010001
1896 003170 005101
1897 003172 010102
1898 003174 005102
1899 003176 010203
1900 003200 005103
1901 003202 010304
1902 003204 005104
1903 003206 010405
1904 003210 005105
1905
1906 003212 074100
1907 003214 074200
1908 003216 074300
1909 003220 074400
1910 003222 074500
1911 003224 005100
1912
1913 003226 001402
1914
1915 003230 000000
1916 003232 000753
1917
1918 003234 020627 001000
1919 003240 001402
1920
1921 003242 000000
1922 003244 000746

```

BEQ BT046 ;BR IF YES

E045: HALT BT045 ;XOR FAILED
BR ;LOCK ON HARD ERROR

; *****
; SBTTL BT046 GPR ADDRESS INTERACTION TEST
; *****

BT046: MOV #125252,R0 ;[R0] = 125252
        MOV R0,R1 ;[R1] = 052525
        COM R1 ;[R2] = 125252
        MOV R1,R2 ;[R3] = 052525
        COM R2 ;[R4] = 125252
        MOV R2,R3 ;[R5] = 052525
        COM R3
        MOV R3,R4
        MOV R4,R5
        COM R5

I046: XOR R1,R0 ;[R0] S/B = 177777
        XOR R2,R0 ;[R0] S/B = 125252
        XOR R3,R0 ;[R0] S/B = 177777
        XOR R4,R0 ;[R0] S/B = 125252
        XOR R5,R0 ;[R0] S/B = 177777
        COM R0 ;[R0] S/B = 000000

BEQ A046 ;BR IF [R0] WAS 000000

EA046: HALT BT046 ;GPR ADDRESSING PROBLEM
BR ;LOCK ON HARD ERROR

A046: CMP SP,#STACK ;DID R6 GET DISTURBED
BEQ BASIC ;BR IF NOT

E2046: HALT BT046 ;R6 ADDRESS PROBLEM
BR ;LOCK ON HARD ERROR

```

```

1923
1924
1925
1926
1927 003246 005037 063254
1928 003252 005037 001012
1929 003256 005037 001126
1930 003262 012701 063236
1931 003266 005021
1932 003270 020174 063254
1933 003274 013374
1934 003278 013786 001000
1935 003302 012737 004030 177770
1936 003310 012737 177777 001074
1937
1938
1939
1940
1941
1942
1943 003316
1944 003316 012700 000000
1945 003322 000257
1946
1947 003324
1948 003324 001002
1949 003326 000000
1950 003330 000774
1951
1952
1953
1954
1955 003332
1956 003332 012700 000001
1957 003336 000264
1958
1959 003340 001001
1960
1961 003342 000402
1962
1963 003344 000000
1964 003346 000773
1965
1966
1967
1968
1969 003350
1970 003350 012700 000002
1971 003354 000264
1972
1973 003356 001402
1974 003356
1975
1976 003360 000000
1977 003362 000774
1978

```

```

;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
BASIC: CLR @#ONCE ;SIGNAL PROGRAM HEADER TO BE PRINTED
CLR @#SERVTL ;CLEAR ERROR COUNT FIRST TIME THROUGH
CLR @#PASS ;CLEAR PASS COUNT FIRST TIME THROUGH
INIT: MOV @#PRIFLG,R1 ;SET UP TO INIT. COUNTERS AND FLAGS
1$: CLR (R1)+ ;CLEAR ONE WORD
CMP R1,#ONCE ;CLEARED ALL FLAGS AND COUNTERS?
BNE 1$ ;BR IF NOT
MOV @#STACK,SP ;SET UP THE STACK POINTER
MOV @#4030,@#UBREAK ;SET SCOPE SYNC FOR COND CODE OPERATE
MOV @#-1,@#SREG5 ;FLAG CURRENT STACK POINTER TO BE TYPED
;IN FIRST ERROR CALL
;*****
;*TEST 0 BASIC "BNE" TEST WITH Z=0
;*****
TST0:
1$: MOV #0,R0 ;LOAD R0 WITH TEST NUMBER
CCC ;MAKE Z=0
2$: RNE TST1 ;TEST THE BNE - IT SHOULD BR
3$: HALT 1$ ;BNE FAILED TO LOAD PC
BR 1$ ;LOCK ON HARD ERROR
;*****
;*TEST 1 BASIC "BNE" TEST WITH Z=1
;*****
TST1:
1$: MOV #1,R0 ;LOAD R0 WITH TEST NUMBER
SEZ ;SET THE "Z" BIT
2$: BNE 3$ ;TEST THE BNE - IT SHOULD NOT BR
BR TST2 ;GO TO NEXT TEST
3$: HALT 1$ ;BNE BRANCHED WITH Z=1
BR 1$ ;LOCK ON HARD ERROR
;*****
;*TEST 2 BASIC "BEQ" TEST WITH Z=1
;*****
TST2:
1$: MOV #2,R0 ;LOAD R0 WITH TEST NUMBER
SEZ ;MAKE Z=1
2$: BEQ TST3 ;TEST THE BEQ - IT SHOULD BR
3$: HALT 1$ ;BEQ FAILED TO LOAD THE PC
BR 1$ ;LOCK ON HARD ERROR
;*****
;*****
;*TEST 3 BASIC "BEQ" TEST WITH Z=0
;*****
TST3:
1$: MOV #3,R0 ;LOAD R0 WITH TEST NUMBER
CCC ;MAKE Z=0
2$: BEQ 3$ ;TEST THE BEQ - IT SHOULD NOT BR
BR TST4 ;GO TO NEXT TEST
3$: HALT 1$ ;BEQ BRANCHED WITH Z=0
BR 1$ ;LOCK ON HARD ERROR
;*****
;*TEST 4 BASIC "BPL" TEST WITH N=1
;*****
TST4:
1$: MOV #4,R0 ;LOAD R0 WITH TEST NUMBER
CLR @#PSW ;CLEAR THE PSW
SEN ;MAKE N=1
2$: BPL 3$ ;TEST THE BPL - IT SHOULDNT BR
BR TST5 ;GO TO NEXT TEST
3$: HALT 1$ ;BPL BRANCHED WITH N=1
BR 1$ ;LOCK ON HARD ERROR
;*****
;*TEST 5 BASIC "BPL" TEST WITH N=0
;*****
TST5:
1$: MOV #5,R0 ;LOAD R0 WITH TEST NUMBER
CLR @#PSW ;CLEAR THE PSW
CCC ;SCOPE SYNC
2$: BPL TST6 ;TEST THE BPL - IT SHOULD BR
3$: HALT 1$ ;BPL FAILED TO LOAD THE PC
BR 1$ ;LOCK ON HARD ERROR
;*****
;*TEST 6 BASIC "MOV (RA),RB" TEST - (RA)=177776
;*****
TST6:
1$: MOV #6,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#PSW,R5 ;SOURCE ADDR = 177776
CLR (R5) ;MAKE [PSW]=00
CLR R3 ;DESTJ=00000
SCC ;MAKE [PSW]=017
2$: MOV (R5),R3 ;TEST THE MOV
3$: CMP R3,#17 ;CORRECT RESULT ?
BEQ TST7 ;BR IF YES

```

```

1979
1980
1981 003364
1982 003364 012700 000003
1983 003370 000257
1984
1985 003372 001401
1986
1987 003374 000402
1988
1989 003376 000000
1990 003400 000773
1991
1992
1993
1994
1995 003402
1996 003402 012700 000004
1997 003406 005037 177776
1998 003412 000270
1999
2000 003414 100001
2001
2002 003416 000402
2003
2004 003420 000000
2005 003422 000771
2006
2007
2008
2009
2010 003424
2011 003424 012700 000005
2012 003430 005037 177776
2013 003434 000257
2014
2015 003436
2016 003436 100002
2017
2018 003440 000000
2019 003442 000772
2020
2021
2022
2023
2024
2025 003444
2026 003450 012705 000006
2027 003452 005037 177776
2028 003454 005037
2029 003460 000277
2030
2031 003462 011503
2032
2033 003464 020327 000017
2034 003470 001402

```

```

;*****
;*TEST 4 BASIC "BPL" TEST WITH N=1
;*****
TST4:
1$: MOV #4,R0 ;LOAD R0 WITH TEST NUMBER
CLR @#PSW ;CLEAR THE PSW
SEN ;MAKE N=1
2$: BPL 3$ ;TEST THE BPL - IT SHOULDNT BR
BR TST5 ;GO TO NEXT TEST
3$: HALT 1$ ;BPL BRANCHED WITH N=1
BR 1$ ;LOCK ON HARD ERROR
;*****
;*TEST 5 BASIC "BPL" TEST WITH N=0
;*****
TST5:
1$: MOV #5,R0 ;LOAD R0 WITH TEST NUMBER
CLR @#PSW ;CLEAR THE PSW
CCC ;SCOPE SYNC
2$: BPL TST6 ;TEST THE BPL - IT SHOULD BR
3$: HALT 1$ ;BPL FAILED TO LOAD THE PC
BR 1$ ;LOCK ON HARD ERROR
;*****
;*TEST 6 BASIC "MOV (RA),RB" TEST - (RA)=177776
;*****
TST6:
1$: MOV #6,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#PSW,R5 ;SOURCE ADDR = 177776
CLR (R5) ;MAKE [PSW]=00
CLR R3 ;DESTJ=00000
SCC ;MAKE [PSW]=017
2$: MOV (R5),R3 ;TEST THE MOV
3$: CMP R3,#17 ;CORRECT RESULT ?
BEQ TST7 ;BR IF YES

```

```

2035
2036 003472 000000
2037 003474 000767
2038
2039
2040
2041 003476
2042 003476 012700 000007
2043 003502 012703 063312
2044 003502 012704 063312
2045 003512 012737 125252 063312
2046 003520 000257
2047
2048 003522 020412
2049
2050 003524 001402
2051
2052 003526 000000
2053 003530 000770
2054
2055
2056
2057 003532
2058 003532 012700 000010
2059 003536 012702 063312
2060 003542 012704 000001
2061 003546 005037 063312
2062 003552 000264
2063
2064 003554 020412
2065
2066 003556 001002
2067
2068 003560 000000
2069 003562 000771
2070
2071
2072
2073
2074 003564
2075 003564 012700 000011
2076 003570 012704 125252
2077 003574 010403
2078 003576 000257
2079
2080 003600 022703 125252
2081
2082 003604 001402
2083
2084 003606 000000
2085 003610 000771
2086
2087 003612 020403
2088 003614 001402
2089
2090 003616 000000

```

```

3S: HALT 1$ ;ERROR-MOV FAILED
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 7 BASIC "CMP RA,(RB)" TEST - (RA) = (DEST)
;*****
TST7:
MOV #7,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #1,R4 ;RESULT S / B = 125252
1S: MOV #125252,@#MBUFO ;MAKE (DEST) = 125252
CCC ;MAKE N:C=0000
2S: CMP R4,(R2) ;TEST THE CMP
REQ TST10 ;;BR IF "Z" GOT SET
3S: HALT 1$ ;ERROR - CMP FAILED TO SET "Z"
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 10 BASIC "CMP RA,(RB)" TEST - (RA) NOT EQUAL TO (DEST)
;*****
TST10:
MOV #10,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
1S: MOV #1,R4 ;RESULT S / B = 000001
CLR @#MBUFO ;MAKE (DEST) = 000000
SEZ ;MAKE N:C=0100
2S: CMP R4,(R2) ;TEST THE CMP
BNE TST11 ;;BR IF "Z" GOT CLEARED
3S: HALT 1$ ;ERROR - CMP FAILED TO CLR "Z"
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 11 BASIC "CMP #N,R" TEST - N = (R)
;*****
TST11:
MOV #11,R0 ;LOAD R0 WITH TEST NUMBER
MOV #125252,R4 ;RESULT S / B = 125252
1S: MOV #4,R3 ;(DEST) = 125252
CCC ;SCOPE SYNC
2S: CMP #125252,R3 ;TEST THE CMP
REQ 4$ ;BR IF N = (R)
3S: HALT 1$ ;CMP FAILED
BR 1$ ;LOCK ON HARD ERROR
4S: CMP R4,R3 ;DID CMP ALTER (DEST)?
BEQ TST12 ;;BR IF NO
5S: HALT ;CMP DELIVERED A RESULT

```

```

2091 003620 000765
2092
2093
2094
2095
2096 003622
2097 003622 012700 000012
2098 003626 005004
2099 003630 010403
2100 003632 000264
2101
2102 003634 022703 000001
2103
2104 003640 001002
2105
2106 003642 000000
2107 003644 000771
2108
2109 003646 020403
2110 003650 001402
2111
2112 003652 000000
2113 003654 000765
2114
2115
2116
2117
2118 003656
2119 003656 012700 000013
2120 003662 012702 063312
2121 003666 012704 177777
2122 003672 005012
2123 003674 000257
2124
2125 003676 010412
2126
2127 003700 020412
2128 003702 001402
2129
2130 003704 000000
2131 003706 000771
2132
2133
2134
2135
2136 003710
2137 003710 012700 000014
2138 003714 012702 063312
2139 003720 012704 177777
2140 003724 005012
2141 003726 000257
2142
2143 003730 012712 177777
2144
2145 003734 020412
2146 003736 001402

```

```

BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 12 BASIC "CMP #N,R" TEST - N NOT EQUAL TO (R)
;*****
TST12:
MOV #12,R0 ;LOAD R0 WITH TEST NUMBER
CLR R4 ;RESULT S / B = 000000
1S: MOV #4,R3 ;(DEST) = 125252
SEZ ;SCOPE SYNC
2S: CMP #1,R3 ;TEST THE CMP
BNE 4$ ;BR IF N NOT EQUAL TO (R)
3S: HALT 1$ ;CMP FAILED
BR 1$ ;LOCK ON HARD ERROR
4S: CMP R4,R3 ;DID CMP ALTER (DEST)?
BEQ TST13 ;;BR IF NO
5S: HALT 1$ ;CMP DELIVERED A RESULT
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 13 BASIC "MOV RA,(RB)" TEST
;*****
TST13:
MOV #13,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #1,R4 ;RESULT S / B = 177777
1S: CLR (R2) ;MAKE (DEST) = 000000
CCC ;SCOPE SYNC - N:C=0000
2S: MOV R4,(R2) ;TEST THE MOV
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TST14 ;;BR IF YES
3S: HALT 1$ ;ERROR - MOV FAILED
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 14 BASIC "MOV #N,(R)" TEST
;*****
TST14:
MOV #14,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #1,R4 ;RESULT S / B = 177777
1S: CLR (R2) ;MAKE (DEST) = 000000
CCC ;SCOPE SYNC
2S: MOV #1,(R2) ;TEST THE MOV
CMP R4,(R2) ;RESULT OK ?
BEQ TST15 ;;BR IF YES

```

```

2147
2148 003740 000000 3S: HALT ;ERROR - MOV FAILED
2149 003742 000770 BR 1$ ;LOCK ON HARD ERROR
2150
;*****
;TEST 15 BASIC "MOVB #N,X(R)" TEST - DEST EVEN
;*****
TST15:
2154 003744 012700 000015 MOV #15,R0 ;LOAD R0 WITH TEST NUMBER
2155 003750 012704 177401 MOV #177401,R4 ;RESULT S / B = 177401
2156 003754 012702 063316 MOV #MBUF1,R2 ;DEST ADDR = MBUF1
2157 003760 012705 063316 MOV #MBUF0,R5 ;BASE DEST ADDR = MBUF0
2158 003764 012712 177777 1$: MOV #-1,(R2) ;DESTJ = 177777
2159 003770 000257 CCC ;SCOPE SYNC
2160
2161 003772 112765 000001 000004 2$: MOVB #1,4(R5) ;TEST THE MOVB
2162 040000 020412 CMP R4,(R2) ;RESULT OK?
2163 040002 001402 BEQ TST16 ;BR IF YES
2164
2165 040004 000000 3$: HALT ;MOVB DELIVERED WRONG RESULT
2166 040006 000766 BR 1$ ;LOCK ON HARD ERROR
2167
;*****
;TEST 16 BASIC "MOVB #N,X(R)" TEST - DEST ODD
;*****
TST16:
2173 004010 012700 000016 MOV #16,R0 ;LOAD R0 WITH TEST NUMBER
2174 004014 012704 000777 MOV #1777,R4 ;RESULT S / B = 777
2175 004020 012702 063316 MOV #MBUF1,R2 ;DEST ADDR = MBUF1
2176 004024 012705 063316 MOV #MBUF0,R5 ;BASE DEST ADDR = MBUF0
2177 004030 012712 177777 1$: MOV #-1,(R2) ;DESTJ = 177777
2178 004034 000257 CCC ;SCOPE SYNC
2179
2180 004036 112765 000001 000005 2$: MOVB #1,5(R5) ;TEST THE MOVB
2181 040044 020412 CMP R4,(R2) ;RESULT OK?
2182 040046 001402 BEQ TST17 ;BR IF YES
2183
2184 040050 000000 3$: HALT ;MOVB DELIVERED WRONG RESULT
2185 040052 000766 BR 1$ ;LOCK ON HARD ERROR
2186
;*****
;TEST 17 BASIC "TST @#A" TEST WITH CAJ GT 0
;*****
TST17:
2192 004054 012700 000017 MOV #17,R0 ;LOAD R0 WITH TEST NUMBER
2193 004054 012702 063316 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
2194 004060 012704 000377 MOV #377,R4 ;RESULT S / B = 377 (NO CHANGE)
2195 004064 012704 000377 1$: MOV R4,(R2) ;DESTJ = 377
2196 004070 010412 CCC ;SCOPE SYNC
2197 004072 000257
2198
2199 004074 005737 063312 2$: TST @#MBUF0 ;TEST THE TST
2200
2201 004100 001401 BEQ 3$ ;BR IF "Z" SET - IT SHOULDN'T BE
2202 004102 100002 BPL TST20 ;BR IF "N" CLEAR - IT SHOULD BE

```

```

2203
2204 004104 000000 3$: HALT ;TST FAILED TO ALTER CODES PROPERLY
2205 004106 000770 BR 1$ ;LOCK ON HARD ERROR
2206
;*****
;TEST 20 BASIC "TST @#A" TEST WITH CAJ LT 0
;*****
TST20:
2209 004110 012700 000020 MOV #20,R0 ;LOAD R0 WITH TEST NUMBER
2210 004114 012702 063312 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
2211 004114 012704 100000 MOV #100000,R4 ;MAKE S / B = 100000
2212 004120 010412 1$: MOV R4,(R2) ;MAKE DESTJ = 100000
2213 004124 010412 CCC ;SCOPE SYNC
2214 004126 000257
2215
2216 004130 005737 063312 2$: TST @#MBUF0 ;TEST THE TST
2217
2218 004134 001401 BEQ 3$ ;BR IF "Z" SET - IT SHOULDN'T BE
2219 004136 100402 BMI 4$ ;BR IF "N" SET - IT SHOULD BE
2220
2221 004140 000000 3$: HALT ;TST FAILED TO ALTER CODES PROPERLY
2222 004142 000770 BR 1$ ;LOCK ON HARD ERROR
2223 004144 020412 4$: CMP R4,(R2) ;DID TST DISTURB DESTJ ?
2224 004146 001402 BEQ TST21 ;BR IF NOT
2225
2226 004150 000000 5$: HALT ;TST DELIVERED A RESULT
2227 004152 000764 BR 1$ ;LOCK ON HARD ERROR
2228
;*****
;TEST 21 BASIC "TST @#A" WITH CAJ = 0
;*****
TST21:
2232 004154 012700 000021 MOV #21,R0 ;LOAD R0 WITH TEST NUMBER
2233 004154 012702 063312 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
2234 004160 012704 100000 CLR R4 ;RESULT S / B = 0 (IT SHOULDN'T CHANGE)
2235 004164 015012 1$: CLR (R2) ;DESTJ = 0
2236 004166 015012 CCC ;SCOPE SYNC - Z=0
2237 004170 000257
2238
2239 004172 005737 063312 2$: TST @#MBUF0 ;TEST THE TST
2240
2241 004176 001402 BEQ 4$ ;BR IF TST SET "Z"
2242
2243 004200 000000 3$: HALT ;TST FAILED TO SET "Z"
2244 004202 000771 BR 1$ ;LOCK ON HARD ERROR
2245
2246 004204 020412 4$: CMP R4,(R2) ;DESTJ STILL = 000000
2247 004206 001402 BEQ TST22 ;BR IF YES
2248
2249 004210 000000 5$: HALT ;TST ALTERED THE DESTJ
2250 004212 000765 BR 1$ ;LOCK ON HARD ERROR
2251
;*****
;TEST 22 BASIC "BIT #N,@#A" WITH BIT SET IN "A"
;*****
TST22:
2255 004214 012700 000022 MOV #22,R0 ;LOAD R0 WITH TEST NUMBER
2256 004214 012702 063312 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
2257 004220 012704 040000 MOV #40000,R4 ;RESULT S / B = 40000

```

```

2259 004230 010412 15: MOV R4,(R2) ;MAKE [DEST] = 40000
2260 004232 000277 SCC ;SCOPE SYNC - Z=1
2261
2262 004234 032737 040000 063312 25: BIT #40000,##MBOFO ;TEST THE BIT
2263
2264 004242 001002 BNE TST23 ;;BR IF Z=0 - IT SHOULD BE
2265
2266 004244 000000 35: HALT ;BIT FAILED TO CLEAR "Z"
2267 004246 000770 BR 1$ ;LOCK ON HARD ERROR
2268
2269 ;*****
2270 ;*TEST 23 BASIC "BIT #N,##A" WITH BIT CLEAR IN "A"
2271 ;*****
2272 TST23:
2273 MOV #23,R0 ;;LOAD R0 WITH TEST NUMBER
2274 MOV #MBOFO,R2 ;DEST ADDR = MBOFO
2275 CLR (R2) ;MAKE [DEST] = 000000
2276 CCC ;SCOPE SYNC - Z=0
2277
2278 004264 032737 040000 063312 25: BIT #40000,##MBOFO ;TEST THE BIT
2279
2280 004272 001402 BEQ 4$ ;BR IF Z=1 - IT SHOULD BE
2281
2282 004274 000000 35: HALT ;BIT FAILED TO SET "Z"
2283 004276 000770 BR 1$ ;LOCK ON HARD ERROR
2284
2285 004300 005712 45: TST (R2) ;DID BIT DELIVER A RESULT
2286 004302 001402 TST24 ;;BR IF NOT
2287
2288 004304 000000 55: HALT ;BIT DISTURBED THE [DEST]
2289 004306 000764 BR 1$ ;LOCK ON HARD ERROR
2290
2291 ;*****
2292 ;*TEST 24 BASIC "TST (R)+ TEST
2293 ;*****
2294 TST24:
2295 MOV #24,R0 ;;LOAD R0 WITH TEST NUMBER
2296 .SBTTL USER CONTROLLED BREAKPOINT -- BIT0
2297 MOV #BIT0,##BPTLOC ;BREAKPOINT HALT SET ??
2298 BEQ +4 ;BR IF NOT
2299 HALT ;BREAK - DEPRESS CONTINUE TO RESTART
2300 MOV #MBOFO,R2 ;INITIAL DEST ADDR = MBOFO
2301 CLR (R2) ;MAKE [DEST] = 000000
2302 CCC ;SCOPE SYNC
2303
2304 004336 005722 25: TST (R2)+ ;TEST THE TST
2305
2306 004340 001402 BEQ 4$ ;BR IF "Z" SET - IT SHOULD BE
2307
2308 004342 000000 35: HALT ;TST FAILED TO SET "Z"
2309 004344 000772 BR 1$ ;LOCK ON HARD ERROR
2310
2311 004346 022702 063314 45: CMP #MBOFO+2,R2 ;DID REG. GET AUTO-INCREMENTED ?
2312 004352 001402 TST25 ;;BR IF YES
2313
2314 004354 000000 55: HALT ;TST FAILED TO UPDATE REGISTER

```

```

2315 004356 000765 BR 1$ ;LOCK ON HARD ERROR
2316
2317 ;*****
2318 ;*TEST 25 BASIC "TST -(R)" TEST
2319 ;*****
2320 TST25:
2321 MOV #25,R0 ;;LOAD R0 WITH TEST NUMBER
2322 MOV #DWTA+6,R2 ;DEST ADDR = DWTA+6
2323 MOV #377,R4 ;RESULT S / B = 377
2324 MOV #DWTA+10,R5 ;BASE DEST ADDR = DWTA+10
2325 SEN ;SCOPE SYNC
2326
2327 004402 005745 25: TST -(R5) ;TEST THE TST
2328
2329 004404 100002 RPL 4$ ;BR IF "N" CLEAR
2330
2331 004406 000000 35: HALT ;TST FAILED TO CLEAR "N"
2332 004410 000771 BR 1$ ;LOCK ON HARD ERROR
2333
2334 004412 020502 45: CMP R5,R2 ;DID DEST REG GET DECREMENTED?
2335 004414 001402 BEQ 6$ ;BR IF YES
2336
2337 004416 000000 55: HALT ;ERROR - TST FAILED TO UPDATE DEST REG
2338 004420 000765 BR 1$ ;LOCK ON HARD ERROR
2339
2340 004422 020412 65: CMP R4,(R2) ;DID TST ALTER [DEST]?
2341 004424 001403 TST26 ;;BR IF NOT
2342
2343 004426 000000 75: HALT ;TST ALTERED [DEST]
2344 004430 010412 MOV R4,(R2) ;RESTORE [DEST]
2345 004432 000760 BR 1$ ;LOCK ON HARD ERROR
2346
2347 ;*****
2348 ;*TEST 26 BASIC "COM ##A" TEST
2349 ;*****
2350 TST26:
2351 MOV #26,R0 ;;LOAD R0 WITH TEST NUMBER
2352 MOV #MBOFO,R2 ;DEST ADDR = MBOFO
2353 CLR R4 ;RESULT S / B = 177777
2354 COM R4
2355 CLR (R2) ;MAKE [DEST] = 000000
2356 CCC ;SCOPE SYNC
2357
2358 004454 005137 063312 25: COM @MBOFO ;TEST THE COM
2359
2360 004460 020412 75: CMP R4,(R2) ;RESULT = 177777 ??
2361 004462 001402 TST27 ;;BR IF YES
2362
2363 004464 000000 35: HALT ;COM DELIVERED THE WRONG RESULT
2364 004466 000770 BR 1$
2365
2366 ;*****
2367 ;*TEST 27 BASIC "INC ##A" TEST
2368 ;*****
2369 TST27:
2370 MOV #27,R0 ;;LOAD R0 WITH TEST NUMBER

```

2371 004474 012702 063312
 2372 004500 012704 000100
 2373 004504 012712 000077
 2374 004510 000257
 2375 004512 005237 063312
 2376
 2377
 2378 004516 020412
 2379 004520 001402
 2380
 2381 004522 000000
 2382 004524 000767
 2383
 2384
 2385
 2386
 2387 004526
 2388 004528 012700 000030
 2389 004532 012703 000001
 2390 004536 000257
 2391
 2392 004540 005303
 2393
 2394 004542 005703
 2395 004544 001402
 2396
 2397 004546 000000
 2398 004550 000770
 2399
 2400
 2401
 2402
 2403 004552
 2404 004554 012700 000031
 2405 004556 012704 177777
 2406 004562 012702 063312
 2407 004566 005015
 2408 004570 000257
 2409
 2410 004572 005337 063312
 2411
 2412 004576 020412
 2413 004600 001402
 2414
 2415 004602 000000
 2416 004604 000770
 2417
 2418
 2419
 2420
 2421 004606
 2422 004606 012700 000032
 2423 004612 012702 063312
 2424 004616 005004
 2425 004620 012705 063312
 2426 004624 012712 177777

```

MOV @#MBUFO,R2 ;DEST ADDR = MBUFO
MOV #100,R4 ;RESULT S / B = 100
MOV #77,(R2) ;CDESTJ = 77
CCC ;SCOPE SYNC

2$: INC @#MBUFO ;TEST THE INC

CMP R4,(R2) ;DID RESULT = 100 ??
BEQ TST30 ;;BR IF YES

3$: HALT 1$ ;INC DELIVERED WRONG RESULT
BR ;LOCK ON HARD ERROR

;*****
;TEST 30 BASIC "DEC RN" TEST
;*****
TST30:
MOV #30,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #1,R3 ;RESULT S / B = 1
CCC ;SCOPE SYNC

2$: DEC R3 ;TEST THE DEC

TST P3 ;RESULT = 000000 ??
BEQ TST31 ;;BR IF YES

3$: HALT 1$ ;DEC DELIVERED THE WRONG RESULT
BR ;LOCK ON HARD ERROR

;*****
;TEST 31 BASIC "DEC @#A" TEST
;*****
TST31:
MOV #31,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #-1,R4 ;RESULT S / B = 177777
MOV @#MBUFO,R2 ;DEST ADDR = MBUFO
CLR (R2) ;MAKE CDESTJ = 000000
CCC ;SCOPE SYNC

2$: DEC @#MBUFO ;TEST THE DEC

CMP R4,(R2) ;DID RESULT = 177777 ??
BEQ TST32 ;;BR IF YES

3$: HALT 1$ ;DEC DELIVERED WRONG RESULT
BR ;LOCK ON HARD ERROR

;*****
;TEST 32 BASIC "CLR X(R)" TESTS
;*****
TST32:
MOV #32,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#MBUFO+2,R2 ;DEST ADDR = MBUFO+2
CLR R4 ;RESULT S / B = 000000
MOV @#MBUFO,R5 ;BASE DEST ADDR = MBUFO
MOV #-1,(R2) ;CDESTJ = 177777

```

2427 004630 000257
 2428 004632 005065 000002
 2429
 2430
 2431 004636 020412
 2432 004640 001402
 2433

```

CCC ;SCOPE SYNC

2$: CLR 2(R5) ;TEST THE CLR

CMP R4,(R2) ;RESULT = 0?
BEQ TST33 ;;BR IF YES

```

2434 004642 000000
 2435 004644 000765
 2436
 2437
 2438
 2439
 2440 004646
 2441 004646 012700 000033
 2442 004652 012703 125252
 2443 004656 000257
 2444
 2445 004660 006303
 2446
 2447 004662 103402
 2448
 2449 004664 000000
 2450 004666 000771
 2451
 2452 004670 022703 052524
 2453 004674 001402
 2454
 2455 004676 000000
 2456 004700 000764
 2457
 2458
 2459
 2460
 2461 004702
 2462 004702 012700 000034
 2463 004706 012703 052525
 2464 004712 000261
 2465
 2466 004714 006303
 2467
 2468 004716 103002
 2469
 2470 004720 000000
 2471 004722 000771
 2472
 2473 004724 022703 125252
 2474 004730 001402
 2475
 2476 004732 000000
 2477 004734 000764
 2478
 2479
 2480
 2481 004736
 2482 004736 012700 000035
 2483 004742 012703 125252
 2484 004746 000257
 2485
 2486 004750 006103
 2487
 2488 004752 103402
 2489

35: HALT BR 1\$;CLR FAILED TO ZERO [DEST]
 ;LOCK ON HARD ERROR.
 ;*****
 ;*TEST 33 BASIC "ASL RN" TEST WITH [DEST]=125252 AND C(0)
 ;*****
 TST33:
 1\$: MOV #33,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #125252,R3 ;MAKE [DEST] = 125252
 CCC ;MAKE C=0
 2\$: ASL R3 ;TEST THE ASL - IT SHOULD SET "C"
 BCS 4\$;BR IF "C" GOT SET
 3\$: HALT BR 1\$;ASL FAILED TO SET "C" BIT
 ;LOCK ON HARD ERROR
 4\$: CMP #52524,R3 ;WAS RESULT = 52524 ??
 BEQ TST34 ;BR IF YES
 5\$: HALT BR 1\$;ASL DELIVERED THE WRONG RESULT
 ;LOCK ON HARD ERROR
 ;*****
 ;*TEST 34 BASIC "ASL RN" TEST WITH [DEST]=052525 AND C(1)
 ;*****
 TST34:
 1\$: MOV #34,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #052525,R3 ;MAKE [DEST] = 052525
 SEC ;MAKE C=1
 2\$: ASL R3 ;TEST THE ASL - IT SHOULD CLR "C"
 BCC 4\$;BR IF "C" GOT CLEARED
 3\$: HALT BR 1\$;ASL FAILED TO CLEAR "C"
 ;LOCK ON HARD ERROR
 4\$: CMP #125252,R3 ;RESULT = 125252 ??
 BEQ TST35 ;BR IF YES
 5\$: HALT BR 1\$;ASL DELIVERED WRONG RESULT
 ;LOCK ON HARD ERROR
 ;*****
 ;*TEST 35 BASIC "ROL RN" TEST WITH [DEST]=125252 AND C(0)
 ;*****
 TST35:
 1\$: MOV #35,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #125252,R3 ;MAKE [DEST] = 125252
 CCC ;MAKE C=0
 2\$: ROL R3 ;TEST THE ROL - IT SHOULD SET C
 BCS 4\$;BR IF "C" GOT SET

2490 004754 000000
 2491 004756 000771
 2492
 2493 004760 022703 052524
 2494 004764 001402
 2495
 2496 004766 000000
 2497 004770 000764
 2498
 2499
 2500
 2501
 2502 004772
 2503 004772 012700 000036
 2504 004776 012703 052524
 2505 005002 000261
 2506
 2507 005004 006103
 2508
 2509 005006 103002
 2510
 2511 005010 000000
 2512 005012 000771
 2513
 2514 005014 022703 125251
 2515 005020 001402
 2516
 2517 005022 000000
 2518 005024 000764
 2519
 2520
 2521
 2522
 2523 005036
 2524 005036 012700 000037
 2525 005032 012702 063330
 2526 005036 012704 000377
 2527 005042 000257
 2528
 2529 005044 105712
 2530
 2531 005046 100402
 2532
 2533 005050 000000
 2534 005052 000773
 2535
 2536 005054 020412
 2537 005056 001403
 2538
 2539 005060 000000
 2540 005062 010412
 2541 005064 000766
 2542
 2543
 2544
 2545 005066

35: HALT BR 1\$;ROL FAILED TO SET "C"
 ;LOCK ON HARD ERROR
 4\$: CMP #052524,R3 ;RESULT = 052524 ??
 BEQ TST36 ;BR IF YES
 5\$: HALT BR 1\$;ROL DELIVERED WRONG RESULT
 ;LOCK ON HARD ERROR
 ;*****
 ;*TEST 36 BASIC "ROL RN" TEST WITH [DEST]=052524 AND C(1)
 ;*****
 TST36:
 1\$: MOV #36,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #052524,R3 ;MAKE [DEST] = 052524
 SEC ;MAKE C=1
 2\$: ROL R3 ;TEST THE ROL - IT SHOULD CLEAR C
 BCC 4\$;BR IF "C" IS CLEAR
 3\$: HALT BR 1\$;ROL FAILED TO CLEAR "C"
 ;LOCK ON HARD ERROR
 4\$: CMP #125251,R3 ;RESULT = 125251 ??
 BEQ TST37 ;BR IF YES
 5\$: HALT BR 1\$;ROL DELIVERED WRONG RESULT
 ;LOCK ON HARD ERROR
 ;*****
 ;*TEST 37 BASIS "TSTB (R)" TEST - EVEN ADDRESS
 ;*****
 TST37:
 1\$: MOV #37,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #DWT+6,R2 ;TEST ADDR = DWT+6
 MOV #377,R4 ;RESULT S / B = 377
 CCC ;SCOPE SYNC
 2\$: TSTB (R2) ;TEST THE TSTB
 BMI 4\$;BR IF "N" SET - IT SHOULD BE
 3\$: HALT BR 1\$;TSTB FAILED TO SET "N"
 ;LOCK ON HARD ERROR
 4\$: CMP R4,(R2) ;DID TSTB DISTURB [DEST]
 BEQ TST40 ;BR IF NOT
 5\$: HALT MOV R4,(R2) ;TSTB ALTERED [DEST]
 BR 1\$;RESTORE [DEST]
 ;LOCK ON HARD ERROR
 ;*****
 ;*TEST 40 BASIS "TSTB (R)" TEST - ODD ADDRESS
 ;*****
 TST40:

2546 005066 012700 000040
2547 005072 012702 064040
2548 005076 012703 177401
2549 005102 012703 064041
2550 005106 000257
2551
2552 005110 105713
2553
2554 005112 100402
2555
2556 005114 000000
2557 005116 000773
2558
2559 005120 020412
2560 005122 001403
2561
2562 005124 000000
2563 005126 010412
2564 005130 000766
2565
2566
2567
2568
2569 005132
2570 005132 012700 000041
2571 005136 012703 063326
2572 005136 012704 177400
2573 005146 000257
2574
2575 005150 105737 063326
2576
2577 005154 001402
2578
2579 005156 000000
2580 005160 000772
2581
2582 005162 020412
2583 005164 001403
2584
2585 005166 000000
2586 005170 010412
2587 005172 000765
2588
2589
2590
2591
2592 005174
2593 005174 012700 000042
2594 005200 012703 063330
2595 005204 012704 000377
2596 005210 000257
2597
2598 005212 105737 063331
2599
2600 005216 001402
2601

```
MOV #40,R0 ;LOAD R0 WITH TEST NUMBER
MOV #DWTB+6,R2 ;DEST ADDR = DWTB+6
MOV #177400,R4 ;RESULT S / B = 177401
MOV #DWTB+,R3 ;DEST ADDR USED = DWTB+7
CCC ;SCOPE SYNC

1$:
2$: TSTB (R3) ;TEST THE TSTB
BMI 4$ ;BR IF "N" SET - IT SHOULD BE

3$: HALT ;TSTB FAILED TO SET "N"
BR 1$ ;LOCK ON HARD ERROR

4$: CMP R4,(R2) ;DID TSTB DISTURB [DEST]
BEQ TST41 ;BR IF NOT

5$: HALT ;TSTB ALTERED [DEST]
MOV R4,(R2) ;RESTORE [DEST]
BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 41 BASIC "TSTB @#A" TEST - EVEN ADDRESS
;*****
TST41:
MOV #41,R0 ;LOAD R0 WITH TEST NUMBER
MOV #DWTB+4,R2 ;DEST ADDR = DWTB+4
MOV #177400,R4 ;RESULT S / B = 177400
CCC ;SCOPE SYNC

1$:
2$: TSTB @#DWTB+4 ;TEST THE TSTB
BEQ 4$ ;BR IF "Z" SET - IT SHOULD BE

3$: HALT ;TSTB FAILED TO SET "Z"
BR 1$ ;LOCK ON HARD ERROR

4$: CMP R4,(R2) ;DID TSTB DISTURB [DEST]?
BEQ TST42 ;BR IF NOT

5$: HALT ;TSTB ALTERED [DEST]
MOV R4,(R2) ;RESTORE [DEST]
BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 42 BASIC "TSTB @#A" TEST - ODD ADDRESS
;*****
TST42:
MOV #42,R0 ;LOAD R0 WITH TEST NUMBER
MOV #DWTB+6,R2 ;DEST ADDR = DWTB+6
MOV #377,R4 ;RESULT S / B = 377
CCC ;SCOPE SYNC

1$:
2$: TSTB @#DWTB+7 ;TEST THE TSTB
BEQ 4$ ;BR IF "Z" SET - IT SHOULD BE
```

2602 005220 000000
2603 005222 000772
2604
2605 005224 020412
2606 005226 001403
2607
2608 005230 000000
2609 005232 010412
2610 005234 000765
2611
2612
2613
2614
2615 005236
2616 005236 012700 000043
2617 005242 010605
2618 005244 012704 177400
2619 005250 010506
2620 005254 000257
2621
2622 005256 105366 000001
2623
2624 005262 020416
2625 005264 001402
2626
2627
2628 005266 000000
2629 005270 000767
2630
2631 005272 010506
2632
2633
2634
2635
2636 005274
2637 005274 012700 000044
2638 005300 005003
2639 005302 000257
2640
2641 005304 013703 063276
2642
2643 005310 022703 063322
2644 005314 001402
2645
2646 005316 000000
2647 005320 000767
2648
2649
2650
2651
2652 005322
2653 005322 012700 000045
2654 005326 012703 063314
2655 005326 012704 123252
2656 005336 012703 063312
2657 005342 005012

```
3$: HALT ;TSTB FAILED TO SET "Z"
BR 1$ ;LOCK ON HARD ERROR

4$: CMP R4,(R2) ;DID TSTB DISTURB [DEST]?
BEQ TST43 ;BR IF NOT

5$: HALT ;TSTB ALTERED [DEST]
MOV R4,(R2) ;RESTORE [DEST]
BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 43 BASIC "DECB 1(SP)"
;*****
TST43:
MOV #43,R0 ;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;SAVE SP
MOV #177400,R4 ;RESULT S / B = 177400
1$:
CLR R5,SP ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: DECB 1(SP) ;TEST THE DECB
CMP R4,(SP) ;RESULT = 177400?
BEQ 4$ ;BR IF YES

3$: HALT ;ERROR - DECB FAILED
BR 1$ ;LOCK ON HARD ERROR

4$: MOV R5,SP ;RESET THE SP

;*****
;TEST 44 BASIC "MOV @#A,R" TEST
;*****
TST44:
MOV #44,R0 ;LOAD R0 WITH TEST NUMBER
1$:
CLR R3 ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV @#ATA,R3 ;TEST THE MOV

3$: CMP #DWTB,R3 ;RESULT = DWTB?
BEQ TST45 ;BR IF YES

3$: HALT ;MOV FAILED TO DELIVER CORRECT RESULT
BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 45 BASIC "MOV #N,X(R)" TEST
;*****
TST45:
MOV #45,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0+2,R2 ;DEST ADDR = MBUF0+2
MOV #125252,R4 ;RESULT S / B = 125252
MOV #MBUF0,R3 ;[R3] = BASE DEST ADDR
1$:
CLP (R2) ;[DEST] = 000000
```



```
2658 005344 000257          CCC                ;SCOPE SYNC
2659
2660 005346 012763 125252 000002 2$:  MOV      #125252,2(R3)  ;TEST THE MOV
2661
2662 005354 020412          CMP      R4,(R2)    ;RESULT OK?
2663 005356 001402          BEQ     TST46       ;BR IF YES
2664
2665 005360 000000          3$:  HALT     ;MOV DELIVERED WRONG RESULT
2666 005362 000765          BR      1$         ;LOCK ON HARD ERROR
2667
2668 ;*****
2669 ;TEST 46 BASIC "MOV #N,(R)" TEST
2670 ;*****
2671 005364          TST46:
2672 005364 012700 000046          MOV      #46,R0     ;LOAD R0 WITH TEST NUMBER
2673 005370 012703 063312          MOV     #MBUF0,R3   ;DEST ADDR = MBUF0
2674 005374 012704 125252          MOV     #125252,R4  ;RESULT S / B = 125252
2675 005400 005013          CLR     (R3)        ;DESTJ = 000000
2676 005402 000257          CCC     ;SCOPE SYNC
2677
2678 005404 012713 125252          2$:  MOV      #125252,(R3) ;TEST THE MOV
2679
2680 005410 020413          CMP     R4,(R3)    ;RESULT OK?
2681 005412 001402          BEQ     TST47       ;BR IF YES
2682
2683 005414 000000          3$:  HALT     ;MOV DELIVERED WRONG RESULT
2684 005416 000770          BR      1$         ;LOCK ON HARD ERROR
2685
2686 ;*****
2687 ;TEST 47 BASIC "MOV (RA)+,RB" TEST
2688 ;*****
2689 005420          TST47:
2690 005420 012700 000047          MOV     #47,R0     ;LOAD R0 WITH TEST NUMBER
2691 005424 012705 063276          MOV     #ATA,R5    ;SRC ADDR = ATA
2692 005430 005003          CLR     R3         ;DESTJ = 000000
2693 005432 000257          CCC     ;SCOPE SYNC
2694
2695 005434 012503          2$:  MOV      (R5)+,R3   ;TEST THE MOV
2696
2697 005436 022703 063322          CMP     #DWTA,R3   ;RESULT OK?
2698 005442 000402          BR      4$         ;BR IF YES
2699
2700 005444 000000          3$:  HALT     ;MOV DELIVERED WRONG RESULT
2701 005446 000766          BR      1$         ;LOCK ON HARD ERROR
2702
2703 005450 022705 063300          4$:  CMP     #ATA+2,R5  ;DID SRC REG GET INCREMENTED?
2704 005454 001402          BEQ     TST50       ;BR IF YES
2705
2706 005456 000000          5$:  HALT     ;MOV FAILED TO UPDATE SRC. REG.
2707 005460 000761          BR      1$         ;LOCK ON HARD ERROR
2708
2709 ;*****
2710 ;TEST 50 BASIC "MOV @#A,@#B"
2711 ;*****
2712 005462          TST50:
2713 005462 012700 000050          MOV     #50,R0     ;LOAD R0 WITH TEST NUMBER
```

```
2714 005466 012702 063316          MOV     #MBUF1,R2   ;DEST ADDR = MBUF1
2715 005472 012704 063322          MOV     #DWTA,R4   ;RESULT S / B = #DWTA
2716 005476 005012          CLR     (R2)        ;MAKE CDESTJ = 000000
2717 005500 000257          CCC     ;SCOPE SYNC
2718
2719 005502 013737 063276 063316          2$:  MOV     @#ATA,@#MBUF1 ;TEST THE MOV
2720 005510 020412          CMP     R4,(R2)    ;DID RESULT = #DWTA ?
2721 005512 001402          BEQ     TST51       ;BR IF YES
2722
2723 005514 000000          3$:  HALT     ;MOV DELIVERED THE WRONG RESULT
2724 005516 000767          BR      1$         ;LOCK ON HARD ERROR
2725
2726 ;*****
2727 ;TEST 51 BASIC "MOV X(R),PC" TEST
2728 ;*****
2729 005520          TST51:
2730 005520 012700 000051          MOV     #51,R0     ;LOAD R0 WITH TEST NUMBER
2731 005524 012705 005532          MOV     #25,R5     ;R5J = 25 (BASE ADDRESS)
2732 005530 000257          CCC     ;SCOPE SYNC
2733
2734 005532 016507 000010          2$:  MOV     4$-2$(R5),PC ;TEST THE MOV - GO TO NEXT TEST VIA 4$
2735
2736 005536 000000          3$:  HALT     ;MOV FAILED TO LOAD THE PC
2737 005540 000771          BR      1$         ;LOCK ON HARD ERROR
2738
2739 005542 005544          4$:  .+2     ;POINTER TO NEXT TEST
2740
2741 ;*****
2742 ;TEST 52 BASIC "MOV @#A,(R)" TEST
2743 ;*****
2744 005544          TST52:
2745 005544 012700 000052          MOV     #52,R0     ;LOAD R0 WITH TEST NUMBER
2746 005548 012704 063322          MOV     #DWTA,R4   ;RESULT S / B = #DWTA
2747 005554 012702 063312          MOV     #MBUF0,R2  ;DEST ADDR = MBUF0
2748 005560 005012          CLR     (R2)        ;MAKE CDESTJ=000000
2749 005562 000257          CCC     ;SCOPE SYNC - Z=0
2750
2751 005564 013712 063276          2$:  MOV     @#ATA,(R2)  ;TEST THE MOV
2752
2753 005570 020412          CMP     R4,(R2)    ;DID RESULT = #DWTA ??
2754 005572 001402          BEQ     TST53       ;BR IF YES
2755
2756 005574 000000          3$:  HALT     ;MOV DELIVERED WRONG RESULT
2757 005576 000770          BR      1$         ;LOCK ON HARD ERROR
2758
2759 ;*****
2760 ;TEST 53 BASIC "MOV X(RA),RB" TEST
2761 ;*****
2762 005600          TST53:
2763 005600 012700 000053          MOV     #53,R0     ;LOAD R0 WITH TEST NUMBER
2764 005604 012705 063276          MOV     #ATA,R5    ;R5J = BASE ADDR FOR SOURCE (ATA)
2765 005610 005003          CLR     R3         ;MAKE CDESTJ = 000000
2766 005612 000257          CCC     ;SCOPE SYNC
2767
2768 005614 016503 000004          2$:  MOV     4(R5),R3   ;TEST THE MOV
```

2770 005620 022703 064630
2771 005624 001402
2772
2773 005626 000000
2774 005630 000767
2775
2776
2777
2778
2779 005632
2780 005632 012700 000054
2781 005636 012702 063312
2782 005642 012704 125252
2783 005642 012705 063314
2784 005652 005012
2785 005654 000257
2786
2787 005656 010445
2788
2789 005660 020412
2790 005662 001402
2791
2792 005664 000000
2793 005666 000767
2794
2795 005670 020205
2796 005672 001402
2797
2798 005674 000000
2799 005676 000763
2800
2801
2802
2803
2804 005700
2805 005700 012700 000055
2806 005704 012704 063312
2807 005710 012702 063312
2808 005714 012705 063314
2809 005720 005012
2810 005722 000257
2811
2812 005724 013745 063276
2813
2814 005730 020412
2815 005732 001402
2816
2817 005734 000000
2818 005736 000766
2819
2820 005740 020502
2821 005742 001402
2822
2823 005744 000000
2824 005746 000762
2825

```

      CMP      #DBTA,R3      ;RESULT = #DBTA ??
      BEQ      T54          ;;BR IF YES
3$:   HALT      1$          ;MOV DELIVERED WRONG RESULT
      BR      1$          ;LOCK ON HARD ERROR
;*****
;TEST 54 BASIC "MOV RA,-(RB)" TEST
;*****
T54:  MOV      #54,R0        ;;LOAD R0 WITH TEST NUMBER
      MOV      #MBUF0,R2    ;;DEST ADDR = MBUF0
      MOV      #DWTA,R4     ;;RESULT S / B = #DWTA
1$:   MOV      #MBUF0+2,R5  ;;INITIAL DEST ADDR = MBUF0+2
      CLR      (R2)        ;;MAKE (DEST) = 000000
      CCC
      2$:   MOV      R4,-(R5) ;TEST THE MOV
      CMP      R4,(R2)     ;RESULT = 125252
      BEQ      4$          ;BR IF YES
      3$:   HALT      1$          ;MOV DELIVERED THE WRONG RESULT
      BR      1$          ;LOCK ON HARD ERROR
      4$:   CMP      R2,R5     ;DID REGISTER GET DECREMENTED ?
      BEQ      T55        ;;BR IF YES
      5$:   HALT      1$          ;MOV FAILED TO UPDATE REGISTER
      BR      1$          ;LOCK ON HARD ERROR
;*****
;TEST 55 BASIC "MOV @#A,-(R)" TEST
;*****
T55:  MOV      #55,R0        ;;LOAD R0 WITH TEST NUMBER
      MOV      #@#A,R4     ;;RESULT S / B = #DWTA
1$:   MOV      #MBUF0,R2    ;;DEST ADDR = MBUF0
      MOV      #MBUF0+2,R5 ;;INITIAL DEST ADDR = MBUF0+2
      CLR      (R2)        ;;MAKE (DEST) = 000000
      CCC
      2$:   MOV      @#ATA,-(R5) ;TEST THE MOV
      CMP      R4,(R2)     ;RESULT = 000000
      BEQ      4$          ;BR IF YES
      3$:   HALT      1$          ;MOV DELIVERED THE WRONG RESULT
      BR      1$          ;LOCK ON HARD ERROR
      4$:   CMP      R5,R2     ;DID DEST REG GET DECREMENTED ??
      BEQ      T56        ;;BR IF YES
      5$:   HALT      1$          ;MOV FAILED TO UPDATE REGISTER
      BR      1$          ;LOCK ON HARD ERROR

```

2826
2827
2828
2829 005750
2830 005750 012700 000056
2831 005754 012702 063312
2832 005760 012704 063322
2833 005764 012705 063276
2834 005770 005012
2835 005772 000257
2836
2837 005774 011537 063312
2838
2839 006000 020412
2840 006002 001402
2841
2842 006004 000000
2843 006006 000770
2844
2845
2846
2847
2848 006010
2849 006010 012700 000057
2850 006014 012702 063312
2851 006020 012704 063322
2852 006024 012705 063300
2853 006030 005012
2854 006032 000257
2855
2856 006034 014537 063312
2857
2858 006040 020412
2859 006042 001402
2860
2861 006044 000000
2862 006046 000766
2863
2864 006050 022705 063276
2865 006054 001402
2866
2867 006056 000000
2868 006060 000761
2869
2870
2871
2872 006062
2873 006062 012700 000060
2874 006066 012705 063276
2875 006072 005003
2876 006074 000257
2877
2878 006076 012503
2879
2880 006100 022703 063322
2881 006104 001402

```

;*****
;TEST 56 BASIC "MOV (R),@#A" TEST
;*****
T56:  MOV      #56,R0        ;;LOAD R0 WITH TEST NUMBER
      MOV      #MBUF0,R2    ;;DEST ADDR = MBUF0
      MOV      #DWTA,R4     ;;RESULT S / B = #DWTA
1$:   MOV      #ATA,R5     ;;SOURCE ADDR = ATA
      CLR      (R2)        ;;MAKE (DEST) = 000000
      CCC
      2$:   MOV      (R5),@#MBUF0 ;TEST THE MOV
      CMP      R4,(R2)     ;RESULT = #DWTA ??
      BEQ      T57        ;;BR IF YES
      3$:   HALT      1$          ;MOV DELIVERED THE WRONG RESULT
      BR      1$          ;LOCK ON HARD ERROR
;*****
;TEST 57 BASIC "MOV -(R),@#A" TEST
;*****
T57:  MOV      #57,R0        ;;LOAD R0 WITH TEST NUMBER
      MOV      #MBUF0,R2    ;;DEST ADDR = MBUF0
1$:   MOV      #DWTA,R4     ;;RESULT S / B = #DWTA
      MOV      #ATA+2,R5   ;;INITIAL SOURCE ADDR = ATA+2
      CLR      (R2)        ;;MAKE (DEST) = 000000
      CCC
      2$:   MOV      -(R5),@#MBUF0 ;TEST THE MOV
      CMP      R4,(R2)     ;RESULT = #DWTA ?
      BEQ      4$          ;BR IF YES
      3$:   HALT      1$          ;MOV DELIVERED THE WRONG RESULT
      BR      1$          ;LOCK ON HARD ERROR
      4$:   CMP      #ATA,R5   ;DID THE SRC REG GET DECREMENTED ?
      BEQ      T58        ;;BR IF YES
      5$:   HALT      1$          ;MOV FAILED TO UPDATE SOURCE REG
      BR      1$          ;LOCK ON HARD ERROR
;*****
;TEST 60 BASIC "MOV (RA),RB" TEST
;*****
T60:  MOV      #60,R0        ;;LOAD R0 WITH TEST NUMBER
1$:   MOV      #ATA,R5     ;;INITIAL SOURCE ADDR = ATA
      CLR      R3          ;;MAKE (DEST) = 000000
      CCC
      2$:   MOV      (R5)+,R3 ;TEST THE MOV
      CMP      #DWTA,R3    ;RESULT = #DWTA ?
      BEQ      4$          ;BR IF YES

```

```

2882 006106 000000 3S: HALT ;MOV DELIVERED WRONG RESULT
2883 006110 000766 BR ;LOCK ON HARD ERROR
2884
2885 006112 022705 063300 4S: CMP #ATA+2,R5 ;DID SOURCE REG GET INCREMENTED
2886 006116 001402 BEQ TST61 ;BR IF YES
2887
2888 006120 000000 5S: HALT ;MOV FAILED TO UPDATE SOURCE REGISTER
2889 006122 000761 BR ;LOCK ON HARD ERROR
2890
2891 ;*****
2892 ;*TEST 61 BASIC "MOV X(RA),RB" TEST
2893 ;*****
2894
2895 006124 012700 000061 TST61: MOV #61,R0 ;LOAD RO WITH TEST NUMBER
2896 006130 012705 063276 MOV #ATA,R5 ;BASE SOURCE ADDR = ATA
2897 006134 005003 1S: CLR R3 ;BASE ADDRESS IN R5
2898 006136 000257 CCC ;SCOPE SYNC
2899
2900 006140 016503 000002 2S: MOV 2(R5),R3 ;TEST THE MOV
2901
2902 006144 022703 064032 CMP #DWTB,R3 ;RESULT = #DWTB ?
2903 006150 001402 BEQ TST62 ;BR IF YES
2904
2905 006152 000000 3S: HALT ;MOV FAILED TO DELIVER CORRECT RESULT
2906 006154 000767 BR ;LOCK ON HARD ERROR
2907
2908 ;*****
2909 ;*TEST 62 BASIC "MOV @X(RA),RB" TEST
2910 ;*****
2911
2912 006156 012700 000062 TST62: MOV #62,R0 ;LOAD RO WITH TEST NUMBER
2913 006162 012705 063312 MOV #DWTB+2,@#MBUF0+2 ;SET UP ADDRESS TABLE MBUF0
2914 006170 005003 1S: CLR R3 ;BASE ADDRESS IN R5
2915 006174 000257 CCC ;MAKE [DEST] = 000000
2916 006176 000257 ;SCOPE SYNC
2917
2918 006200 017503 000002 2S: MOV @2(R5),R3 ;TEST THE MOV
2919
2920 006204 022703 177777 CMP #-1,R3 ;RESULT = 177777
2921 006210 001402 BEQ TST63 ;BR IF YES
2922
2923 006212 000000 3S: HALT ;MOV DELIVERED THE WRONG RESULT
2924 006214 000767 BR ;LOCK ON HARD ERROR
2925
2926 ;*****
2927 ;*TEST 63 BASIC "MOV (R)+,X(R)" TEST
2928 ;*****
2929
2930 006216 012700 000063 TST63: MOV #63,R0 ;LOAD RO WITH TEST NUMBER
2931 006218 012704 000063 MOV #13252,R4 ;RESULT S / B = 125252
2932 006226 012703 063320 MOV #MBUF1+5,R2 ;FINAL DEST ADDR = MBUF1+2
2933 006232 010437 063312 MOV R4,@#MBUF0 ;SOURCE OPERAND = 125252
2934 006236 012705 063312 1S: MOV #MBUF0,R5 ;[R5] = INITIAL SRC ADDR = MBUF0
2935 006242 005012 CCC ;MAKE [DEST] = 000000
2936 006244 000257 ;SCOPE SYNC
2937

```

```

2938 006246 012565 000004 2S: MOV (R5)+,4(R5) ;TEST THE MOV
2939
2940 006252 020412 3S: CMP R4,(R2) ;RESULT = 125252 ?
2941 006254 001402 BEQ 4S ;BR IF YES
2942
2943 006256 000000 3S: HALT ;MOV DELIVERED WRONG RESULT
2944 006260 000766 BR ;LOCK ON HARD ERROR
2945
2946 006262 022705 063314 4S: CMP #MBUF0+2,R5 ;DID REGISTER GET INCREMENTED ?
2947 006266 001402 BEQ TST64 ;BR IF YES
2948
2949 006270 000000 5S: HALT ;MOV FAILED TO UPDATE REGISTER
2950 006272 000761 BR ;LOCK ON HARD ERROR
2951
2952 ;*****
2953 ;*TEST 64 BASIC "CMP R,@#A" TEST WITH [R] = [A]
2954 ;*****
2955
2956 006274 012700 000064 TST64: MOV #64,R0 ;LOAD RO WITH TEST NUMBER
2957
2958 006300 032737 000002 063234 .SBITL USER CONTROLLED BREAKPOINT -- BIT1
2959 006306 001401 BIT #BIT1,@#BPTLOC ;BREAKPOINT HALT SET ??
2960 006310 000000 BEQ .+4 ;BR IF NOT
2961 006312 012702 HALT ;BREAK - DEPRESS CONTINUE TO RESTART
2962 006316 012704 125252 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
2963 006322 010405 1S: MOV #125252,R4 ;RESULT S / B = 125252
2964 006324 010412 MOV R4,R5 ;[R5] = SOURCE OP = 125252
2965 006326 000257 CCC ;MAKE [DEST] = 125252
2966 ;SCOPE SYNC
2967
2968 006330 020537 063312 2S: CMP R5,@#MBUF0 ;TEST THE CMP
2969
2970 006334 001402 BEQ 4S ;BR IF "Z" WAS SET - IT SHOULD BE
2971
2972 006336 000000 3S: HALT ;CMP FAILED TO SET "Z"
2973 006340 000770 BR ;LOCK ON HARD ERROR
2974
2975 006342 020412 4S: CMP R4,(R2) ;IS RESULT STILL = 125252 ?
2976 006344 001402 BEQ TST65 ;BR IF YES
2977
2978 006346 000000 5S: HALT ;CMP ALTERED [DEST]
2979 006350 000764 BR ;LOCK ON HARD ERROR
2980
2981 ;*****
2982 ;*TEST 65 BASIC "CMP R,@#A" WITH [R] NOT EQUAL TO [A]
2983 ;*****
2984
2985 006352 012700 000065 TST65: MOV #65,R0 ;LOAD RO WITH TEST NUMBER
2986 006356 012702 063312 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
2987 006362 012704 125252 MOV #125252,R4 ;MAKE RESULT S / B = 125252
2988 006366 005005 1S: CLR R5 ;[R5] = SOURCE OP = 000000
2989 006370 010412 MOV R4,(R2) ;MAKE [DEST] = 125252
2990 006372 000277 SCC ;SCOPE SYNC - MAKE Z=1
2991
2992 006374 020537 063312 2S: CMP R5,@#MBUF0 ;TEST THE CMP
2993 006400 001002 BNE TST66 ;BR IF Z=0 - IT SHOULD BE

```

```
2994
2995 006402 000000
2996 006404 000770
2997
2998
2999
3000
3001 006406
3002 006406 012700 000066
3003 006410 012704 063312
3004 006412 012704 177777
3005 006422 005012
3006 006424 000257
3007
3008 006426 052737 177777 063312
3009
3010 006434 020412
3011 006436 001402
3012
3013 006440 000000
3014 006442 000767
3015
3016
3017
3018
3019 006444
3020 006444 012700 000067
3021 006450 012702 063312
3022 006454 012704 000377
3023 006460 012712 177777
3024 006464 000257
3025
3026 006466 042737 177700 063312
3027
3028 006474 020412
3029 006476 001402
3030
3031 006500 000000
3032 006502 000766
3033
3034
3035
3036
3037 006504
3038 006510 012700 000070
3039 006514 005003
3040 006512 005103
3041 006514 000257
3042
3043 006516 042703 177400
3044
3045 006522 022703 000377
3046 006526 001402
3047
3048 006530 000000
3049 006532 000766

3S: HALT ;CMP FAILED TO CLEAR "Z"
BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 66 BASIC "BIS #N,@#A" TEST - N=177777,[A]=000000
;*****
TST66:
MOV #66,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #1,R4 ;RESULT S / B = 177777
1$: CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2$: BIS #-1,@#MBUF0 ;TEST THE BIS
CMP R4,(R2) ;RESULT OK?
BEQ TST67 ;BR IF YES
3$: HALT ;BIS FAILED TO SET ALL BITS IN BITFLG
BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 67 BASIC "BIC #N,@#A" TEST
;*****
TST67:
MOV #67,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #77,R4 ;RESULT S / B = 77
1$: MOV #-1,(R2) ;MAKE [DEST] = 177777
CCC ;SCOPE SYNC
2$: BIC #177700,@#MBUF0 ;TEST THE BIC
CMP R4,(R2) ;DID RESULT = 77 ?
BEQ TST70 ;BR IF YES
3$: HALT ;BIC DELIVERED THE WRONG RESULT
BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 70 BASIC "BIC #N,R" TEST
;*****
TST70:
MOV #70,R0 ;LOAD R0 WITH TEST NUMBER
CLR R3 ;[DEST] = 177777
COM R3
CCC ;SCOPE SYNC
2$: BIC #177400,R3 ;TEST THE BIC
CMP #377,R3 ;RESULT OK?
BEQ TST71 ;BR IF YES
3$: HALT ;BIC FAILED TO CLEAR HI-BYTE
BR 1$ ;LOCK ON HARD ERROR
```

```
3050
3051
3052
3053
3054 006534
3055 006534 012700 000071
3056 006536 012704 000357
3057 006544 010605
3058 006546 010506
3059 006550 012746 000377
3060 006554 005746
3061 006556 000257
3062
3063 006560 042766 000020 000002
3064
3065 006566 010602
3066 006570 005722
3067 006572 020412
3068 006574 001402
3069
3070 006576 000000
3071 006600 000762
3072
3073 006602 010506
3074
3075
3076
3077
3078 006604
3079 006604 012700 000072
3080 006610 012704 000002
3081 006614 000257
3082
3083 006616 062703 000002
3084
3085 006622 022703 000004
3086 006626 001402
3087
3088 006630 000000
3089 006632 000766
3090
3091
3092
3093
3094 006634
3095 006634 012700 000073
3096 006640 012702 063312
3097 006644 012704 000004
3098 006650 012712 000002
3099 006654 000257
3100
3101 006656 062712 000002
3102
3103 006662 020412
3104 006664 001402
3105

;*****
;TEST 71 BASIC "BIC #N,2(SP)" TEST
;*****
TST71:
MOV #71,R0 ;LOAD R0 WITH TEST NUMBER
MOV #357,R4 ;RESULT S / B = 357
MOV SP,R5 ;SAVE SP
1$: MOV #5,SP ;RESET SP FOR ERROR LOOP
MOV #377,-(SP) ;[DEST] = 377 PUT ON STACK
TST -(SP) ;DECREMENT SP
CCC ;SCOPE SYNC
2$: BIC #20,2(SP) ;TEST THE BIC - CLEAR BIT 4
MOV SP,R2 ;[R2] = DEST ADDR
TST (R2)
CMP R4,(R2) ;RESULT = 357?
BEQ 4$ ;BR IF YES
3$: HALT ;BIC FAILED TO CLR BIT2 OF DEST
BR 1$ ;LOCK ON HARD ERROR
4$: MOV R5,SP

;*****
;TEST 72 BASIC "ADD #N,RN" TEST
;*****
TST72:
MOV #72,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2,R3 ;MAKE [DEST] = 2
CCC ;SCOPE SYNC
2$: ADD #2,R3 ;TEST THE ADD
CMP #4,R3 ;RESULT = 4 ?
BEQ TST73 ;BR IF YES
3$: HALT ;ADD DELIVERED THE WRONG RESULT
BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 73 BASIC "ADD #N,(R)" TEST
;*****
TST73:
MOV #73,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #3,R4 ;RESULT S / B = 4
1$: MOV #3,(R2) ;MAKE [DEST] = 2
CCC ;SCOPE SYNC
2$: ADD #2,(R2) ;TEST THE ADD
CMP R4,(R2) ;RESULT = 4 ?
BEQ TST74 ;BR IF YES
```

```

3106 006666 000000
3107 006670 000767
3108
3109
3110
3111
3112 006672
3113 006672 012700 000074
3114 006672 012704 000002
3115 006702 012702 063314
3116 006706 012705 063312
3117 006713 000257
3118 006714 000257
3119
3120 006716 062765 000002 000002
3121
3122 006724 020412
3123 006726 001402
3124
3125 006730 000000
3126 006732 000765
3127
3128
3129
3130
3131 006734
3132 006734 012700 000075
3133 006740 012704 177400
3134 006744 010605
3135 006746 010602
3136 006750 005742
3137 006752 010506
3138 006754 010446
3139 006756 000257
3140
3141 006760 122726 000000
3142
3143 006764 001402
3144
3145 006766 000000
3146 006770 000770
3147
3148 006772 020506
3149 006774 001402
3150
3151 006776 000000
3152 007000 000764
3153
3154 007002 020412
3155 007004 001402
3156
3157 007006 000000
3158 007010 000760
3159
3160
3161

```

```

3S: HALT ;ADD DELIVERED THE WRONG RESULT
BR ;LOCK ON HARD ERROR

;*****
;TEST 74 BASIC "ADD #N,(R)" TEST
;*****
TST74:
MOV #74,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2,R4 ;RESULT S / B = 2
MOV #MBUF0+2,R2 ;DEST ADDR = MBUF0 + 2
1S: MOV #MBUF0,R5 ;BASE DEST ADDR = MBUF0
CLR (R2) ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC

2S: ADD #2,(R5) ;TEST THE ADD

CMP #4,(R2) ;RESULT = 2 ?
BEQ TST75 ;BR IF YES

3S: HALT ;ADD DELIVERED THE WRONG RESULT
BR ;LOOP ON HARD ERROR

;*****
;TEST 75 BASIC "CMPB #N,(SP)+" TEST
;*****
TST75:
MOV #75,R0 ;LOAD R0 WITH TEST NUMBER
MOV #177400,R4 ;RESULT S / B = 177400
MOV SP,R2 ;SAVE SP
MOV SP,R2 ;SET UP DEST ADDR
TST -(R2) ;R2 CONTAINS DEST ADDR
1S: MOV R5,SP ;RESET SP FOR ERROR LOOP
MOV R4,-(SP) ;MAKE [DEST] = 177400
CCC ;SCOPE SYNC - "Z" = 0

2S: CMPB #0,(SP)+ ;TEST THE CMPB
BEQ 4$ ;BR IF "Z" SET - IT SHOULD BE

3S: HALT ;CMPB FAILED TO SET "Z"
BR ;LOCK ON HARD ERROR

4S: CMP R5,SP ;DID SP GET UPDATED BY 2?
BEQ 6$ ;BR IF YES

5S: HALT ;CMPB FAILED TO UPDATE SP PROPERLY
BR ;LOCK ON HARD ERROR

6S: CMP R4,(R2) ;[DEST] ALTERED?
BEQ TST76 ;BR IF NOT

7S: HALT ;CMPB MODIFIED [DEST]
BR ;LOCK ON HARD ERROR.

;*****
;TEST 76 BASIC "CMPB (RA)+,(RB)+" - SRC AND DEST EVEN

```

```

3162 007012
3163 007012 012700 000076
3164 007016 012704 177777
3165 007022 012702 063324
3166 007026 012705 063330
3167 007032 010203
3168 007034 000257
3169
3170 007036 122523
3171
3172 007040 001402
3173
3174 007042 000000
3175 007044 000770
3176
3177 007046 022703 063325
3178 007052 001402
3179
3180 007054 000000
3181 007056 000763
3182
3183 007060 022705 063331
3184 007064 001402
3185
3186 007066 000000
3187 007070 000756
3188
3189 007072 020412
3190 007074 001403
3191
3192 007076 000000
3193 007100 010412
3194 007102 000751
3195
3196
3197
3198
3199
3200 007104
3201 007110 012700 000077
3202 007110 012704 177777
3203 007110 012702 063324
3204 007110 012705 063327
3205 007112 012703 063325
3206 007130 000257
3207
3208 007132 122523
3209
3210 007134 001402
3211
3212 007136 000000
3213 007140 000767
3214
3215 007142 022703 063326
3216 007146 001402
3217

```

```

;*****
;TEST 76 BASIC "CMPB (RA)+,(RB)+" - SRC AND DEST EVEN
;*****
TST76:
MOV #76,R0 ;LOAD R0 WITH TEST NUMBER
MOV #1,R4 ;RESULT S / B = 177777
MOV #DMTA+2,R2 ;DEST ADDR = DMTA+2
1S: MOV #DMTA+6,R5 ;SRC ADDR = DMTA+6
MOV R2,R3 ;R3 GETS DEST ADDR
CCC ;SCOPE SYNC

2S: CMPB (R5)+,(R3)+ ;TEST THE CMPB
REQ 4$ ;BR IF "Z" = 1 - IT SHOULD BE

3S: HALT ;CMPB FAILED TO SET "Z"
BR ;LOCK ON HARD ERROR

4S: CMP #DMTA+3,R3 ;DID DEST REG GET UPDATED?
BEQ 6$ ;BR IF YES

5S: HALT ;CMPB FAILED TO UPDATE DEST REG
BR ;LOCK ON HARD ERROR

6S: CMP #DMTA+7,R5 ;DID SRC REG GET UPDATED?
BEQ 8$ ;BR IF YES

7S: HALT ;CMPB FAILED TO UPDATE SRC REG
BR ;LOCK ON HARD ERROR

8S: CMP R4,(R2) ;DID [DEST] GET ALTERED?
BEQ TST77 ;BR IF NOT

9S: HALT ;CMPB DELIVERED A RESULT
BR R4,(R2) ;RESTORE [DEST]
;LOCK ON HARD ERROR

;*****
;TEST 77 BASIC "CMPB (RA)+,(RB)+" - SRC AND DEST ODD
;*****
TST77:
MOV #77,R0 ;LOAD R0 WITH TEST NUMBER
MOV #1,R4 ;RESULT S / B = 177777
MOV #DMTA+2,R2 ;DEST ADDR = DMTA+2
1S: MOV #DMTA+5,R5 ;SRC ADDR = DMTA+5
MOV #DMTA+3,R3 ;R3 GETS DEST ADDR+1
CCC ;SCOPE SYNC

2S: CMPB (R5)+,(R3)+ ;TEST THE CMPB
BEQ 4$ ;BR IF "Z" = 1 - IT SHOULD BE

3S: HALT ;CMPB FAILED TO SET "Z"
BR ;LOCK ON HARD ERROR

4S: CMP #DMTA+4,R3 ;DID DEST REG GET UPDATED?
BEQ 6$ ;BR IF YES

```

3218 007150 000000
3219 007152 000762
3220
3221 007154 022705 063330
3222 007160 001402
3223
3224 007162 000000
3225 007164 000755
3226
3227 007166 020412
3228 007170 001403
3229
3230 007172 000000
3231 007174 010412
3232 007176 000750
3233
3234
3235
3236
3237 007200
3238 007200 012700 000100
3239 007204 012704 177400
3240 007210 012702 063326
3241 007214 012705 063330
3242 007220 012703 063327
3243 007224 000257
3244
3245 007226 122523
3246
3247 007230 001402
3248
3249 007232 000000
3250 007234 000767
3251
3252 007236 022703 063330
3253 007242 001402
3254
3255 007244 000000
3256 007246 000762
3257
3258 007250 022705 063331
3259 007254 001402
3260
3261 007256 000000
3262 007260 000755
3263
3264 007262 020412
3265 007264 001403
3266
3267 007266 000000
3268 007270 010412
3269 007272 000750
3270
3271
3272
3273

```
5$: HALT ;CMPB FAILED TO UPDATE DEST REG
    BR ;LOCK ON HARD ERROR
6$: CMP #DWTA+6,R5 ;DID SRC REG GET UPDATED?
    BEQ 8$ ;BR IF YES
7$: HALT ;CMPB FAILED TO UPDATE SRC REG
    BR 1$ ;LOCK ON HARD ERROR
8$: CMP R4,(R2) ;DID [DEST] GET ALTERED?
    BEQ T$T100 ;BR IF NOT
9$: HALT ;CMPB DELIVERED A RESULT
    MOV ;RESTORE [DEST]
    BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 100 BASIC "CMPB (RA)+,(RB)+" - SRC / EVEN,DEST / ODD
;*****
T$T100: MOV #100,R0 ;LOAD R0 WITH TEST NUMBER
        MOV #177400,R4 ;RESULT S / B = 177400
        MOV #DWTA+4,R2 ;DEST ADDR = DWTA+4
1$: MOV #DWTA+6,R5 ;SRC ADDR = DWTA+6
    MOV #DWTA+5,R3 ;R3 GETS DEST ADDR
    CCC ;SCOPE SYNC
2$: CMPB (R5)+,(R3)+ ;TEST THE CMPB
    BEQ 4$ ;BR IF "Z" = 1 - IT SHOULD BE
3$: HALT ;CMPB FAILED TO SET "Z"
    BR 1$ ;LOCK ON HARD ERROR
4$: CMP #DWTA+6,R3 ;DID DEST REG GET UPDATED?
    BEQ 6$ ;BR IF YES
5$: HALT ;CMPB FAILED TO UPDATE DEST REG
    BR 1$ ;LOCK ON HARD ERROR
6$: CMP #DWTA+7,R5 ;DID SRC REG GET UPDATED?
    BEQ 8$ ;BR IF YES
7$: HALT ;CMPB FAILED TO UPDATE SRC REG
    BR 1$ ;LOCK ON HARD ERROR
8$: CMP R4,(R2) ;DID [DEST] GET ALTERED?
    BEQ T$T101 ;BR IF NOT
9$: HALT ;CMPB DELIVERED A RESULT
    MOV ;RESTORE [DEST]
    BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 101 BASIC "CMPB (RA)+,(RB)+" - SRC / ODD,DEST / EVEN
;*****
```

3274 007274 012700 000101
3275 007300 012704 177777
3276 007304 012702 063324
3277 007310 012705 063327
3278 007314 010203
3279 007316 000257
3280
3281 007320 122523
3282
3283 007322 001402
3284
3285 007324 000000
3286 007326 000770
3287
3288 007330 022703 063325
3289 007334 001402
3290
3291 007336 000000
3292 007340 000763
3293
3294 007342 022705 063330
3295 007346 001402
3296
3297 007350 000000
3298 007352 000756
3299
3300 007354 020412
3301 007356 001403
3302
3303 007360 000000
3304 007362 010412
3305 007364 000751
3306
3307
3308
3309
3310
3311 007366 012700 000102
3312 007370 012702 063316
3313 007376 012703 063312
3314 007400 012704 177400
3315 007406 012705 063330
3316 007412 012702 177777
3317
3318 007420 112563 000004
3319
3320 007424 020412
3321 007426 001402
3322
3323 007430 000000
3324 007432 000765
3325
3326 007434 022705 064631
3327 007440 001402

```
T$T101: MOV #101,R0 ;LOAD R0 WITH TEST NUMBER
        MOV #-1,R4 ;RESULT S / B = 177777
        MOV #DWTA+2,R2 ;DEST ADDR = DWTA+2
1$: MOV #DWTA+5,R5 ;SRC ADDR = DWTA+5
    MOV R2,R3 ;R3 GETS DEST ADDR
    CCC ;SCOPE SYNC
2$: CMPB (R5)+,(R3)+ ;TEST THE CMPB
    BEQ 4$ ;BR IF "Z" = 1 - IT SHOULD BE
3$: HALT ;CMPB FAILED TO SET "Z"
    BR 1$ ;LOCK ON HARD ERROR
4$: CMP #DWTA+3,R3 ;DID DEST REG GET UPDATED?
    BEQ 6$ ;BR IF YES
5$: HALT ;CMPB FAILED TO UPDATE DEST REG
    BR 1$ ;LOCK ON HARD ERROR
6$: CMP #DWTA+6,R5 ;DID SRC REG GET UPDATED?
    BEQ 8$ ;BR IF YES
7$: HALT ;CMPB FAILED TO UPDATE SRC REG
    BR 1$ ;LOCK ON HARD ERROR
8$: CMP R4,(R2) ;DID [DEST] GET ALTERED?
    BEQ T$T102 ;BR IF NOT
9$: HALT ;CMPB DELIVERED A RESULT
    MOV ;RESTORE [DEST]
    BR 1$ ;LOCK ON HARD ERROR

;*****
;TEST 102 BASIC "MOV (RA)+,(RB)+" - SRC EVEN / DEST EVEN
;*****
T$T102: MOV #102,R0 ;LOAD R0 WITH TEST NUMBER
        MOV #MBUF1,R2 ;DEST ADDR = MBUF1
        MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0
        MOV #177400,R4 ;RESULT S / B = 177400
1$: MOV #DWTA,R5 ;SRC ADDR = DWTA
    MOV #-1,(R2) ;[DEST] = 177777
    CCC ;SCOPE SYNC
2$: MOV (R5)+,4(R3) ;TEST THE MOV
3$: CMP R4,(R2) ;RESULT OK?
    BEQ 4$ ;BR IF YES
4$: HALT ;MOV DELIVERED WRONG RESULT
    BR 1$ ;LOCK ON HARD ERROR
5$: CMP #DWTA+1,R5 ;DID SRC REG GET INCREMENTED BY +1
    BEQ T$T103 ;BR IF YES
```

007442 000000
007444 000760

007446 012700 000103
007448 012702 063316
007450 012703 063316
007462 012704 000777
007466 012705 064635
007472 012712 177777
007476 000257

007500 112563 000005

007504 020412
007506 001402

007510 000000
007512 000765

007514 022705 064636
007520 001402

007522 000000
007524 000760

007526 012700 000104
007528 012702 063316
007532 012703 063316
007542 012704 000377
007546 012705 064630
007552 012712 177777
007556 000257

007560 112563 000005

007564 020412
007566 001402

007570 000000
007572 000765

007574 022705 064631
007600 001402

007602 000000
007604 000760

```
5$: HALT ;MOVB FAILED TO UPDATE SRC REG
    BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 103 BASIC "MOVB (RA)+,X(RB) - SRC ODD / DEST ODD
;*****
TST103:
MOV #103,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF1,R2 ;DEST ADDR = MBUF1
MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0
MOV #777,R4 ;RESULT S / B = 777
1$: MOV #DBTB+1,R5 ;SRC ADDR = DBTB+1
    MOV #-1,(R2) ;DESTJ = 177777
    CCC ;SCOPE SYNC
2$: MOVB (R5)+,5(R3) ;TEST THE MOVB
    CMP R4,(R2) ;RESULT OK?
    BEQ 4$ ;BR IF YES
3$: HALT ;MOV DELIVERED WRONG RESULT
    BR 1$ ;LOCK ON HARD ERROR
4$: CMP #DBTB+2,R5 ;DID SRC REG GET INCREMENTED BY +1
    BEQ 5$ ;BR IF YES
5$: HALT ;MOVB FAILED TO UPDATE SRC REG
    BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 104 BASIC "MOVB (RA)+,X(RB) - SRC EVEN / DEST ODD
;*****
TST104:
MOV #104,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF1,R2 ;DEST ADDR = MBUF1
MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
1$: MOV #DBTA,R5 ;SRC ADDR = DBTA
    MOV #-1,(R2) ;DESTJ = 177777
    CCC ;SCOPE SYNC
2$: MOVB (R5)+,5(R3) ;TEST THE MOVB
    CMP R4,(R2) ;RESULT OK?
    BEQ 4$ ;BR IF YES
3$: HALT ;MOV DELIVERED WRONG RESULT
    BR 1$ ;LOCK ON HARD ERROR
4$: CMP #DBTA+1,R5 ;DID SRC REG GET INCREMENTED BY +1
    BEQ 5$ ;BR IF YES
5$: HALT ;MOVB FAILED TO UPDATE SRC REG
    BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 105 BASIC "MOVB (RA)+,X(RB) - SRC ODD / DEST EVEN
;*****
```

007606 012700 000105
007608 012702 063316
007612 012703 063316
007616 012704 177401
007626 012705 064635
007632 012712 177777
007636 000257

007640 112563 000004

007644 020412
007646 001402

007650 000000
007652 000765

007654 022705 064636
007660 001402

007662 000000
007664 000760

007666 012700 000106
007668 012702 063316
007672 012704 177401
007676 012705 064032
007706 012703 177777
007710 012712 177777
007714 000257

007716 116523 000002

007722 020412
007724 001402

007726 000000
007730 000766

007732 022703 063313
007736 001402

007740 000000
007742 000761

007744 012700 000107
007750 012702 063312
007754 012704 177401

```
TST105:
MOV #105,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF1,R2 ;DEST ADDR = MBUF1
MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0
MOV #17401,R4 ;RESULT S / B = 177401
1$: MOV #DBTB+1,R5 ;SRC ADDR = DBTB+1
    MOV #-1,(R2) ;DESTJ = 177777
    CCC ;SCOPE SYNC
2$: MOVB (R5)+,4(R3) ;TEST THE MOVB
    CMP R4,(R2) ;RESULT OK?
    BEQ 4$ ;BR IF YES
3$: HALT ;MOV DELIVERED WRONG RESULT
    BR 1$ ;LOCK ON HARD ERROR
4$: CMP #DBTB+2,R5 ;DID SRC REG GET INCREMENTED BY +1
    BEQ 5$ ;BR IF YES
5$: HALT ;MOVB FAILED TO UPDATE SRC REG
    BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 106 BASIC "MOVB 2(RA),(RB)+ TEST - SRC EVEN / DEST EVEN
;*****
TST106:
MOV #106,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177401,R4 ;RESULT S / B = 177401
MOV #DWTB,R5 ;SRC ADDR = DWTB
1$: MOV #R2,R3 ;R3 GETS DEST ADDR
    MOV #-1,(R3) ;DESTJ = 177400
    CCC ;SCOPE SYNC
2$: MOVB 2(R5),(R3)+ ;TEST THE MOVB
    CMP R4,(R2) ;RESULT OK?
    BEQ 4$ ;BR IF YES
3$: HALT ;MOVB DELIVERED WRONG RESULT
    BR 1$ ;LOCK ON HARD ERROR
4$: CMP #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED?
    BEQ 5$ ;BR IF YES
5$: HALT ;MOVB FAILED TO AUTO INCREMENT DEST REG
    BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 107 BASIC "MOVB 2(RA),(RB)+ TEST - SRC ODD / DEST EVEN
;*****
TST107:
MOV #107,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177401,R4 ;RESULT S / B = 177401
```

3442 007760 012705 064634
3443 007764 012703 177777
3444 007766 012713 177777
3445 007772 000257
3446
3447 007774 116523 000001
3448
3449 010000 020412
3450 010002 001402
3451
3452 010004 000000
3453 010006 000766
3454
3455 010010 022703 063313
3456 010014 001402
3457
3458 010016 000000
3459 010020 000761
3460
3461
3462
3463
3464 010022
3465 010022 012700 000110
3466 010026 012702 063313
3467 010032 012704 000777
3468 010036 012705 064032
3469 010042 012703 063313
3470 010046 012712 177777
3471 010052 000257
3472
3473 010054 116523 000002
3474
3475 010060 020412
3476 010062 001402
3477
3478 010064 000000
3479 010066 000765
3480
3481 010070 022703 063314
3482 010074 001402
3483
3484 010076 000000
3485 010100 000760
3486
3487
3488
3489
3490 010102
3491 010106 012700 000111
3492 010106 012702 063313
3493 010112 012704 000777
3494 010116 012705 064634
3495 010122 012703 063313
3496 010126 012712 177777
3497 010132 000257

1\$: MOV #DWB,R5 ;SRC ADDR = DWB
MOV R2,R3 ;R3 GETS DEST ADDR
MOV #-1,(R3) ;DESTJ = 177777
CCC ;SCOPE SYNC
2\$: MOVB 1(R5),(R3)+ ;TEST THE MOVB
CMP R4,(R2) ;RESULT OK?
BEQ 4\$;BR IF YES
3\$: HALT ;MOVB DELIVERED WRONG RESULT
BR 1\$;LOCK ON HARD ERROR
4\$: CMP #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED?
BEQ TST110 ;BR IF YES
5\$: HALT ;MOVB FAILED TO AUTO INCREMENT DEST REG
BR 1\$;LOCK ON HARD ERROR
;*****
;TEST 110 BASIC "MOVB 2(RA),(RB)+ " TEST - SRC EVEN / DEST ODD
;*****
TST110:
MOV #110,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #777,R4 ;RESULT S / B = 777
MOV #DWTB,R5 ;SRC ADDR = DWTB
1\$: MOV #MBUF0+1,R3 ;R3 GETS DEST ADDR
MOV #-1,(R2) ;DESTJ = 177777
CCC ;SCOPE SYNC
2\$: MOVB 2(R5),(R3)+ ;TEST THE MOVB
CMP R4,(R2) ;RESULT OK?
BEQ 4\$;BR IF YES
3\$: HALT ;MOVB DELIVERED WRONG RESULT
BR 1\$;LOCK ON HARD ERROR
4\$: CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED?
BEQ TST111 ;BR IF YES
5\$: HALT ;MOVB FAILED TO AUTO INCREMENT DEST REG
BR 1\$;LOCK ON HARD ERROR
;*****
;TEST 111 BASIC "MOVB 2(RA),(RB)+ " TEST - SRC ODD / DEST ODD
;*****
TST111:
MOV #111,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #777,R4 ;RESULT S / B = 777
MOV #DWB,R5 ;SRC ADDR = DWB
1\$: MOV #MBUF0+1,R3 ;R3 GETS DEST ADDR = MBUF0+1
MOV #-1,(R2) ;DESTJ = 177777
CCC ;SCOPE SYNC

3498 010134 116523 000001
3499
3500 010140 020412
3501 010142 001402
3502
3503 010144 000000
3504 010146 000765
3505
3506 010150 022703 063314
3507 010154 001402
3508
3509 010156 000000
3510 010160 000760
3511
3512
3513
3514 010162
3515 010162 012700 000112
3516 010166 012705 063313
3517 010172 005003
3518 010174 000257
3519
3520 010176 114503
3521
3522 010200 022703 177777
3523 010204 001402
3524
3525 010206 000000
3526 010210 000766
3527
3528 010212 022705 063330
3529 010216 001402
3530
3531 010220 000000
3532 010222 000761
3533
3534
3535 010224
3536 010224 012700 000113
3537 010230 012705 063330
3538 010234 005003
3539 010236 000257
3540
3541 010240 114503
3542
3543 010242 022703 177777
3544 010246 001402
3545
3546 010250 000000
3547 010252 000766
3548
3549 010254 022705 063327
3550 010260 001402

2\$: MOVB 1(R5),(R3)+ ;TEST THE MOVB
CMP R4,(R2) ;RESULT OK?
BEQ 4\$;BR IF YES
3\$: HALT ;MOVB DELIVERED WRONG RESULT
BR 1\$;LOCK ON HARD ERROR
4\$: CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED?
BEQ TST112 ;BR IF YES
5\$: HALT ;MOVB FAILED TO AUTO INCREMENT DEST REG
BR 1\$;LOCK ON HARD ERROR
;*****
;TEST 112 BASIC "MOVB -(RA),RB" TEST - SRC EVEN ADDR
;*****
TST112:
MOV #112,R0 ;LOAD R0 WITH TEST NUMBER
MOV #DWTB+7,R5 ;SRC ADDR = DWTB+7
CLR R3 ;DESTJ = 000000
CCC ;SCOPE SYNC
2\$: MOVB -(R5),R3 ;TEST THE MOVB
CMP #-1,R3 ;RESULT OK?
BEQ 4\$;BR IF YES
3\$: HALT ;MOVB FAILED - WRONG RESULT
BR 1\$;LOCK ON HARD ERROR
4\$: CMP #DWTB+6,R5 ;SRC REG GET DECREMENTED?
BEQ TST113 ;BR IF YES
5\$: HALT ;MOVB FAILED TO UPDATE SRC REG
BR 1\$;LOCK ON HARD ERROR
;*****
;TEST 113 BASIC "MOVB -(RA),RB" TEST - SRC ODD ADDR
;*****
TST113:
MOV #113,R0 ;LOAD R0 WITH TEST NUMBER
MOV #DWTB+6,R5 ;SRC ADDR = DWTB+6
CLR R3 ;DESTJ = 000000
CCC ;SCOPE SYNC
2\$: MOVB -(R5),R3 ;TEST THE MOVB
CMP #-1,R3 ;RESULT OK?
BEQ 4\$;BR IF YES
3\$: HALT ;MOVB FAILED - WRONG RESULT
BR 1\$;LOCK ON HARD ERROR
4\$: CMP #DWTB+5,R5 ;SRC REG GET DECREMENTED?
BEQ TST114 ;BR IF YES


```

3554 010262 000000
3555 010264 000761
3556
3557
3558
3559
3560
3561 010266 012700 000114
3562 010268 010506
3563 010270 012704 177400
3564 010272 010506
3565 010300 012703 064630
3566 010302 012746 177777
3567 010304 010502
3568 010314 005726
3569 010316 000257
3570
3571
3572 010320 112346
3573
3574 010322 022703 064631
3575 010326 001402
3576
3577 010330 000000
3578 010332 000762
3579
3580 010334 020412
3581 010336 001402
3582
3583 010340 000000
3584 010342 000756
3585
3586 010344 020206
3587 010346 001402
3588
3589 010350 000000
3590 010352 000752
3591
3592 010354 010506
3593
3594
3595
3596
3597 010356 012700 000115
3598 010358 010506
3599 010360 012704 177400
3600 010362 010506
3601 010370 012703 064035
3602 010372 012746 177777
3603 010400 010502
3604 010402 005726
3605 010404 000257
3606 010406
3607
3608 010410 112346
3609
    
```

```

5$: HALT ;MOVB FAILED TO UPDATE SRC REG
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 114 BASIC "MOVB (RA)+,-(SP)" TEST - SRC ADDR EVEN
;*****
TST114:
MOV #114,R0 ;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;SAVE SP
MOV #17400,R4 ;RESULT S / B = 177400
R5,SP ;RESET SP FOR ERROR LOOP
1$: MOV #DBTA,R3 ;SRC ADDR = DBTA
MOV #-1,(SP) ;DEST = 177777
MOV SP,R2 ;R2 GETS DEST ADDR
TST (SP)+ ;RESET SP
CCC ;SCOPE SYNC

2$: MOVB (R3)+,-(SP) ;TEST THE MOVB

CMP #DBTA+1,R3 ;DID MOVB INCREMENT SRC REG?
BEQ 4$ ;BR IF YES

3$: HALT ;MOVB FAILED TO UPDATE SRC REG
BR 1$ ;LOCK ON HARD ERROR

4$: CMP R4,(R2) ;RESULT OK?
BEQ 6$ ;BR IF YES

5$: HALT ;MOVB FAILED TO DELIVER CORRECT RESULT
BR 1$ ;LOCK ON HARD ERROR

6$: CMP R2,SP ;DID SP GET PUSHED BY 2 ?
BEQ 8$ ;BR IF YES

7$: HALT ;MOVB FAILED TO PUSH SP PROPERLY
BR 1$ ;LOCK ON HARD ERROR

8$: MOV R5,SP ;RESET SP IN CASE OF ERROR

;*****
;TEST 115 BASIC "MOVB (RA)+,-(SP)" TEST - SRC ADDR ODD
;*****
TST115:
MOV #115,R0 ;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;SAVE SP
MOV #17400,R4 ;RESULT S / B = 177400
R5,SP ;RESET SP FOR ERROR LOOP
1$: MOV #DWTB+3,R3 ;SRC ADDR = DWTB+3
MOV #-1,(SP) ;DEST = 177777
MOV SP,R2 ;R2 GETS DEST ADDR
TST (SP)+ ;RESET SP
CCC ;SCOPE SYNC

2$: MOVB (R3)+,-(SP) ;TEST THE MOVB
    
```

```

3610 010412 022703 064036
3611 010416 001402
3612
3613 010420 000000
3614 010422 000762
3615
3616 010424 020412
3617 010426 001402
3618
3619 010430 000000
3620 010432 000756
3621
3622 010434 020206
3623 010436 001402
3624
3625 010440 000000
3626 010442 000752
3627
3628 010444 010506
3629
3630
3631
3632 010446 012700 000116
3633 010448 012702 063312
3634 010450 012704 000001
3635 010452 012705 064032
3636 010454 005726
3637 010456 000257
3638 010470
3639
3640
    
```

```

CMP #DWTB+4,R3 ;DID MOVB INCREMENT SRC REG?
BEQ 4$ ;BR IF YES

3$: HALT ;MOVB FAILED TO UPDATE SRC REG
BR 1$ ;LOCK ON HARD ERROR

4$: CMP R4,(R2) ;RESULT OK?
BEQ 6$ ;BR IF YES

5$: HALT ;MOVB FAILED TO DELIVER CORRECT RESULT
BR 1$ ;LOCK ON HARD ERROR

6$: CMP R2,SP ;DID SP GET PUSHED BY 2
BEQ 8$ ;BR IF YES

7$: HALT ;MOVB FAILED TO PUSH SP
BR 1$ ;LOCK ON HARD ERROR

8$: MOV R5,SP ;RESET SP IN CASE OF ERROR

;*****
;TEST 116 BASIC "MOVB X(R),@#A" TEST - SRC EVEN / DEST EVEN
;*****
TST116:
MOV #116,R0 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #1,R4 ;RESULT S / B = 1
MOV #DWTB,R5 ;BASE SRC ADDR = DWTB
1$: CLR (R2) ;DEST = 000000
CCC ;SCOPE SYNC
    
```

```

3641 010472 116537 000006 063312 2$: MOVB 6(R5),@#MBUFO ;TEST THE MOVX
3642 010500 020412 ;RESULT OK?
3643 010502 001402 CMQB R4,(R2) ;BR IF YES
3644 ;
3645 ;
3646 010504 000000 3$: HALT ;MOVX DELIVERED WRONG RESULT
3647 010506 000767 BR 1$ ;LOCK ON HARD ERROR
3648 ;
3649 ;
3650 ;
3651 010510 ;
3652 010510 012700 000117 TST117: MOV #117,R0 ;LOAD R0 WITH TEST NUMBER
3653 010514 012702 063312 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
3654 010520 012704 000001 MOV #16,R4 ;RESULT S / B = 1
3655 010524 012705 064634 MOV #DBTB,R5 ;BASE SRC ADDR = DBTB
3656 010530 005012 1$: CLR (R2) ;DESTJ = 000000
3657 010532 000257 CCC ;SCOPE SYNC
3658 ;
3659 010534 116537 000001 063312 2$: MOVX 1(R5),@#MBUFO ;TEST THE MOVX
3660 ;
3661 010542 020412 ;RESULT OK?
3662 010544 001402 CMQB R4,(R2) ;BR IF YES
3663 ;
3664 010546 000000 3$: HALT ;MOVX DELIVERED WRONG RESULT
3665 010550 000767 BR 1$ ;LOCK ON HARD ERROR
3666 ;
3667 ;
3668 ;
3669 010552 ;
3670 010552 012700 000120 TST120: MOV #120,R0 ;LOAD R0 WITH TEST NUMBER
3671 010556 012702 063312 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
3672 010562 012704 000400 MOV #400,R4 ;RESULT S / B = 400
3673 010566 012705 064032 MOV #DBTB,R5 ;BASE SRC ADDR = DBTB
3674 010572 005012 1$: CLR (R2) ;DESTJ = 000000
3675 010574 000257 CCC ;SCOPE SYNC
3676 ;
3677 010576 116537 000006 063313 2$: MOVX 6(R5),@#MBUFO+1 ;TEST THE MOVX
3678 ;
3679 010604 020412 ;RESULT OK?
3680 010606 001402 CMQB R4,(R2) ;BR IF YES
3681 ;
3682 010610 000000 3$: HALT ;MOVX DELIVERED WRONG RESULT
3683 010612 000767 BR 1$ ;LOCK ON HARD ERROR
3684 ;
3685 ;
3686 ;
3687 010614 ;
3688 010614 012700 000121 TST121: MOV #121,R0 ;LOAD R0 WITH TEST NUMBER
3689 ;
3690 010620 032737 000004 063234 -SBTTL USER CONTROLLED BREAKPOINT -- BIT2 ;BREAKPOINT HALT SET ??
3691 010626 001401 BEQ #BIT2,@#BPTLCC ;BR IF NOT
3692 010630 000000 HALT ;BREAK - DEPRESS CONTINUE TO RESTART
3693 010632 012702 063312 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
3694 010636 012704 000400 MOV #400,R4 ;RESULT S / B = 400
3695 010642 012705 064634 MOV #DBTB,R5 ;BASE SRC ADDR = DBTB
3696 010646 005012 1$: CLR (R2) ;DESTJ = 000000

```

```

3697 010650 000257 CCC ;SCOPE SYNC
3698 ;
3699 010652 116537 000001 063313 2$: MOVX 1(R5),@#MBUFO+1 ;TEST THE MOVX
3700 ;
3701 010660 020412 ;RESULT OK?
3702 010662 001402 CMQB R4,(R2) ;BR IF YES
3703 ;
3704 010664 000000 3$: HALT ;MOVX DELIVERED WRONG RESULT
3705 010666 000767 BR 1$ ;LOCK ON HARD ERROR
3706 ;
3707 ;
3708 ;
3709 ;
3710 010670 ;
3711 010670 012700 000122 TST122: MOV #122,R0 ;LOAD R0 WITH TEST NUMBER
3712 010674 000257 CCC ;CLEAR ALL FLAGS
3713 ;
3714 010676 001404 2$: BEQ 3$ ;NO BR SHOULD OCCUR-FLAG=0
3715 010700 100403 BMI 3$ ;NO BR SHOULD OCCUR-FLAG=0
3716 010702 102402 BVS 3$ ;NO BR SHOULD OCCUR-FLAG=0
3717 010704 103401 BCS 3$ ;NO BR SHOULD OCCUR-FLAG=0
3718 010706 000402 BR TST123 ;GO TO NEXT TEST
3719 ;
3720 010710 000000 3$: HALT ;ONE OF ABOVE BR'S FAILED
3721 010712 000770 BR 1$ ;ERROR LOOP RETURN
3722 ;
3723 ;
3724 ;
3725 ;
3726 ;
3727 ;
3728 ;
3729 ;
3730 010714 ;
3731 010714 012700 000123 TST123: MOV #123,R0 ;LOAD R0 WITH TEST NUMBER
3732 010720 000277 SCC ;MAKE N:C = 1111
3733 ;
3734 010722 001402 21$: BEQ 22$ ;TEST THE BEQ-IT SHOULD BR
3735 ;
3736 010724 000000 3$: HALT ;BEQ FAILED
3737 010726 000774 BR 1$ ;ERROR LOOP RETURN
3738 ;
3739 ;
3740 010730 100402 22$: BMI 23$ ;TEST THE BMI-IT SHOULD BR
3741 ;
3742 010732 000000 5$: HALT ;BMI FAILED
3743 010734 000771 BR 1$ ;ERROR LOOP RETURN
3744 ;
3745 010736 102402 23$: BVS 24$ ;TEST THE BVS-IT SHOULD BR
3746 ;
3747 010740 000000 7$: HALT ;BVS FAILED
3748 010742 000766 BR 1$ ;ERROR LOOP RETURN
3749 ;
3750 ;
3751 010744 ;
3752 010744 103402 24$: BCS TST124 ;TEST THE BCS-IT SHOULD BR
3753 ;
3754 010746 000000 9$: HALT ;BCS FAILED
3755 010750 000763 BR 1$ ;ERROR LOOP RETURN
3756 ;
3757 ;
3758 ;
3759 ;
3760 ;
3761 ;
3762 ;
3763 ;
3764 ;
3765 ;
3766 ;
3767 ;
3768 ;
3769 ;
3770 ;
3771 ;
3772 ;
3773 ;
3774 ;
3775 ;
3776 ;
3777 ;
3778 ;
3779 ;
3780 ;
3781 ;
3782 ;
3783 ;
3784 ;
3785 ;
3786 ;
3787 ;
3788 ;
3789 ;
3790 ;
3791 ;
3792 ;
3793 ;
3794 ;
3795 ;
3796 ;
3797 ;
3798 ;
3799 ;
3800 ;
3801 ;
3802 ;
3803 ;
3804 ;
3805 ;
3806 ;
3807 ;
3808 ;
3809 ;
3810 ;
3811 ;
3812 ;
3813 ;
3814 ;
3815 ;
3816 ;
3817 ;
3818 ;
3819 ;
3820 ;
3821 ;
3822 ;
3823 ;
3824 ;
3825 ;
3826 ;
3827 ;
3828 ;
3829 ;
3830 ;
3831 ;
3832 ;
3833 ;
3834 ;
3835 ;
3836 ;
3837 ;
3838 ;
3839 ;
3840 ;
3841 ;
3842 ;
3843 ;
3844 ;
3845 ;
3846 ;
3847 ;
3848 ;
3849 ;
3850 ;
3851 ;
3852 ;
3853 ;
3854 ;
3855 ;
3856 ;
3857 ;
3858 ;
3859 ;
3860 ;
3861 ;
3862 ;
3863 ;
3864 ;
3865 ;
3866 ;
3867 ;
3868 ;
3869 ;
3870 ;
3871 ;
3872 ;
3873 ;
3874 ;
3875 ;
3876 ;
3877 ;
3878 ;
3879 ;
3880 ;
3881 ;
3882 ;
3883 ;
3884 ;
3885 ;
3886 ;
3887 ;
3888 ;
3889 ;
3890 ;
3891 ;
3892 ;
3893 ;
3894 ;
3895 ;
3896 ;
3897 ;
3898 ;
3899 ;
3900 ;
3901 ;
3902 ;
3903 ;
3904 ;
3905 ;
3906 ;
3907 ;
3908 ;
3909 ;
3910 ;
3911 ;
3912 ;
3913 ;
3914 ;
3915 ;
3916 ;
3917 ;
3918 ;
3919 ;
3920 ;
3921 ;
3922 ;
3923 ;
3924 ;
3925 ;
3926 ;
3927 ;
3928 ;
3929 ;
3930 ;
3931 ;
3932 ;
3933 ;
3934 ;
3935 ;
3936 ;
3937 ;
3938 ;
3939 ;
3940 ;
3941 ;
3942 ;
3943 ;
3944 ;
3945 ;
3946 ;
3947 ;
3948 ;
3949 ;
3950 ;
3951 ;
3952 ;
3953 ;
3954 ;
3955 ;
3956 ;
3957 ;
3958 ;
3959 ;
3960 ;
3961 ;
3962 ;
3963 ;
3964 ;
3965 ;
3966 ;
3967 ;
3968 ;
3969 ;
3970 ;
3971 ;
3972 ;
3973 ;
3974 ;
3975 ;
3976 ;
3977 ;
3978 ;
3979 ;
3980 ;
3981 ;
3982 ;
3983 ;
3984 ;
3985 ;
3986 ;
3987 ;
3988 ;
3989 ;
3990 ;
3991 ;
3992 ;
3993 ;
3994 ;
3995 ;
3996 ;
3997 ;
3998 ;
3999 ;
4000 ;

```

```

3753
3754 010752
3755 010752 012700 000124
3756
3757 010756 000262
3758
3759 010760 102001
3760 010762 000402
3761
3762 010764 000000
3763 010766 000773
3764
3765
3766
3767
3768 010770
3769 010770 012700 000125
3770
3771 010774 000242
3772
3773 010776
3774 010776 102002
3775
3776 011000 000000
3777 011002 000774
3778
3779
3780
3781
3782 011004
3783 011004 012700 000126
3784
3785 011010 000257
3786
3787 011012
3788 011012 002002
3789
3790 011014 000000
3791 011016 000774
3792
3793
3794
3795
3796 011020
3797 011020 012700 000127
3798
3799 011024 000257
3800 011026 000262
3801
3802 011030 002001
3803 011032 000402
3804
3805 011034 000000
3806 011036 000772
3807
3808

```

```

*****
TST124:
MOV #124,R0 ;;LOAD R0 WITH TEST NUMBER
1$: SEV ;;MAKE V=1
2$: BVC 3$ ;;TEST THE BVC-IT SHOULDN'T BR
BR TST125 ;;GO TO NEXT TEST
3$: HALT 1$ ;;BVC FAILED
BR ;;ERROR LOOP RETURN
*****
;*TEST 125 BASIC BVC TEST WITH V=0
*****
TST125:
MOV #125,R0 ;;LOAD R0 WITH TEST NUMBER
1$: CLV ;;MAKE V=0
2$: BVC TST126 ;;TEST THE BVC-IT SHOULD BR
3$: HALT 1$ ;;BVC FAILED
BR ;;ERROR LOOP RETURN
*****
;*TEST 126 BASIC BGE TEST WITH N,V = 00
*****
TST126:
MOV #126,R0 ;;LOAD R0 WITH TEST NUMBER
1$: CCC ;;MAKE N:C = 0000
2$: BGE TST127 ;;TEST THE BGE-IT SHOULD BR
3$: HALT 1$ ;;BGE FAILED
BR ;;ERROR LOOP RETURN
*****
;*TEST 127 BASIC BGE TEST WITH N,V = 01
*****
TST127:
MOV #127,R0 ;;LOAD R0 WITH TEST NUMBER
1$: CCC SEV ;;CLEAR FLAGS
;MAKE N,V = 01
2$: BGE 3$ ;;TEST THE BGE-IT SHOULDN'T BR
BR TST130 ;;GO TO NEXT TEST
3$: HALT 1$ ;;BGE FAILED
BR ;;ERROR LOOP RETURN
*****
;

```

```

3809
3810
3811 011040
3812 011040 012700 000130
3813
3814 011044 000257
3815 011046 000270
3816
3817 011050 002001
3818 011052 000402
3819
3820 011054 000000
3821 011056 000772
3822
3823
3824
3825
3826 011060
3827 011060 012700 000131
3828
3829 011064 000257
3830 011066 000272
3831
3832 011070
3833 011070 002002
3834
3835 011072 000000
3836 011074 000773
3837
3838
3839
3840
3841 011076
3842 011076 012700 000132
3843
3844 011102 000257
3845
3846 011104 002401
3847 011106 000402
3848
3849
3850 011110 000000
3851 011112 000773
3852
3853
3854
3855
3856 011114
3857 011114 012700 000133
3858
3859 011120 000257
3860 011122 000262
3861
3862 011124
3863 011124 002402
3864 011126 000000

```

```

*****
;*TEST 130 BASIC BGE TEST WITH N,V = 10
*****
TST130:
MOV #130,R0 ;;LOAD R0 WITH TEST NUMBER
1$: CCC SEN ;;CLEAR FLAGS
;MAKE N,V = 10
2$: BGE 3$ ;;TEST THE BGE-IT SHOULDN'T BR
BR TST131 ;;GO TO NEXT TEST
3$: HALT 1$ ;;BGE FAILED
BR ;;ERROR LOOP RETURN
*****
;*TEST 131 BASIC BGE TEST WITH N,V = 11
*****
TST131:
MOV #131,R0 ;;LOAD R0 WITH TEST NUMBER
1$: CCC 272 ;;CLEAR FLAGS
;MAKE N,V = 11
2$: BGE TST132 ;;TEST THE BGE-IT SHOULD BR
3$: HALT 1$ ;;BGE FAILED
BR ;;ERROR LOOP RETURN
*****
;*TEST 132 BASIC BLT TEST WITH N,V = 00
*****
TST132:
MOV #132,R0 ;;LOAD R0 WITH TEST NUMBER
1$: CCC ;;CLEAR FLAGS
2$: BLT 3$ ;;TEST THE BLT-IT SHOULDN'T BR
BR TST133 ;;GO TO NEXT TEST
3$: HALT 1$ ;;BLT FAILED
BR ;;ERROR LOOP RETURN
*****
;*TEST 133 BASIC BLT TEST WITH N,V = 01
*****
TST133:
MOV #133,R0 ;;LOAD R0 WITH TEST NUMBER
1$: CCC SEV ;;CLEAR FLAGS
;MAKE N,V = 01
2$: BLT TST134 ;;TEST THE BLT-IT SHOULD BR
3$: HALT ;;BLT FAILED

```

3865 011130 000773
3866
3867
3868
3869
3870 011132
3871 011132 012700 000134
3872
3873 011136 000257
3874 011140 000270
3875
3876 011142
3877 011142 002402
3878
3879 011144 000000
3880 011146 000773
3881
3882
3883
3884
3885 011150
3886 011150 012700 000135
3887
3888 011154 000257
3889 011156 000272
3890
3891 011160 002401
3892 011162 000402
3893
3894 011164 000000
3895 011166 000772
3896
3897
3898
3899
3900 011170
3901 011170 012700 000136
3902
3903 011174 000257
3904 011176 000266
3905
3906 011200 003001
3907 011202 000402
3908
3909 011204 000000
3910 011206 000772
3911
3912
3913
3914
3915 011210
3916 011210 012700 000137
3917
3918 011214 000257
3919 011216 000262
3920

```
BR 1$ ;ERROR LOOP RETURN
;*****
;TEST 134 BASIC BLT TEST WITH N,V = 10
;*****
TST134: MOV #134,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC ;CLEAR FLAGS
SEN ;SET N - N,V = 10
2$:
3$: BLT TST135 ;TEST THE BLT-IT SHOULD BR
HALT 1$ ;BLT FAILED
BR ;ERROR LOOP RETURN
;*****
;TEST 135 BASIC BLT TEST WITH N,V = 11
;*****
TST135: MOV #135,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC ;CLEAR FLAGS
272 ;MAKE N,V = 11
2$: BLT 3$ ;TEST THE BLT-IT SHOULDN'T BR
BR TST136 ;GO TO NEXT TEST
3$: HALT ;BLT FAILED
BR 1$ ;ERROR LOOP RETURN
;*****
;TEST 136 BASIC BGT TEST WITH Z = 1 AND N,V = 01
;*****
TST136: MOV #136,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC ;CLEAR FLAGS
266 ;SET Z AND V
2$: BGT 3$ ;TEST THE BGT-IT SHOULDN'T BR
BR TST137 ;GO TO NEXT TEST
3$: HALT ;BGT FAILED
BR 1$ ;ERROR LOOP RETURN
;*****
;TEST 137 BASIC BGT TEST WITH Z = 0 AND N,V = 01
;*****
TST137: MOV #137,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC ;CLEAR FLAGS
SEV ;SET V
```

3921 011220 003001
3922 011222 000402
3923
3924 011224 000000
3925 011226 000772
3926
3927
3928
3929
3930 011230
3931 011230 012700 000140
3932
3933 011234 000257
3934 011236 000264
3935
3936 011240 003001
3937 011242 000402
3938
3939 011244 000000
3940 011246 000772
3941
3942
3943
3944
3945
3946 011250
3947 011250 012700 000141
3948 011254 000257
3949
3950 011256
3951 011258 003002
3952
3953 011260 000000
3954 011262 000774
3955
3956
3957
3958
3959 011264
3960 011264 012700 000142
3961
3962 011270 000257
3963 011272 000266
3964
3965 011274 003001
3966 011276 000402
3967
3968 011300 000000
3969 011302 000772
3970
3971
3972
3973
3974
3975 011304
3976 011304 012700 000143

```
2$: BGT 3$ ;TEST THE BGT-IT SHOULD NOT BR
BR TST140 ;GO TO SCOPE LOOP EXIT
3$: HALT ;BGT FAILED
BR 1$ ;ERROR LOOP RETURN
;*****
;TEST 140 BASIC BGT TEST WITH Z = 1 AND N,V = 00
;*****
TST140: MOV #140,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC ;CLEAR FLAGS
SEZ ;SET Z
2$: BGT 3$ ;TEST THE BGT-IT SHOULD NOT BR
BR TST141 ;GO TO SCOPE LOOP EXIT
3$: HALT ;BGT FAILED
BR 1$ ;ERROR LOOP RETURN
;*****
;TEST 141 BASIC BGT TEST WITH Z = 0 AND N,V = 00
;*****
TST141: MOV #141,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC ;CLEAR FLAGS
2$: BGT TST142 ;TEST THE BGT - IT SHOULD BR
3$: HALT ;BGT FAILED
BR 1$ ;ERROR LOOP RETURN
;*****
;TEST 142 BASIC BGT TEST WITH Z = 1 AND N,V = 01
;*****
TST142: MOV #142,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC ;CLEAR FLAGS
266 ;MAKE N,V = 01 AND Z = 1
2$: BGT 3$ ;TEST THE BGT-IT SHOULDN'T BR
BR TST143 ;GO TO NEXT TEST
3$: HALT ;BGT FAILED
BR 1$ ;ERROR LOOP RETURN
;*****
;TEST 143 BASIC BGT TEST WITH Z = 1 AND N,V = 10
;*****
TST143: MOV #143,R0 ;LOAD RO WITH TEST NUMBER
```

```

3977 011310 000257
3978 011312 000274
3979
3980 011314 003001
3981 011316 000402
3982
3983 011320 000000
3984 011322 000772
3985
3986
3987
3988
3989
3990 011324 012700 000144
3991
3992 011330 000257
3993 011332 000276
3994
3995 011334 003001
3996 011336 000402
3997
3998 011340 000000
3999 011342 000772
4000
4001
4002
4003
4004 011344 012700 000145
4005 011344
4006
4007 011350 000257
4008 011352 000272
4009
4010 011354 003002
4011 011354
4012
4013 011356 000000
4014 011360 000773
4015
4016
4017
4018
4019 011362 012700 000146
4020 011362
4021 011366 000257
4022
4023 011370 003002
4024 011370 101002
4025
4026 011372 000000
4027 011374 000774
4028
4029
4030
4031
4032

```

```

1$: CCC 274 ;CLEAR FLAGS
;MAKE Z = 1 AND N,V = 10
2$: BGT 3$ ;TEST THE BGT-IT SHOULDN'T BR
BR TST144 ;GO TO NEXT TEST
3$: HALT 1$ ;BGT FAILED
BR ;ERROR LOOP RETURN
;*****
;TEST 144 BASIC BGT TEST WITH Z = 1 AND N,V = 11
;*****
TST144: MOV #144,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC 276 ;CLEAR FLAGS
;MAKE Z = 1 AND N,V = 11
2$: BGT 3$ ;TEST THE BGT-IT SHOULD NOT BR
BR TST145 ;GO TO NEXT TEST
3$: HALT 1$ ;BGT FAILED
BR ;ERROR LOOP RETURN
;*****
;TEST 145 BASIC BGT TEST WITH Z=0 AND N,V=11
;*****
TST145: MOV #145,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC 272 ;CLEAR FLAGS
;MAKE N:C=1010
2$: BGT TST146 ;TEST THE BGT - IT SHOULD BR
3$: HALT 1$ ;BGT FAILED
BR ;ERROR LOOP RETURN
;*****
;TEST 146 BASIC BHI TEST WITH Z,C = 00
;*****
TST146: MOV #146,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC ;MAKE Z,C = 00
2$: BHI TST147 ;TEST THE BHI-IT SHOULD BR
3$: HALT 1$ ;BHI FAILED
BR ;ERROR LOOP RETURN
;*****
;TEST 147 BASIC BHI TEST WITH Z,C = 01
;*****

```

```

4033 011376 012700 000147
4034 011376
4035
4036 011402 000257
4037 011404 000261
4038
4039 011406 101001
4040 011410 000402
4041
4042 011412 000000
4043 011414 000772
4044
4045
4046
4047
4048 011416 012700 000150
4049 011416
4050
4051 011422 000257
4052 011424 000264
4053
4054 011426 101001
4055 011430 000402
4056
4057 011432 000000
4058 011434 000772
4059
4060
4061
4062
4063 011436 012700 000151
4064 011436
4065
4066 011442 000257
4067 011444 000265
4068
4069 011446 101001
4070 011450 000402
4071
4072 011452 000000
4073 011454 000772
4074
4075
4076
4077
4078 011456 012700 000152
4079 011456 012704 177776
4080 011456 012704 000002
4081 011456 000257
4082 011456 000257
4083 011474 000266
4084
4085 011476 005403
4086
4087 011500 100003
4088 011502 001402

```

```

TST147: MOV #147,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC SEC ;CLEAR FLAGS
;MAKE Z,C = 01
2$: BHI 3$ ;TEST THE BHI-IT SHOULD NOT BR
BR TST150 ;GO TO NEXT TEST
3$: HALT 1$ ;BHI FAILED
BR ;ERROR LOOP RETURN
;*****
;TEST 150 BASIC BHI TEST WITH Z,C = 10
;*****
TST150: MOV #150,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC SEZ ;CLEAR FLAGS
;MAKE Z,C = 10
2$: BHI 3$ ;TEST THE BHI-IT SHOULD NOT BR
BR TST151 ;GO TO NEXT TEST
3$: HALT 1$ ;BHI FAILED
BR ;ERROR LOOP RETURN
;*****
;TEST 151 BASIC BHI TEST WITH Z,C = 11
;*****
TST151: MOV #151,R0 ;LOAD RO WITH TEST NUMBER
1$: CCC 265 ;CLEAR FLAGS
;MAKE Z,C = 11
2$: BHI 3$ ;TEST THE BHI-IT SHOULDN'T BR
BR TST152 ;GO TO NEXT TEST
3$: HALT 1$ ;BHI FAILED
BR ;ERROR LOOP RETURN
;*****
;TEST 152 BASIC NEG MODE 0 TEST : (DEST) GT 0
;*****
TST152: MOV #152,R0 ;LOAD RO WITH TEST NUMBER
MOV #2,R4 ;RESULT S / B = 177776
MOV #2,R3 ;INITIAL IDESTJ = 2
CCC ;CLEAR FLAGS
266 ;MAKE N:C = 0110
2$: NEG R3 ;TEST THE NEG
3$: BPL 3$ ;DID N:C = 1001?
BEQ 3$

```

```

4089 011504 102401
4090 011506 103402
4091
4092 011510 000000
4093 011512 000765
4094
4095 011514 020304
4096 011516 001402
4097
4098 011520 000000
4099 011522 000761
4100
4101
4102
4103
4104 011524
4105 011524 012700 000153
4106 011530 012704 000002
4107 011534 012702 063312
4108 011540 012712 000004
4109 011544 000257
4110
4111 011546 162737 000002 063312 2$: SUB #2,@#MBUF0 ;TEST THE SUB
4112
4113 011554 020412
4114 011556 011403
4115 011560 011403
4116 011562 000000
4117 011564 000765
4118
4119
4120
4121
4122 011566
4123 011566 012700 000154 063312
4124 011572 012737 000002
4125 011600 012703 000004
4126 011604 000257
4127
4128 011606 163703 063312 2$: SUB @#MBUF0,R3 ;TEST THE SUB
4129
4130 011612 020403
4131 011614 001402
4132
4133 011616 000000
4134 011620 000767
4135
4136
4137
4138
4139 011622
4140 011622 012700 000155
4141 011626 010605
4142 011630 010506
4143 011632 012703 011552
4144 011636 012746 177777

```

```

4145 011642 000277
4146
4147 011644 000203 2$: RTS R3 ;TEST THE RTS - GO TO 4$
4148
4149 011646 000000 3$: HALT BR 1$ ;RTS FAILED TO LOAD THE PC
4150 011650 000767 ;LOCK ON ERROR
4151
4152 011652 100003 4$: BPL 5$ ;N:C = 1111 ?
4153 011654 001002 BNE 6$
4154 011656 102001 BVC 6$
4155 011660 103402 BCS 6$
4156
4157 011662 000000 5$: HALT BR 1$ ;RTS ALTERED CODES - CLEARED ONE
4158 011664 000761 ;LOCK ON ERROR
4159
4160 011666 020327 177777 6$: CMP R3,#-1 ;DID R3 GET LOADED FROM STACK ?
4161 011672 001402 BEQ 8$ ;BR IF YES
4162
4163 011674 000000 7$: HALT BR 1$ ;RTS FAILED TO LOAD REG
4164 011676 000754 ;LOCK ON ERROR
4165
4166 011700 020506 8$: CMP R5,SP ;DID RTS POP THE STACK POINTER ?
4167 011702 001402 BEQ T$T156 ;BR IF YES
4168
4169 011704 000000 9$: HALT BR 1$ ;RTS FAILED TO POP SP
4170 011706 000750 ;LOCK ON ERROR
4171
4172
4173
4174
4175 011710
4176 011710 012700 000156
4177 011714 010605
4178 011716 010506 011734
4179 011720 012746
4180 011724 000257
4181
4182 011726 000207 2$: RTS PC ;TEST THE RTS - GO TO 4$
4183
4184 011730 000000 3$: HALT BR 1$ ;RTS FAILED TO LOAD PC
4185 011732 000771 ;LOCK ON HARD ERROR
4186
4187 011734 020605 4$: CMP SP,R5 ;DID SP GET POPPED ?
4188 011736 001402 BEQ T$T157 ;BR IF YES
4189
4190 011740 000000 5$: HALT BR 1$ ;RTS FAILED TO UPDATE SP
4191 011742 000765 ;LOCK ON HARD ERROR
4192
4193
4194
4195
4196 011744
4197 011744 012700 000157
4198
4199 011750 032737 000010 063234 -SBTTL USER CONTROLLED BREAKPOINT -- BIT3
4200 011756 001401 BIT #BIT3,@#BPTLOC ;BREAKPOINT HALT SET ??
;BR IF NOT

```

```

4201 011760 000000          HALT          ;BREAK - DEPRESS CONTINUE TO RESTART
4202 011760 010605          MOV           ;SAVE ORIGINAL SP
4203 011760 010506          MOV           ;SAVE SP FOR ERROR LOOP
4204 011766 000257          CCC          ;SCOPE SYNC
4205
4206 011770 004737 012000    2$:   JSR           PC,@#4$ ;TEST THE JSR - GO TO 4$
4207
4208 011774 000000          HALT          ;JSR FAILED TO LOAD PC
4209 011776 000772          BR           1$ ;LOCK ON HARD ERROR
4210
4211 012000 022726 011774    4$:   CMP           #3$, (SP)+ ;DID JSR SAVE OLD PC ON STACK ?
4212 012004 001402          BEQ          TST160 ;;BR IF YES
4213
4214 012006 000000          HALT          ;JSR FAILED TO SAVE OLD PC
4215 012010 000765          BR           1$ ;LOCK ON HARD ERROR
4216
;*****
;TEST 160 BASIC "RTI" TEST - N:C=0000
;*****
4217
4218
4219
4220 012012 000000          TST160:
4221 012012 012700 000160    1$:   MOV           #160,R0 ;;LOAD R0 WITH TEST NUMBER
4222 012016 010605          MOV           SP,R5 ;;SAVE THE SP
4223 012020 010506          MOV           R5,SP ;;RESET THE SP FOR ERROR LOOP
4224 012022 012746 000357    MOV           #357,-(SP) ;NEW PSW = 357
4225 012022 012749 012949    MOV           #357,-(SP) ;NEW PC = 4$
4226 012032 005037 177776    CLR          @#PSW ;MAKE [PSW] = 000
4227 012036 000257          CCC          ;MAKE N:C=0000
4228
4229 012040 000002          2$:   RTI           ;TEST THE RTI - GO TO 4$
4230
4231 012042 000000          HALT          ;RTI FAILED TO LOAD PC
4232 012044 000765          BR           1$ ;LOOP ON HARD ERROR
4233
4234 012046 013702 177776    4$:   MOV           @#PSW,R2 ;SAVE THE [PSW] IN R2
4235 012052 022702 000357    CMP           #357,R2 ;WAS [PSW] = 357 ?
4236 012056 001404          BEQ          6$ ;BR IF YES
4237
4238 012060 010237 177776    5$:   MOV           R2,@#PSW ;RESTORE THE ERROR PSW
4239 012064 000000          HALT          ;RTI FAILED TO LOAD PSW
4240 012066 000754          BR           1$ ;LOCK ON HARD ERROR
4241
4242 012070 020605          6$:   CMP           SP,R5 ;DID SP GET UPDATED OK ?
4243 012072 001402          BEQ          TST161 ;;BR IF YES
4244
4245 012074 000000          HALT          ;RTI FAILED TO UPDATE THE SP
4246 012076 000750          BR           1$ ;LOCK ON HARD ERROR
4247
;*****
;TEST 161 BASIC "RTI" TEST WITH N:C=1111
;*****
4248
4249
4250
4251 012100 012700 000161    TST161:
4252 012100 012700          MOV           #161,R0 ;;LOAD R0 WITH TEST NUMBER
4253 012104 010605          MOV           SP,R5 ;;SAVE THE SP IN R5
4254 012106 010506          MOV           R5,SP ;;RESET SP FOR ERROR LOOP
4255 012108 005046          CLR          -(SP) ;NEW PSW = 000000
4256 012112 012746 012130    MOV           #4$,-(SP) ;NEW PC = 4$

```

```

4257 012116 012737 000357 177776    MOV           #357,@#PSW ;MAKE OLD PSW = 357
4258 012124 000240          NOP          ;SCOPE SYNC
4259
4260 012126 000002          2$:   RTI           ;TEST THE RTI - GO TO 4$
4261
4262 012130 013702 177776    4$:   MOV           @#PSW,R2 ;GET THE PSW
4263 012134 022702 000000    CMP           #0,R2 ;WAS [PSW]=000
4264 012140 001404          BEQ          TST162 ;;BR IF YES
4265
4266 012142 010237 177776    3$:   MOV           R2,@#PSW ;RESTORE ERROR PSW
4267 012146 000000          HALT          ;RTI FAILED TO CLEAR PSW
4268 012150 000756          BR           1$ ;LOCK ON HARD ERROR
4269
;*****
;TEST 162 BASIC "IOT" TEST -VERIFY LOADING PSW WITH 357
;*****
4270
4271
4272
4273 012152 012700 000162    TST162:
4274 012152 010605          MOV           #162,R0 ;;LOAD R0 WITH TEST NUMBER
4275 012156 010605          MOV           SP,R5 ;;SAVE THE SP
4276 012160 010506          MOV           R5,SP ;;RESET SP FOR ERROR LOOP
4277 012162 012737 012220 000020    MOV           #357,@#20 ;SET UP IOT VECTOR
4278 012162 012737 000357 000022    MOV           #357,@#22
4279 012176 012766 177776    MOV           #1,-2(SP) ;IOT SHOULD CHANGE -1 TO 0
4280 012204 005037 177776    MOV           @#PSW ;MAKE [PSW] = 000
4281 012210 000257          CLR          @#PSW ;SCOPE SYNC
4282
4283 012212 000004          2$:   IOT           ;TEST THE IOT
4284
4285 012214 000000          HALT          ;IOT FAILED TO LOAD PC
4286 012216 000760          BR           1$ ;LOCK ON HARD ERROR
4287
4288 012220 013702 177776    4$:   MOV           @#PSW,R2 ;GET THE PSW
4289 012224 022702 000357    CMP           #357,R2 ;DID IOT LOAD A 357 ?
4290 012230 001404          BEQ          6$ ;BR IF YES
4291
4292 012232 010237 177776    5$:   MOV           R2,@#PSW ;RESTORE ERROR PSW
4293 012236 000000          HALT          ;IOT FAILED TO LOAD PSW
4294 012240 000747          BR           1$ ;LOCK ON HARD ERROR
4295
4296 012242 022726 012214    6$:   CMP           #3$, (SP)+ ;DID IOT SAVE OLD PC ?
4297 012246 001404          BEQ          8$ ;BR IF YES
4298
4299 012250 010237 177776    7$:   MOV           R2,@#PSW ;RESTORE ERROR PSW
4300 012254 000000          HALT          ;IOT FAILED TO SAVE OLD PC
4301 012256 000740          BR           1$ ;LOCK ON HARD ERROR
4302
4303 012260 005726          8$:   TST          (SP)+ ;DID IOT SAVE OLD PSW ?
4304 012262 001404          BEQ          TST163 ;;BR IF YES
4305
4306 012264 010237 177776    9$:   MOV           R2,@#PSW ;RESTORE ERROR PSW
4307 012266 000000          HALT          ;IOT FAILED TO SAVE OLD PSW
4308 012272 000732          BR           1$ ;LOCK ON HARD ERROR
4309
;*****
;TEST 163 BASIC "IOT" TEST - VERIFY LINKAGE TO SCOPE SERVICE
;*****
4310
4311
4312

```

```

4313 012274 012700 000163
4314 012274 010605
4315 012300 010506
4316 012300 010506
4317 012300 005037 063244
4318 012310 012737 061612 000020
4319 012316 005037 000022
4320 012322 000257
4321
4322 012324 000004
4323
4324 012326 005137 063244
4325 012332 001402
4326
4327 012334 000000
4328 012336 000761
4329
4330 012340 010506
4331
4332
4333
4334
4335 012342 012700 000164
4336 012342 010605
4337 012350 010506
4338 012350 012737 012410 000020
4339 012350 005037 000022
4340 012366 012766 177777 177776
4341 012374 005037 177776
4342 012400 000257
4343
4344 012402 000004
4345
4346 012404 000000
4347 012406 000760
4348
4349 012410 013702 177776
4350 012414 013702 000357
4351 012420 001404
4352
4353 012422 010237 177776
4354 012426 000000
4355 012430 000747
4356
4357 012432 022726 012404
4358 012436 001404
4359
4360 012440 010237 177776
4361 012444 000000
4362 012446 000740
4363
4364 012450 005726
4365 012452 001404
4366
4367 012454 010237 177776
4368 012460 000000

```

```

TST163:
MOV #163,R0 ;LOAD R0 WITH TEST NUMBER
SP,R5 ;SAVE THE SP
R5,SP ;RESET SP FOR ERROR LOOP
1$: CLR @SCOFLG ;TRAP SERVICE WILL COM "SCOFLG"
MOV #SCOPEA,@#20 ;SET UP IOT VECTOR
CLR @#22
CCC ;SCOPE SYNC
2$: SCOPE ;TEST THE IOT
COM @SCOFLG ;SCOFLG SHOULD BECOME 000000
BEQ 4$ ;BR IF IT DID
3$: HALT ;IOT FAILED TO LINK TO SCOPE SERVICE
BR 1$ ;LOCK ON HARD ERROR
4$: MOV R5,SP ;RESET SP IN CASE OF ERROR
;*****
;TEST 164 BASIC "IOT" TEST - VERIFY LOADING PSW WITH 357
;*****
TST164:
MOV #164,R0 ;LOAD R0 WITH TEST NUMBER
SP,R5 ;SAVE THE SP
R5,SP ;RESET SP FOR ERROR LOOP
1$: CLR @#20 ;SET UP IOT VECTOR
MOV #357,@#22
MOV #-1,@#2(SP) ;IOT SHOULD CHANGE -1 TO 0
CLR @#PSW ;MAKE [PSW] = 000
CCC ;SCOPE SYNC
2$: IOT ;TEST THE IOT
3$: HALT ;IOT FAILED TO LOAD PC
BR 1$ ;LOCK ON HARD ERROR
4$: MOV @#PSW,R2 ;GET THE PSW
CMP #357,R2 ;DID IOT LOAD A 357 ?
BEQ 6$ ;BR IF YES
5$: MOV R2,@#PSW ;RESTORE ERROR PSW
HALT ;IOT FAILED TO LOAD PSW
BR 1$ ;LOCK ON HARD ERROR
6$: CMP #35,(SP)+ ;DID IOT SAVE OLD PC ?
BEQ 7$ ;BR IF YES
7$: MOV R2,@#PSW ;RESTORE ERROR PSW
HALT ;IOT FAILED TO SAVE OLD PC
BR 1$ ;LOCK ON HARD ERROR
8$: TST (SP)+ ;DID IOT SAVE OLD PSW ?
BEQ TST165 ;BR IF YES
9$: MOV R2,@#PSW ;RESTORE ERROR PSW
HALT ;IOT FAILED TO SAVE OLD PSW

```

```

4369 012462 000732
4370
4371
4372
4373
4374 012464
4375 012464 012700 000165
4376 012470 010605
4377 012472 010506
4378 012474 012737 012520 000020
4379 012502 005037 000022
4380 012506 012737 000340 177776
4381 012514 000277
4382
4383 012516 000004
4384
4385 012520 013702 177776
4386 012524 001404
4387
4388 012526 010237 177776
4389 012532 000000
4390 012534 000756
4391
4392 012536 010506
4393
4394
4395
4396
4397
4398 012540 012700 000166
4399 012544 010605
4400 012546 010506
4401 012550 005037 063236
4402 012554 005037 000036
4403 012560 012737 062202 000034
4404 012566 000257
4405
4406 012570 104401
4407
4408 012572 012737 063166 000034
4409 012600 012737 000340 000036
4410 012606 005137 063236
4411 012612 001402
4412
4413 012614 000000
4414 012616 000753
4415
4416
4417
4418
4419 012620
4420 012620 012700 000167
4421 012624 010605
4422 012626 010506
4423 012630 012737 062040 000030
4424 012636 005037 000032

```

```

BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 165 BASIC IOT TEST - VERIFY LOADING PSW WITH 000
;*****
TST165:
MOV #165,R0 ;LOAD R0 WITH TEST NUMBER
SP,R5 ;SAVE THE SP
R5,SP ;RESET SP FOR ERROR LOOP
1$: CLR @#20 ;SET UP IOT VECTOR
MOV #340,@#PSW ;MAKE [PSW] = 340
SCC ;MAKE N:C=1111
2$: IOT ;TEST THE IOT
4$: MOV @#PSW,R2 ;GET THE [PSW]
BEQ 6$ ;BR IF [PSW] = 000
3$: MOV R2,@#PSW ;RESTORE THE ERROR PSW
HALT ;IOT FAILED TO CLEAR THE PSW
BR 1$ ;LOCK ON HARD ERROR
6$: MOV R5,SP ;RESET THE SP BEFORE CONTINUING
;*****
;TEST 166 BASIC "TRAP" TEST - LINKAGE TO PRINT ROUTINE
;*****
TST166:
MOV #166,R0 ;LOAD R0 WITH TEST NUMBER
SP,R5 ;SAVE THE SP
R5,SP ;RESET SP FOR ERROR LOOP
1$: CLR @PRIFLG ;INITIALIZE TEST FLAG
MOV #36,@#34 ;SET UP THE "TRAP" VECTOR
CCC ;SCOPE SYNC
2$: TYPE ;TEST THE TRAP
MOV #STRAP,@#34 ;SETUP TRAP VECTOR
MOV #34,@#36
COM @PRIFLG ;SHOULD MAKE [PRIFLG] = 000000
BEQ TST167 ;BR IF IT DID
3$: HALT ;TRAP FAILED TO LINK TO PRINT SERV.
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 167 BASIC "EMT" TEST - LINKAGE TO ERROR SERVICE
;*****
TST167:
MOV #167,R0 ;LOAD R0 WITH TEST NUMBER
SP,R5 ;SAVE THE SP
R5,SP ;RESET SP FOR ERROR LOOP
1$: MOV #ERRA,@#30 ;SET UP THE EMT VECTOR
CLR @#32

```


4425 012642 005037 063240
 4426 012646 000257
 4427
 4428 012650 104000
 4429
 4430 012652 005137 063240
 4431 012656 001402
 4432
 4433 012660 000000
 4434 012662 000761
 4435
 4436
 4437
 4438 012664
 4439 012664 012700 000170
 4440 012670 010605
 4441 012673 012737 061114 000010
 4442 012700 012737 000340 000012
 4443 012705 010506
 4444 012710 005037 063246
 4445
 4446 012714 000257
 4447
 4448 012716 000007
 4449
 4450 012720 005137 063246
 4451 012724 001402
 4452
 4453 012726 000000
 4454 012730 000766
 4455
 4456 012732 012737 061122 000010
 4457 012740 012737 000340 000012
 4458
 4459
 4460
 4461
 4462 012746
 4463 012746 012700 000171
 4464 012752 010605
 4465 012753 012737 061212 000004
 4466 012762 012737 000340 000006
 4467 012770 010506
 4468 012772 005037 063250
 4469
 4470 012776 000257
 4471
 4472 013000 005737 177700
 4473
 4474 013004 005137 063250
 4475 013010 001402
 4476
 4477 013012 000000
 4478 013014 000765
 4479
 4480

```

CLR @#ERRFLG ;EMT SERVICE WILL COM (ERRFLG)
CCC ;SCOPE SYNC
2$: ERROR ;TEST THE EMT
COM @#ERRFLG ;DID EMT SERV. COM ERRFLG?
BEQ TST170 ;BR IF YES
3$: HALT ;EMT DID NOT LINK PROPERLY
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 170 BASIC TEST OF RSVD INSTR. TRAP LINKAGE
;*****
TST170:
MOV #170,R0 ;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;SAVE THE SP
MOV #RSVFLG,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
MOV #340,@#12
1$: CLR @#RSVFLG ;RESET SP FOR ERROR LOOP
;INITIALIZE TEST FLAG THAT WILL GET
;COMPLEMENTED BY TRAP SERVICE
;SCOPE SYNC
2$: 000007 ;FORCE RSVD INSTR. TRAP
COM @#RSVFLG ;TEST FLAG SHOULD GO TO 000000
BEQ 4$ ;BR IF TRAP SPRUNG
3$: HALT ;RSVD INSTR. TRAP FAILED
BR 1$ ;LOCK ON HARD ERROR
4$: MOV #RSERR,@#10 ;SET UP RSVD INSTR TRAP VECTOR TO POINT
MOV #340,@#12 ;TO ERROR SERVICE ROUTINE
;*****
;TEST 171 BASIC TEST OF BUS TIMEOUT TRAP LINKAGE
;*****
TST171:
MOV #171,R0 ;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;SAVE THE SP
MOV #BTEST,@#4 ;SET UP THE BUS ERROR VECTOR
MOV #340,@#6
1$: MOV RS,SP ;RESET SP FOR ERROR LOOP
CLR @#BERFLG ;INITIALIZE TEST FLAG THAT WILL GET
;COMPLEMENTED BY TRAP SERVICE
;SCOPE SYNC
2$: TST @#177700 ;FORCE BUS TIMEOUT USING R0 ADDR.
COM @#BERFLG ;TEST FLAG SHOULD GO TO 000000
BEQ TST172 ;BR IF TRAP SPRUNG
3$: HALT ;BUS ERROR FAILED TO SPRING TRAP
BR 1$ ;LOCK ON HARD ERROR
;*****

```

4481
 4482
 4483 013016
 4484 013016 012700 000172
 4485 013022 005067 050264
 4486 013026 005367 050260
 4487 013032 001375
 4488 013034 012737 013074 000004
 4489 013042 012737 000340 000006
 4490 013050 010605
 4491 013052 010506
 4492 013054 012702 177560
 4493 013060 000257
 4494
 4495 013062 005722
 4496 013064 005722
 4497 013066 005722
 4498 013070 005712
 4499
 4500 013072 000403
 4501
 4502 013074 005742
 4503 013076 000000
 4504 013100 000764
 4505
 4506 013102 012737 061220 000004
 4507 013110 012737 000340 000006
 4508
 4509
 4510
 4511 013116
 4512 013122 012700 000173
 4513 013122 012702 177564
 4514 013126 012704 000200
 4515 013132 005012
 4516 013134 005001
 4517 013136 000257
 4518
 4519 013140 020412
 4520
 4521 013142 001405
 4522 013144 005301
 4523 013146 001374
 4524
 4525 013150 011203
 4526 013152 000000
 4527 013154 000766
 4528
 4529
 4530
 4531
 4532 013156
 4533 013156 012700 000174
 4534 013162 012702 177564
 4535 013162 000200
 4536 013162 005012

```

;*****
;TEST 172 BASIC TEST FOR ACCESSING DL11 REGISTERS
;*****
TST172:
MOV #172,R0 ;LOAD R0 WITH TEST NUMBER
CLR MBUFO ;INIT STALL COUNTER
DEC MBUFO ;COUNT THE TIMER
BNE 11$ ;BR IF NO TIMEOUT
MOV #340,@#4 ;SET UP
MOV SP,R5 ;BUS TIMEOUT VECTOR
MOV RS,SP ;SAVE TH SP
MOV #RCSR,R2 ;RESET SP FOR ERROR LOOP
CCC ;CR2 = STARTING DL11 ADDR.
;SCOPE SYNC
2$: TST (R2)+ ;REFERENCE DL11 - RCSR
TST (R2)+ ;REFERENCE DL11 - RDDR
TST (R2)+ ;REFERENCE DL11 - XCSR
TST (R2) ;REFERENCE DL11 - XDBR
BR 4$ ;GO TO NEXT TEST
3$: TST -(R2) ;BAD ADDRESS IN R2
HALT ;ONE OF DL11 ADDR'S CAUSED TIME OUT
BR 1$ ;LOCK ON HARD ERROR
4$: MOV #BERR,@#4 ;SET UP BUS ERROR VECTOR TO POINT
MOV #340,@#6 ;TO ERROR SERVICE ROUTINE
;*****
;TEST 173 BASIC TEST OF DL11 - XCSR - READY(1)
;*****
TST173:
MOV #173,R0 ;LOAD R0 WITH TEST NUMBER
MOV #XCSR,R2 ;DEST ADDR = XCSR
MOV #200,R4 ;RESULT S / B = 200
CLR (R2) ;CLEAR CDEST
CLR R1 ;SET UP TIMEOUT COUNTER
CCC ;SCOPE SYNC
2$: CMP R4,(R2) ;TEST READY BIT - IT SHOULD BE SET
BEQ TST174 ;BR IF IT WAS
DEC R1 ;TICK-TOCK GOES THE TIMER
BNE 2$ ;BR IF NOT A TIMEOUT
3$: MOV (R2),R3 ;GET THE WAS DATA
HALT ;READY BIT IN XCSR FAILED ON A (0)
BR 1$ ;LOCK ON HARD ERROR
;*****
;TEST 174 BASIC TEST OF DL11 - XCSR - MAINT BIT (0)
;*****
TST174:
MOV #174,R0 ;LOAD R0 WITH TEST NUMBER
MOV #XCSR,R2 ;DEST ADDR = XCSR
MOV #200,R4 ;RESULT S / B = 200
CLR (R2) ;CLEAR MAINT. BIT

```

```

4537 013174 000257          CCC                ;SCOPE SYNC
4538 013176 020412          2$: CMP            R4,(R2)        ;TEST MAINT(0)
4539 013200 001403          BEQ            TST175          ;;BR IF MAINT BIT CLEAR
4540 013202 011203          MOV            (R2),R3        ;GET THE WAS DATA
4541 013204 000000          HALT           1$            ;CAN'T CLEAR MAINT BIT
4542 013206 000771          BR             1$            ;LOCK ON HARD ERROR
4543
4544 *****
4545 ;*TEST 175 BASIC TEST OF DL11 XCSR - MAINT BIT = 1
4546 ;*****
4547 TST175:
4548 MOV            #175,R0        ;LOAD R0 WITH TEST NUMBER
4549 MOV            #XCSR,R2      ;DEST ADDR = XCSR
4550 MOV            #204,R4       ;RESULT S / = 204
4551 MOV            #4,(R2)      ;SET THE MAINT. BIT
4552 CCC                ;SCOPE SYNC
4553
4554 013232 020412          2$: CMP            R4,(R2)        ;TEST MAINT.(1)
4555 013234 001403          BEQ            TST176          ;;BR IF IT WAS
4556 013236 011203          MOV            (R2),R3        ;GET THE WAS DATA
4557 013240 000000          HALT           1$            ;CAN'T SET MAINT BIT IN XCSR
4558 013242 000770          BR             1$            ;LOCK O HARD ERROR
4559
4560 *****
4561 ;*TEST 176 BASIC DL11 OUT / IN ECHO TEST (MAINT MODE)
4562 ;THIS ROUTINE USES THE MAINTENANCE MODE FEATURE OF THE DL11 TO
4563 ;TURN AROUND A STRING OF 8 CHARACTERS TO THE DL11. THIS STRING CONSISTS
4564 ;OF ALTERNATING NULL / DELETE CHARS WHICH ARE NON PRINTING. THE 8 CHARS
4565 ;ARE OUTPUT THEN READ BACK INTO A CORE BUFFER AND THEN THE INPUT AND
4566 ;OUTPUT CORE BUFFERS ARE CHECKED FOR EQUIVALENCE. IF AN ERROR IS DET-
4567 ;ECTED DURING THE COMPARISON THE ROUTINE HALTS WITH THE WAS AND S / B
4568 ;DATA IN R3 AND R4 RESPECTFULLY. A TIMER IS EMPLOYED TO PREVENT THE
4569 ;TEST FROM HANGING IF RECEIVER DONE DOES NOT RESPOND.
4570 ;*****
4571 TST176:
4572 MOV            #176,R0        ;LOAD R0 WITH TEST NUMBER
4573 MOV            #RCSR,R2      ;R2 POINTS TO DL11 - START ADDR
4574 TSTB          2(R2)          ;REFERENCE DL11 INPUT DATA BUFFER TWICE
4575 TSTB          2(R2)          ;TO FLUSH RCVR "DONE" BIT
4576 MOV            #INBUF,R3     ;R3 POINTS TO CORE INPUT BUFFER
4577 MOV            #OBUF,R4      ;R4 POINTS TO CORE OUTPUT BUFFER
4578 MOV            #10,R5        ;R5 WILL COUNT 8 CHARS OUTPUT
4579 MOV            #4,(R2)      ;TURN ON MAINT MODE
4580
4581 013306 005001          1$: CLR            R1            ;R1 USED AS TIMEOUT COUNTER
4582 013310 112462          MOVB          (R4)+,6(R2)     ;LOAD OUTPUT BUFFER IN DL11
4583 013314 108713          TSTB          (R2)          ;RECEIVER DONE SET ?
4584 013316 004714          BNE           3$            ;R1 YES
4585 013320 005301          DEC            R1            ;COUNT THE TIMER
4586 013322 001374          BNE           2$            ;BR IF NO TIMEOUT
4587
4588 013244 012700 000176          MOV            #176,R0        ;LOAD R0 WITH TEST NUMBER
4589 013250 012702 177560          MOV            #RCSR,R2      ;R2 POINTS TO DL11 - START ADDR
4590 013254 105762 000002          TSTB          2(R2)          ;REFERENCE DL11 INPUT DATA BUFFER TWICE
4591 013260 105762 000002          TSTB          2(R2)          ;TO FLUSH RCVR "DONE" BIT
4592 013264 012703 063256          MOV            #INBUF,R3     ;R3 POINTS TO CORE INPUT BUFFER
4593 013270 012704 063256          MOV            #OBUF,R4      ;R4 POINTS TO CORE OUTPUT BUFFER
4594 013274 012705 000010          MOV            #10,R5        ;R5 WILL COUNT 8 CHARS OUTPUT
4595 013300 012762 000004 000004          MOV            #4,(R2)      ;TURN ON MAINT MODE
4596
4597 013306 005001          1$: CLR            R1            ;R1 USED AS TIMEOUT COUNTER
4598 013310 112462          MOVB          (R4)+,6(R2)     ;LOAD OUTPUT BUFFER IN DL11
4599 013314 108713          TSTB          (R2)          ;RECEIVER DONE SET ?
4600 013316 004714          BNE           3$            ;R1 YES
4601 013320 005301          DEC            R1            ;COUNT THE TIMER
4602 013322 001374          BNE           2$            ;BR IF NO TIMEOUT
4603
4604 013244 012700 000176          MOV            #176,R0        ;LOAD R0 WITH TEST NUMBER
4605 013250 012702 177560          MOV            #RCSR,R2      ;R2 POINTS TO DL11 - START ADDR
4606 013254 105762 000002          TSTB          2(R2)          ;REFERENCE DL11 INPUT DATA BUFFER TWICE
4607 013260 105762 000002          TSTB          2(R2)          ;TO FLUSH RCVR "DONE" BIT
4608 013264 012703 063256          MOV            #INBUF,R3     ;R3 POINTS TO CORE INPUT BUFFER
4609 013270 012704 063256          MOV            #OBUF,R4      ;R4 POINTS TO CORE OUTPUT BUFFER
4610 013274 012705 000010          MOV            #10,R5        ;R5 WILL COUNT 8 CHARS OUTPUT
4611 013300 012762 000004 000004          MOV            #4,(R2)      ;TURN ON MAINT MODE
4612
4613 013306 005001          1$: CLR            R1            ;R1 USED AS TIMEOUT COUNTER
4614 013310 112462          MOVB          (R4)+,6(R2)     ;LOAD OUTPUT BUFFER IN DL11
4615 013314 108713          TSTB          (R2)          ;RECEIVER DONE SET ?
4616 013316 004714          BNE           3$            ;R1 YES
4617 013320 005301          DEC            R1            ;COUNT THE TIMER
4618 013322 001374          BNE           2$            ;BR IF NO TIMEOUT

```

```

4593 013324 000000          HALT           6$            ;DL11 FAILED TO RESPOND IN TIME
4594 013326 000750          BR             6$            ;LOCK ON HARD ERROR
4595
4596 013330 116223 000002          3$: MOVB          2(R2),(R3)+   ;READ THE DL11 INPUT BUFFER INTO CORE
4597 013334 005305          DEC            R3            ;COUNT ONE CHAR
4598 013336 001363          BNE           1$            ;BR IF NOT DONE 8 CHARS
4599
4600 013340 005062 000004          CLR            4(R2)          ;TURN OFF MAINT. MODE
4601 013344 012702 000010          MOV            #10,R5        ;RESET CHAR COUNTER
4602 013348 012703 063256          MOV            #INBUF,R3     ;RESET INBUF POINTER
4603 013354 012704 063256          MOV            #OBUF,R4      ;RESET OUTBUF POINTER
4604
4605 013360 122324          4$: CMPB          (R3)+,(R4)+   ;INPUT = OUTPUT ??
4606 013364 005305          BNE           5$            ;BR IF NOT
4607 013366 005305          DEC            R3            ;COUNT ONE CHECKED
4608 013368 001374          BNE           4$            ;BR UNTIL 8 DONE
4609 013370 000410          BR             CITST         ;GO TO NEXT TEST
4610
4611 013372 114303          5$: MOVB          -(R3),R3      ;WAS DATA IN R3 [BITS 7:0]
4612 013374 114404          MOVB          -(R4),R4      ;S / B DATA IN R4 [BITS 7:0]
4613 013376 042703 177400          BIC            #177400,R3    ;STRIP OFF BITS <15:08>
4614 013402 042704 177400          BIC            #177400,R4    ;"
4615 013406 000000          HALT           6$            ;RECEIVED DATA NOT EQUAL TO OUTPUT DATA
4616 013410 000717          BR             6$            ;LOCK ON HARD ERROR

```

```

4617
4618
4619
4620
4621 013412 012737 061260 000020
4622 013420 005037 000022
4623 013424 012737 061620 000030
4624 013432 012737 003140 000032
4625 013430 012737 063166 000034
4626 013446 012737 000340 000036
4627 013454 012737 060664 000024
4628 013462 012737 000340 000026
4629 013470 105737 001141
4630 013474 100093
4631 013476 012737 001142 001040
4632 013504 032777 010000 165326
4633 013512 001007
4634 013514 005737 063254
4635 013520 001004
4636 013522 005137 063254
4637 013526 104401
4638 013530 065141
4639 013532 005037 177776
4640 013536 012737 003316
4641 013544 012737 000040 001110
4642 013552 010037 001124

```

```

;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
;*****/COMPREHENSIVE INSTRUCTION TESTS/*****
;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
CITST: MOV #SCOPE,@#20 ;SET UP IOT VECTOR
CLR @#22
MOV #ERROR,@#30 ;SET UP EMT VECTOR
MOV #TRAP,@#34 ;SET UP TRAP VECTOR
MOV #PWRDN,@#24 ;SET UP POWER FAIL VECTOR
MOV #340,@#26
TSTB #SENVM ;DO NOT SIZE BIT SET?
BPL 3$ ;BR IF NOT - USE HARDWARE SWITCH REG
MOV #SWREG,@SWR ;USE APT SWITCH REG
RIT #SW12,@SWR ;INHIBIT PRINTING INTRO. I.D. MESSAGE?
BNE IS ;BR IF YES
TST @ONCE ;FIRST TIME INTO "CIT" TESTS?
BNE IS ;BR IF NOT - PRINT ID ONLY ONCE
CDM @ONCE ;SET FLAG TO INHIBIT PRINTING AGAIN
TYPE ;IDENTIFY THIS PROGRAM
IDENT1 ;ADDR OF THE ID MESSAGE
CLR @#PSW ;SET CPU PRIORITY TO LEVEL 000
MOV #TST0,@#SLPADR ;INITIALIZE SCOPE LOOP RETURN
MOV #40,@#TIMES ;ITERATE ON BIT SECTION 32 TIMES
MOV #0,@#TESTN ;PREVENT MISSED TEST ERROR ON
;FIRST SCOPE CALL

```

```

;*****
;TEST 177 BCC TEST WITH C=1
;*****

```

```

TST177: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #177,R0 ;LOAD R0 WITH TEST NUMBER
MOV #R2S,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
SEC ;MAKE C=1
2$: BCC 3$ ;TEST THE BCC, IT SHOULDN'T BR
BR TST200 ;GO TO SCOPE EXIT
3$: ERROR 5 ;BCC FAILED

```

```

;*****
;TEST 200 BCC TEST WITH C=0
;*****

```

```

TST200: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #200,R0 ;LOAD R0 WITH TEST NUMBER
MOV #R2S,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLC ;MAKE C=0
2$: BCC TST201 ;;TEST THE BCC-IT SHOULD BR
3$: ERROR 5 ;BCC FAILED

```

```

4673
4674
4675
4676 013620 000004
4677 013622 012700 000201
4678 013624 013701 013644
4679 013626 012704 000017
4680 013632 012704 000017
4681 013636 012702 177776
4682
4683 013642 000277
4684
4685 013644 103004
4686
4687 013646 013703 177776
4688 013648 020304
4689 013654 001401
4690
4691 013656 104001
4692
4693
4694
4695
4696 013660 000004
4697 013660 012700 000202
4698 013662 012701 013704
4699 013666 013701 013704
4700 013672 012704 000017
4701 013676 012702 177776
4702
4703 013702 000277
4704
4705 013704 000401
4706
4707 013706 104005
4708
4709 013710 013703 177776
4710 013714 020304
4711 013716 001401
4712
4713 013720 104001
4714
4715
4716
4717
4718 013722 000004
4719 013722 012700 000203
4720 013724 012701 013744
4721 013730 013701 013744
4722 013734 005004 177776
4723 013736 012702
4724
4725 013742 000257
4726
4727 013744 103404
4728

```

```

;*****
;TEST 201 VERIFY NO BRANCH MICROROUTINE DOES NOT CLR FLAGS
;*****
TST201: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #201,R0 ;LOAD R0 WITH TEST NUMBER
MOV #R2S,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #R4,R4 ;S/B PSW
MOV #PSW,R2 ;DEST = PSW FOR ERROR CALL
;MAKE N:C = 1111
2$: BCC 3$ ;TEST THE BCC-IT SHOULDN'T BR
MOV @#PSW,R3 ;GET WAS FLAGS
CMP R3,R4 ;N:C = 1111?
BEQ TST202 ;;BR IF YES
3$: ERROR 1 ;NO BRANCH MICROROUTINE ALTERED CODES

```

```

;*****
;TEST 202 VERIFY BRANCH MICROROUTINE DOES NOT CLR FLAGS
;*****

```

```

TST202: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #202,R0 ;LOAD R0 WITH TEST NUMBER
MOV #R2S,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #R4,R4 ;S/B PSW
MOV #PSW,R2 ;DEST = PSW FOR ERROR CALL
;MAKE N:C = 1111
2$: BR 4$ ;TEST THE BR
3$: ERROR 5 ;JUST IN CASE THE BR DIDN'T WORK
4$: MOV @#PSW,R3 ;GET THE FLAGS
CMP R3,R4 ;N:C = 1111?
BEQ TST203 ;;BR IF YES
5$: ERROR 1 ;BRANCH MICROROUTINE ALTERED CODES

```

```

;*****
;TEST 203 VERIFY NO BRANCH MICROROUTINE DOES NOT SET FLAGS
;*****

```

```

TST203: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #203,R0 ;LOAD R0 WITH TEST NUMBER
MOV #R2S,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;PSW S/B = 0
MOV #PSW,R2 ;DEST = PSW FOR ERROR CALL
;MAKE N:C = 0000
2$: BCS 3$ ;TEST THE BCS-IT SHOULDN'T BR

```

```

4729 013746 013703 177776
4730 013752 005703
4731 013754 001401
4732
4733 013756 104001
4734
4735
4736
4737
4738 013760
4739 013760 000004
4740 013762 012700 000204
4741 013766 013701 014002
4742 013772 005004
4743 013774 012702 177776
4744
4745 014000 000257
4746 014002 000401
4747
4748 014004 104005
4749
4750
4751 014006 013703 177776
4752 014012 005703
4753 014014 001401
4754
4755 014016 104001
4756
4757
4758
4759
4760 014020
4761 014020 000004
4762 014022 012700 000205
4763 014026 013701 014034
4764 014032 000257
4765
4766 014034 003401
4767 014036 000401
4768
4769 014040 104005
4770
4771
4772
4773
4774 014042
4775 014042 000004
4776 014044 012700 000206
4777 014050 013701 014060
4778 014054 000257
4779 014056 000264
4780
4781 014060
4782 014060 003401
4783
4784 014062 104005

```

```

MOV R4 @PSW,R3 ;GET FLAGS
TST R3 ;N:C = 0000
BEQ TST204 ;;BR IF YES
3$: ERROR 1 ;NO BRANCH MICROROUTINE-ALTERED CODES
;*****
;TEST 204 VERIFY BRANCH MICROROUTINE DOES NOT SET FLAGS
;*****
TST204:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #204,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;PSW S/B = 0
MOV #PSW,R2 ;DEST = PSW FOR ERROR CALL
CCC ;MAKE N:C = 0000
2$: BR 4$ ;TEST THE BR
3$: ERROR 5 ;JUST IN CASE THE BR DIDN'T WORK
4$: MOV @#PSW,R3 ;GET FLAGS
TST R3 ;N:C = 0000
BEQ TST205 ;;BR IF YES
5$: ERROR 1 ;BRANCH MICROROUTINE ALTERED CODES.
;*****
;TEST 205 BLE TEST WITH Z = 0 AND N,V = 00
;*****
TST205:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #205,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;CLEAR FLAGS
CCC ;CLEAR FLAGS
2$: BLE 3$ ;TEST THE BLE-IT SHOULDN'T BR
BR TST206 ;;GO TO SCOPE EXIT
3$: ERROR 5 ;BLE FAILED
;*****
;TEST 206 BLE TEST WITH Z = 1 AND N,V = 00
;*****
TST206:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #206,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CCC ;CLEAR FLAGS
SEZ ;SET Z = 1
2$: BLE TST207 ;;TEST THE BLE-IT SHOULD BR
3$: ERROR 5 ;BLE FAILED

```

```

4785
4786
4787
4788
4789 014064
4790 014064 000004
4791 014066 012700 000207
4792 014072 013701 014102
4793 014076 000257
4794 014100 000262
4795
4796 014102
4797 014102 003401
4798
4799 014104 104005
4800
4801
4802
4803
4804 014106
4805 014106 000004
4806 014110 012700 000210
4807 014114 013701 014124
4808 014120 000257
4809 014122 000270
4810
4811 014124
4812 014124 003401
4813
4814 014126 104005
4815
4816
4817
4818
4819 014130
4820 014130 000004
4821 014132 012700 000211
4822 014136 013701 014146
4823 014142 000257
4824 014144 000272
4825
4826 014146 003401
4827 014150 000401
4828
4829 014152 104005
4830
4831
4832
4833 014154
4834 014154 000004
4835 014156 012700 000212
4836 014162 013701 014170
4837 014166 000257
4838
4839
4840 014170 101401

```

```

;*****
;TEST 207 BLE TEST WITH Z = 0 AND N,V = 01
;*****
TST207:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #207,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;CLEAR FLAGS
SEV ;MAKE Z = 0 AND N,V = 01
2$: BLE TST210 ;;TEST THE BLE-IT SHOULD BR
3$: ERROR 5 ;BLE FAILED
;*****
;TEST 210 BLE TEST WITH Z = 0 AND N,V = 10
;*****
TST210:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #210,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;CLEAR FLAGS
SEN ;MAKE Z = 0 AND N,V = 10
2$: BLE TST211 ;;TEST THE BLE-IT SHOULD BR
3$: ERROR 5 ;BLE FAILED
;*****
;TEST 211 BLE TEST WITH Z = 0 AND N,V = 11
;*****
TST211:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #211,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;CLEAR FLAGS
272 ;MAKE Z = 0 AND N,V = 11
2$: BLE 3$ ;TEST THE BLE-IT SHOULDN'T BR
BR TST212 ;;GO TO SCOPE EXIT
3$: ERROR 5 ;BLE FAILED
;*****
;TEST 212 BLOS TEST WITH Z,C = 00
;*****
TST212:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #212,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;MAKE Z,C = 00
2$: BLOS 3$ ;TEST THE BLOS-IT SHOULDN'T BR

```

4841 014172 000401
 4842
 4843 014174 104005
 4844
 4845
 4846
 4847
 4848 014176
 4849 014176 000004
 4850 014200 012700 000213
 4851 014204 013701 014214
 4852 014210 000257
 4853 014212 000261
 4854
 4855 014214
 4856 014214 101401
 4857
 4858 014216 104005
 4859
 4860
 4861
 4862
 4863 014220
 4864 014220 000004
 4865 014220 012700 000214
 4866 014226 013701 014236
 4867 014232 000257
 4868 014234 000264
 4869
 4870
 4871 014236 101401
 4872 014236 104005
 4873
 4874
 4875
 4876
 4877
 4878 014242
 4879 014242 000004
 4880 014244 012700 000215
 4881 014250 013701 014260
 4882 014254 000257
 4883 014256 000265
 4884
 4885 014260
 4886 014260 101401
 4887
 4888 014262 104005
 4889
 4890
 4891
 4892
 4893
 4894 014264
 4895 014266 000004
 4896 014272 013701 000216 014310

```

BR TST213 ;GO TO SCOPE EXIT
3$: ERROR 5 ;BLOS FAILED
;*****
;TEST 213 BLOS TEST WITH Z,C = 01
;*****
TST213:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #213,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CCC ;CLEAR FLAGS
SEC ;MAKE Z,C = 01
2$: BLOS TST214 ;TEST THE BLOS-IT SHOULD BR
3$: ERROR 5 ;BLOS FAILED
;*****
;TEST 214 BLOS TEST WITH Z,C = 10
;*****
TST214:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #214,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CCC ;CLEAR FLAGS
SEZ ;MAKE Z,C = 10
2$: BLOS TST215 ;TEST THE BLOS-IT SHOULD BR
3$: ERROR 5 ;BLOS FAILED
;*****
;TEST 215 BLOS TEST WITH Z,C = 11
;*****
TST215:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #215,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CCC ;CLEAR FLAGS
265 ;MAKE Z,C = 11
2$: BLOS TST216 ;TEST THE BLOS-IT SHOULD BR
3$: ERROR 5 ;BLOS FAILED
;*****
;TEST 216 SXT MODE 0 TEST WITH N = 0 AND C = 1
;*****
TST216:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #216,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

```

4897 014276 005004
 4898 014300 012703 177777
 4899 014304 000257
 4900 014306 000263
 4901
 4902 014310 006703
 4903
 4904 014312 100403
 4905 014314 101403
 4906 014316 102401
 4907 014320 103401
 4908
 4909 014322 104002
 4910
 4911 014324 005703
 4912 014326 001401
 4913
 4914 014330 104002
 4915
 4916
 4917
 4918
 4919 014332
 4920 014332 000004
 4921 014334 012700 000217
 4922 014340 013701 014366
 4923
 4924 014344 032737 000020 063234
 4925 014352 001401
 4926 014354 000000
 4927
 4928 014356 005004
 4929 014360 012703 177777
 4930 014364 000257
 4931
 4932 014366 006703
 4933 014370 103001
 4934
 4935 014372 104002
 4936
 4937
 4938
 4939
 4940
 4941 014374
 4942 014376 000004
 4943 014402 012700 000220
 4944 014406 013701 014416 177777
 4945 014408 012704
 4946 014412 005003
 4947 014414 000277
 4948
 4949 014416 006703
 4950
 4951 014420 100003
 4952 014424 001402
 4953 014424 102401

```

CLR R4 ;RESULT S / B = 0
MOV #-1,R3 ;INITIAL DEST. OP = 177777
CCC ;CLEAR CODES
263 ;N:C = 0011
2$: SXT R3 ;TEST THE SXT
BNI 35 ;DID SXT MAKE N:C = 0101?
BVS 35
BCS 45
3$: ERROR 2 ;SXT FAILED TO ALTER CODES PROPERLY
4$: TST R3 ;DID RESULT = 0?
BEQ TST217 ;BR IF IT DID
5$: ERROR 2 ;SXT DELIVERED WRONG RESULT TO R3
;*****
;TEST 217 SXT MODE 0 TEST WITH N = 0 AND C = 0
;*****
TST217:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #217,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
.SBTL USER CONT ROLLED BREAKPOINT -- BIT4 ;BREAKPOINT HALT SET ??
BIT #BIT4,#BPTLOC ;BR IF NOT
BEQ .+4 ;BREAK - DEPRESS CONTINUE TO RESTART
HALT
CLR R4 ;RESULT S / B = 0
MOV #-1,R3 ;INITIAL DEST OP = 177777
CCC ;CLEAR N:C
2$: SXT R3 ;TEST THE SXT
BCC TST220 ;BR IF "C" STILL CLEAR
3$: ERROR 2 ;SXT AFFECTED "C" BIT
;*****
;TEST 220 SXT MODE 0 TEST WITH N = 1 AND C = 1
;*****
TST220:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #220,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S / B = 177777
CLR R3 ;INITIAL DEST OP = 0
SCC ;MAKE N:C = 1111
2$: SXT R3 ;TEST THE SXT
BPL 35 ;N:C = 1001?
BEQ 35
BVS 35

```

4953 014426 103401
 4954 014430 104002
 4955 014432 010305
 4958 014434 005105
 4959 014436 001401
 4960 014440 104002
 4961
 4962
 4963
 4964
 4965
 4966 014442 000004
 4967 014444 012700 000221
 4968 014450 013701 014466
 4969 014454 012704 177777
 4970 014460 005003
 4971 014462 000257
 4972 014464 000276
 4973
 4974
 4975 014466 006703
 4976 014470 103001
 4977
 4978 014472 104002
 4979
 4980
 4981
 4982
 4983 014474 000004
 4984 014476 012700 000222
 4985 014502 013701 014524
 4986 014506 012702 063312
 4987 014512 005004
 4988 014514 012712 177777
 4989 014518 006724
 4990 014522 000263
 4991
 4992
 4993 014524 006712
 4994
 4995
 4996 014526 100403
 4997 014530 001002
 4998 014532 102401
 4999 014534 103401
 5000
 5001 014536 104001
 5002
 5003 014540 005712
 5004 014542 001401
 5005
 5006 014544 104001
 5007
 5008 014546 012702 063312
 5009 014552 013701 014566

BCS 4\$
 3\$: ERROR 2 ;SXT FAILED TO ALTER CODES PROPERLY
 4\$: MOV R3,R5 ;GET RESULT
 COM R5 ;COMPLEMENT IT-SHOULD GO TO 0
 BEQ TST221 ;BR IF RESULT OF SXT = 1
 5\$: ERROR 2 ;SXT DELIVERED WRONG RESULT.
 ;*****
 ;*TEST 221 SXT MODE 0 TEST WITH N = 1 AND C = 0
 ;*****
 TST221: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #221,R0 ;LOAD RO WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #-1,R4 ;RESULT S / B = 177777
 CLR R3 ;INITIAL DESTJ = 0
 CC ;CLEAR FLAGS
 276 ;MAKE N:C = 1110
 2\$: SXT R3 ;TEST THE SXT
 BCC TST222 ;BR IF "C" UNAFFECTED
 3\$: ERROR 2 ;SXT SET "C" BIT
 ;*****
 ;*TEST 222 SXT MODE 1 AND 2 TEST WITH N = 0 AND C = 1
 ;*****
 TST222: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #222,R0 ;LOAD RO WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUF0,R2 ;R2 POINTS TO DEST OP
 CLR R4 ;RESULT S / B = 0
 MOV #-1,(R2) ;INITIAL CDESTJ = 177777
 CC ;CLEAR CODES
 263 ;MAKE N:C = 0011
 2\$: SXT (R2) ;TEST THE SXT - DM1
 BMI 3\$
 BNE 3\$;N:C = 0101
 BVS 3\$
 BCS 4\$
 3\$: ERROR 1 ;SXT FAILED TO ALTER CODES PROPERLY
 4\$: TST (R2) ;DID RESULT = 0?
 BEQ 11\$;BR IF YES
 5\$: ERROR 1 ;SXT SHOULD HAVE ZEROED CDESTJ
 11\$: MOV #MBUF0,R2 ;DEST ADDR = MBUF0
 MOV #125,R1 ;LOAD R1 WITH TEST INSTR WORD

5009 014556 012712 177777
 5010 014562 000257
 5011 014564 000263
 5012
 5013 014566 006722
 5014
 5015 014570 100403
 5016 014572 001002
 5017 014574 102401
 5018 014576 103401
 5019
 5020 014600 104001
 5021
 5022 014602 005737 063312
 5023 014606 001401
 5024
 5025 014610 104001
 5026
 5027 014612 020227 063314
 5028 014616 001401
 5029
 5030 014620 104001
 5031
 5032
 5033 014622 000004
 5034 014624 012700 000223
 5035 014630 013701 014650
 5036 014634 005004
 5037 014636 012702 063312
 5038 014642 012712 177777
 5039 014646 000257
 5040
 5041
 5042 014650 006712
 5043 014652 103001
 5044
 5045 014654 104001
 5046
 5047
 5048
 5049
 5050
 5051
 5052 014656 000004
 5053 014658 012700 000224
 5054 014664 013701 014704
 5055 014670 012704 177777
 5056 014674 012702 063312
 5057 014700 005012
 5058 014702 000277
 5059
 5060 014704 006712
 5061
 5062
 5063 014706 100003
 5064 014710 001402

MOV #-1,(R2) ;INITIAL CDESTJ = 177777
 CCC ;CLEAR CODES
 263 ;MAKE N:C = 0011
 12\$: SXT (R2)+ ;TEST SXT - DM2
 BMI 7\$
 BNE 7\$;N:C = 0101 ?
 BVS 7\$
 BCS 6\$
 7\$: ERROR 1 ;SXT FAILED TO ALTER CODES PROPERLY
 6\$: TST #MBUF0 ;DID RESULT GET ZEROED ?
 BEQ 8\$;BR IF YES
 9\$: ERROR 1 ;SXT FAILED TO ZERO CDESTJ
 8\$: CMP R2,#MBUF0+2 ;WAS IT REALLY MODE 2 ?
 BEQ TST223 ;BR IF YES
 ERROR 1 ;SXT FAILED TO AUTO INCREMENT
 ;*****
 ;*TEST 223 SXT MODE 1 TEST WITH N = 0 AND C = 0
 ;*****
 TST223: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #223,R0 ;LOAD RO WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 CLR R4 ;RESULT S / B = 0
 MOV #MBUF0,R2 ;R2 POINTS TO DEST OP
 MOV #-1,(R2) ;INITIAL CDESTJ = 177777
 CCC ;CLEAR "C" BIT
 2\$: SXT (R2) ;TEST THE SXT
 BCC TST224 ;BR IF "C" UNDISTURBED
 3\$: ERROR 1 ;SXT SET THE "C" BIT
 ;*****
 ;*TEST 224 SXT MODE 1 TEST WITH N = 1 AND C = 1
 ;*****
 TST224: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #224,R0 ;LOAD RO WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #-1,R4 ;RESULT S / B = 177777
 MOV #MBUF0,R2 ;R2 POINTS TO DEST OP
 CLR (R2) ;INITIAL CDESTJ = 0
 SCC ;MAKE N:C = 1111
 2\$: SXT (R2) ;TEST THE SXT
 BPL 3\$
 BEQ 3\$;N:C = 1001?

5065 014712 102401
5066 014714 103401
5067
5068 014716 104001
5069
5070 014720 021204
5071 014722 001401
5072
5073 014724 104001
5074
5075
5076
5077
5078 014726
5079 014726 000004
5080 014730 012700 000225
5081 014734 013701 014756
5082 014740 012704 177400
5083 014744 012702 063312
5084 014750 005012
5085 014752 000257
5086 014754 000276
5087
5088 014756 006712
5089 014760 103001
5090
5091 014762 104001
5092
5093
5094
5095
5096 014764
5097 014764 000004
5098 014766 012700 000226
5099 014770 013701 015012
5100 014776 012704 177400
5101 015002 012703 000377
5102 015006 000257
5103 015010 000273
5104
5105 015012 000303
5106
5107 015014 100403
5108 015016 001002
5109 015020 102401
5110 015022 103001
5111
5112 015024 104002
5113
5114 015026 020403
5115 015030 001401
5116
5117 015032 104002
5118
5119
5120

```

BVS 3$
BCS 4$
3$: ERROR 1 ;SXT FAILED TO ALTER CODES PROPERLY
4$: CMP #R2,R4 ;RESULT = 177777?
BEQ #1,R4 ;BR IF YES
5$: ERROR 1 ;SXT DELIVERED WRONG RESULT
;*****
;4TEST 225 SXT MODE 1 TEST WITH N = 1 AND C = 0
;*****
4ST225: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #225,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S / B = 177777
MOV #NBUF0,R2 ;R2 POINTS TO DEST OP
CLR (R2) ;INITIAL DESTJ = 0
CCC ;CLEAR FLAGS
276 ;MAKE N:C = 1110
2$: SXT (R2) ;TEST THE SXT
BCC #T226 ;BR IF "C" UNAFFECTED
3$: ERROR 1 ;SXT SET THE "C" BIT
;*****
;4TEST 226 SWAB MODE 0 TEST WITH POS. RESULT
;*****
4ST226: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #226,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177400,R4 ;RESULT S / B = 177400
MOV #377,R3 ;INITIAL DEST OP = 377
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
2$: SWAB R3 ;TEST THE SWAB
BMI 3$
BNE 3$ ;N:C = 0100
BVS 3$
BCC 4$
3$: ERROR 2 ;SWAB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT?
BEQ #T227 ;BR IF YES
5$: ERROR 2 ;SWAB DELIVERED WRONG RESULT
;*****
;4TEST 227 SWAB MODE 0 TEST WITH NEG. RESULT
;*****
```

5121 015034
5122 015034 000004
5123 015036 012700 000227
5124 015042 013701 015062
5125 015046 012704 000377
5126 015052 012703 177400
5127 015056 000257
5128 015060 000267
5129
5130 015062 000303
5131
5132 015064 100003
5133 015066 001401
5134 015070 102401
5135 015072 103001
5136
5137 015074 104002
5138
5139 015076 020403
5140 015100 001401
5141
5142 015102 104002
5143
5144
5145
5146
5147
5148 015104
5149 015104 000004
5150 015106 012700 000230
5151 015112 013701 015136
5152 015116 012704 177400
5153 015122 012702 063312
5154 015126 012712 000377
5155 015132 000257
5156 015134 000273
5157
5158 015136 000312
5159
5160 015140 100403
5161 015142 001002
5162 015144 102401
5163 015146 103001
5164
5165 015150 104001
5166
5167 015152 020412
5168 015154 001401
5169
5170 015156 104001
5171
5172 015160 013701 015200
5173 015164 012702 063312
5174 015170 012712 000377
5175 015174 000257
5176 015176 000273

```

;*****
;4TEST 227: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #227,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #377,R4 ;RESULT S / B = 377
MOV #177400,R3 ;INITIAL DEST OP = 177400
CCC ;CLEAR FLAGS
267 ;MAKE N:C = 0111
2$: SWAB R3 ;TEST THE SWAB
BPL 3$
BEQ 3$ ;DID SWAB MAKE N:C = 1000
BVS 3$
BCC 4$
3$: ERROR 2 ;SWAB FAILED TO ALTER CODES PROPERLY
4$: CMP #R4,R3 ;DID SWAB DELIVER CORRECT RESULT?
BEQ #T230 ;BR IF OK
5$: ERROR 2 ;SWAB DELIVERED WRONG RESULT
;*****
;4TEST 230 SWAB MODE 1 AND 2 TEST WITH POS. RESULT
;*****
4ST230: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #230,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177400,R4 ;RESULT S / B = 177400
MOV #NBUF0,R2 ;R2 POINTS TO DEST OP
MOV #377,(R2) ;SET UP DEST OP = 377
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
2$: SWAB (R2) ;TEST THE SWAB - DM1
BMI 3$
BNE 3$ ;N:C = 0100
BVS 3$
BCC 4$
3$: ERROR 1 ;SWAB FAILED TO ALTER CODES PROPERLY
4$: CMP #R4,(R2) ;CORRECT RESULT?
BEQ 5$ ;BR IF OK
5$: ERROR 1 ;SWAB DELIVERED WRONG RESULT
5$: MOV #205,R1 ;LOAD R1 WITH TEST INSTR. WORD
MOV #NBUF0,R2 ;R2 POINTS TO DEST OP
MOV #377,(R2) ;DESTJ = 000377
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
```

```

5177 015200 000322 20$: SWAB (R2)+ ;TEST THE SWAB - DM2
5178 015202 100403 BMI 7$ ;N:C = 0100
5179 015204 001002 BNE 7$
5180 015206 102401 BVS 7$
5181 015210 103001 BCC 6$
5182 015212 104001 7$: ERROR 1 ;SWAB FAILED TO SET CODES PROPERLY
5183 015214 020437 063312 6$: CMP R4,#MBUFO ;CORRECT RESULT ?
5184 015220 001401 BEQ 8$ ;BR IF YES
5185 015222 104001 9$: ERROR 1 ;SWAB DELIVERED THE WRONG RESULT
5186 015224 020227 063314 8$: CMP R2,#MBUFO+2 ;DID AUTO INCREMENT OCCUR ?
5187 015230 001401 BEQ TS4231 ;BR IF YES
5188 015232 104001 ERROR 1 ;SWAB FAILED TO AUTO INC REG.
;*****
;TEST 231 SWAB MODE 1 TEST WITH NEG. RESULT
;*****
;T231:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #231,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #377,R4 ;RESULT S / R = 377
MOV #MBUFO,R2 ;R2 POINTS TO DEST OP
MOV #177400,(R2) ;SET UP DEST. OP = 177400
CCC ;CLEAR FLAGS
267 ;MAKE N:C = 0111
5190 015266 000312 2$: SWAB (R2) ;TEST THE SWAB
5191 015270 100003 RPL 3$
5192 015272 001402 BEQ 3$ ;N:C = 1000?
5193 015274 102401 BVS 3$
5194 015276 103001 BCC 4$
5195 015300 104001 3$: ERROR 1 ;SWAB FAILED TO ALTER CODES PROPERLY
5196 015302 020412 4$: CMP R4,(R2) ;CORRECT RESULT?
5197 015304 001401 BEQ TS1232 ;BR IF YES
5198 015306 104001 5$: ERROR 1 ;SWAB DELIVERED WRONG RESULT
;*****
;TEST 232 NEG MODE 0 TEST : [DEST] = 0
;*****
;T232:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #232,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 0
CLR R3 ;INITIAL [DEST] = 0
5200 015310 000004 000231
5201 015312 012700 000232
5202 015314 013701 015332
5203 015316 013701 015332
5204 015318 005004
5205 015320 005003
5206 015324 005003

```

```

5233 015326 000257 CCC ;CLEAR FLAGS
5234 015330 000273 273 ;MAKE N:C = 1011
5235 015332 005403 2$: NEG R3 ;TEST THE NEG
5236 015334 100403 BMI 3$
5237 015336 001002 BNE 3$ ;N:C = 0100 ONLY "Z" SET?
5238 015340 102401 BVS 3$
5239 015342 103001 BCC 4$
5240 015344 104002 3$: ERROR 2 ;NEG FAILED TO ALTER CODES PROPERLY
5241 015346 020304 4$: CMP R3,R4 ;WAS RESULT = 0
5242 015350 001401 BEQ TS4233 ;BR IF YES
5243 015352 104002 5$: ERROR 2 ;NEG DELIVERED WRONG RESULT
;*****
;TEST 233 NEG MODE 0 TEST : [DEST] LT 0
;*****
;T233:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #233,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #2,R4 ;RESULT S / B = 2
MOV #-2,R3 ;INITIAL [DEST] = 177776
CCC ;CLEAR FLAGS
276 ;MAKE N:C = 1110
5245 015402 005403 2$: NEG R3 ;TEST THE NEG
5246 015404 100403 BMI 3$
5247 015406 001402 BEQ 3$ ;N:C = 0001?
5248 015410 102401 BVS 3$
5249 015412 103401 BCS 4$
5250 015414 104002 3$: ERROR 2 ;NEG FAILED TO ALTER CODES PROPERLY
5251 015416 020304 4$: CMP R3,R4 ;RESULT = 2?
5252 015420 001401 BEQ TS4234 ;BR IF YES
5253 015422 104002 5$: ERROR 2 ;NEG DELIVERED WRONG RESULT
;*****
;TEST 234 NEG MODE 0 TEST : [DEST] = 100000 (B)
;*****
;T234:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #234,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100000,R4 ;RESULT S / B = 100000
MOV R4,R3 ;INITIAL [DEST] = 100000
CCC ;CLEAR FLAGS
SEZ ;MAKE N:C = 010000
5255 015450 005403 2$: NEG R3 ;TEST THE NEG

```



```

52 89
53 90 015452 100003
54 91 015453 001402
55 92 015454 102001
56 93 015455 103401
57 94
58 95 015462 104002
59 96
60 97 015464 020304
61 98 015466 001401
62 99
63 00 015470 104002
64 01
65 02
66 03
67 04
68 05 015472
69 06 015473 000004
70 07 015474 012700 000235
71 08 015500 013701 015520
72 09 015504 012702 063312
73 10 015510 005004
74 11 015512 005012
75 12 015514 000257
76 13 015516 000273
77 14
78 15 015520 005412
79 16 015522 100403
80 17 015524 001002
81 18 015526 102401
82 19 015530 103001
83 20
84 21
85 22 015532 104001
86 23
87 24 015534 021204
88 25 015536 001401
89 26
90 27 015540 104001
91 28
92 29
93 30
94 31 015542
95 32 015543 000004
96 33 015544 012700 000236
97 34 015550 013701 015574
98 35 015554 012702 063312
99 36 015556 012704 177776
00 37 015560 012704 177776
01 38 015564 012712 000002
02 39 015570 000257
03 40 015572 000266
04 41
05 42 015574 005412
06 43
07 44 015576 100003

```

```

BPL 3$
BEQ 3$ ;N:C = 1011?
BVC 3$
BCS 4$
3$: ERROR 2 ;NEG FAILED TO ALTER CODES PROPERLY
4$: CMP R3,R4 ;RESULT STILL 100000?
BEQ TST235 ;;BR IF YES
5$: ERROR 2 ;NEG DELIVERED WRONG RESULT
;*****
;TEST 235 NEG MODE 1 TEST : [DEST] = 0
;*****
TST235: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #235,R0 ;;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #B0F0,R2 ;R2 POINTS TO DEST OP
CLR R4 ;RESULT S / B = 0
CLR (R2) ;INITIAL [DEST] = 0
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
2$: NEG (R2) ;TEST THE NEG
BMI 3$
BNE 3$ ;N:C = 0100?
BVS 3$
BCC 4$
3$: ERROR 1 ;NEG FAILED TO ALTER CODES PROPERLY
4$: CMP (R2),R4 ;RESULT = 0?
BEQ TST236 ;;BR IF YES
5$: ERROR 1 ;NEG DELIVERED WRONG RESULT
;*****
;TEST 236 NEG MODE 1 TEST : [DEST] GT 0
;*****
TST236: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #236,R0 ;;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #B0F0,R2 ;R2 POINTS TO DEST OP
MOV #2,R4 ;RESULT S / B = 177776
MOV #2,(R2) ;INITIAL [DEST] = 2
CCC ;CLEAR FLAGS
266 ;MAKE N:C = 0110
2$: NEG (?) ;TEST THE NEG
BPL 3$

```

```

53 45 015600 001402
54 46 015602 102401
55 47 015604 103401
56 48
57 49 015606 104001
58 50 015610 021204
59 51 015612 001401
60 52
61 53 015614 104001
62 54
63 55
64 56 015616
65 57 015616 000004
66 58 015620 013700 000237
67 59 015624 013701 015650
68 60 015630 012702 063312
69 61 015634 012704 000002
70 62 015640 012712 177776
71 63 015644 000257
72 64 015646 000276
73 65
74 66 015650 005412
75 67
76 68 015652 100403
77 69 015654 001402
78 70 015656 102401
79 71 015660 103401
80 72
81 73
82 74 015662 104001
83 75
84 76 015664 021204
85 77 015666 001401
86 78
87 79
88 80 015670 104001
89 81
90 82
91 83
92 84 015672
93 85 015672 000004
94 86 015674 012700 000240
95 87 015700 013701 015722
96 88 015704 012702 063312
97 89 015710 012704 100000
98 90 015714 018412
99 91 015716 000257
00 92 015720 000264
01 93
02 94 015722 005412
03 95
04 96 015724 100003
05 97 015726 001402
06 98 015730 102001

```

```

BEQ 3$
BVS 3$ ;N:C = 1001?
BCS 4$
3$: ERROR 1 ;NEG FAILED TO ALTER CODES PROPERLY
4$: CMP (R2),R4 ;CORRECT RESULT?
REQ TST237 ;;BR IF YES
5$: ERROR 1 ;NEG DELIVERED WRONG RESULT
;*****
;TEST 237 NEG MODE 1 TEST : [DEST] LT 0
;*****
TST237: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #237,R0 ;;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #B0F0,R2 ;R2 POINTS TO DEST OP
MOV #2,R4 ;RESULT S / B = 2
MOV #2,(R2) ;INITIAL [DEST] = 177776
CCC ;CLEAR FLAGS
276 ;MAKE N:C = 1110
2$: NEG (R2) ;TEST THE NEG
BMI 3$
BEQ 3$ ;N:C = 0001?
BVS 3$
BCS 4$
3$: ERROR 1 ;NEG FAILED TO ALTER CODES PROPERLY
4$: CMP (R2),R4 ;CORRECT RESULT = 2?
BEQ TST240 ;;BR IF YES
5$: ERROR 1 ;NEG DELIVERED WRONG RESULT
;*****
;TEST 240 NEG MODE 1 TEST : [DEST] = 100000 (8)
;*****
TST240: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #240,R0 ;;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #B0F0,R2 ;R2 POINTS TO DEST OP
MOV #100000,R4 ;RESULT S / B = 100000
MOV R4,(R2) ;INITIAL [DEST] = 100000
CCC ;CLEAR FLAGS
SEZ ;MAKE N:Z = 0100
2$: NEG (R2) ;TEST THE NEG
BPL 3$
BEQ 3$ ;N:C = 1011?
BVC 3$

```

5401 015732 103401
5402
5403 015734 104001
5404
5405 015736 021204
5406 015740 001401
5407
5408 015742 104001
5409
5410
5411
5412
5413 015744
5414 015744 000004
5415 015746 012700 000241
5416 015752 013701 015772
5417 015758 012704 052525
5418 015762 012703 125252
5419 015766 000257
5420 015770 000276
5421
5422 015772 006003
5423
5424 015774 100403
5425 015776 001402
5426 016000 102401
5427 016002 103001
5428
5429 016004 104002
5430
5431 016006 020403
5432 016010 001401
5433
5434 016012 104002
5435
5436
5437
5438
5439 016014
5440 016014 000004
5441 016016 012700 000242
5442 016022 013701 016040
5443 016026 005004
5444 016030 012703 000001
5445 016034 000257
5446 016036 000270
5447
5448 016040 006003
5449
5450 016042 100403
5451 016044 001002
5452 016046 101001
5453 016050 103401
5454
5455 016052 104002
5456

BCS 4\$
3\$: ERROR 1 ;NEG FAILED TO ALTER CODES PROPERLY
4\$: CMP R(2),R4 ;CORRECT RESULT = 10000?
BEQ TST241 ;;BR IF YES
5\$: ERROR 1 ;NEG DELIVERED WRONG RESULT
;*****
;TEST 241 ROR TEST - DMO - N:C = 1110
;*****
TST241: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #241,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #525,R4 ;RESULT S / B = 52525
MOV #12525,R3 ;[DEST] = 125252
CCC ;CLEAR FLAGS
276 ;N:C = 1111
2\$: ROR R3 ;TEST THE ROR
BMI 3\$;N:C = 0000 ?
BEQ 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST242 ;;BR IF YES
5\$: ERROR 2 ;ROR DELIVERED THE WRONG RESULT
;*****
;TEST 242 ROR TEST - DMO - N:C = 1000
;*****
TST242: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #242,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #1,R3 ;[DEST] =
CCC ;CLEAR FLAGS
SEN ;N:C = 1000
2\$: ROR R3 ;TEST THE ROR
BMI 3\$;N:C = 0111 ?
BNE 3\$
BCS 4\$
3\$: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY

5457 016054 020403
5458 016056 001401
5459
5460 016060 104002
5461
5462
5463
5464
5465 016062
5466 016062 000004
5467 016064 012700 000243
5468 016070 013701 016110
5469 016074 012704 052525
5470 016078 012703 052525
5471 016104 000257
5472 016106 000267
5473
5474 016110 006003
5475
5476 016112 100003
5477 016114 001402
5478 016116 102401
5479 016120 103401
5480
5481 016122 104002
5482
5483 016124 020403
5484 016126 001401
5485
5486 016130 104002
5487
5488
5489
5490
5491 016132
5492 016132 000004
5493 016134 012700 000244
5494 016140 014701 016156
5495 016144 005004
5496 016146 012703 000001
5497 016152 000257
5498 016154 000270
5499
5500 016156 006003
5501
5502 016160 100403
5503 016162 001002
5504 016164 102001
5505 016166 103401
5506
5507 016170 104002
5508
5509 016172 020403
5510 016174 001401
5511
5512 016176 104002

4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST243 ;;BR IF YES
5\$: ERROR 2 ;ROR DELIVERED THE WRONG RESULT
;*****
;TEST 243 ROR TEST - DMO - N:C = 0111
;*****
TST243: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #243,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #525,R4 ;RESULT S / B = 125252
MOV #52525,R3 ;[DEST] = 52525
CCC ;CLEAR FLAGS
267 ;N:C = 0111
2\$: ROR R3 ;TEST THE ROR
BPL 3\$;N:C = 1001 ?
BEQ 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST244 ;;BR IF YES
5\$: ERROR 2 ;ROR DELIVERED THE WRONG RESULT
;*****
;TEST 244 ASR TEST - DMO - N:C = 1000
;*****
TST244: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #244,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #1,R3 ;[DEST] = 1
CCC ;CLEAR FLAGS
SEN ;N:C = 1000
2\$: ROR R3 ;TEST THE ROR
BMI 3\$;N:C = 0111 ?
BNE 3\$
BCS 4\$
3\$: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST245 ;;BR IF YES
5\$: ERROR 2 ;ROR DELIVERED THE WRONG RESULT

```

5513
5514
5515
5516
5517
5518 016200
5519 016200 000004
5520 016202 012700 000245
5521 016206 013701 016226
5522 016212 012704 152525
5523 016215 016703 125252
5524 016224 000265
5525
5526 016226 006003
5527
5528 016230 100003
5529 016232 001402
5530 016234 102001
5531 016236 103001
5532
5533 016240 104002
5534
5535 016242 020403
5536 016244 001401
5537
5538 016246 104002
5539
5540
5541
5542
5543 016250
5544 016250 000004
5545 016252 012700 000246
5546 016252 012701 016276
5547 016252 012704 052525
5548 016266 012703
5549 016272 000257
5550 016274 000274
5551
5552 016276 006003
5553
5554 016300 100403
5555 016302 001402
5556 016304 102001
5557 016306 103401
5558
5559 016310 104002
5560
5561 016312 020403
5562 016314 001401
5563
5564 016316 104002
5565
5566
5567
5568
;*****
; *TEST 245 ASR TEST - DM0 - N:C = 0101
;*****
TST245:
SCOPE #245,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV #52525,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #125252,R3 ;RESULT S / B = 152525
CCC ;DESTJ = 125252
CLC ;CLEAR FLAGS
N:C = 0101

2S: ROR R3 ;TEST THE ROR

BPL 3S ;N:C = 1010 ?
BEQ 3S
BVC 3S
BCC 4S

3S: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY

4S: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST246 ;BR IF YES

5S: ERROR 2 ;ROR DELIVERED THE WRONG RESULT
;*****
; *TEST 246 ASR TEST - DM0 - N:C = 1100
;*****
TST246:
SCOPE #246,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV #52525,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #125252,R3 ;RESULT S / B = 25252
CCC ;DESTJ = 52525
CLC ;CLEAR FLAGS
N:C = 1100

2S: ROR R3 ;TEST THE ROR

BMI 3S ;N:C = 0011 ?
BEQ 3S
BVC 3S
BCC 4S

3S: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY

4S: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST247 ;BR IF YES

5S: ERROR 2 ;ROR DELIVERED THE WRONG RESULT
;*****
; *TEST 247 ROR TEST - DM1 - N:C = 1110
;*****

```

```

5569 016320
5570 016320 000004
5571 016322 012700 000247
5572 016326 013701 016352
5573 016332 012702 063312
5574 016336 012704 052525
5575 016342 012712 125252
5576 016350 000276
5577 016352
5578
5579 016352 006012
5580
5581 016354 100403
5582 016356 001402
5583 016360 102401
5584 016362 103001
5585
5586 016364 104001
5587
5588 016366 020412
5589 016370 001402
5590 016372 011203
5591 016374 104001
5592
5593
5594
5595
5596 016376
5597 016376 000004
5598 016400 012700 000250
5599 016400 013701 016426
5600 016410 012702 063312
5601 016414 005004
5602 016416 012712 000001
5603 016422 000257
5604 016424 000270
5605
5606 016426 006012
5607
5608 016430 100403
5609 016432 001002
5610 016434 102001
5611 016436 103401
5612
5613 016440 104001
5614
5615 016442 020412
5616 016444 001402
5617
5618 016446 011203
5619 016450 104001
5620
5621
5622
5623
5624 016452
;*****
; *TEST 247 ROR TEST - DM1 - N:C = 1110
;*****
TST247:
SCOPE #247,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #52525,R4 ;DEST ADDR = MBUF0
MOV #125252,(R2) ;RESULT S / B = 52525
CCC ;DESTJ = 125252
CLC ;CLEAR FLAGS
N:C = 1110

2S: ROR (R2) ;TEST THE ROR

BMI 3S ;N:C = 0000 ?
BEQ 3S
BVC 3S
BCC 4S

3S: ERROR 1 ;ROR FAILED TO ALTER CODES PROPERLY

4S: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST250 ;BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ROR DELIVERED WRONG RESULT
;*****
; *TEST 250 ROR TEST - DM1 - N:C = 1000
;*****
TST250:
SCOPE #250,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;DEST ADDR = MBUF0
MOV #1,(R2) ;RESULT S / B = 000000
CCC ;DESTJ = 1
SEN ;CLEAR FLAGS
N:C = 1000

2S: ROR (R2) ;TEST THE ROR

BMI 3S ;N:C = 0111 ?
BNE 3S
BVC 3S
BCC 4S

3S: ERROR 1 ;ROR FAILED TO ALTER CODES PROPERLY

4S: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST251 ;BR IF YES

5S: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ROR DELIVERED WRONG RESULT
;*****
; *TEST 251 ROR TEST - DM1 - N:C = 0111
;*****
TST251:

```

675 016452 000004
676 016454 012700 000251
677 016460 013701 016504
678 016464 012702 063312
679 016470 012703 063312
680 016500 006257 052525
681 016502 000267
682 016504 006012
683 016506 100003
684 016510 001402
685 016512 102401
686 016514 103401
687 016516 104001
688 016520 020412
689 016522 001402
690 016524 011203
691 016526 104001
692 016530 000004
693 016532 012700 000252
694 016536 013701 016560
695 016542 012702 063312
696 016546 005004
697 016550 012749 000001
698 016554 000257
699 016556 000270
700 016560 006012
701 016562 100403
702 016564 001402
703 016566 102001
704 016570 103401
705 016572 104001
706 016574 020412
707 016576 001402
708 016600 011203
709 016602 104001
710 016604

```
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #251,R0 ;LOAD R0 WITH TEST NUMBER
MOV #251,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #12525,R4 ;RESULT S / B = 125252
MOV #52525,(R2) ;DESTJ = 52525
CCC ;CLEAR FLAGS
267 ;N:C = 0111

2$: ROR (R2) ;TEST THE ROR

BPL 3$ ;N:C = 1001 ?
BEQ 3$
BVC 3$
BCS 4$

3$: ERROR 1 ;ROR FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TS1252 ;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ROR DELIVERED WRONG RESULT

;*****
;TEST 252 ASR TEST - DM1 - N:C = 1000
;*****
TST252:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #252,R0 ;LOAD R0 WITH TEST NUMBER
MOV #252,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
MOV #1,(R2) ;DESTJ = 1
CCC ;CLEAR FLAGS
SPW ;N:C = 1000

2$: ROR (R2) ;TEST THE ROR

BMI 3$ ;N:C = 0111 ?
BEQ 3$
BVC 3$
BCS 4$

3$: ERROR 1 ;ROR FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TS1253 ;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ROR DELIVERED WRONG RESULT

;*****
;TEST 253 ASR TEST - DM1 - N:C = 1100
;*****
TST253:
```

681 016604 000004
682 016606 012700 000253
683 016612 013701 016636
684 016616 012702 063312
685 016622 012703 025252
686 016626 012712 052525
687 016632 000257
688 016634 000274
689 016636 006012
690 016640 100403
691 016642 001402
692 016644 102001
693 016646 103401
694 016650 104001
695 016652 020412
696 016654 001402
697 016656 011203
698 016660 104001
699 016662
700 016662 000004
701 016664 012700 000254
702 016670 013701 016714
703 016674 012702 063312
704 016700 012704 152525
705 016704 012712 125252
706 016710 000257
707 016712 000265
708 016714 006012
709 016716 100003
710 016720 001402
711 016722 102001
712 016724 103001
713 016726 104001
714 016730 020412
715 016732 001402
716 016734 011203
717 016736 104001
718 016740

```
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #253,R0 ;LOAD R0 WITH TEST NUMBER
MOV #253,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #25252,R4 ;RESULT S / B = 252522
MOV #52525,(R2) ;DESTJ = 52525
CCC ;CLEAR FLAGS
274 ;N:C = 1100

2$: ROR (R2) ;TEST THE ROR

BMI 3$ ;N:C = 0011 ?
BEQ 3$
BVC 3$
BCS 4$

3$: ERROR 1 ;ROR FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TS1254 ;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ROR DELIVERED WRONG RESULT

;*****
;TEST 254 ASR TEST - DM1 - N:C = 0101
;*****
TST254:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #254,R0 ;LOAD R0 WITH TEST NUMBER
MOV #254,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #152525,R4 ;RESULT S / B = 152525
MOV #125252,(R2) ;DESTJ = 125252
CCC ;CLEAR FLAGS
265 ;N:C = 0101

2$: ROR (R2) ;TEST THE ROR

BPL 3$ ;N:C = 1010 ?
BEQ 3$
BVC 3$
BCS 4$

3$: ERROR 1 ;ROR FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TS1255 ;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ROR DELIVERED WRONG RESULT

;*****
;TEST 255 RORB TEST - DM2 - EVEN ADDRESS
;*****
TST255:
```

5737 016740 000004
5738 016742 012700 000255
5739 016746 013701 016772
5740 016752 012702 063312
5741 016756 012704 000177
5742 016762 012203
5743 016764 012712 000377
5744 016770 000257
5745
5746 016772 106023
5747
5748 016774 103401
5749
5750 016776 104001
5751
5752 017000 022703 063313
5753 017004 001401
5754
5755 017006 104005
5756
5757 017010 020412
5758 017012 001402
5759
5760 017014 011203
5761 017016 104001
5762
5763
5764
5765
5766 017020
5767 017020 000004
5768 017022 012700 000256
5769 017026 013701 017054
5770 017036 012702 063312
5771 017038 012704 000377
5772 017042 010203
5773 017044 012712 000376
5774 017050 000257
5775 017052 000261
5776
5777 017054 106013
5778
5779 017056 103001
5780
5781 017060 104001
5782
5783 017062 020412
5784 017064 001402
5785
5786 017066 011203
5787 017070 104001
5788
5789
5790
5791
5792 017072

```
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #255,R0 ;LOAD R0 WITH TEST NUMBER
MOV #255,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177,R4 ;RESULT S / B = 177
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #377,(R2) ;DESTJ = 377
CCC ;SCOPE SYNC "C" = 0

2$: RORB (R3)+ ;TEST THE RORB
BCS 4$ ;BR IF ROR SET "C"
3$: ERROR 1 ;ROR FAILED TO SET "C"
4$: CMP #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED ?
BEQ 6$ ;BR IF YES
5$: ERROR 5 ;RORB FAILED TO UPDATE DEST REG
6$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ T$T256 ;BR IF YES
7$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;RORB DELIVERED WRONG RESULT

;*****
;TEST 256 RORB TEST - DM1 - EVEN ADDRESS
;*****
T$T256:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #256,R0 ;LOAD R0 WITH TEST NUMBER
MOV #256,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #376,(R2) ;DESTJ = 376
CCC ;CLEAR FLAGS
SEC ;SCOPE SYNC - SET "C"

2$: RORB (R3) ;TEST THE RORB
BCC 4$ ;BR IF "C" CLR - IT SHOULD BE
3$: ERROR 1 ;RORB FAILED TO CLR "C"
4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ T$T257 ;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;RORB DELIVERED WRONG RESULT

;*****
;TEST 257 RORB TEST - DM2 - ODD ADDRESS
;*****
T$T257:
```

5793 017072 000004
5794 017074 012700 000257
5795 017100 013701 017142
5796
5797 017104 032737 000040 063234
5798 017112 001401
5799 017114 000000
5800 017116 012704 063313
5801 017118 012704 077777
5802 017126 012705 063312
5803 017132 010203
5804 017134 012715 177777
5805 017140 000257
5806
5807 017142 106023
5808
5809 017144 103401
5810
5811 017146 104001
5812
5813 017150 022703 063314
5814 017154 001401
5815
5816 017156 104005
5817
5818 017160 020415
5819 017162 001402
5820
5821 017164 011503
5822 017166 104001
5823
5824
5825
5826
5827 017170
5828 017170 000004
5829 017172 012700 000260
5830 017176 013701 017226
5831 017202 012702 063313
5832 017206 012704 177777
5833 017212 012705 063312
5834 017216 010203
5835 017220 012715 177377
5836 017224 000261
5837
5838 017226 106023
5839
5840 017230 103001
5841
5842 017232 104001
5843
5844 017234 020415
5845 017236 001402
5846
5847 017240 011503
5848 017242 104001

```
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #257,R0 ;LOAD R0 WITH TEST NUMBER
MOV #257,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
.SBTTL USER CONTROLLED BREAKPOINT -- BITS
BIT #BITS,@#BPTLOC ;BREAKPOINT HALT SET ??
BEQ .+4 ;BR IF NOT
HALT ;BREAK - DEPRESS CONTINUE TO RESTART
MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
MOV #17777,R4 ;RESULT S / B = 17777
MOV #MBUF0,R5 ;POINT R5 TO CHECK RESULT
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R5) ;DESTJ = 17777
CCC ;SCOPE SYNC - "C" = 0

2$: RORB (R3)+ ;TEST THE RORB
BCS 4$ ;BR IF "C" IS SET - IT SHOULD BE
3$: ERROR 1 ;RORB FAILED TO SET "C"
4$: CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED ?
BEQ 6$ ;BR IF YES
5$: ERROR 5 ;RORB FAILED TO UPDATE DEST REG
6$: CMP R4,(R5) ;CORRECT RESULT ?
BEQ T$T260 ;BR IF YES
7$: MOV (R5),R3 ;GET THE WAS DATA
ERROR 1 ;RORB DELIVERED WRONG RESULT

;*****
;TEST 260 RORB TEST - DM1 - ODD ADDRESS
;*****
T$T260:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #260,R0 ;LOAD R0 WITH TEST NUMBER
MOV #260,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
MOV #-1,R4 ;RESULT S / B = 17777
MOV #MBUF0,R5 ;POINT R5 TO CHECK RESULT
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #177377,(R5) ;DESTJ = 177377
SEC ;SCOPE SYNC - SET "C"

2$: RORB (R3)+ ;TEST THE RORB
BCC 4$ ;BR IF "C" CLEAR - IT SHOULD BE
3$: ERROR 1 ;RORB FAILED TO CLEAR "C"
4$: CMP R4,(R5) ;CORRECT RESULT ?
BEQ T$T261 ;BR IF YES
5$: MOV (R5),R3 ;GET THE WAS DATA
ERROR 1 ;RORB DELIVERED WRONG RESULT
```

```

*****
TEST 261 ASRB TEST - DM2 - ODD ADDRESS
*****
TST261:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #261,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
MOV #377,R4 ;RESULT S / B = 377
MOV #MBUF0,R5 ;POINT R5 TO CHECK RESULT
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #777,(R5) ;DESTJ = 777
CCC ;SCOPE SYNC "C" = 0

25: ASRB (R3)+ ;TEST THE ASRB
BCS 45 ;BR IF CARRY SET - IT SHOULD BE

35: ERROR 1 ;ASRB FAILED TO SET THE CARRY
45: CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED ?
BEQ 65 ;BR IF YES

55: ERROR 5 ;ASRB FAILED TO UPDATE DEST REG
65: CMP R4,(R5) ;CORRECT RESULT ?
BEQ TST262 ;BR IF YES

75: MOV (R5),R3 ;GET THE WAS DATA
ERROR 1 ;ASRB DELIVERED WRONG RESULT

*****
TEST 262 ASRB TEST - DM1 - ODD ADDRESS
*****
TST262:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #262,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
MOV #140377,R4 ;RESULT S / B = 140377
MOV #MBUF0,R5 ;POINT R5 TO CHECK RESULT
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #100377,(R5) ;DESTJ = 100377
SEC ;SCOPE SYNC - "C" = 1

25: ASRB (R3) ;TEST THE ASRB
BCC 45 ;BR IF CARRY CLEAR - IT SHOULD BE

35: ERROR 1 ;ASRB FAILED TO CLEAR THE CARRY
45: CMP R4,(R5) ;CORRECT RESULT ?
BEQ TST263 ;BR IF YES

75: MOV (R5),R3 ;GET THE WAS DATA

```

```

5905 017402 104001
5906
5907
5908
5909
5910 017404
5911 017404 000004
5912 017406 012700 000263
5913 017412 013701 017436
5914 017416 012702 063312
5915 017422 012704 000077
5916 017426 012703 000177
5917 017434 006257
5918
5919
5920 017436 106223
5921
5922 017440 103401
5923
5924 017442 104001
5925
5926 017444 022703 063313
5927 017450 001401
5928
5929 017452 104005
5930
5931 017454 020412
5932 017456 001402
5933
5934 017460 011203
5935 017462 104001
5936
5937
5938
5939
5940 017464
5941 017464 000004
5942 017466 012700 000264
5943 017472 013701 017516
5944 017476 012702 063312
5945 017502 012704 000303
5946 017506 012703 000206
5947 017510 012712
5948 017514 000261
5949
5950 017516 106213
5951
5952 017520 103001
5953
5954 017522 104001
5955
5956 017524 020412
5957 017526 001402
5958
5959 017530 011203
5960 017532 104001

```

```

55: ERROR 1 ;ASRB DELIVERED WRONG RESULT

*****
TEST 263 ASRB TEST - DM2 - EVEN ADDRESS
*****
TST263:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #263,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #776,R4 ;RESULT S / B = 776
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #177,(R2) ;DESTJ = 177
CCC ;SCOPE SYNC - "C" = 0

25: ASRB (R3)+ ;TEST THE ASRB
BCS 45 ;BR IF "C" = 1 - IT SHOULD BE

35: ERROR 1 ;ASRB FAILED TO SET "C"
45: CMP #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED ?
BEQ 65 ;BR IF YES

55: ERROR 5 ;ASRB FAILED TO UPDATE DEST REG
65: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST264 ;BR IF YES

75: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ASRB DELIVERED WRONG RESULT

*****
TEST 264 ASRB TEST - DM1 - EVEN ADDRESS
*****
TST264:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #264,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #303,R4 ;RESULT S / B = 303
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #206,(R2) ;DESTJ = 206
SEC ;SCOPE SYNC - "C" = 1

25: ASRB (R3) ;TEST THE CLRASRB
BCC 45 ;BR IF CARRY CLEAR - ITSHOULD BE

35: ERROR 1 ;ASRB FAILED TO CLEAR THE CARRY
45: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST265 ;BR IF YES

55: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ASRB DELIVERED WRONG RESULT

```



```

6077 017772 012704 125252      MOV #125252,R4 ;RESULT S / R = 125252
6078 017776 012703 052525      MOV #52525,R3 ;CDESTJ = 52525
6079 020002 000257      CCC ;CLEAR CODES
6080 020004 000266      266 ;N:C = 0110
6081 020006 005103      2$: COM R3 ;TEST THE COM
6082 020010 100003      BPL 3$ ;N:C = 1001 ?
6083 020012 001402      BEQ 3$
6084 020014 102401      BVS 3$
6085 020016 103401      BCS 4$
6086 020020 104002      3$: ERROR 2 ;COM FAILED TO ALTER THE CODES PROPERLY
6087 020022 020403      4$: CMP R4,R3 ;RESULT OK ?
6088 020024 001401      BEQ TST272 ;;BR IF YES
6089 020026 104002      5$: ERROR 2 ;COM DELIVERED THE WRONG RESULT
6090
6091
6092 ;*****
6093 ;*TEST 272 COM DMO TEST - N:C = 1001
6094 ;*****
6095 TST272:
6096 020030 000004      SCOPE ;CALL THE SCOPE LOOP UTILITY
6097 020032 012700 000272      MOV #272,R0 ;;LOAD R0 WITH TEST NUMBER
6098 020034 013701 020054      MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
6099 020036 005004      CLR R4 ;RESULT S / R = 000000
6100 020038 012703 177777      MOV #-1,R3 ;CDESTJ = 177777
6101 020040 000257      CCC ;CLEAR CODES
6102 020042 000271      271 ;N:C = 1001
6103
6104 020054 005103      2$: COM R3 ;TEST THE COM
6105 020056 100403      BMI 3$ ;N:C = 0101 ?
6106 020058 101402      BNE 3$
6107 020060 102401      BVS 3$
6108 020062 103401      BCS 4$
6109 020064 104002      3$: ERROR 2 ;COM FAILED TO ALTER THE CODES PROPERLY
6110
6111 020066 104002      4$: CMP R4,R3 ;RESULT OK ?
6112 020070 020403      BEQ TST273 ;;BR IF YES
6113 020072 001401      5$: ERROR 2 ;COM DELIVERED THE WRONG RESULT
6114
6115 020074 104002
6116
6117 ;*****
6118 ;*TEST 273 INC DMO TEST - N:C = 1011
6119 ;*****
6120 TST273:
6121 020076 000004      SCOPE ;CALL THE SCOPE LOOP UTILITY
6122 020078 012700 000273      MOV #273,R0 ;;LOAD R0 WITH TEST NUMBER
6123 020104 013701 020122      MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
6124 020106 005004      CLR R4 ;RESULT S / R = 000000
6125 020108 012703 177777      MOV #-1,R3 ;CDESTJ = 177777
6126 020110 000257      CCC ;CLEAR CODES
6127 020112 000273      273 ;N:C = 1011
6128

```

```

6129 020122 005203      2$: INC R3 ;TEST THE INC
6130
6131 020124 100403      BMI 3$ ;N:C = 0101 ?
6132 020126 001002      BNE 3$
6133 020130 102401      BVS 3$
6134 020132 103401      BCS 4$
6135 020134 104002      3$: ERROR 2 ;INC FAILED TO ALTER THE CODES PROPERLY
6136 020136 020403      4$: CMP R4,R3 ;RESULT OK ?
6137 020140 001401      BEQ TST274 ;;BR IF YES
6138 020142 104002      5$: ERROR 2 ;INC DELIVERED THE WRONG RESULT
6139
6140 ;*****
6141 ;*TEST 274 INC DMO TEST - N:C = 0100
6142 ;*****
6143 TST274:
6144 020144 000004      SCOPE ;CALL THE SCOPE LOOP UTILITY
6145 020146 012700 000274      MOV #274,R0 ;;LOAD R0 WITH TEST NUMBER
6146 020152 013701 020172      MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
6147 020154 012704 100000      MOV #100000,R4 ;RESULT S / R = 100000
6148 020156 012703 077777      MOV #77777,R3 ;CDESTJ = 77777
6149 020158 000257      CCC ;CLEAR CODES
6150 020160 000264      264 ;N:C = 0100
6151
6152 020172 005203      2$: INC R3 ;TEST THE INC
6153 020174 100003      BPL 3$ ;N:C = 1010 ?
6154 020176 001402      BEQ 3$
6155 020180 102001      BVC 3$
6156 020202 103001      BCC 4$
6157 020204 104002      3$: ERROR 2 ;INC FAILED TO ALTER THE CODES PROPERLY
6158 020206 020403      4$: CMP R4,R3 ;RESULT OK ?
6159 020210 001401      BEQ TST275 ;;BR IF YES
6160 020212 104002      5$: ERROR 2 ;INC DELIVERED THE WRONG RESULT
6161
6162 ;*****
6163 ;*TEST 275 DEC DMO TEST - N:C = 1011
6164 ;*****
6165 TST275:
6166 020214 000004      SCOPE ;CALL THE SCOPE LOOP UTILITY
6167 020216 012700 000275      MOV #275,R0 ;;LOAD R0 WITH TEST NUMBER
6168 020222 013701 020240      MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
6169 020224 005004      CLR R4 ;RESULT S / R = 000000
6170 020226 012703 000001      MOV #-1,R3 ;CDESTJ = 1
6171 020228 000257      CCC ;CLEAR CODES
6172 020230 000273      273 ;N:C = 1011
6173
6174 020240 005303      2$: DEC R3 ;TEST THE DEC
6175 020242 100403      BMI 3$ ;N:C = 0101 ?
6176

```



```

6185 020244 001002
6186 020246 102401
6187 020250 103401
6188
6189 020252 104002
6190
6191 020254 020403
6192 020256 001401
6193
6194 020260 104002
6195
6196
6197
6198
6199 020262
6200 020262 000004
6201 020264 012700 000276
6202 020270 013701 020310
6203 020274 012704 077777
6204 020300 011703 100000
6205 020304 000257
6206 020306 000274
6207
6208 020310 005303
6209
6210 020312 100403
6211 020314 001402
6212 020316 102001
6213 020320 103001
6214
6215 020322 104002
6216
6217 020324 020403
6218 020326 001401
6219
6220 020330 104002
6221
6222
6223
6224
6225 020332
6226 020332 000004
6227 020334 012700 000277
6228 020340 013701 020354
6229 020344 012704 177777
6230 020350 005003
6231 020352 000257
6232
6233 020354 005303

```

```

BNE 3$
BVS 4$
BCS 4$
3$: ERROR 2 ;DEC FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,R3 ;RESULT OK ?
BEQ T$276 ;;BR IF YES
5$: ERROR 2 ;DEC DELIVERED THE WRONG RESULT
;*****
;TEST 276 DEC DMO TEST - N:C = 1100
;*****
T$276: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #276,R0 ;LOAD RO WITH TEST NUMBER
MOV #276,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #17777,R4 ;RESULT S / B = 17777
MOV #100000,R3 ;DESTJ = 100000
CCC ;CLEAR CODES
274 ;N:C = 1100
2$: DEC R3 ;TEST THE DEC
BMI 3$ ;N:C = 0010 ?
BEQ 3$
BVC 3$
BCC 4$
3$: ERROR 2 ;DEC FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,R3 ;RESULT OK ?
BEQ T$277 ;;BR IF YES
5$: ERROR 2 ;DEC DELIVERED THE WRONG RESULT
;*****
;TEST 277 DEC DMO TEST - N:C = 0000
;*****
T$277: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #277,R0 ;LOAD RO WITH TEST NUMBER
MOV #277,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S / B = 177777
CLR R3 ;DESTJ = 000000
CCC ;CLEAR CODES
2$: DEC R3 ;TEST THE DEC

```

```

6234
6235 020356 100003
6236 020360 001402
6237 020362 102401
6238 020364 103001
6239
6240 020366 104002
6241
6242 020370 020403
6243 020372 001401
6244
6245 020374 104002
6246
6247
6248
6249
6250 020376
6251 020376 000004
6252 020400 012700 000300
6253 020404 013701 020422
6254 020410 005003 100000
6255 020412 011703
6256 020416 000257
6257 020420 000270
6258
6259 020422 006303
6260
6261 020424 100403
6262 020426 001002
6263 020430 102001
6264 020432 103401
6265
6266 020434 104002
6267
6268 020436 020403
6269 020440 001401
6270
6271 020442 104002
6272
6273
6274
6275
6276 020444
6277 020444 000004
6278 020446 012700 000301
6279 020452 013701 020472
6280 020456 012704 100000
6281 020462 012703 040000
6282 020466 000257
6283 020470 000265
6284
6285 020472 006303
6286
6287 020474 100003
6288 020476 001402
6289 020500 102001

```

```

BPL 3$ ;N:C = 1000 ?
BEQ 3$
BVS 3$
BCC 4$
3$: ERROR 2 ;DEC FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,R3 ;RESULT OK ?
BEQ T$300 ;;BR IF YES
5$: ERROR 2 ;DEC DELIVERED THE WRONG RESULT
;*****
;TEST 300 ASL DMO TEST - N:C = 1000
;*****
T$300: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #300,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #100000,R3 ;DESTJ = 100000
CCC ;CLEAR CODES
SEN ;N:C = 1000
2$: ASL R3 ;TEST THE ASL
BMI 3$ ;N:C = 0111 ?
BNE 3$
BVC 3$
BCS 4$
3$: ERROR 2 ;ASL FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,R3 ;RESULT OK ?
BEQ T$301 ;;BR IF YES
5$: ERROR 2 ;ASL DELIVERED THE WRONG RESULT
;*****
;TEST 301 ASL DMO TEST - N:C = 0101
;*****
T$301: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #301,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100000,R4 ;RESULT S / B = 100000
MOV #40000,R3 ;DESTJ = 40000
CCC ;CLEAR CODES
265 ;N:C = 0101
2$: ASL R3 ;TEST THE ASL
BPL 3$ ;N:C = 1010 ?
BEQ 3$
BVC 3$

```

```

(299) 020502 103001
(299) 020502 103001
(299) 020502 103001
(299) 020506 020403
(299) 020510 001401
(299) 020512 104002
(299) 020514 000004
(299) 020516 012700
(299) 020522 013701
(299) 020526 005004
(299) 020530 005003
(299) 020532 000257
(299) 020534 000262
(299) 020536 006303
(299) 020540 100403
(299) 020542 001002
(299) 020544 102401
(299) 020546 103001
(299) 020550 104002
(299) 020552 020403
(299) 020554 001401
(299) 020556 104002
(299) 020560 000004
(299) 020562 012700
(299) 020566 013701
(299) 020570 012704
(299) 020576 012703
(299) 020602 000257
(299) 020604 000265
(299) 020606 006103
(299) 020610 100403
(299) 020612 001402
(299) 020614 102001
(299) 020616 103401
(299) 020620 104002

```

```

3$: BCC 4$
4$: CMP R4,R3 ;ASL FAILED TO ALTER THE CODES PROPERLY
   BEQ T$T302 ;RESULT OK ?
   ;BR IF YES
5$: ERROR 2 ;ASL DELIVERED THE WRONG RESULT

;*****
;TEST 302 ASL DMO TEST - N:C = 0010
;*****
T$T302:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #302,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
CLR R3 ;CDESTJ = 000000
CCC ;CLEAR CODES
SEV ;N:C = 0010

2$: ASL R3 ;TEST THE ASL
   BMI 3$ ;N:C = 0100 ?
   BNE 3$
   BVS 3$
   BCC 4$

3$: ERROR 2 ;ASL FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,R3 ;RESULT OK ?
   BEQ T$T303 ;BR IF YES
5$: ERROR 2 ;ASL DELIVERED THE WRONG RESULT

;*****
;TEST 303 ROL DMO TEST - N:C = 1101
;*****
T$T303:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #303,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #525,R4 ;RESULT S / B = 52525
MOV #12525,R3 ;CDESTJ = 125252
CCC ;CLEAR CODES
SEV ;N:C = 1101

2$: ROL R3 ;TEST THE ROL
   BMI 3$ ;N:C = 0011 ?
   BEQ 3$
   BVC 3$
   BCS 4$

3$: ERROR 2 ;ROL FAILED TO ALTER THE CODES PROPERLY

```

```

6346 020622 020403
6347 020624 001401
6348 020626 104002
6349 020630 000004
6350 020632 012700
6351 020636 013701
6352 020642 012704
6353 020646 012703
6354 020652 000257
6355 020654 000265
6356 020656 006103
6357 020660 100003
6358 020662 001402
6359 020664 102001
6360 020666 103001
6361 020670 104002
6362 020672 020403
6363 020674 001401
6364 020676 104002
6365 020700 000004
6366 020702 012700
6367 020706 013701
6368 020712 005004
6369 020714 005003
6370 020716 000257
6371 020720 000262
6372 020722 006103
6373 020724 100403
6374 020726 001002
6375 020730 102401
6376 020732 103001
6377 020734 104002
6378 020736 020403
6379 020740 001401
6380 020742 104002

```

```

4$: CMP R4,R3 ;RESULT OK ?
   BEQ T$T304 ;BR IF YES
5$: ERROR 2 ;ROL DELIVERED THE WRONG RESULT

;*****
;TEST 304 ROL DMO TEST - N:C = 0101
;*****
T$T304:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #304,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #525,R4 ;RESULT S / B = 125253
MOV #12525,R3 ;CDESTJ = 525252
CCC ;CLEAR CODES
SEV ;N:C = 0101

2$: ROL R3 ;TEST THE ROL
   BPL 3$ ;N:C = 1010 ?
   BEQ 3$
   BVC 3$
   BCS 4$

3$: ERROR 2 ;ROL FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,R3 ;RESULT OK ?
   BEQ T$T305 ;BR IF YES
5$: ERROR 2 ;ROL DELIVERED THE WRONG RESULT

;*****
;TEST 305 ROL DMO TEST - N:C = 0010
;*****
T$T305:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #305,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
CLR R3 ;CDESTJ = 000000
CCC ;CLEAR CODES
SEV ;N:C = 0010

2$: ROL R3 ;TEST THE ROL
   BMI 3$ ;N:C = 0100 ?
   BNE 3$
   BVS 3$
   BCC 4$

3$: ERROR 2 ;ROL FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,R3 ;RESULT OK ?
   BEQ T$T306 ;BR IF YES
5$: ERROR 2 ;ROL DELIVERED THE WRONG RESULT

```

```

6402
6403
6404
6405 020744
6406 020744 000004
6407 020746 012700 000306
6408 020752 013701 020772
6409 020752 013701 100000
6410 020762 012703 077777
6411 020766 000257
6412 020770 000265
6413
6414 020772 005503
6415
6416 020774 100003
6417 020776 001402
6418 021000 102001
6419 021002 103001
6420
6421 021004 104002
6422
6423 021006 020403
6424 021010 001401
6425
6426 021012 104002
6427
6428
6429
6430
6431 021014
6432 021016 000004
6433 021016 012700 000307
6434 021022 013701 021040
6435 021026 005004
6436 021030 012703 177777
6437 021034 000257
6438 021036 000273
6439
6440 021040 005503
6441
6442 021042 100403
6443 021044 001002
6444 021046 102401
6445 021050 103401
6446
6447 021052 104002
6448
6449 021054 020403
6450 021056 001401
6451
6452 021060 104002
6453
6454
6455
6456
6457 021062

```

```

*****
;TEST 306 ADC DMO TEST - N:C = 0101
*****
TST306:
SCOPE
MOV #306,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD RO WITH TEST NUMBER
MOV #1,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100000,R4 ;RESULT S / B = 100000
MOV #17777,R3 ;DEST = 17777
CCC ;CLEAR CODES
265 ;N:C = 0101
25: ADC R3 ;TEST THE ADC
BPL 35 ;N:C = 1010 ?
BEQ 35
BVC 35
BCC 45
35: ERROR 2 ;ADC FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,R3 ;RESULT OK ?
BEQ TST307 ;BR IF YES
55: ERROR 2 ;ADC DELIVERED THE WRONG RESULT
*****
;TEST 307 ADC DMO TEST - N:C = 1011
*****
TST307:
SCOPE
MOV #307,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD RO WITH TEST NUMBER
CLR R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R3 ;RESULT S / B = 000000
MOV #17777 ;DEST = 17777
CCC ;CLEAR CODES
273 ;N:C = 1011
25: ADC R3 ;TEST THE ADC
BMI 35 ;N:C = 0101 ?
BNE 35
BVS 45
BCS 45
35: ERROR 2 ;ADC FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,R3 ;RESULT OK ?
BEQ TST310 ;BR IF YES
55: ERROR 2 ;ADC DELIVERED THE WRONG RESULT
*****
;TEST 310 ADC DMO TEST - N:C = 1010
*****
TST310:

```

```

6458 021062 000004
6459 021064 012700 000310
6460 021070 013701 021140
6461 021074 012704 177777
6462 021100 012703 177777
6463 021104 000257
6464 021106 000272
6465
6466 021110 005503
6467
6468 021112 100003
6469 021114 001402
6470 021116 102401
6471 021120 103001
6472
6473 021122 104002
6474
6475 021124 020403
6476 021126 001401
6477
6478 021130 104002
6479
6480
6481
6482
6483 021132
6484 021132 000004
6485 021134 012700 000311
6486 021140 013701 021156
6487 021144 005004
6488 021146 012703 000001
6489 021152 000257
6490 021154 000273
6491
6492 021156 005603
6493
6494 021160 100403
6495 021162 001002
6496 021164 102401
6497 021166 103001
6498
6499 021170 104002
6500
6501 021172 020403
6502 021174 001401
6503
6504 021176 104002
6505
6506
6507
6508
6509 021200
6510 021200 000004
6511 021202 012700 000312
6512 021206 013701 021226
6513 021212 012704 077777

```

```

SCOPE
MOV #310,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD RO WITH TEST NUMBER
MOV #-1,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R3 ;RESULT S / B = 177777
MOV #17777 ;DEST = 17777
CCC ;CLEAR CODES
272 ;N:C = 1010
25: ADC R3 ;TEST THE ADC
RPL 35 ;N:C = 1000 ?
BEQ 35
BVS 35
BCC 45
35: ERROR 2 ;ADC FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,R3 ;RESULT OK ?
BEQ TST311 ;BR IF YES
55: ERROR 2 ;ADC DELIVERED THE WRONG RESULT
*****
;TEST 311 SRC DMO TEST - N:C = 1011
*****
TST311:
SCOPE
MOV #311,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD RO WITH TEST NUMBER
CLR R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R3 ;RESULT S / B = 000000
MOV #1 ;DEST = +1
CCC ;CLEAR CODES
273 ;N:C = 1011
25: SBC R3 ;TEST THE SBC
BMI 35 ;N:C = 0100 ?
BNE 35
BVS 45
BCC 45
35: ERROR 2 ;SBC FAILED TO ALTER THE CODES PROPERLY
45: CMP R4,R3 ;RESULT OK ?
BEQ TST312 ;BR IF YES
55: ERROR 2 ;SRC DELIVERED THE WRONG RESULT
*****
;TEST 312 SRC DMO TEST - N:C = 0101
*****
TST312:
SCOPE
MOV #312,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD RO WITH TEST NUMBER
MOV #077777,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #077777 ;RESULT S / B = 077777

```

```

6570 021216 012703 100000
6571 021224 000265
6572 021226 005603
6573 021230 100403
6574 021234 102401
6575 021236 103001
6576 021240 104002
6577 021242 020403
6578 021244 001401
6579 021246 104002
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599

```

```

MOV #100000,R3 ;CDEST = 100000
CLR R3 ;CLEAR CODES
MFC #0101 ;N:C = 0101

2$: SBC R3 ;TEST THE SBC
BMI 35 ;N:C = 0010 ?
BNE 35
BVS 35
BCC 45

3$: ERROR 2 ;SBC FAILED TO ALTER THE CODES PROPERLY

4$: CMP R4,R3 ;RESULT OK ?
BEQ T$T313 ;BR IF YES

5$: ERROR 2 ;SBC DELIVERED THE WRONG RESULT

;*****
;TEST 313 SBC DMO TEST - N:C = 1110
;*****
T$T313:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #313,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R2 ;RESULT S / B = 1
CLR R3 ;CDEST = {
CCC ;CLEAR CODES
MFC #1110 ;N:C = 1110

2$: SBC R3 ;TEST THE SBC
BMI 35 ;N:C = 0000 ?
BNE 35
BVS 35
BCC 45

3$: ERROR 2 ;SBC FAILED TO ALTER THE CODES PROPERLY

4$: CMP R4,R3 ;RESULT OK ?
BEQ T$T314 ;BR IF YES

5$: ERROR 2 ;SBC DELIVERED THE WRONG RESULT

;*****
;TEST 314 SBC DMO TEST - N:C = 0111
;*****
T$T314:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #314,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R2 ;RESULT S / B = 177777
CLR R3 ;CDEST = 000000
CCC ;CLEAR CODES
MFC #0111 ;N:C = 0111

```

```

6570 021344 005603
6571 021346 100003
6572 021350 001402
6573 021354 102401
6574 021354 103401
6575 021356 104002
6576 021360 020403
6577 021362 001401
6578 021364 104002
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590
6591
6592
6593
6594
6595
6596
6597
6598
6599
6600
6601
6602
6603
6604
6605
6606
6607
6608
6609
6610
6611
6612
6613
6614
6615
6616
6617
6618
6619
6620
6621
6622
6623
6624
6625

```

```

2$: SBC R3 ;TEST THE SBC
BPL 35 ;N:C = 1001 ?
BNE 35
BVS 35
BCC 45

3$: ERROR 2 ;SBC FAILED TO ALTER THE ERROR CODES PROPERLY

4$: CMP R4,R3 ;RESULT OK ?
BEQ T$T315 ;BR IF YES

5$: ERROR 2 ;SBC DELIVERED THE WRONG RESULT

;*****
;TEST 315 TST DMO TEST - N:C = 1011
;*****
T$T315:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #315,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
CLR (R2) ;CDEST = 000000
CCC ;CLEAR CODES
MFC #1011 ;N:C=1011

2$: TST (R2) ;TEST THE TST
BMI 35 ;N:C = 0100 ?
BNE 35
BVS 35
BCC 45

3$: ERROR 1 ;TST FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;RESULT OK ?
BEQ T$T316 ;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;TST ALTERED THE CDEST

;*****
;TEST 316 TST DMO TEST - N:C = 0100
;*****
T$T316:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #316,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 177777
COM R4 ;CDEST = 177777
MOV #-1,(R2) ;CLEAR CODES
CCC ;N:C=0100
MFC #0100 ;N:C=0100

```

```

6626 021472 005712
6627 021474 100003
6628 021474 100003
6629 021476 001402
6630 021500 102401
6631 021502 103001
6632 021504 104001
6633 021506 020412
6634 021510 001402
6635 021512 011203
6636 021514 104001
6637 021512 011203
6638 021514 104001
6639 021514 104001
6640
6641
6642
6643
6644
6645 021516 000004
6646 021520 012700 000317
6647 021524 013701 021560
6648
6649 021530 032737 000100 063234
6650 021536 001401
6651 021540 000000
6652 021542 012702 063312
6653 021546 005004
6654 021550 012712 177777
6655 021554 000257
6656 021556 000273
6657
6658 021560 005012
6659
6660 021562 100403
6661 021564 001002
6662 021568 102401
6663 021570 103001
6664
6665 021572 104001
6666
6667 021574 020412
6668 021576 001402
6669
6670 021600 011203
6671 021602 104001
6672
6673
6674
6675
6676 021604
6677 021606 000004
6678 021606 012700 000320
6679 021612 013701 021632
6680 021618 012702 063312
6681 021622 005004

```

```

2$: TST (R2) ;TEST THE TST
;N:C = 1000 ?
BPL 3$
BEQ 3$
BVS 3$
BCC 4$
3$: ERROR 1 ;TST FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;RESULT OK ?
BEQ TST317 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;TST ALTERED THE CDESTJ
;*****
;TEST 317 CLR DM1 TEST - N:C = 1011
;*****
TST317: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #317,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #R2,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
.SBRTL USER CONTROLLED BREAKPOINT -- BIT6
BIT #BIT6,@#BPTLOC ;BREAKPOINT HALT SET ??
BEQ +4 ;BR IF NOT
HALT ;BREAK - DEPRESS CONTINUE TO RESTART
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
MOV #-1,(R2) ;CDESTJ = 177777
CCC ;CLEAR CODES
273 ;N:C = 1011
2$: CLR (R2) ;TEST THE CLR
;N:C = 0100 ?
BMI 3$
BNE 3$
BVS 3$
BCC 4$
3$: ERROR 1 ;CLR FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,(R2) ;RESULT OK ?
BEQ TST320 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CLR DELIVERED THE WRONG RESULT
;*****
;TEST 320 CLR DM2 TEST - N:C = 0000
;*****
TST320: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #320,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #R2,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000

```

```

6682 021624 013712 063324
6683 021630 000257
6684
6685 021632 005022
6686
6687 021634 100403
6688 021636 001002
6689 021640 102401
6690 021642 103001
6691
6692 021644 104001
6693
6694 021646 022702 063314
6695 021652 001401
6696
6697 021654 104005
6698
6699 021656 020442
6700 021660 001402
6701
6702 021662 011203
6703 021664 104001
6704
6705
6706
6707
6708 021666
6709 021666 000004
6710 021670 012700 000321
6711 021674 013701 021720
6712 021700 012702 063312
6713 021704 012704 125252
6714 021710 012712 052525
6715 021714 000257
6716 021716 000266
6717
6718 021720 005112
6719
6720 021722 100003
6721 021724 001402
6722 021726 102401
6723 021730 103401
6724
6725 021732 104001
6726 021734 020412
6727 021736 001402
6728
6729 021740 011203
6730 021742 104001
6731
6732
6733
6734
6735 021744
6736 021744 000004
6737 021746 012700 000322

```

```

MOV @#DWTA+2,(R2) ;CDESTJ = 177777
CCC ;CLEAR CODES
2$: CLR (R2)+ ;TEST THE CLR
;N:C = 0100 ?
BMI 3$
BNE 3$
BVS 3$
BCC 4$
3$: ERROR 1 ;CLR FAILED TO ALTER THE CODES PROPERLY
4$: CMP #MBUF0+2,R2 ;DID CLR INCREMENT DEST REG
BEQ 6$ ;BR IF YES
5$: ERROR 5 ;CLR FAILED TO UPDATE DEST REG
6$: CMP R4,-(R2) ;RESULT OK ?
BEQ TST321 ;;BR IF YES
7$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CLR DELIVERED THE WRONG RESULT
;*****
;TEST 321 COM DM1 TEST - N:C = 0110
;*****
TST321: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #321,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #R2,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,R4 ;RESULT S / B = 125252
MOV #52525,(R2) ;CDESTJ = 52525
CCC ;CLEAR CODES
266 ;N:C = 0110
2$: COM (R2) ;TEST THE CLR
;N:C = 1001 ?
BPL 3$
BEQ 3$
BVS 3$
BCS 4$
3$: ERROR 1 ;COM FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,(R2) ;RESULT OK ?
BEQ TST322 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;COM DELIVERED THE WRONG RESULT
;*****
;TEST 322 COM DM1 TEST - N:C = 1001
;*****
TST322: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #322,R0 ;;LOAD R0 WITH TEST NUMBER

```

6738 021752 013701 021774
 6739 021756 012702 063312
 6740 021762 005004
 6741 021764 012712 177777
 6742 021770 000257
 6743 021772 000271
 6744
 6745 021774 005112
 6746
 6747 021776 100403
 6748 022000 001002
 6749 022002 102401
 6750 022004 103401
 6751
 6752 022006 104001
 6753 022010 020412
 6754 022012 001402
 6755
 6756 022014 011203
 6757 022016 104001
 6758
 6759
 6760
 6761
 6762 022020
 6763 022022 000004
 6764 022022 012700 000323
 6765 022026 013701 022050
 6766 022032 012702 063312
 6767 022036 005004
 6768 022040 012712 177777
 6769 022044 003751
 6770 022046 000273
 6771
 6772 022050 005212
 6773
 6774 022052 100403
 6775 022054 001002
 6776 022056 102401
 6777 022060 103401
 6778
 6779 022062 104001
 6780 022064 020412
 6781 022066 001402
 6782
 6783 022070 011203
 6784 022072 104001
 6785
 6786
 6787
 6788
 6789 022074
 6790 022074 000004
 6791 022076 012700 000324
 6792 022078 013701 022126
 6793 022106 012702 063312

MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
 CLR R4 ;RESULT S / B = 000000
 MOV #-1,(R2) ;RESULT = 177777
 CCC ;CLEAR CODES
 271 ;N:C = 1001
 25: COM (R2) ;TEST THE COM
 BMI 35 ;N:C = 0101 ?
 BNE 35
 BVS 35
 BCS 45
 35: ERROR 1 ;COM FAILED TO ALTER THE CODES PROPERLY
 45: CMP R4,(R2) ;RESULT OK ?
 BEQ TS#323 ;BR IF YES
 55: MOV (R2),R3 ;GET THE WAS DATA
 ERROR 1 ;COM DELIVERED THE WRONG RESULT
 ;*****
 ;TEST 323 INC DM1 TEST - N:C = 1011
 ;*****
 TST323: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #323,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
 CLR R4 ;RESULT S / B = 000000
 MOV #-1,(R2) ;RESULT = 177777
 CCC ;CLEAR CODES
 273 ;N:C = 1011
 25: INC (R2) ;TEST THE INC
 BMI 35 ;N:C = 0101 ?
 BNE 35
 BVS 35
 BCS 45
 35: ERROR 1 ;INC FAILED TO ALTER THE CODES PROPERLY
 45: CMP R4,(R2) ;RESULT OK ?
 BEQ TS#324 ;BR IF YES
 55: MOV (R2),R3 ;GET THE WAS DATA
 ERROR 1 ;INC DELIVERED THE WRONG RESULT
 ;*****
 ;TEST 324 INC DM1 TEST - N:C = 0100
 ;*****
 TST324: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #324,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUFO,R2 ;DEST ADDR = MBUFO

6794 022112 012704 100000
 6795 022116 012712 077777
 6796 022122 000257
 6797 022124 000264
 6798
 6799 022126 005212
 6800
 6801 022130 100003
 6802 022132 001402
 6803 022134 102001
 6804 022136 103001
 6805
 6806 022140 104001
 6807 022142 020412
 6808 022144 001402
 6809
 6810 022146 011203
 6811 022150 104001
 6812
 6813
 6814
 6815
 6816 022152
 6817 022152 000004
 6818 022154 012700 000325
 6819 022160 013701 022202
 6820 022164 012702 063312
 6821 022170 005004
 6822 022172 012712 000001
 6823 022174 003751
 6824 022200 000273
 6825
 6826 022202 005312
 6827
 6828 022204 100403
 6829 022206 001002
 6830 022210 102401
 6831 022212 103401
 6832
 6833 022214 104001
 6834 022216 020412
 6835 022220 001402
 6836
 6837 022222 011203
 6838 022224 104001
 6839
 6840
 6841
 6842
 6843 022226
 6844 022226 000004
 6845 022226 012700 000326
 6846 022230 013701 022260
 6847 022240 012702 063312
 6848 022244 012704 077777
 6849 022250 012712 100000

MOV #100000,R4 ;RESULT S / B = 100000
 MOV #77777,(R2) ;DEST = 77777
 CCC ;CLEAR CODES
 264 ;N:C = 0100
 25: INC (R2) ;TEST THE INC
 BPL 35 ;N:C = 1010 ?
 BEQ 35
 BVC 35
 BCC 45
 35: ERROR 1 ;INC FAILED TO ALTER THE CODES PROPERLY
 45: CMP R4,(R2) ;RESULT OK ?
 BEQ TS#325 ;BR IF YES
 55: MOV (R2),R3 ;GET THE WAS DATA
 ERROR 1 ;INC DELIVERED THE WRONG RESULT
 ;*****
 ;TEST 325 DEC DM1 TEST - N:C = 1011
 ;*****
 TST325: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #325,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
 CLR R4 ;RESULT S / B = 000000
 MOV #1,(R2) ;RESULT = 1
 CCC ;CLEAR CODES
 273 ;N:C = 1011
 25: DEC (R2) ;TEST THE DEC
 BMI 35 ;N:C = 0101 ?
 BNE 35
 BVS 35
 BCS 45
 35: ERROR 1 ;DEC FAILED TO ALTER THE CODES PROPERLY
 45: CMP R4,(R2) ;RESULT OK ?
 BEQ TS#326 ;BR IF YES
 55: MOV (R2),R3 ;GET THE WAS DATA
 ERROR 1 ;DEC DELIVERED THE WRONG RESULT
 ;*****
 ;TEST 326 DEC DM1 TEST - N:C = 1100
 ;*****
 TST326: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #326,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
 MOV #77777,R4 ;RESULT S / B = 77777
 MOV #100000,(R2) ;DEST = 100000

```

6850 022254 000257
6851 022256 000274
6852 022260 005312
6853 022262 100403
6854 022264 001402
6855 022266 102001
6856 022270 103001
6860 022272 104001
6861 022274 020412
6862 022276 001402
6863 022300 011203
6864 022302 104001
6865 022304 000004
6866 022306 012700
6867 022312 013701
6868 022316 012702
6869 022322 012704
6870 022330 000257
6871 022332 005312
6872 022334 100003
6873 022336 001402
6874 022340 102401
6875 022342 103001
6876 022344 104001
6877 022346 020412
6878 022350 001402
6879 022352 011203
6880 022354 104001
6881 022356 000004
6882 022360 012700
6883 022364 013701
6884 022370 012702
6885 022374 005004
6886 022376 012712
6887 022402 000257
6888 022404 000270

```

```

      CCC          ;CLEAR CODES
      274         ;N:C = 1100
2$:   DEC        (R2)      ;TEST THE DEC
      BMI        3$       ;N:C = 0010 ?
      BEQ        3$
      BVC        3$
      BCC        4$
3$:   ERROR      1         ;DEC FAILED TO ALTER THE CODES PROPERLY
4$:   CMP        R4,(R2)   ;RESULT OK ?
      BEQ        TS$327   ;;BR IF YES
5$:   MOV        (R2),R3   ;GET THE WAS DATA
      ERROR      1         ;DEC DELIVERED THE WRONG RESULT
;*****
;TEST 327      DEC DM1 TEST - N:C = 0000
;*****
TS$327:  SCOPE          ;CALL THE SCOPE LOOP UTILITY
      MOV        #327,R0   ;LOAD R0 WITH TEST NUMBER
      MOV        #25,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV        #MBUF0,R2 ;DEST ADDR = MBUF0
      MOV        #1,R4     ;RESULT S / B = 177777
      CLR        (R2)     ;DESTJ = 000000
      CCC        ;CLEAR CODES
2$:   DEC        (R2)      ;TEST THE DEC
      BPL        3$       ;N:C = 1000 ?
      BEQ        3$
      BVC        3$
      BCC        4$
3$:   ERROR      1         ;DEC FAILED TO ALTER THE CODES PROPERLY
4$:   CMP        R4,(R2)   ;RESULT OK ?
      BEQ        TS$330   ;;BR IF YES
5$:   MOV        (R2),R3   ;GET THE WAS DATA
      ERROR      1         ;DEC DELIVERED THE WRONG RESULT
;*****
;TEST 330      ASL DM1 TEST - N:C = 1000
;*****
TS$330:  SCOPE          ;CALL THE SCOPE LOOP UTILITY
      MOV        #330,R0   ;LOAD R0 WITH TEST NUMBER
      MOV        #25,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV        #MBUF0,R2 ;DEST ADDR = MBUF0
      MOV        #100000,R4 ;RESULT S / B = 000000
      CLR        (R2)     ;DESTJ = 100000
      MOV        #100000,(R2) ;CLEAR CODES
      CCC        ;N:C = 1000
      SEN

```

```

6906 022406 006312
6907 022410 100403
6908 022412 001002
6909 022414 102001
6910 022416 103401
6911 022420 104001
6912 022422 020412
6913 022424 001402
6914 022426 011203
6915 022430 104001
6916 022432 000004
6917 022434 012700
6918 022440 013701
6919 022444 012702
6920 022450 012704
6921 022454 000257
6922 022462 000265
6923 022464 006312
6924 022466 100003
6925 022470 001402
6926 022472 102001
6927 022474 103001
6928 022476 104001
6929 022500 020412
6930 022502 001402
6931 022504 011203
6932 022506 104001
6933 022510 000004
6934 022512 012700
6935 022516 013701
6936 022522 012702
6937 022526 005004
6938 022530 005012
6939 022534 000257
6940 022536 000262
6941 022538 006312
6942 022540 006312

```

```

2$:   ASL        (R2)      ;TEST THE ASL
      BMI        3$       ;N:C = 0111 ?
      BNE        3$
      BVC        3$
      BCS        4$
3$:   ERROR      1         ;ASL FAILED TO ALTER THE CODES PROPERLY
4$:   CMP        R4,(R2)   ;RESULT OK ?
      BEQ        TS$331   ;;BR IF YES
5$:   MOV        (R2),R3   ;GET THE WAS DATA
      ERROR      1         ;ASL DELIVERED THE WRONG RESULT
;*****
;TEST 331      ASL DM1 TEST - N:C = 0101
;*****
TS$331:  SCOPE          ;CALL THE SCOPE LOOP UTILITY
      MOV        #331,R0   ;LOAD R0 WITH TEST NUMBER
      MOV        #25,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV        #MBUF0,R2 ;DEST ADDR = MBUF0
      MOV        #100000,R4 ;RESULT S / B = 100000
      MOV        #40000,(R2) ;DESTJ = 40000
      CCC        ;CLEAR CODES
      265         ;N:C = 0101
2$:   ASL        (R2)      ;TEST THE ASL
      BPL        3$       ;N:C = 1010 ?
      BEQ        3$
      BVC        3$
      BCC        4$
3$:   ERROR      1         ;ASL FAILED TO ALTER THE CODES PROPERLY
4$:   CMP        R4,(R2)   ;RESULT OK ?
      BEQ        TS$332   ;;BR IF YES
5$:   MOV        (R2),R3   ;GET THE WAS DATA
      ERROR      1         ;ASL DELIVERED THE WRONG RESULT
;*****
;TEST 332      ASL DM1 TEST - N:C = 0010
;*****
TS$332:  SCOPE          ;CALL THE SCOPE LOOP UTILITY
      MOV        #332,R0   ;LOAD R0 WITH TEST NUMBER
      MOV        #25,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV        #MBUF0,R2 ;DEST ADDR = MBUF0
      CLR        R4       ;RESULT S / B = 000000
      CLR        (R2)     ;DESTJ = 000000
      CCC        ;CLEAR CODES
      SEV          ;N:C = 0010
2$:   ASL        (R2)      ;TEST THE ASL

```

6962 022540 100403
6963 022541 001002
6964 022542 102401
6965 022546 103001
6966
6967 022550 194001
6968 022551 001402
6969 022554 001402
6970
6971 022556 011203
6972 022560 104001
6973
6974
6975
6976
6977 022562
6978 022562 000004
6979 022564 012700 000333
6980 022564 012701 022614
6981 022574 012702 024312
6982 022500 012704 052525
6983 022604 012712 125252
6984 022610 000257
6985 022612 000275
6986
6987 022614 006112
6988
6989 022616 100403
6990 022620 001402
6991 022624 103001
6992 022624 103401
6993
6994 022626 104001
6995 022630 020412
6996 022632 001402
6997
6998 022634 011203
6999 022636 104001
7000
7001
7002
7003
7004
7005 022640
7006 022640 000004
7007 022642 012700 000334
7008 022646 013701 022672
7009 022652 012702 053344
7010 022652 012704 125252
7011 022656 000257
7012 022670 000265
7013
7014 022672 006112
7015
7016 022674 100003
7017 022676 001402

BMI 3\$;N:C = 0100 ?
BNE 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 1 ;ASL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ T\$T333 ;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ASL DELIVERED THE WRONG RESULT
;*****
;TEST 333 ROL DM1 TEST - N:C = 1101
;*****
;T333: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #333,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #M000,R2 ;DEST ADDR = M000
MOV #52525,R4 ;RESULT S / B = 52525
MOV #125252,(R2) ;DESTJ = 125252
CCC ;CLEAR CODES
275 ;N:C = 1101
2\$: ROL (R2) ;TEST THE ROL
BMI 3\$;N:C = 0011 ?
BNE 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 1 ;ROL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ T\$T334 ;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ROL DELIVERED THE WRONG RESULT
;*****
;TEST 334 ROL DM1 TEST - N:C = 0101
;*****
;T334: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #334,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #M000,R2 ;DEST ADDR = M000
MOV #52525,(R2) ;RESULT S / B = 52525
CCC ;CLEAR CODES
265 ;N:C = 0101
2\$: ROL (R2) ;TEST THE ROL
BPL 3\$;N:C = 1010 ?
BEQ 3\$

7018 022700 102001
7019 022702 103001
7020
7021 022704 104001
7022 022706 020412
7023 022710 001402
7024
7025 022712 011203
7026 022714 104001
7027
7028
7029
7030
7031 022716
7032 022716 000004
7033 022716 012700 000335
7034 022716 013701 022744
7035 022730 012702 063312
7036 022734 005004
7037 022736 005012
7038 022740 000257
7039 022742 000262
7040
7041 022744 006112
7042
7043 022746 100403
7044 022750 001002
7045 022752 102401
7046 022754 103001
7047
7048 022756 104001
7049 022760 020412
7050 022762 001402
7051
7052 022764 011203
7053 022766 104001
7054
7055
7056
7057
7058 022770
7059 022770 000004
7060 022772 012700 000336
7061 022776 013701 023022
7062 023002 012702 063312
7063 023006 012704 100000
7064 023012 012712 077777
7065 023016 000257
7066 023020 000265
7067
7068 023022 005512
7069
7070 023024 100003
7071 023026 001402
7072 023030 102001
7073 023032 103001

BVC 3\$
BCC 4\$
3\$: ERROR 1 ;ROL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ T\$T335 ;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ROL DELIVERED THE WRONG RESULT
;*****
;TEST 335 ROL DM1 TEST - N:C = 0010
;*****
;T335: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #335,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #M000,R2 ;DEST ADDR = M000
CLR R4 ;RESULT S / B = 000000
CLR (R2) ;DESTJ = 000000
CCC ;CLEAR CODES
SEV ;N:C = 0010
2\$: ROL (R2) ;TEST THE ROL
BMI 3\$;N:C = 0100 ?
BNE 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 1 ;ROL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ T\$T336 ;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;ROL DELIVERED THE WRONG RESULT
;*****
;TEST 336 ADC DM1 TEST - N:C = 0101
;*****
;T336: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #336,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #M000,R2 ;DEST ADDR = M000
MOV #100000,R4 ;RESULT S / B = 100000
MOV #77777,(R2) ;DESTJ = 77777
CCC ;CLEAR CODES
265 ;N:C = 0101
2\$: ADC (R2) ;TEST THE ADC
BPL 3\$;N:C = 1010 ?
BEQ 3\$
BVS 3\$
BCC 4\$

7074
7075 023034 104001
7076 023036 020412
7077 023040 001402
7078
7079 023042 011203
7080 023044 104001
7081
7082
7083
7084
7085 023046
7086 023046 000004
7087 023050 012700 000337
7088 023054 012701 023076
7089 023060 012702 063312
7090 023064 005004
7091 023066 012712 177777
7092 023072 000273
7093 023074 000273
7094
7095 023076 005512
7096
7097 023100 100403
7098 023102 001002
7099 023104 102401
7100 023106 103401
7101
7102 023110 104001
7103 023112 020412
7104 023114 001402
7105
7106 023116 011203
7107 023120 104001
7108
7109
7110
7111
7112 023122
7113 023122 000004
7114 023124 012700 000340
7115 023130 012701 023154
7116 023134 012702 063312
7117 023140 012704 177777
7118 023144 012712 177777
7119 023150 000257
7120 023152 000272
7121
7122 023154 005512
7123
7124 023156 100003
7125 023160 001402
7126 023162 102401
7127 023164 103001
7128
7129 023166 104001

```

3S: ERROR 1 ;ADC FAILED TO ALTER THE CODES PROPERLY
4S: CMP R4,(R2) ;RESULT OK ?
    BEQ TST337 ;;BR IF YES
5S: MOV (R2),R3 ;GET THE WAS DATA
    ERROR 1 ;ADC DELIVERED THE WRONG RESULT
;*****
;#TEST 337 ADC DMI TEST - N:C = 1011
;*****
TST337: SCOPE ;CALL THE SCOPE LOOP UTILITY
        MOV #337,R0 ;LOAD R0 WITH TEST NUMBER
        MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV #MBUF0,R2 ;DEST ADDR = MBUF0
        CLR R4 ;RESULT S / B = 000000
        MOV #-1,(R2) ;DESTJ = 177777
        CCC ;CLEAR CODES
        T73 ;N:C = 1011
2S: ADC (R2) ;TEST THE ADC
        BMI 3S ;N:C = 0101 ?
        RNE 3S
        BVS 3S
        BCS 4S
3S: ERROR 1 ;ADC FAILED TO ALTER THE CODES PROPERLY
4S: CMP R4,(R2) ;RESULT OK ?
    BEQ TST340 ;;BR IF YES
5S: MOV (R2),R3 ;GET THE WAS DATA
    ERROR 1 ;ADC DELIVERED THE WRONG RESULT
;*****
;#TEST 340 ADC DMI TEST - N:C = 1010
;*****
TST340: SCOPE ;CALL THE SCOPE LOOP UTILITY
        MOV #340,R0 ;LOAD R0 WITH TEST NUMBER
        MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV #MBUF0,R2 ;DEST ADDR = MBUF0
        CLR R4 ;RESULT S / B = 177777
        MOV #-1,(R2) ;DESTJ = 177777
        CCC ;CLEAR CODES
        T72 ;N:C = 1010
2S: ADC (R2) ;TEST THE ADC
        BPL 3S ;N:C = 1000 ?
        BEQ 3S
        BVS 3S
        BCC 4S
3S: ERROR 1 ;ADC FAILED TO ALTER THE CODES PROPERLY

```

7130 023170 020412
7131 023172 001402
7132
7133 023174 011203
7134 023176 104001
7135
7136
7137
7138
7139
7140 023200
7141 023202 000004
7142 023206 012700 000341
7143 023212 012701 023230
7144 023216 012702 063312
7145 023220 005004
7146 023224 000273 000001
7147 023226 000273
7148
7149 023230 005612
7150
7151 023232 100403
7152 023234 001002
7153 023236 102401
7154 023240 103001
7155
7156 023242 104001
7157 023244 020412
7158 023246 001402
7159
7160 023250 011203
7161 023252 104001
7162
7163
7164
7165
7166 023254
7167 023254 000004
7168 023256 012700 000342
7169 023262 012701 023306
7170 023266 012702 063312
7171 023272 012704 077777
7172 023276 012712 100000
7173 023302 000257
7174 023304 000265
7175
7176 023306 005612
7177
7178 023310 100403
7179 023312 001402
7180 023314 102401
7181 023316 103001
7182
7183 023320 104001
7184 023322 020412
7185 023324 001402

```

4S: CMP R4,(R2) ;RESULT OK ?
    BEQ TST341 ;;BR IF YES
5S: MOV (R2),R3 ;GET THE WAS DATA
    ERROR 1 ;ADC DELIVERED THE WRONG RESULT
;*****
;#TEST 341 SRC DMI TEST - N:C = 1011
;*****
TST341: SCOPE ;CALL THE SCOPE LOOP UTILITY
        MOV #341,R0 ;LOAD R0 WITH TEST NUMBER
        MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV #MBUF0,R2 ;DEST ADDR = MBUF0
        CLR R4 ;RESULT S / B = 000000
        MOV #1,(R2) ;DESTJ = +1
        CCC ;CLEAR CODES
        T73 ;N:C = 1011
2S: SBC (R2) ;TEST THE SBC
        BMI 3S ;N:C = 0100 ?
        RNE 3S
        BVS 3S
        BCC 4S
3S: ERROR 1 ;SBC FAILED TO ALTER THE CODES PROPERLY
4S: CMP R4,(R2) ;RESULT OK ?
    BEQ TST342 ;;BR IF YES
5S: MOV (R2),R3 ;GET THE WAS DATA
    ERROR 1 ;SBC DELIVERED THE WRONG RESULT
;*****
;#TEST 342 SRC DMI TEST - N:C = 0101
;*****
TST342: SCOPE ;CALL THE SCOPE LOOP UTILITY
        MOV #342,R0 ;LOAD R0 WITH TEST NUMBER
        MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV #MBUF0,R2 ;DEST ADDR = MBUF0
        MOV #077777,R4 ;RESULT S / B = 077777
        MOV #100000,(R2) ;DESTJ = 100000
        CCC ;CLEAR CODES
        T65 ;N:C = 0101
2S: SBC (R2) ;TEST THE SBC
        BMI 3S ;N:C = 0010 ?
        BEQ 3S
        BVS 3S
        BCC 4S
3S: ERROR 1 ;SBC FAILED TO ALTER THE CODES PROPERLY
4S: CMP R4,(R2) ;RESULT OK ?
    BEQ TST343 ;;BR IF YES

```

```

7186 023326 011203
7187 023330 104001
7188
7189
7190
7191
7192
7193
7194 023332 000004
7195 023334 012700 000343
7196 023334 013701 023364
7197 023334 012702 063312
7198 023334 012704 000001
7199 023334 012714 000001
7200 023360 000257
7201 023362 000276
7202
7203 023364 005612
7204
7205 023366 100403
7206 023370 014017
7207 023372 014017
7208 023374 103001
7209
7210 023376 104001
7211 023400 020412
7212 023402 001402
7213
7214 023404 011203
7215 023406 104001
7216
7217
7218
7219
7220 023410
7221 023410 000004
7222 023412 012700 000344
7223 023416 013701 023440
7224 023416 013703 063312
7225 023416 012704 177777
7226 023432 005012
7227 023434 000257
7228 023436 000267
7229
7230 023440 005612
7231
7232 023442 100003
7233 023444 001402
7234 023446 102401
7235 023450 103401
7236
7237 023452 104001
7238 023454 020412
7239 023456 001402
7240
7241 023460 011203

```

```

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;SBC DELIVERED THE WRONG RESULT
;*****
;TEST 343 SBC DMI TEST - N:C = 1110
;*****
TST343: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #343,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #1,R4 ;RESULT S / B = 1
MOV #1,(R2) ;DESTJ = 17777
CCC ;CLEAR CODES
276 ;N:C = 1110

2$: SBC (R2) ;TEST THE SBC
BMI 3$ ;N:C = 0000 ?
BEQ 3$
BVS 3$
BCC 4$

3$: ERROR 1 ;SBC FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,(R2) ;RESULT OK ?
BEQ TST344 ;BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;SBC DELIVERED THE WRONG RESULT
;*****
;TEST 344 SBC DMI TEST - N:C = 0111
;*****
TST344: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #344,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #1,R4 ;RESULT S / B = 17777
CLR (R2) ;DESTJ = 000000
CCC ;CLEAR CODES
267 ;N:C = 0111

2$: SBC (R2) ;TEST THE SBC
BPL 3$ ;N:C = 1001 ?
BEQ 3$
BVS 3$
BCS 4$

3$: ERROR 1 ;SBC FAILED TO ALTER THE CODES PROPERLY
4$: CMP R4,(R2) ;RESULT OK ?
BEQ TST345 ;BR IF YES
MOV (R2),R3 ;GET THE WAS DATA

```

```

7242 023462 104001
7243
7244
7245
7246
7247 023464
7248 023464 000004
7249 023472 012700 000345
7250 023472 013701 023362
7251 023476 012704 177776
7252 023502 012703 177402
7253 023506 000257
7254 023510 000266
7255
7256 023512 105403
7257
7258 023514 100003
7259 023516 001402
7260 023520 014017
7261 023522 103401
7262
7263 023524 104002
7264
7265 023526 020403
7266 023530 001401
7267
7268 023532 104002
7269
7270
7271
7272 023534
7273 023534 000004
7274 023536 012700 000346
7275 023542 013701 023362
7276 023546 012704 177400
7277 023552 012703 177400
7278 023556 000257
7279 023560 000263
7280
7281 023562 105403
7282
7283 023564 100403
7284 023566 001002
7285 023570 102401
7286 023572 103001
7287
7288 023574 104002
7289
7290 023576 020403
7291 023600 001401
7292
7293
7294 023602 104002
7295
7296
7297

```

```

5$: ERROR 1 ;SBC DELIVERED THE WRONG RESULT
;*****
;TEST 345 NEGB - MODE 0 TEST - N:C = 0110
;*****
TST345: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #345,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177776,R4 ;RESULT S / B = 376 (LO BYTE)
MOV #177402,R3 ;DESTJ = 177402
CCC ;CLEAR FLAGS
266 ;N:C = 0110

2$: NEGB R3 ;TEST THE NEGB
BPL 3$ ;N:C = 1001
BEQ 3$
BVS 3$
BCS 4$

3$: ERROR 2 ;NEGB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST346 ;BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
ERROR 2 ;NEGB DELIVERED THE WRONG RESULT
;*****
;TEST 346 NEGB - MODE 0 TEST - N:C = 0011
;*****
TST346: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #346,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177400,R4 ;RESULT S / B = 000 (LO BYTE)
MOV #177400,R3 ;DESTJ = 177400
CCC ;CLEAR FLAGS
263 ;N:C = 0011

2$: NEGB R3 ;TEST THE NEGB
BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$

3$: ERROR 2 ;NEGB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST347 ;BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
ERROR 2 ;NEGB DELIVERED THE WRONG RESULT
;*****
;TEST 347 NEGB - MODE 0 TEST - N:C = 1101
;*****

```

```

7298
7299 023604
7300 023604 000004
7301 023606 012700 000347
7302 023612 013701 023632
7303 023616 012704 177600
7304 023616 012704 177600
7305 023626 000257
7306 023630 000275
7307
7308 023632 105403
7309
7310 023634 100003
7311 023636 001402
7312 023640 102001
7313 023642 103401
7314
7315 023644 104002
7316
7317 023646 020403
7318 023650 001401
7319
7320 023652 104002
7321
7322
7323
7324
7325 023654
7326 023654 000004
7327 023659 012700 000350
7328 023662 013701 023702
7329 023666 012704 177400
7330 023672 012703 177777
7331 023676 000257
7332 023700 000275
7333
7334 023702 105003
7335
7336 023704 100403
7337 023706 001002
7338 023710 103401
7339 023712 103001
7340
7341 023714 104002
7342
7343 023716 020403
7344 023720 001401
7345
7346 023722 104002
7347
7348
7349
7350
7351 023724
7352 023724 000004 000351
7353 023726 012700

```

```

*****
TST347: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #347,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177600,R4 ;RESULT S / B = 200 (LO BYTE)
MOV #177600,R3 ;DEST1 = 17600
CCC ;CLEAR FLAGS
ZFS ;N:C = 1101
2$: NEGB R3 ;TEST THE NEGB
;N:C = 1011
BPL 3$
BEQ 3$
BVC 3$
BCS 4$
3$: ERROR 2 ;NEGB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST350 ;BR IF YES
5$: ERROR 2 ;NEGB DELIVERED THE WRONG RESULT
*****
TST350: CLRBB - MODE 0 TEST - N:C = 1011
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #350,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177400,R4 ;RESULT S / B = 000 (LO BYTE)
MOV #-1,R3 ;DEST1 = 177777
CCC ;CLEAR FLAGS
ZFS ;N:C = 1011
2$: CLRBB R3 ;TEST THE CLRBB
;N:C = 0100 ?
BMI 3$
BNE 3$
BVS 3$
BCC 4$
3$: ERROR 2 ;CLRBB FAILED TO SET CODES PROPERLY
4$: CMP R4,R3 ;RESULT CORRECT ?
BEQ TST351 ;BR IF YES
5$: ERROR 2 ;CLRBB DELIVERED THE WRONG RESULT
*****
TST351: CLRBB - MODE 0 TEST - N:C = 0100
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #351,R0 ;LOAD R0 WITH TEST NUMBER

```

```

7354 023732 013701 023752
7355 023742 012704 177400
7356 023746 000257 177777
7357 023746 000257
7358 023750 000264
7359
7360 023752 105003
7361
7362 023754 100403
7363 023756 001002
7364 023760 102401
7365 023762 103001
7366
7367 023764 104002
7368
7369 023766 020403
7370 023770 001401
7371
7372 023772 104002
7373
7374
7375
7376
7377 023774
7378 023774 000004
7379 023776 012700 000352
7380 024002 013701 024032
7381 024006 012702 063313
7382 024012 012704 000377
7383 024016 012705 063312
7384 024022 012703 177777
7385 024024 012705
7386 024030 000257
7387
7388 024032 105023
7389
7390 024034 022703 063314
7391 024040 001401
7392
7393 024042 104005
7394
7395 024044 020415
7396 024046 001402
7397
7398 024050 011503
7399 024052 104001
7400
7401
7402
7403
7404 024054
7405 024054 000004
7406 024056 012700 000353
7407 024062 013701 024112
7408 024066 012702 063312
7409 024072 012704 000377

```

```

MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177400,R4 ;RESULT S / B = 000 (LO BYTE)
MOV #-1,R3 ;DEST1 = 177777
CCC ;CLEAR FLAGS
SEZ ;N:C = 0100
2$: CLRBB R3 ;TEST THE CLRBB
;N:C = 0100 ?
BMI 3$
BNE 3$
BVS 3$
BCC 4$
3$: ERROR 2 ;CLRBB FAILED TO SET CODES PROPERLY
4$: CMP R4,R3 ;RESULT CORRECT ?
BEQ TST352 ;BR IF YES
5$: ERROR 2 ;CLRBB DELIVERED THE WRONG RESULT
*****
TST352: CLRBB TEST - DM2 - ODD ADDRESS
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #352,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
MOV #377,R4 ;RESULT S / B = 377
MOV #MBUF0,R5 ;POINT R5 TO CHECK RESULT
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R5) ;DEST1 = 177777
CCC ;SCOPE SYNC
2$: CLRBB (R3)+ ;TEST THE CLRBB
;DID DEST REG GET INCREMENTED ?
;BR IF YES
CMP #MBUF0+2,R3
BEQ 4$
3$: ERROR 5 ;CLRBB FAILED TO UPDATE DEST REG
4$: CMP R4,(R5) ;CORRECT RESULT ?
BEQ TST353 ;BR IF YES
5$: MOV (R5),R3 ;GET THE WAS DATA
ERROR 1 ;CLRBB DELIVERED WRONG RESULT
*****
TST353: CLRBB TEST - DM1 - ODD ADDRESS
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #353,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
MOV #377,R4 ;RESULT S / B = 377

```

```

7410 024076 012705 063312      MOV    #MBUF0,R5      ;POINT R5 TO CHECK RESULT
7411 024102 010203          MOV    R2,R3          ;R3 CONTAINS DEST ADDR
7412 024104 012715 177777      MOV    #-1,(R5)      ;[DEST] = 177777
7413 024110 000257          CCC                    ;SCOPE SYNC
7414
7415 024112 105013          2$:   CLR B    (R3)    ;TEST THE CLR B
7416
7417 024114 020415          CMP    R4,(R5)       ;CORRECT RESULT ?
7418 024116 001402          BEQ                    ;BR IF YES
7419
7420 024120 011503          3$:   MOV    (R5),R3    ;GET THE WAS DATA
7421 024122 104001          ERROR 1              ;CLR B DELIVERED WRONG RESULT
7422
;*****
;TEST 354 CLR B TEST - DM2 - EVEN ADDRESS
;*****
TST354:
7423 024124          SCOPE                ;CALL THE SCOPE LOOP UTILITY
7424 024124 000004          MOV    #354,R0       ;LOAD R0 WITH TEST NUMBER
7425 024124 012700          MOV    #206,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
7426 024124 013701 000354      MOV    #MBUF0,R2     ;DEST ADDR = MBUF0
7427 024124 012703 024156      MOV    #177400,R4    ;RESULT S / B = 177400
7428 024124 012704 063312      MOV    R2,R3         ;R3 CONTAINS DEST ADDR
7429 024124 012704 177400      MOV    #1,(R2)       ;[DEST] = 177777
7430 024124 012704 177400      CCC                    ;SCOPE SYNC
7431
7432 024156 105023          2$:   CLR B    (R3)+    ;TEST THE CLR B
7433
7434 024160 022703 063313      CMP    #MBUF0+1,R3   ;DID DEST REG GET INCREMENTED ?
7435 024164 001401          BEQ    4$            ;BR IF YES
7436
7437 024166 104005          3$:   ERROR 5          ;CLR B FAILED TO UPDATE DEST REG
7438
7439 024170 020412          4$:   CMP    R4,(R2)    ;CORRECT RESULT ?
7440 024172 001402          BEQ                    ;BR IF YES
7441
7442 024174 011203          5$:   MOV    (R2),R3     ;GET THE WAS DATA
7443 024176 104001          ERROR 1              ;CLR B DELIVERED WRONG RESULT
7444
;*****
;TEST 355 CLR B TEST - DM1 - EVEN ADDRESS
;*****
TST355:
7445 024200          SCOPE                ;CALL THE SCOPE LOOP UTILITY
7446 024200 000004          MOV    #355,R0       ;LOAD R0 WITH TEST NUMBER
7447 024200 012700          MOV    #225,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
7448 024200 013701 000355      MOV    #MBUF0,R2     ;DEST ADDR = MBUF0
7449 024200 012703 024332      MOV    #177400,R4    ;RESULT S / B = 177400
7450 024200 012704 063312      MOV    R2,R3         ;R3 CONTAINS DEST ADDR
7451 024200 012704 177400      MOV    #1,(R2)       ;[DEST] = 177777
7452 024200 000257          CCC                    ;SCOPE SYNC
7453
7454 024232 105013          2$:   CLR B    (R3)    ;TEST THE CLR B
7455
7456 024234 020412          CMP    R4,(R2)       ;CORRECT RESULT ?
7457 024236 001402          BEQ    T$T356       ;BR IF YES

```

```

7466 024240 011203          3$:   MOV    (R2),R3    ;GET THE WAS DATA
7467 024242 104001          ERROR 1              ;CLR B DELIVERED WRONG RESULT
7468
;*****
;TEST 356 NEGB TEST - DM2 - ODD ADDRESS
;*****
TST356:
7469 024244          SCOPE                ;CALL THE SCOPE LOOP UTILITY
7470 024244 000004          MOV    #356,R0       ;LOAD R0 WITH TEST NUMBER
7471 024244 012700          MOV    #225,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
7472 024244 013701 000356      MOV    #MBUF0+1,R2   ;DEST ADDR = MBUF0+1
7473 024244 012703 024302      MOV    #777,R4        ;RESULT S / B = 777
7474 024244 012704 063313      MOV    #MBUF0,R5     ;POINT R5 TO CHECK RESULT
7475 024244 012705 063312      MOV    R2,R3         ;R3 CONTAINS DEST ADDR
7476 024244 012715 177777      MOV    #-1,(R5)     ;[DEST] = 177777
7477 024244 000257          CCC                    ;SCOPE SYNC
7478
7479 024302 105423          2$:   NEGB   (R3)+    ;TEST THE NEGB
7480
7481 024304 022703 063314      CMP    #MBUF0+2,R3   ;DID DEST REG GET INCREMENTED ?
7482 024310 001401          BEQ    4$            ;BR IF YES
7483
7484 024312 104005          3$:   ERROR 5          ;NEGB FAILED TO UPDATE DEST REG
7485
7486 024314 020415          4$:   CMP    R4,(R5)    ;CORRECT RESULT ?
7487 024316 001402          BEQ                    ;BR IF YES
7488
7489 024320 011503          5$:   MOV    (R5),R3     ;GET THE WAS DATA
7490 024322 104001          ERROR 1              ;NEGB DELIVERED WRONG RESULT
7491
;*****
;TEST 357 NEGB TEST - DM1 - ODD ADDRESS
;*****
TST357:
7492 024324          SCOPE                ;CALL THE SCOPE LOOP UTILITY
7493 024324 000004          MOV    #357,R0       ;LOAD R0 WITH TEST NUMBER
7494 024324 012700          MOV    #225,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
7495 024324 013701 000357      MOV    #MBUF0+1,R2   ;DEST ADDR = MBUF0+1
7496 024324 012703 024374      MOV    #777,R4        ;RESULT S / B = 777
7497 024324 012704 063313      MOV    #MBUF0,R5     ;POINT R5 TO CHECK RESULT
7498 024324 012705 063312      MOV    R2,R3         ;R3 CONTAINS DEST ADDR
7499 024324 012715 177777      MOV    #-1,(R5)     ;[DEST] = 177777
7500 024324 000257          CCC                    ;SCOPE SYNC
7501
7502 024374 105413          2$:   NEGB   (R3)      ;TEST THE NEGB
7503
7504 024376 020415          CMP    R4,(R5)       ;CORRECT RESULT ?
7505 024378 001402          BEQ    T$T360       ;BR IF YES
7506
7507 024402 011503          MOV    (R5),R3       ;GET THE WAS DATA

```

```

7521 024404 104001
7522
7523
7524
7525
7526 024406
7527 024406 000004
7528 024410 012700 000360
7529 024414 013701 024440
7530 024420 012702 063312
7531 024424 012704 177401
7532 024430 010203
7533 024432 012712 177777
7534 024436 000257
7535
7536 024440 105423
7537
7538 024442 022703 063313
7539 024446 001401
7540
7541 024450 104005
7542
7543 024452 020412
7544 024454 001402
7545
7546 024456 011203
7547 024460 104001
7548
7549
7550
7551 024462
7552 024462 000004
7553 024464 012700 000361
7554 024464 012700 024514
7555 024470 013701 063312
7556 024474 012702 177401
7557 024500 012704
7558 024504 010203
7559 024506 012712 177777
7560 024512 000257
7561
7562 024514 105413
7563
7564 024516 020412
7565 024520 001402
7566
7567 024522 011203
7568 024524 104001
7569
7570
7571
7572
7573 024526
7574 024526 000004
7575 024530 012700 000362
7576 024534 013701 024556

```

```

3$: ERROR 1 ;NEGB DELIVERED WRONG RESULT
;*****
;TEST 360 NEGB TEST - DM2 - EVEN ADDRESS
;*****
TST360:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #360,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177401,R4 ;RESULT S / B = 177401
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R2) ;DESTJ = 177777
CCC ;SCOPE SYNC

2$: NEGB (R3)+ ;TEST THE NEGB

CMP #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED ?
BEQ 4$ ;BR IF YES

3$: ERROR 5 ;NEGB FAILED TO UPDATE DEST REG

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST361 ;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;NEGB DELIVERED WRONG RESULT
;*****
;TEST 361 NEGB TEST - DM1 - EVEN ADDRESS
;*****
TST361:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #361,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177401,R4 ;RESULT S / B = 177401
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R2) ;DESTJ = 177777
CCC ;SCOPE SYNC

2$: NEGB (R3) ;TEST THE NEGB

CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST362 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;NEGB DELIVERED WRONG RESULT
;*****
;TEST 362 ADD TEST - SMO,DMO - N:C = 1010
;*****
TST362:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #362,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

```

```

7577 024540 005004
7578 024542 012705 177777
7579 024546 012703 000001
7580 024552 000257
7581 024554 000272
7582
7583 024556 060503
7584
7585 024560 100403
7586 024562 001002
7587 024564 102401
7588 024566 103401
7589
7590 024570 104002
7591
7592 024572 020403
7593 024574 001401
7594
7595 024576 104002
7596
7597
7598
7599
7600 024600
7601 024600 000004
7602 024602 012700 000363
7603 024606 013701 024632
7604 024612 012704 100006
7605 024616 012705 077777
7606 024622 012703 000007
7607 024626 000257
7608 024630 000265
7609
7610 024632 060503
7611
7612 024634 100003
7613 024636 001402
7614 024640 102001
7615 024642 103001
7616
7617 024644 104002
7618
7619 024646 020403
7620 024650 001401
7621
7622 024652 104002
7623
7624
7625
7626
7627 024654
7628 024654 000004
7629 024656 012700 000364
7630 024662 013701 024702
7631 024666 012704 063322
7632 024672 012705 063276

```

```

CLR R4 ;RESULT S / B = 000000
MOV #-1,R5 ;SRC DPR = 177777
MOV #+1,R3 ;DESTJ = +1
CCC ;CLEAR FLAGS
272 ;N:C = 1010

2$: ADD R5,R3 ;TEST THE ADD

BMI 3$ ;N:C = 0101
BNE 3$
BVS 3$
BCS 4$

3$: ERROR 2 ;ADD FAILED TO ALTER CODES PROPERLY

4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST363 ;BR IF YES

5$: ERROR 2 ;ADD DELIVERED THE WRONG RESULT
;*****
;TEST 363 ADD TEST - SMO,DMO - N:C = 0101
;*****
TST363:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #363,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100006,R4 ;RESULT S / B = 100006
MOV #177777,R5 ;SRC DPR = 177777
MOV #7,R3 ;DESTJ = 7
CCC ;CLEAR FLAGS
265 ;N:C = 0101

2$: ADD R5,R3 ;TEST THE ADD

BPL 3$ ;N:C = 1010
BEQ 3$
BVC 3$
BCC 4$

3$: ERROR 2 ;ADD FAILED TO ALTER CODES PROPERLY

4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST364 ;BR IF YES

5$: ERROR 2 ;ADD DELIVERED THE WRONG RESULT
;*****
;TEST 364 ADD SM1,DMO TEST
;*****
TST364:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #364,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #DWTA,R4 ;RESULT S / B = #DWTA
MOV #ATA,R5 ;SOURCE ADDR = ATA

```

```

7633 024676 005003
7634 024700 000257
7635 024702 061503
7636 024704 020403
7637 024706 001401
7638 024710 104002
7639 024712 022705 063276
7640 024716 001401
7641 024720 104005
7642 *****
7643 ;*TEST 365 ADD SM2,DMO TEST
7644 ;*TEST 365:
7645 SCOPE #365,R0 ;CALL THE SCOPE LOOP UTILITY
7646 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
7647 MOV #DWT,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
7648 MOV #ATA,R5 ;RESULT S / B = #DWT
7649 CLR R3 ;SOURCE ADDR = ATA
7650 CCC ;DESTJ = 0
7651 ;SCOPE SYNC
7652 2$: ADD (R5)+,R3 ;TEST THE ADD - SM2,DMO
7653 CMP R4,R3 ;RESULT = #DWT
7654 BEQ 4$ ;BR IF YES
7655 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7656 4$: CMP #ATA+2,R5 ;DID ADD AUTO INCREMENT SOURCE REG?
7657 BEQ #TST366 ;BR IF YES
7658 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.
7659 *****
7660 ;*TEST 366 ADD SM3,DMO TEST
7661 ;*TEST 366:
7662 SCOPE #366,R0 ;CALL THE SCOPE LOOP UTILITY
7663 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
7664 MOV #DWT,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
7665 MOV #ATA+10,R5 ;RESULT S / B = #DWT
7666 MOV R4,@#MBUF0 ;RS POINTS TO SOURCE ADDR
7667 CLR R3 ;[SOURCE] = #DWT
7668 CCC ;DESTJ = 0
7669 ;SCOPE SYNC
7670 2$: ADD @(R5)+,R3 ;TEST THE ADD - SM3,DMO
7671 CMP R4,@#MBUF0 ;RESULT = #DWT
7672 BEQ 4$ ;BR IF YES
7673 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7674 4$: CMP #ATA+10,R5 ;DID ADD AUTO INCREMENT SOURCE REG?
7675 BEQ #TST366 ;BR IF YES
7676 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.
7677 *****
7678 ;*TEST 367 ADD SM4,DMO TEST
7679 ;*TEST 367:
7680 SCOPE #367,R0 ;CALL THE SCOPE LOOP UTILITY
7681 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
7682 MOV #DWT,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
7683 MOV #ATA+2,R5 ;RESULT S / B = #DWT
7684 CLR R3 ;SOURCE ADDR = ATA
7685 CCC ;DESTJ = 0
7686 ;SCOPE SYNC
7687 2$: ADD -(R5),R3 ;TEST THE ADD - SM4,DMO
7688 CMP R4,R3 ;RESULT = #DWT
7689 BEQ 4$ ;BR IF YES
7690 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7691 4$: CMP #ATA+12,R5 ;DID ADD AUTO INCREMENT SOURCE REG?
7692 BEQ #TST367 ;BR IF YES
7693 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.
7694 *****
7695 ;*TEST 368 ADD SM5,DMO TEST
7696 ;*TEST 368:
7697 SCOPE #368,R0 ;CALL THE SCOPE LOOP UTILITY
7698 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
7699 MOV #DWT,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
7700 MOV #ATA+10,R5 ;RESULT S / B = #DWT
7701 MOV R4,@#MBUF0 ;RS POINTS TO SOURCE ADDR
7702 CLR R3 ;[SOURCE] = #DWT
7703 CCC ;DESTJ = 0
7704 ;SCOPE SYNC
7705 2$: ADD @-(R5),R3 ;TEST THE ADD - SM5,DMO
7706 CMP R4,@#MBUF0 ;RESULT = #DWT
7707 BEQ 4$ ;BR IF YES
7708 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7709 4$: CMP #ATA+10,R5 ;DID ADD DECREMENT SOURCE REG?
7710 BEQ #TST371 ;BR IF YES
7711 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.
7712 *****
7713 ;*TEST 369 ADD SM6,DMO TEST
7714 ;*TEST 369:
7715 SCOPE #369,R0 ;CALL THE SCOPE LOOP UTILITY
7716 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
7717 MOV #DWT,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
7718 MOV #ATA+10,R5 ;RESULT S / B = #DWT
7719 MOV R4,@#MBUF0 ;RS POINTS TO SOURCE ADDR
7720 CLR R3 ;[SOURCE] = #DWT
7721 CCC ;DESTJ = 0
7722 ;SCOPE SYNC
7723 2$: ADD @-(R5),R3 ;TEST THE ADD - SM6,DMO
7724 CMP R4,@#MBUF0 ;RESULT = #DWT
7725 BEQ 4$ ;BR IF YES
7726 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7727 4$: CMP #ATA+10,R5 ;DID ADD DECREMENT SOURCE REG?
7728 BEQ #TST371 ;BR IF YES
7729 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.
7730 *****
7731 ;*TEST 370 ADD SM7,DMO TEST
7732 ;*TEST 370:
7733 SCOPE #370,R0 ;CALL THE SCOPE LOOP UTILITY
7734 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
7735 MOV #DWT,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
7736 MOV #ATA+12,R5 ;RESULT S / B = #DWT
7737 MOV R4,@#MBUF0 ;RS POINTS TO SOURCE ADDR
7738 CLR R3 ;[SOURCE] = #DWT
7739 CCC ;DESTJ = 0
7740 ;SCOPE SYNC
7741 2$: ADD @-(R5),R3 ;TEST THE ADD - SM7,DMO
7742 CMP R4,@#MBUF0 ;RESULT = #DWT
7743 BEQ 4$ ;BR IF YES
7744 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7745 4$: CMP #ATA+12,R5 ;DID ADD AUTO INCREMENT SOURCE REG?
7746 BEQ #TST370 ;BR IF YES
7747 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.

```

```

7689 025032 104002
7690 025034 022705 063310
7691 025040 001401
7692 025042 104005
7693 *****
7694 ;*TEST 367 ADD SM4,DMO TEST
7695 ;*TEST 367:
7696 SCOPE #367,R0 ;CALL THE SCOPE LOOP UTILITY
7697 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
7698 MOV #DWT,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
7699 MOV #ATA+2,R5 ;RESULT S / B = #DWT
7700 CLR R3 ;SOURCE ADDR = ATA
7701 CCC ;DESTJ = 0
7702 ;SCOPE SYNC
7703 2$: ADD -(R5),R3 ;TEST THE ADD - SM4,DMO
7704 CMP R4,R3 ;RESULT = #DWT
7705 BEQ 4$ ;BR IF YES
7706 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7707 4$: CMP #ATA,R5 ;DID SOURCE REG GET DECREMENTED?
7708 BEQ #TST370 ;BR IF YES
7709 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.
7710 *****
7711 ;*TEST 370 ADD SM5,DMO TEST
7712 ;*TEST 370:
7713 SCOPE #370,R0 ;CALL THE SCOPE LOOP UTILITY
7714 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
7715 MOV #DWT,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
7716 MOV #ATA+12,R5 ;RESULT S / B = #DWT
7717 MOV R4,@#MBUF0 ;RS POINTS TO SOURCE ADDR
7718 CLR R3 ;[SOURCE] = #DWT
7719 CCC ;DESTJ = 0
7720 ;SCOPE SYNC
7721 2$: ADD @-(R5),R3 ;TEST THE ADD - SM5,DMO
7722 CMP R4,@#MBUF0 ;RESULT = #DWT
7723 BEQ 4$ ;BR IF YES
7724 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7725 4$: CMP #ATA+10,R5 ;DID ADD DECREMENT SOURCE REG?
7726 BEQ #TST371 ;BR IF YES
7727 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.
7728 *****
7729 ;*TEST 371 ADD SM6,DMO TEST
7730 ;*TEST 371:
7731 SCOPE #371,R0 ;CALL THE SCOPE LOOP UTILITY
7732 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
7733 MOV #DWT,R4 ;LOAD R1 WITH TEST INSTRUCTION WORD
7734 MOV #ATA+10,R5 ;RESULT S / B = #DWT
7735 MOV R4,@#MBUF0 ;RS POINTS TO SOURCE ADDR
7736 CLR R3 ;[SOURCE] = #DWT
7737 CCC ;DESTJ = 0
7738 ;SCOPE SYNC
7739 2$: ADD @-(R5),R3 ;TEST THE ADD - SM6,DMO
7740 CMP R4,@#MBUF0 ;RESULT = #DWT
7741 BEQ 4$ ;BR IF YES
7742 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7743 4$: CMP #ATA+10,R5 ;DID ADD DECREMENT SOURCE REG?
7744 BEQ #TST371 ;BR IF YES
7745 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.

```

7745
7746
7747
7748
7750 025166 000004
7751 025170 012700 000371
7752 025174 013701 025214
7753 025200 012704 063312
7754 025204 012705 063276
7755 025210 005003
7756 025212 000257
7757
7758 025214 066503 000010
7759
7760 025220 020403
7761 025222 001401
7762
7763 025224 104002
7764
7765
7766
7767
7768 025226 000004
7769 025226 012700 000372
7770 025230 012701 025260
7771 025234 012704 063312
7772 025240 012705 063276
7773 025244 012705 063312
7774 025250 010437 063312
7775 025254 005003
7776 025256 000257
7777
7778 025260 067503 000010
7779
7780 025264 020403
7781 025266 001401
7782
7783 025270 104002
7784
7785
7786
7787
7788 025272 000004
7789 025274 012700 000373
7790 025300 013701 025324
7791 025304 012702 063312
7792 025310 012704 063322
7793 025314 012705 063276
7794 025322 000257
7795
7796 025324 061512
7797
7798 025326 020412
7799
7800

```

;*****
;TEST 371 ADD SM6,DM0 TEST
;*****
TST371:  SCOPE                ;CALL THE SCOPE LOOP UTILITY
        MOV #371,R0         ;LOAD R0 WITH TEST NUMBER
        MOV #25,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV @MBUF0,R4      ;RESULT S / B = MBUF0
        MOV #ATA,R5        ;BASE SOURCE ADDR = ATA
        CLR R3             ;[DEST] = 0
        CCC               ;SCOPE SYNC

2$:     ADD 10(R5),R3       ;TEST THE ADD - SM6,DM0

        CMP R4,R3         ;RESULT = MBUF0?
        BEQ TST372       ;BR IF YES

3$:     ERROR 2           ;ADD DELIVERED WRONG RESULT

;*****
;TEST 372 ADD SM7,DM0 TEST
;*****
TST372:  SCOPE                ;CALL THE SCOPE LOOP UTILITY
        MOV #372,R0         ;LOAD R0 WITH TEST NUMBER
        MOV #25,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV @MBUF0,R4      ;RESULT S / B = MBUF0
        MOV #ATA,R5        ;BASE SOURCE ADDR = ATA
        CLR R3             ;[SOURCE] = #DWTA
        CCC               ;[DEST] = 0
        CCC               ;SCOPE SYNC

2$:     ADD @10(R5),R3     ;TEST THE ADD - SM7,DM0

        CMP R4,R3         ;RESULT = #DWTA?
        BEQ TST373       ;BR IF YES

3$:     ERROR 2           ;ADD DELIVERED WRONG RESULT

;*****
;TEST 373 ADD SM1,DM1 TEST
;*****
TST373:  SCOPE                ;CALL THE SCOPE LOOP UTILITY
        MOV #373,R0         ;LOAD R0 WITH TEST NUMBER
        MOV #25,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV @MBUF0,R2      ;DEST ADDR = MBUF0
        MOV #DWTA,R4       ;RESULT S / B = #DWTA
        MOV #ATA,R5        ;SOURCE ADDR = ATA
        CLR (R2)           ;[DEST] = 0
        CCC               ;SCOPE SYNC

2$:     ADD (R5),(R2)      ;TEST THE ADD - SM1,DM1

        CMP R4,(R2)       ;RESULT = #DWTA?

```

7801 025330 001402
7802
7803 025332 011203
7804 025334 104001
7805
7806
7807
7808
7809 025336 000004
7810 025336 012700 000374
7811 025340 012701 025370
7812 025344 012702 063312
7813 025350 012704 063322
7814 025354 012705 063276
7815 025360 012705 063276
7816 025364 005012
7817 025366 000257
7818
7819 025370 062512
7820
7821 025372 020412
7822 025374 001402
7823
7824 025376 011203
7825 025400 104001
7826
7827
7828
7829
7830 025402 000004
7831 025402 012700 000375
7832 025404 013701 025436
7833 025410 012702 063312
7834 025414 012704 063322
7835 025420 012705 063276
7836 025424 012705 063276
7837 025430 010203
7838 025432 005012
7839 025434 000257
7840
7841 025436 061523
7842
7843 025440 020412
7844 025442 001406
7845
7846 025444 010337 063316
7847 025450 011203
7848 025452 104001
7849
7850 025454 013703 063316
7851 025460 022703 063314
7852 025464 001401
7853
7854 025466 104005
7855
7856

```

        BEQ TST374         ;BR IF YES

3$:     MOV (R2),R3       ;GET WAS DATA
        ERROR 1           ;ADD DELIVERED WRONG RESULT

;*****
;TEST 374 ADD SM2,DM1 TEST
;*****
TST374:  SCOPE                ;CALL THE SCOPE LOOP UTILITY
        MOV #374,R0         ;LOAD R0 WITH TEST NUMBER
        MOV #25,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV @MBUF0,R2      ;DEST ADDR = MBUF0
        MOV #DWTA,R4       ;RESULT S / B = #DWTA
        MOV #ATA,R5        ;SOURCE ADDR = ATA
        CLR (R2)           ;[DEST] = 0
        CCC               ;SCOPE SYNC

2$:     ADD (R5)+,(R2)    ;TEST THE ADD - SM2,DM1

        CMP R4,(R2)       ;RESULT = #DWTA?
        BEQ TST375       ;BR IF YES

3$:     MOV (R2),R3       ;GET WAS DATA
        ERROR 1           ;ADD DELIVERED WRONG RESULT

;*****
;TEST 375 ADD SM1,DM2 TEST
;*****
TST375:  SCOPE                ;CALL THE SCOPE LOOP UTILITY
        MOV #375,R0         ;LOAD R0 WITH TEST NUMBER
        MOV #25,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV @MBUF0,R2      ;DEST ADDR = MBUF0
        MOV #DWTA,R4       ;RESULT S / B = #DWTA
        MOV #ATA,R5        ;SOURCE ADDR = ATA
        MOV #R3,R4         ;[R3] = DEST ADDR
        CLR (R2)           ;[DEST] = 0
        CCC               ;SCOPE SYNC

2$:     ADD (R5),(R3)+    ;TEST THE ADD - SM1,DM2

        CMP R4,(R2)       ;RESULT = #DWTA?
        BEQ 4$           ;BR IF YES

3$:     MOV R3,@MBUF1     ;SAVE UPDATED DEST ADDR
        MOV (R2),R3       ;GET WAS DATA
        ERROR 1           ;ADD DELIVERED WRONG RESULT

4$:     MOV @MBUF1,R3     ;RESTORE UPDATED DEST ADDR
        CMP @MBUF0+2,R3   ;DID ADD INCREMENT DEST REG
        BEQ TST376       ;BR IF YES

5$:     ERROR 5           ;ADD FAILED TO UPDATE DEST REG

;*****

```



```

7969
7970 026022 104005
7971
7972
7973
7974
7975 026024
7976 026024 000004
7977 026026 012700 000402
7978 026032 013701 026062
7979 026036 012702 063312
7980 026042 012704 063322
7981 026046 012705 063276
7982 026052 012703 063314
7983 026056 005012
7984 026060 000257
7985
7986 026062 061543
7987
7988 026064 020412
7989 026066 001406
7990
7991 026070 010337 063316
7992 026074 011203
7993 026076 104001
7994
7995 026100 013703 063316
7996 026104 020302
7997 026106 001401
7998
7999 026110 104005
8000
8001
8002
8003
8004
8005 026112
8006 026112 000004
8007 026114 012700 000403
8008 026120 013701 026150
8009 026130 012702 063312
8010 026134 012704 063322
8011 026134 012705 063276
8012 026140 012703 063310
8013 026144 005012
8014 026146 000257
8015
8016 026150 061553
8017 026152 020412
8018 026154 001406
8019
8020 026156 010337 063316
8021 026158 011203
8022 026164 104001
8023
8024 026166 013703 063316

```

```

5$: ERROR 5 ;ADD FAILED TO UPDATE DEST REG.
;*****
;TEST 402 ADD SM2,DM4 TEST
;*****
TST402:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #402,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #BUF0,R2 ;DEST ADDR = MBUF0
MOV #DWT4,R4 ;RESULT S / B = #DWT4
MOV #ATA,R5 ;SOURCE ADDR = ATA
MOV #BUF0+2,R3 ;R3 POINTS TO DEST ADDR +2
CLR (R2) ;CDESTJ = 0
CCC ;SCOPE SYNC

2$: ADD (R5),-(R3) ;TEST THE ADD - SM2,DM4
CMP R4,(R2) ;RESULT = #DWT4?
BEQ 4$ ;BR IF YES

3$: MOV R3,@#MBUF1 ;SAVE R3
MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;ADD DELIVERED WRONG RESULT

4$: MOV @#MBUF1,R3 ;RESTORE R3
CMP R3,R2 ;DID ADD INCREMENT DEST REG?
BEQ TST403 ;BR IF YES

5$: ERROR 5 ;ADD FAILED TO UPDATE DEST REG.
;*****
;TEST 403 ADD SM1,DM5 TEST
;*****
TST403:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #403,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #BUF0,R2 ;DEST ADDR = MBUF0
MOV #DWT4,R4 ;RESULT S / B = #DWT4
MOV #ATA,R5 ;SOURCE ADDR = ATA
MOV #ATA+12,R3 ;R3 CONTAINS ADDR OF DEST ADDR PLUS 2
CLR (R2) ;CDESTJ = 0
CCC ;SCOPE SYNC

2$: ADD (R5),@(R3) ;TEST THE ADD - SM1,DM5
CMP R4,(R2) ;RESULT = #DWT4?
BEQ 4$ ;BR IF YES

3$: MOV R3,@#MBUF1 ;SAVE R3
MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;ADD DELIVERED WRONG RESULT

MOV @#MBUF1,R3 ;RESTORE R3

```

```

8025 026172 022703 063306
8026 026176 001401
8027
8028 026200 104005
8029
8030
8031
8032
8033 026202
8034 026202 000004
8035 026204 012700 000404
8036 026210 013701 026240
8037 026214 012702 063312
8038 026214 012704 063322
8039 026214 012705 063276
8040 026230 012703 063310
8041 026234 005012
8042 026236 000257
8043
8044 026240 062553
8045
8046 026242 020412
8047 026244 001406
8048
8049 026246 010337 063316
8050 026252 011203
8051 026254 104001
8052
8053 026256 013703 063316
8054 026262 022703 063306
8055 026266 001401
8056
8057 026270 104005
8058
8059
8060
8061
8062 026272
8063 026272 000004
8064 026274 012700 000405
8065 026300 013701 026330
8066 026304 012702 063316
8067 026310 012704 063322
8068 026314 012705 063276
8069 026320 012703 063312
8070 026324 005012
8071 026326 000257
8072
8073 026330 061563 000004
8074
8075 026334 020412
8076 026336 001402
8077
8078 026340 011203
8079 026342 104001
8080

```

```

4$: CMP #ATA+10,R3 ;DID ADD DECREMENT DEST REG?
BEQ TST404 ;BR IF YES

5$: ERROR 5 ;ADD FAILED TO UPDATE DEST REG.
;*****
;TEST 404 ADD SM2,DM5 TEST
;*****
TST404:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #404,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #BUF0,R2 ;DEST ADDR = MBUF0
MOV #DWT4,R4 ;RESULT S / B = #DWT4
MOV #ATA,R5 ;SOURCE ADDR = ATA
MOV #ATA+12,R3 ;R3 CONTAINS ADDR OF DEST ADDR PLUS 2
CLR (R2) ;CDESTJ = 0
CCC ;SCOPE SYNC

2$: ADD (R5)+,@(R3) ;TEST THE ADD - SM2,DM5
CMP R4,(R2) ;RESULT = #DWT4?
BEQ 4$ ;BR IF YES

3$: MOV R3,@#MBUF1 ;SAVE R3
MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;ADD DELIVERED WRONG RESULT

4$: MOV @#MBUF1,R3 ;RESTORE R3
CMP #ATA+10,R3 ;DID ADD DECREMENT DEST REG?
BEQ TST405 ;BR IF YES

5$: ERROR 5 ;ADD FAILED TO UPDATE DEST REG
;*****
;TEST 405 ADD SM1,DM6 TEST
;*****
TST405:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #405,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #BUF0+4,R2 ;DEST ADDR = MBUF0+4
MOV #DWT4,R4 ;RESULT S / B = #DWT4
MOV #ATA,R5 ;SOURCE ADDR = ATA
MOV #BUF0,R3 ;R3 = BASE DEST ADDR
CLR (R2) ;CDESTJ = 0
CCC ;SCOPE SYNC

2$: ADD (R5),4(R3) ;TEST THE ADD - SM1,DM6
CMP R4,(R2) ;RESULT = #DWT4?
BEQ TST406 ;BR IF YES

3$: MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;ADD DELIVERED WRONG RESULT

```



```

8193 026642 104002
8194
8195 026644 020403
8196 026646 001401
8197
8198 026650 104002
8199
8200
8201
8202
8203 026652
8204 026654 000004
8205 026656 013700 000413
8206 026660 013701 026704
8207 026664 012704 177777
8208 026670 012705 125252
8209 026674 012703 052525
8210 026700 002656
8211 026702 000266
8212
8213 026704 074503
8214
8215 026706 100003
8216 026710 001402
8217 026714 103401
8218 026714 103001
8219
8220 026716 104002
8221
8222 026720 020403
8223 026722 001401
8224
8225 026724 104002
8226
8227
8228
8229
8230 026726
8231 026726 000004
8232 026730 012700 000414
8233 026734 013701 026704
8234 026740 012704 177777
8235 026744 012705 052525
8236 026750 012703 125252
8237 026754 000257
8238 026756 000271
8239
8240 026760 074503
8241
8242 026762 100003
8243 026764 001402
8244 026766 103401
8245 026770 103401
8246
8247 026772 104002
8248

```

```

3$: ERROR 2 ;XOR FAILED TO SET FLAGS PROPERLY
4$: CMP R4,R3 ;RESULT CORRECT?
BEQ TST413 ;;BR IF YES
5$: ERROR 2 ;XOR DELIVERED THE WRONG RESULT
;*****
;TEST 413 "XOR RA,RB" TEST - A=125252,B=052525 N:C=0110
;*****
TST413: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #413,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S/B = 177777
MOV #125252,R5 ;MASK=125252
MOV #052525,R3 ;DESTJ = 052525
CCC ;SCOPE SYNC
266 ;MAKE N:C=0110
2$: XOR R5,R3 ;TEST THE XOR
BPL 3$ ;N:C=1000 ??
BEQ 3$
BVS 3$
BCS 4$
3$: ERROR 2 ;XOR FAILED TO SET FLAGS PROPERLY
4$: CMP R4,R3 ;RESULT CORRECT?
BEQ TST414 ;;BR IF YES
5$: ERROR 2 ;XOR DELIVERED THE WRONG RESULT
;*****
;TEST 414 "XOR RA,RB" TEST - A=052525,B=125252 N:C=1001
;*****
TST414: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #414,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S/B = 177777
MOV #52525,R5 ;MASK=052525
MOV #125252,R3 ;DESTJ = 125252
CCC ;SCOPE SYNC
271 ;MAKE N:C=1001
2$: XOR R5,R3 ;TEST THE XOR
BPL 3$ ;N:C=1001 ??
BEQ 3$
BVS 3$
BCS 4$
3$: ERROR 2 ;XOR FAILED TO SET FLAGS PROPERLY

```

```

8249 026774 020403
8250 026776 001401
8251
8252 027000 104002
8253
8254
8255
8256
8257 027002
8258 027002 000004
8259 027004 012700 000415
8260 027010 013701 027032
8261 027014 005004
8262 027016 005005
8263 027020 012702 063312
8264 027024 005012
8265 027026 000257
8266 027030 000272
8267
8268 027032 074512
8269
8270 027034 100403
8271 027036 001002
8272 027040 102401
8273 027042 103001
8274
8275 027044 104001
8276
8277 027046 020412
8278 027050 001402
8279
8280 027052 011203
8281 027054 104001
8282
8283
8284
8285
8286 027056
8287 027056 000004
8288 027060 012700 000416
8289 027064 013701 027112
8290 027070 005004
8291 027072 012705 177777
8292 027076 012702 063312
8293 027102 012712 177777
8294 027106 000257
8295 027110 000265
8296
8297 027112 074512
8298
8299 027114 100403
8300 027116 001002
8301 027120 102401
8302 027122 103401
8303
8304 027124 104001

```

```

4$: CMP R4,R3 ;RESULT CORRECT?
BEQ TST415 ;;BR IF YES
5$: ERROR 2 ;XOR DELIVERED THE WRONG RESULT
;*****
;TEST 415 "XOR RA,(RB)" TEST - A=B=000000 N:C=1010
;*****
TST415: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #415,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,P1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S/B = 000000
CLR R5 ;MASK = 000000
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR (R2) ;DESTJ = 000000
CCC ;SCOPE SYNC
272 ;MAKE N:C=1010
2$: XOR R5,(R2) ;TEST THE XOR
BMI 3$ ;N:C = 0100 ??
BNE 3$
BVS 3$
BCS 4$
3$: ERROR 1 ;XOR FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;RESULT CORRECT?
BEQ TST416 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;XOR DELIVERED THE WRONG RESULT
;*****
;TEST 416 "XOR RA,(RB)" TEST - A=B=177777 N:C=0101
;*****
TST416: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #416,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S/B = 000000
MOV #-1,R5 ;MASK = 177777
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR #-1,(R2) ;DESTJ = 177777
CCC ;SCOPE SYNC
265 ;MAKE N:C=0101
2$: XOR R5,(R2) ;TEST THE XOR
BMI 3$ ;N:C = 0101 ??
BNE 3$
BVS 3$
BCS 4$
3$: ERROR 1 ;XOR FAILED TO ALTER CODES PROPERLY

```

8305
8306 027126 020412
8307 027130 001402
8308
8309 027132 011203
8310 027134 104001
8311
8312
8313
8314
8315 027136
8316 027136 000004
8317 027140 012700 000417
8318 027144 013701 027206
8319
8320 027150 032737 000400 063234
8321 027156 001401
8322 027160 000000
8323 027162 012704 177777
8324 027166 012705 125252
8325 027172 012702 063312
8326 027172 012712 052525
8327 027202 000257
8328 027204 000266
8329
8330 027206 074512
8331
8332 027210 100003
8333 027212 001402
8334 027214 102401
8335 027216 103001
8336
8337 027220 104001
8338
8339 027222 020412
8340 027224 001402
8341
8342 027226 011203
8343 027230 104001
8344
8345
8346
8347
8348 027232
8349 027234 000004
8350 027240 012700 000420
8351 027240 013701 027270
8352 027244 012704 177777
8353 027250 012705 052525
8354 027254 012702 063312
8355 027260 012712 125252
8356 027266 000271
8357
8358
8359 027270 074512
8360

4\$: CMP R4,(R2) ;RESULT CORRECT?
BEQ T5T417 ;;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;XOR DELIVERED THE WRONG RESULT
;*****
;*TEST 417 "XOR RA,(RB)" TEST - A=125252,B=052525 N:C=0110
;*****
T5T417: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #417,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
;SBTTL USER CONTROLLED BREAKPOINT -- BIT8
BEQ #BIT8,#BPTLOC ;BREAKPOINT HALT SET ??
HALT .+4 ;BR IF NOT
MOV #1,R4 ;BREAK - DEPRESS CONTINUE TO RESTART
MOV #125252,R5 ;RESULT S/B = 177777
MOV #MBUF0,R2 ;MASK = 125252
MOV #052525,(R2) ;DEST ADDR = MBUF0
CCC ;DEST = 052525
266 ;SCOPE SYNC
;MAKE N:C=0110
2\$: XOR R5,(R2) ;TEST THE XOR
BPL 3\$;N:C = 1000 ??
BEQ 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 1 ;XOR FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT CORRECT?
BEQ T5T420 ;;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;XOR DELIVERED THE WRONG RESULT
;*****
;*TEST 420 "XOR RA,(RB)" TEST - A=052525,B=125252 N:C=1001
;*****
T5T420: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #420,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S/B = 177777
MOV #52525,R5 ;MASK = 052525
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,(R2) ;DEST = 125252
CCC ;SCOPE SYNC
271 ;MAKE N:C=1001
2\$: XOR R5,(R2) ;TEST THE XOR

8361 027272 100003
8362 027274 001402
8363 027276 102401
8364 027300 103401
8365
8366 027302 104001
8367
8368 027304 020412
8369 027306 001402
8370
8371 027310 011203
8372 027312 104001
8373
8374
8375
8376
8377 027314
8378 027314 000004
8379 027316 012700 000421
8380 027322 013701 027342
8381 027326 005004
8382 027330 012703 052525
8383 027334 010305
8384 027336 000257
8385 027340 000273
8386
8387 027342 160503
8388
8389 027344 100403
8390 027346 001002
8391 027350 102401
8392 027352 103001
8393
8394 027354 104002
8395
8396 027356 020304
8397 027360 001401
8398
8399 027362 104002
8400
8401
8402
8403
8404 027364
8405 027364 000004
8406 027366 012700 000422
8407 027372 013701 027412
8408 027376 005004
8409 027400 012703 125252
8410 027404 010305
8411 027406 000257
8412 027410 000273
8413
8414 027412 160503
8415
8416 027414 100403

BPL 3\$;N:C = 1001 ??
BEQ 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 1 ;XOR FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT CORRECT?
BEQ T5T421 ;;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;XOR DELIVERED THE WRONG RESULT
;*****
;*TEST 421 SUB TEST SMO,DMO - (SRC) = (DEST) = +,+
;*****
T5T421: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #421,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 0
MOV #052525,R3 ;[R3] = DEST OP = 52525
MOV R3,R5 ;[R5] = SRC OP = 52525
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
2\$: SUB R5,R3 ;TEST THE SUB
BMI 3\$;DID N:C = 0100
BNE 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R3,R4 ;WAS RESULT = 0?
BEQ T5T422 ;;BR IF YES
5\$: ERROR 2 ;SUB DELIVERED WRONG RESULT
;*****
;*TEST 422 SUB TEST SMO,DMO - (SRC) = (DEST) = -,-
;*****
T5T422: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #422,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 0
MOV #125252,R3 ;[R3] = DEST OP = 125252
MOV R3,R5 ;[R5] = SOURCE OP = 125252
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
2\$: SUB R5,R3 ;TEST THE SUB
BMI 3\$

8417 027416 001002
8418 027420 102401
8419 027422 103001
8420
8421 027424 104002
8422
8423 027426 020304
8424 027430 001401
8425
8426 027432 104002
8427
8428
8429
8430
8431 027434 000004
8432 027436 012700 000423
8433 027440 012701 027466
8434 027442 012704 000002
8435 027446 012703 000001
8436 027452 012705 177777
8437 027456 000257
8438 027462 000267
8439 027464 000276
8440
8441 027466 160503
8442
8443 027470 100403
8444 027472 001402
8445 027474 102401
8446 027476 103401
8447
8448 027500 104002
8449
8450 027502 020304
8451 027504 001401
8452
8453 027506 104002
8454
8455
8456
8457
8458 027510 000004
8459 027512 012700 000424
8460 027516 012701 027542
8461 027520 012704 177777
8462 027526 012703 000001
8463 027532 012705 000001
8464 027536 000257
8465 027540 000267
8466
8467 027542 160503
8468
8469
8470 027544 100003
8471 027546 001402
8472 027550 102401

BNE 35 ;N:C = 0100?
BVS 35
BCC 45
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R3,R4 ;RESULT = 0?
BEQ TS423 ;;BR IF YES
5\$: ERROR 2 ;SUB DELIVERED WRONG RESULT
;*****
;TEST 423 SUB TEST SMO,DMO - (SRC) = (DEST) = -,+
;*****
TST423: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #423,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S / B = 2
MOV #1,R3 ;CR3] = DEST OP = 1
MOV #-1,R5 ;CR5] = SRC OP = -1
CCC ;CLEAR FLAGS
276 ;MAKE N:C = 1110
2\$: SUB R5,R3 ;TEST THE SUB
BNI 35 ;N:C = 0001
BVS 35
BCS 45
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R3,R4 ;RESULT = +2?
BEQ TS424 ;;BR IF YES
5\$: ERROR 2 ;SUB DELIVERED WRONG RESULT
;*****
;TEST 424 SUB TEST SMO,DMO (SRC) = (DEST) = +,-
;*****
TST424: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #424,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S / B = -2
MOV #-1,R3 ;CR3] = [DEST] = -1
MOV #1,R5 ;CR5] = [SOURCE] = +1
CCC ;CLEAR FLAGS
267 ;MAKE N:C = 0111
2\$: SUB R5,R3 ;TEST THE SUB
BPL 35 ;N:C = 1000
BEQ 35
BVS 35

8473 027552 103001
8474
8475 027554 104002
8476
8477 027556 020403
8478 027560 001401
8479
8480 027562 104002
8481
8482
8483
8484
8485 027564 000004
8486 027566 012700 000425
8487 027568 012701 027616
8488 027572 012704 077777
8489 027576 012703 100000
8490 027602 012705 000001
8491 027606 000257
8492 027612 000274
8493
8494
8495 027616 160503
8496
8497
8498 027620 100403
8499 027622 001402
8500 027624 102001
8501 027626 103001
8502
8503 027630 104002
8504
8505 027632 020304
8506 027634 001401
8507
8508 027636 104002
8509
8510
8511
8512 027640
8513 027640 000004
8514 027640 012700 000426
8515 027646 012701 027674
8516 027652 012702 063312
8517 027656 012704 177777
8518 027662 012705 000001
8519 027666 005012
8520 027670 000257
8521 027672 000266
8522
8523 027674 160512
8524
8525 027676 100003
8526 027700 001402
8527 027702 102401
8528 027704 103401

BCC 45
3\$: ERROR 2 ;SUB DID NOT ALTER CODES PROPERLY
4\$: CMP R4,R3 ;RESULT = -2?
BEQ TS425 ;;BR IF YES
5\$: ERROR 2 ;SUB DELIVERED WRONG RESULT
;*****
;TEST 425 SUB TEST SMO,DMO - "V" BIT SETS
;*****
TST425: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #425,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #7777,R4 ;RESULT = 7777?
MOV #100000,R3 ;CR3] = DEST OP = 100000
MOV #1,R5 ;CR5] = SRC OP = 1
CCC ;CLEAR FLAGS
274 ;MAKE N:C = 1100
2\$: SUB R5,R3 ;TEST THE SUB
BNI 35 ;N:C = 0011 ("V" BIT SHOULD SET)
BEQ 35
BVC 35
BCC 45
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R3,R4 ;RESULT = 7777?
BEQ TS426 ;;BR IF YES
5\$: ERROR 2 ;SUB DELIVERED WRONG RESULT
;*****
;TEST 426 SUB TEST - SMO,DMI - N:C = 0110
;*****
TST426: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #426,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #-1,R4 ;RESULT S / B = 17777
MOV #1,R5 ;SRC OPR = +1
CLR (R2) ;[DEST] = 000000
CCC ;CLEAR FLAGS
266 ;N:C = 0110
2\$: SUB R5,(R2) ;TEST THE SUB
BPL 35 ;N:C = 1001
BEQ 35
BVS 35
BCS 45

8529 027706 104001
8530 027710 020412
8531 027712 061402
8532 027714 011203
8533 027716 104001
8534 027720
8535 027722 012700
8536 027726 013701 000427
8537 027732 012702 063312
8538 027736 005004
8539 027740 012705 177777
8540 027744 012712 177777
8541 027750 000257
8542 027752 000272
8543 027754 160512
8544 027756 100403
8545 027760 001002
8546 027762 102401
8547 027764 103001
8548 027766 104001
8549 027770 020412
8550 027772 001402
8551 027774 011203
8552 027776 104001
8553 030000
8554 030002 012700 000430
8555 030006 013701 030034
8556 030012 012702 063312
8557 030016 012704 177777
8558 030022 012705 000001
8559 030026 012712 100000
8560 030032 000257
8561 030034 160512
8562 030036 100403
8563 030040 001402
8564 030042 102001

```
3$: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
   BEQ TS427 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
   ERROR 1 ;SUB DELIVERED THE WRONG RESULT
;*****
;TEST 427 SUB TEST - SM0,DM1 - N:C = 1010
;*****
TST427:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #427,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
MOV #1,R5 ;SRC OPR = 177777
MOV #1,(R2) ;DESTJ = 177777
CCC ;CLEAR FLAGS
272 ;N:C = 1010
2$: SUB R5,(R2) ;TEST THE SUB
   BMI 3$ ;N:C = 0100
   BNE 3$
   BVS 3$
   BCC 4$
3$: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
   BEQ TS430 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
   ERROR 1 ;SUB DELIVERED THE WRONG RESULT
;*****
;TEST 430 SUB TEST - SM0,DM1 - N:C = 0000
;*****
TST430:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #430,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 77777
MOV #1,R5 ;SRC OPR = +1
MOV #10000,(R2) ;DESTJ = 100000
CCC ;CLEAR FLAGS
2$: SUB R5,(R2) ;TEST THE SUB
   BMI 3$ ;N:C = 0010
   BEQ 3$
   BVC 3$
```

8585 030044 103001
8586 030046 104001
8587 030050 020412
8588 030052 061402
8589 030054 011203
8590 030056 104001
8591 030060
8592 030062 012700 000431
8593 030066 013701 030110
8594 030072 012704 177777
8595 030076 012705 064034
8596 030102 005003
8597 030104 000257
8598 030106 000266
8599 030110 161503
8600 030112 100003
8601 030114 001402
8602 030116 102401
8603 030120 103401
8604 030122 104002
8605 030124 020403
8606 030126 001401
8607 030130 104002
8608 030132
8609 030134 012700 000432
8610 030140 013701 030160
8611 030144 000004
8612 030146 012705 063324
8613 030152 011503
8614 030154 000257
8615 030156 000272
8616 030160 161503
8617 030162 100403
8618 030164 001002
8619 030166 102401
8620 030170 103001

```
BCC 4$
3$: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
   BEQ TS431 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
   ERROR 1 ;SUB DELIVERED THE WRONG RESULT
;*****
;TEST 431 SUB TEST - SM1,DM0 - N:C = 0110
;*****
TST431:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #431,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S / B = 177777
MOV DWTB+2,R5 ;SRC ADDR = DWTB+2
CLR R3 ;DESTJ = 000000
CCC ;CLEAR FLAGS
266 ;N:C = 0110
2$: SUB (R5),R3 ;TEST THE SUB
   BPL 3$ ;N:C = 1001
   BEQ 3$
   BVS 3$
   BCS 4$
3$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
   BEQ TS432 ;;BR IF YES
5$: ERROR 2 ;SUB DELIVERED THE WRONG RESULT
;*****
;TEST 432 SUB TEST - SM1,DM0 - N:C = 1010
;*****
TST432:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #432,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV DMTA+2,R5 ;SRC ADDR = DMTA+2
MOV (R5),R3 ;DESTJ = 177777
CCC ;CLEAR FLAGS
272 ;N:C = 1010
2$: SUB (R5),R3 ;TEST THE SUB
   BMI 3$ ;N:C = 0100
   BNE 3$
   BVS 3$
   BCC 4$
```

```

8641
8642 030172 104002
8643
8644 030174 020403
8645 030176 001401
8646
8647 030200 104002
8648
8649
8650
8651
8652 030202
8653 030202 000004
8654 030204 012700 000433
8655 030210 012701 030236
8656 030214 012704 077777
8657 030220 012705 063316
8658 030224 012703 100000
8659 030230 012715 000001
8660 030234 000257
8661
8662 030236 161503
8663
8664 030240 100403
8665 030242 001402
8666 030244 102001
8667 030246 103001
8668
8669 030250 104002
8670
8671 030252 020403
8672 030254 001401
8673
8674 030256 104002
8675
8676
8677
8678
8679 030260
8680 030260 000004
8681 030262 012700 000434
8682 030266 013701 030320
8683 030272 012702 063312
8684 030276 012704 177777
8685 030302 012705 063316
8686 030306 012715 000001
8687 030312 005012
8688 030314 000257
8689 030316 000266
8690
8691 030320 161512
8692
8693 030322 100003
8694 030324 001402
8695 030326 102401
8696 030330 103401

```

```

3$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ T5T433 ;;BR IF YES
5$: ERROR 2 ;SUB DELIVERED THE WRONG RESULT
;*****
;TEST 433 SUB TEST - SM1,DM0 - N:C = 0000
;*****
TST433: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #433,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S / B = 77777
MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
MOV #100000,R3 ;[DEST] = 100000
MOV #+1,(R5) ;SRC OPR = +1
CCC ;CLEAR FLAGS
2$: SUB (R5),R3 ;TEST THE SUB
BMI 3$ ;N:C = 0010
BEQ 3$
BVC 3$
BC 4$
3$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ T5T434 ;;BR IF YES
5$: ERROR 2 ;SUB DELIVERED THE WRONG RESULT
;*****
;TEST 434 SUB SM1,DM1 TEST - N:C = 0110
;*****
TST434: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #434,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #1,R4 ;RESULT S / B = 177777
MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
MOV #+1,(R5) ;[SOURCE] = 000001
CLR (R2) ;[DEST] = 000000
CCC ;CLEAR FLAGS
266 ;N:C = 0110
2$: SUB (R5),(R2) ;TEST THE SUB
BPL 3$ ;N:C = 1001 ?
BEQ 3$
BVS 3$
BCS 4$

```

```

8697
8698 030332 104001
8699
8700 030334 020412
8701 030336 001402
8702
8703 030340 011203
8704 030342 104001
8705
8706
8707
8708
8709 030344
8710 030344 000004
8711 030346 012700 000435
8712 030352 013701 030406
8713 030352 012702 063312
8714 030352 012704 177777
8715 030366 012705 063316
8716 030372 012715 000001
8717 030376 005012
8718 030400 010203
8719 030402 000257
8720 030404 000266
8721
8722 030406 161523
8723
8724 030410 100003
8725 030412 001402
8726 030414 102401
8727 030416 103401
8728
8729 030420 104005
8730
8731 030422 020412
8732 030424 001402
8733
8734 030426 011203
8735 030430 104001
8736
8737
8738
8739
8740 030432
8741 030432 000004
8742 030434 012700 000436
8743 030440 013701 030464
8744 030444 012702 063312
8745 030450 012704 125252
8746 030454 010205
8747 030456 012712 052526
8748 030462 000257
8749
8750 030464 005425
8751
8752 030466 020412

```

```

3$: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ T5T435 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;SUB DELIVERED THE WRONG RESULT
;*****
;TEST 435 SUB SM1,DM2 TEST - N:C = 0110
;*****
TST435: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #435,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #1,R4 ;RESULT S / B = 177777
MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
MOV #+1,(R5) ;[SOURCE] = 000001
CLR (R2) ;[DEST] = 000000
MOV R2,R3 ;R3 GETS DEST ADDR
CCC ;CLEAR FLAGS
266 ;N:C = 0110
2$: SUB (R5),(R3)+ ;TEST THE SUB
BPL 3$ ;N:C = 1001 ?
BEQ 3$
BVS 3$
BCS 4$
3$: ERROR 5 ;SUB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ T5T436 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;SUB DELIVERED THE WRONG RESULT
;*****
;TEST 436 NEG DM2 TEST
;*****
TST436: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #436,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,R4 ;RESULT S / B = 125252
MOV R2,R5 ;[RS] = DEST ADDR
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC
2$: NEG (R5)+ ;TEST THE NEG - MODE 2
CMP R4,(R2) ;RESULT = 125252?

```

8753 030470 001402
8754
8755 030472 011203
8756 030474 104001
8758 030476 022705 063314
8759 030502 001401
8760
8761 030504 104005
8762
8763
8764
8765
8766 030506
8767 030506 000004
8768 030510 012700 000437
8769 030514 013701 030542
8770 030520 012701 063312
8771 030524 012704 125252
8772 030530 012705 063306
8773 030534 012712 052526
8774 030540 000257
8775
8776 030542 005435
8777
8778 030544 020412
8779 030546 001402
8780
8781 030550 011203
8782 030552 104001
8783
8784 030554 022705 063310
8785 030560 001401
8786
8787 030562 104005
8788
8789
8790
8791
8792 030564
8793 030566 000004 000440
8794 030572 012700 030620
8795 030572 013701 030620
8796 030576 012702 063312
8797 030602 012704 125252
8798 030606 012705 063314
8799 030612 012712 052526
8800 030616 000257
8801
8802 030620 005445
8803
8804 030622 020412
8805 030624 001402
8806
8807 030626 011203
8808 030630 104001

BEQ 4\$;BR IF YES
3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;NEG DELIVERED WRONG RESULT
4\$: CMP #MBUF0+2,R5 ;DID REG. GET AUTO INCREMENTED?
BEQ TST437 ;;BR IF YES
5\$: ERROR 5 ;NEG FAILED TO UPDATE REG.
;*****
;TEST 437 NEG DM3 TEST
;*****
TST437: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #437,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,R4 ;RESULT S / B = 125252
MOV #ATA+10,R5 ;[ATA+10] = MBUF0
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC
2\$: NEG @(R5)+ ;TEST THE NEG - MODE 3
CMP R4,(R2) ;RESULT = 125252?
BEQ 4\$;BR IF YES
3\$: MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;NEG DELIVERED WRONG RESULT
4\$: CMP #ATA+12,R5 ;DID REG GET AUTO INCREMENTED?
BEQ TST440 ;;BR IF YES
5\$: ERROR 5 ;NEG FAILED TO UPDATE REG.
;*****
;TEST 440 NEG DM4 TEST
;*****
TST440: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #440,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,R4 ;RESULT S / B = 125252
MOV #MBUF0+2,R5 ;[R5] = DEST ADDR + 2
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC
2\$: NEG -(R5) ;TEST THE NEG - MODE 4
CMP R4,(R2) ;RESULT = 125252?
BEQ 4\$;BR IF YES
3\$: MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;NEG DELIVERED WRONG RESULT

8809
8810 030632 020502
8811 030634 001401
8812
8813 030636 104005

4\$: CMP R5,R2 ;DID REG GET AUTO INCREMENTED?
BEQ TST441 ;;BR IF YES
5\$: ERROR 5 ;NEG FAILED TO UPDATE REG


```

8814
8815
8816
8817
8818 030640
8819 030640 000004
8820 030642 012700 000441
8821 030646 012701 030674
8822 030652 012702 063312
8823 030656 012704 125252
8824 030662 012705 063310
8825 030666 012712 052526
8826 030672 000257
8827
8828 030674 005455
8829
8830 030676 020412
8831 030700 001402
8832
8833 030702 011203
8834 030704 104001
8835
8836 030706 022705 063306
8837 030712 001401
8838
8839 030714 104005
8840
8841
8842
8843
8844 030716
8845 030716 000004
8846 030720 012700 000442
8847 030724 013701 030752
8848 030730 012703 063312
8849 030734 012704 125252
8850 030740 012705 063310
8851 030744 012712 052526
8852 030750 000257
8853
8854 030752 005465 000002
8855
8856 030756 020412
8857 030760 001402
8858
8859 030762 011203
8860 030764 104001
8861
8862
8863
8864
8865 030766
8866 030766 000004
8867 030770 012700 000443
8868 030774 013701 031022
8869 031000 012702 063312

```

```

;*****
;TEST 441 NEG DM5 TEST
;*****
TST441:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #441,R0 ;LOAD R0 WITH TEST NUMBER
MOV #22,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,R4 ;RESULT S / B = 125252
MOV #ATA+12,R5 ;[R5] = (ADR OF MBUF0) +2
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC

2$: NEG @-(R5) ;TEST THE NEG - MODE 5

CMP R4,(R2) ;RESULT = 125252?
BEQ 4$ ;BR IF YES

MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;NEG DELIVERED WRONG RESULT

3$:

4$: CMP #ATA+10,R5 ;DID NEG UPDATE REG
ERROR TST442 ;BR IF YES

5$: ERROR 5 ;NEG FAILED TO UPDATE REG

;*****
;TEST 442 NEG DM6 TEST
;*****
TST442:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #442,R0 ;LOAD R0 WITH TEST NUMBER
MOV #22,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,R4 ;RESULT S / B = 125252
MOV #MBUF0,R5 ;[R5] = BASE ADDR
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC

2$: NEG 2(R5) ;TEST THE NEG - MODE 6

CMP R4,(R2) ;RESULT = 125252?
BEQ TST443 ;BR IF YES

MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;NEG DELIVERED WRONG RESULT

3$:

;*****
;TEST 443 NEG DM7 TEST
;*****
TST443:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #443,R0 ;LOAD R0 WITH TEST NUMBER
MOV #22,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0

```

```

8870 031004 012704 125252
8871 031010 012705 063276
8872 031014 012712 052526
8873 031020 000257
8874
8875 031022 005475 000010
8876
8877 031026 020412
8878 031030 001402
8879
8880 031032 011203
8881 031034 104001
8882
8883
8884
8885
8886 031036
8887 031036 000004
8888 031040 012700 000444
8889 031044 013701 031074
8890 031050 005004
8891 031052 005104
8892 031054 012702 063312
8893 031060 012705 063324
8894 031064 010203
8895 031066 005012
8896 031070 000257
8897 031072 000264
8898
8899 031074 011513
8900
8901 031076 100003
8902 031100 001402
8903 031102 102401
8904 031104 103001
8905
8906 031106 104001
8907
8908 031110 020412
8909 031112 001403
8910
8911 031114 005003
8912 031116 051203
8913 031120 104001
8914
8915
8916
8917
8918 031122
8919 031122 000004
8920 031124 012700 000445
8921 031130 013701 031160
8922 031134 005004
8923 031136 005104
8924 031140 012702 063312
8925 031144 012705 063324

```

```

MOV #125252,R4 ;RESULT S / B = 125252
MOV #ATA,R5 ;[R5] = BASE ADDR
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC

2$: NEG @10(R5) ;TEST THE NEG - MODE 7

CMP R4,(R2) ;RESULT = 125252?
BEQ TST444 ;BR IF YES

MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;NEG DELIVERED WRONG RESULT

3$:

;*****
;TEST 444 MOV SM1,DM1 TEST - N:C = 0100
;*****
TST444:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #444,R0 ;LOAD R0 WITH TEST NUMBER
MOV #22,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 17777
COM R4 ;DEST ADDR = MBUF0
MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
MOV R2,R3 ;BASE DEST ADDR = MBUF0
CLR (R2) ;MAKE [DEST] = 000000
CCC ;CLEAR FLAGS
264 ;N:C = 0100

2$: MOV (R5),(R3) ;TEST THE MOV - SM1,DM1

BPL 3$ ;N:C = 1000 ?
BEQ 3$
BVS 3$
BCC 4$

3$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ TST445 ;BR IF YES

CLR R3 ;GET THE WAS DATA
BIS (R2),R3 ;MOV DELIVERED THE WRONG RESULT
ERROR 1

;*****
;TEST 445 MOV SM2,DM1 TEST - N:C = 0100
;*****
TST445:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #445,R0 ;LOAD R0 WITH TEST NUMBER
MOV #22,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 17777
COM R4 ;DEST ADDR = MBUF0
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2

```

```

8926 031150 010203      MOV     R2,R3      ;BASE DEST ADDR = MBUFO
8927 031152 005012      CLR     (R2)       ;MAKE [DEST] = 000000
8928 031154 000257      CCC     264        ;CLEAR FLAGS
8929 031156 000264      264              ;N:C = 0100
8930
8931 031160 012513      2$:  MOV     (R5)+,(R3) ;TEST THE MOV - SM2,DM1
8932
8933 031162 100003      BPL     3$         ;N:C = 1000 ?
8934 031164 001402      BEQ     3$
8935 031166 102401      BVS     4$
8936 031170 103001      BCC     4$
8937
8938 031172 104001      3$:  ERROR   1      ;MOV FAILED TO ALTER CODES PROPERLY
8939
8940 031174 020412      4$:  CMP     R4,(R2) ;RESULT CORRECT ??
8941 031176 001403      BEQ     T$446     ;BR IF YES
8942
8943 031200 005003      CLR     R3        ;GET THE WAS DATA
8944 031202 051203      BIS     (R2),R3
8945 031204 104001      5$:  ERROR   1      ;MOV DELIVERED THE WRONG RESULT
8946
8947 ;*****
8948 ;*TEST 446 MOV SM1,DM1 TEST - N:C = 1011
8949 ;*****
8950 T$446:
8951 SCOPE
8952 MOV     #446,R0    ;CALL THE SCOPE LOOP UTILITY
8953 MOV     @#25,R1    ;LOAD R0 WITH TEST NUMBER
8954 CLR     R4         ;LOAD R1 WITH TEST INSTRUCTION WORD
8955 MOV     #MBUFO,R2 ;RESULT S / B = 000000
8956 MOV     #DMTA,R5  ;DEST ADDR = MBUFO
8957 MOV     R2,R3     ;SOURCE ADDR = DMTA
8958 CLR     (R2)      ;BASE DEST ADDR = MBUFO
8959 COM     (R2)      ;MAKE [DEST] = 177777
8960 CCC     273       ;CLEAR FLAGS
8961 273           ;N:C = 1011
8962
8963 031244 011513      2$:  MOV     (R5),(R3) ;TEST THE MOV - SM1,DM1
8964
8965 031246 100403      BMI     3$         ;N:C = 0101 ?
8966 031250 001002      BNE     3$
8967 031252 102401      BVS     3$
8968 031254 103401      BCS     4$
8969
8970 031256 104001      3$:  ERROR   1      ;MOV FAILED TO ALTER CODES PROPERLY
8971
8972 031260 020412      4$:  CMP     R4,(R2) ;RESULT CORRECT ??
8973 031262 001403      BEQ     T$447     ;BR IF YES
8974
8975 031264 005003      CLR     R3        ;GET THE WAS DATA
8976 031266 051203      BIS     (R2),R3
8977 031270 104001      5$:  ERROR   1      ;MOV DELIVERED THE WRONG RESULT
8978
8979 ;*****
8980 ;*TEST 447 MOV SM2,DM1 TEST - N:C = 1011
8981 ;*****

```

```

8982 031272 000004      MOV     #447,R0    ;CALL THE SCOPE LOOP UTILITY
8983 031274 012700 000447    MOV     @#25,R1    ;LOAD R0 WITH TEST NUMBER
8984 031276 013701 031330    MOV     R4         ;LOAD R1 WITH TEST INSTRUCTION WORD
8985 031300 005004      CLR     R4         ;RESULT S / B = 000000
8986 031304 005004 063312    MOV     #MBUFO,R2 ;DEST ADDR = MBUFO
8987 031306 012705 063322    MOV     #DMTA,R5  ;SOURCE ADDR = DMTA
8988 031310 010203      MOV     R2,R3     ;BASE DEST ADDR = MBUFO
8989 031312 005012      CLR     (R2)      ;MAKE [DEST] = 177777
8990 031314 005012      COM     (R2)
8991 031316 000257      CCC     273       ;CLEAR FLAGS
8992 031318 000273      273           ;N:C = 1011
8993
8994 031330 012513      2$:  MOV     (R5)+,(R3) ;TEST THE MOV - SM2,DM1
8995
8996 031332 100403      BMI     3$         ;N:C = 0101 ?
8997 031334 001002      BNE     3$
8998 031336 102401      BVS     3$
8999 031340 103401      BCS     4$
9000
9001 031342 104001      3$:  ERROR   1      ;MOV FAILED TO ALTER CODES PROPERLY
9002
9003 031344 020412      4$:  CMP     R4,(R2) ;RESULT CORRECT ??
9004 031346 001403      BEQ     T$450     ;BR IF YES
9005
9006 031350 005003      CLR     R3        ;GET THE WAS DATA
9007 031352 051203      BIS     (R2),R3
9008 031354 104001      5$:  ERROR   1      ;MOV DELIVERED THE WRONG RESULT
9009
9010 ;*****
9011 ;*TEST 450 MOV SM1,DM2 TEST - N:C = 0100
9012 ;*****
9013 T$450:
9014 SCOPE
9015 MOV     #450,R0    ;CALL THE SCOPE LOOP UTILITY
9016 MOV     @#25,R1    ;LOAD R0 WITH TEST NUMBER
9017 CLR     R4         ;LOAD R1 WITH TEST INSTRUCTION WORD
9018 MOV     R4         ;RESULT S / B = 177777
9019 COM     R4
9020 MOV     #MBUFO,R2 ;DEST ADDR = MBUFO
9021 MOV     #DMTA+2,R5 ;SOURCE ADDR = DMTA
9022 MOV     R2,R3     ;BASE DEST ADDR = MBUFO
9023 CLR     (R2)      ;MAKE [DEST] = 000000
9024 CCC     264       ;CLEAR FLAGS
9025 264           ;N:C = 0100
9026
9027 031414 011523      2$:  MOV     (R5),(R3)+ ;TEST THE MOV - SM1,DM2
9028
9029 031416 100003      BPL     3$         ;N:C = 1000 ?
9030 031420 001402      BEQ     3$
9031 031422 102401      BVS     3$
9032 031424 103001      BCC     4$
9033
9034 031426 104001      3$:  ERROR   1      ;MOV FAILED TO ALTER CODES PROPERLY
9035
9036 031430 022703 063314    4$:  CMP     #MBUFO+2,R3 ;DID MOV INCREMENT DEST REG ?
9037 031434 001401      BEQ     6$        ;BR IF YES

```

9038			
9039	031436	104005	
9040			
9041	031440	020412	
9042	031442	001403	
9043			
9044	031444	005003	
9045	031446	051203	
9046	031450	104001	
9047			
9048			
9049			
9050			
9051	031452		
9052	031452	000004	
9053	031454	012700	000451
9054	031460	013701	031510
9055	031464	005004	
9056	031466	005104	
9057	031470	012702	063312
9058	031474	012705	063324
9059	031500	010203	
9060	031502	005012	
9061	031504	000257	
9062	031506	000264	
9063			
9064	031510	012523	
9065			
9066	031512	100003	
9067	031514	001402	
9068	031516	102401	
9069	031520	103001	
9070			
9071	031522	104001	
9072			
9073	031524	022703	063314
9074	031530	001401	
9075			
9076	031532	104005	
9077			
9078	031534	020412	
9079	031536	001403	
9080			
9081	031540	005003	
9082	031542	051203	
9083	031544	104001	
9084			
9085			
9086			
9087			
9088	031546		
9089	031546	000004	
9090	031546	012700	000452
9091	031554	013701	031606
9092	031560	005004	
9093	031562	005104	

```

5$: ERROR 5 ;MOV FAILED TO UPDATE DEST REG
6$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ TST451 ;BR IF YES
CLR R4 ;
COM R4 ;
7$: CLR R3 ;GET THE WAS DATA
BIS (R2),R3 ;
ERROR 1 ;MOV DELIVERED THE WRONG RESULT
;*****
;T451: MOV SM2,DM2 TEST - N:C = 0100
;*****
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #451,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4 ;
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #DMTA+2,R5 ;SOURCE ADDR = DMTA+2
MOV R2,R3 ;BASE DEST ADDR = MBUF0
CLR (R2) ;MAKE [DEST] = 000000
CCC ;CLEAR FLAGS
264 ;N:C = 0100
2$: MOV (R5)+,(R3)+ ;TEST THE MOV - SM2,DM2
RPL 3$ ;
BEQ 3$ ;N:C = 1000 ?
BVS 3$ ;
BCC 4$ ;
3$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY
4$: CMP #MBUF0+2,R3 ;DID MOV INCREMENT DEST REG ?
BEQ 6$ ;BR IF YES
5$: ERROR 5 ;MOV FAILED TO UPDATE DEST REG
6$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ TST452 ;BR IF YES
CLR R3 ;
BIS (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT
;*****
;T452: MOV SM1,DM3 TEST - N:C = 0100
;*****
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #452,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4 ;

```

9094	031564	012702	063312
9095	031570	012706	063324
9096	031574	022703	063306
9097	031600	005012	
9098	031602	000257	
9099	031604	000264	
9100			
9101	031606	011533	
9102			
9103	031610	100003	
9104	031612	001402	
9105	031614	102401	
9106	031616	103001	
9107			
9108	031620	104001	
9109			
9110	031622	022703	063310
9111	031626	001401	
9112			
9113	031630	104005	
9114			
9115	031632	020412	
9116	031634	001403	
9117			
9118	031636	005003	
9119	031640	051203	
9120	031642	104001	
9121			
9122			
9123			
9124			
9125			
9126	031644	000004	
9127	031644	012700	000453
9128	031652	013701	031704
9129	031656	005004	
9130	031660	005104	
9131	031662	012705	063312
9132	031666	012705	063324
9133	031672	012703	063306
9134	031676	005012	
9135	031700	000257	
9136	031702	000264	
9137			
9138	031704	012533	
9139			
9140	031706	100003	
9141	031710	001402	
9142	031712	102401	
9143	031714	103001	
9144			
9145	031716	104001	
9146			
9147	031720	022703	063310
9148	031724	001401	
9149			

```

MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #DMTA+2,R5 ;SOURCE ADDR = DMTA+2
MOV #ATA+10,R3 ;BASE DEST ADDR = ATA+10
CLR (R2) ;MAKE [DEST] = 000000
CCC ;CLEAR FLAGS
264 ;N:C = 0100
2$: MOV (R5)+,(R3)+ ;TEST THE MOV - SM1,DM3
RPL 3$ ;
BEQ 3$ ;N:C = 1000 ?
BVS 3$ ;
BCC 4$ ;
3$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY
4$: CMP #ATA+12,R3 ;DID MOV INCREMENT DEST REG ?
BEQ 6$ ;BR IF YES
5$: ERROR 5 ;MOV FAILED TO UPDATE DEST REG
6$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ TST453 ;BR IF YES
CLR R3 ;
BIS (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT
;*****
;T453: MOV SM2,DM3 TEST - N:C = 0100
;*****
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #453,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4 ;
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #DMTA+2,R5 ;SOURCE ADDR = DMTA+2
MOV #ATA+10,R3 ;BASE DEST ADDR = ATA+10
CLR (R2) ;MAKE [DEST] = 000000
CCC ;CLEAR FLAGS
264 ;N:C = 0100
2$: MOV (R5)+,(R3)+ ;TEST THE MOV - SM2,DM3
RPL 3$ ;
BEQ 3$ ;N:C = 1000 ?
BVS 3$ ;
BCC 4$ ;
3$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY
4$: CMP #ATA+12,R3 ;DID MOV INCREMENT DEST REG ?
BEQ 6$ ;BR IF YES

```

9150 031726 104005
 9151 031730 020412
 9152 031732 001403
 9153 031734 005003
 9154 031736 051203
 9155 031740 104001
 9156
 9157
 9158
 9159
 9160
 9161
 9162 031742 000004
 9163 031744 012700 000454
 9164 031746 013701 032002
 9165 031750 005004
 9166 031754 005104
 9167 031756 005104
 9168 031760 012702 063312
 9169 031764 012705 063314
 9170 031770 012703
 9171 031774 005012
 9172 031776 000257
 9173 032000 000264
 9174
 9175
 9176 032002 011543
 9177 032004 100003
 9178 032006 001402
 9179 032010 102401
 9180 032012 103001
 9181
 9182 032014 104001
 9183
 9184 032016 020203
 9185 032020 001401
 9186
 9187 032022 104005
 9188
 9189 032024 020412
 9190 032026 001403
 9191
 9192 032030 005003
 9193 032032 051203
 9194 032034 104001
 9195
 9196
 9197
 9198
 9199 032036 000004
 9200 032038 012700 000455
 9201 032040 013701 032076
 9202 032044 005004
 9203 032050 005104
 9204 032054 012702 063312
 9205

5\$: ERROR 5 ;MOV FAILED TO UPDATE DEST REG
 6\$: CMP R4,(R2) ;RESULT CORRECT ??
 BEQ T3T454 ;BR IF YES
 CLR R3 ;GET THE WAS DATA
 7\$: BIS (R2),R3
 ERROR 1 ;MOV DELIVERED THE WRONG RESULT
 ;*****
 ;*TEST 454 MOV SM1,DM4 TEST - N:C = 0100
 ;*****
 TST454: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #454,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 CLR R4 ;RESULT S / B = 17777
 COM R4
 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
 MOV #DMTA+2,R5 ;SOURCE ADDR = DMTA+2
 MOV #MBUF0+2,R3 ;BASE DEST ADDR = MBUF0+2
 CLR (R2) ;MAKE [DEST] = 000000
 CCC ;CLEAR FLAGS
 264 ;N:C = 0100
 2\$: MOV (R5),-(R3) ;TEST THE MOV - SM1,DM4
 BPL 3\$;N:C = 1000 ?
 BEQ 3\$
 BVS 3\$
 BCC 4\$
 3\$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY
 4\$: CMP R2,R3 ;DID MOV DECREMENT DEST REG ?
 BEQ 6\$;BR IF YES
 5\$: ERROR 5 ;MOV FAILED TO UPDATE DEST REG
 6\$: CMP R4,(R2) ;RESULT CORRECT ??
 BEQ T3T455 ;BR IF YES
 CLR R3 ;GET THE WAS DATA
 7\$: BIS (R2),R3
 ERROR 1 ;MOV DELIVERED THE WRONG RESULT
 ;*****
 ;*TEST 455 MOV SM2,DM4 TEST - N:C = 0100
 ;*****
 TST455: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #455,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 CLR R4 ;RESULT S / B = 17777
 COM R4
 MOV #MBUF0,R2 ;DEST ADDR = MBUF0

9206 032060 012705 063324
 9207 032064 012703 063314
 9208 032070 005012
 9209 032074 000257
 9210 032076 000264
 9211
 9212 032076 012543
 9213
 9214 032100 100003
 9215 032104 001402
 9216 032108 102401
 9217 032106 103001
 9218
 9219 032110 104001
 9220
 9221 032112 020203
 9222 032114 001401
 9223
 9224 032116 104005
 9225
 9226 032120 020412
 9227 032122 001403
 9228
 9229 032124 005003
 9230 032126 051203
 9231 032130 104001
 9232
 9233
 9234
 9235
 9236 032132 000004
 9237 032134 012700 000456
 9238 032136 013701 032204
 9239 032140 005004
 9240
 9241 032144 032737 001000 063234
 9242 032152 001401
 9243 032154 000000
 9244 032156 005004
 9245 032160 005104
 9246 032162 012702 063312
 9247 032166 012705 063324
 9248 032172 012703 063310
 9249 032176 005012
 9250 032200 000257
 9251 032202 000264
 9252
 9253 032204 011553
 9254
 9255 032206 100003
 9256 032210 001402
 9257 032214 102401
 9258 032216 103001
 9259
 9260 032216 104001
 9261

MOV #DMTA+2,R5 ;SOURCE ADDR = DMTA+2
 MOV #MBUF0+2,R3 ;BASE DEST ADDR = MBUF0+2
 CLR (R2) ;MAKE [DEST] = 000000
 CCC ;CLEAR FLAGS
 264 ;N:C = 0100
 2\$: MOV (R5)+,-(R3) ;TEST THE MOV - SM2,DM4
 BPL 3\$;N:C = 1000 ?
 BEQ 3\$
 BVS 3\$
 BCC 4\$
 3\$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY
 4\$: CMP R2,R3 ;DID MOV INCREMENT DEST REG ?
 BEQ 6\$;BR IF YES
 5\$: ERROR 5 ;MOV FAILED TO UPDATE DEST REG
 6\$: CMP R4,(R2) ;RESULT CORRECT ??
 BEQ T3T456 ;BR IF YES
 CLR R3 ;GET THE WAS DATA
 7\$: BIS (R2),R3
 ERROR 1 ;MOV DELIVERED THE WRONG RESULT
 ;*****
 ;*TEST 456 MOV SM1,DM5 TEST - N:C = 0100
 ;*****
 TST456: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #456,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 .SBTTL USER CONTROLLED BREAKPOINT -- BIT9
 BIT #BIT9,@#BPTLOC ;BREAKPOINT HALT SET ??
 BEQ +4 ;BR IF NOT
 HALT ;BREAK - DEPRESS CONTINUE TO RESTART
 CLR R4 ;RESULT S / B = 17777
 COM R4
 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
 MOV #DMTA+2,R5 ;SOURCE ADDR = DMTA+2
 MOV #ATA+12,R3 ;BASE DEST ADDR = ATA+12
 CLR (R2) ;MAKE [DEST] = 000000
 CCC ;CLEAR FLAGS
 264 ;N:C = 0100
 2\$: MOV (R5),@(R3) ;TEST THE MOV - SM1,DM5
 BPL 3\$;N:C = 0100 ?
 BEQ 3\$
 BVS 3\$
 BCC 4\$
 3\$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY

9262	032220	022703	063306
9263	032224	001401	
9264			
9265	032226	104005	
9266			
9267	032230	020412	
9268	032232	001403	
9269			
9270	032234	005003	
9271	032236	051203	
9272	032240	104001	
9273			
9274			
9275			
9276			
9277	032242		
9278	032242	000004	
9279	032244	012700	000457
9280	032250	013701	032302
9281	032254	005004	
9282	032256	005104	
9283	032260	012702	063312
9284	032264	012705	063324
9285	032270	012703	063310
9286	032274	005012	
9287	032276	000257	
9288	032300	000264	
9289			
9290	032302	012553	
9291			
9292	032304	100003	
9293	032306	001402	
9294	032310	102401	
9295	032312	103001	
9296			
9297	032314	104001	
9298			
9299	032316	022703	063306
9300	032322	001401	
9301			
9302	032324	104005	
9303			
9304	032326	020412	
9305	032330	001403	
9306			
9307	032332	005003	
9308	032334	051203	
9309	032336	104001	
9310			
9311			
9312			
9313			
9314			
9315	032340	000004	
9316	032342	012700	000460
9317	032346	013701	032400

```

4$:   CMP   #ATA+10,R3      ;DID MOV DECREMENT DEST REG ?
      BEQ   6$              ;BR IF YES

5$:   ERROR 5              ;MOV FAILED TO UPDATE DEST REG

6$:   CMP   R4,(R2)         ;RESULT CORRECT ??
      BEQ   TS457           ;BR IF YES

      CLR   R3              ;GET THE WAS DATA
      BIS   (R2),R3        ;MOV DELIVERED THE WRONG RESULT

7$:   ERROR 1              ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 457   MOV SM2,DM5 TEST - N:C = 0100
;*****
TS457: SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV   #457,R0         ;LOAD R0 WITH TEST NUMBER
      CLR   R4#25,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV   R4              ;RESULT S / B = 177777
      COM   R4
      MOV   #MBUF0,R2      ;DEST ADDR = MBUF0
      MOV   #DMTA+2,R5     ;SOURCE ADDR = DMTA+2
      MOV   #ATA+12,R3     ;BASE DEST ADDR = ATA+12
      CLR   (R2)           ;MAKE DESTJ = 00000
      CCC   264            ;CLEAR FLAGS
      BCC   264            ;N:C = 1000

2$:   MOV   (R5)+,@-(R3)   ;TEST THE MOV - SM2,DM5
      BPL   3$             ;N:C = 1000 ?
      BEQ   3$
      BVS   3$
      BCC   4$

3$:   ERROR 1              ;MOV FAILED TO ALTER CODES PROPERLY

4$:   CMP   #ATA+10,R3      ;DID MOV DECREMENT DEST REG ?
      BEQ   6$              ;BR IF YES

5$:   ERROR 5              ;MOV FAILED TO UPDATE DEST REG

6$:   CMP   R4,(R2)         ;RESULT CORRECT ??
      BEQ   TS460           ;BR IF YES

      CLR   R3              ;GET THE WAS DATA
      BIS   (R2),R3        ;MOV DELIVERED THE WRONG RESULT

7$:   ERROR 1              ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 460   MOV SM1,DM6 TEST - N:C = 0100
;*****
TS460: SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV   #460,R0         ;LOAD R0 WITH TEST NUMBER
      MOV   @#25,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD

```

9318	032352	005004	
9319	032354	005104	
9320	032356	012702	063320
9321	032360	013705	063324
9322	032362	013703	063312
9323	032372	005012	
9324	032374	000257	
9325	032376	000264	
9326			
9327	032400	011563	000006
9328			
9329	032404	100003	
9330	032406	001402	
9331	032410	102401	
9332	032412	103001	
9333			
9334	032414	104001	
9335			
9336	032416	020412	
9337	032420	001403	
9338			
9339	032422	005003	
9340	032424	051203	
9341	032426	104001	
9342			
9343			
9344			
9345			
9346	032430		
9347	032430	000004	
9348	032432	012700	000461
9349	032436	013701	032470
9350	032442	005004	
9351	032444	005104	
9352	032446	012702	063320
9353	032452	012705	063324
9354	032456	012703	063312
9355	032462	005012	
9356	032464	000257	
9357	032466	000264	
9358			
9359	032470	012563	000006
9360			
9361	032474	100003	
9362	032476	001402	
9363	032500	102401	
9364	032502	103001	
9365			
9366	032504	104001	
9367			
9368	032506	020412	
9369	032510	001403	
9370			
9371	032512	005003	
9372	032514	051203	
9373	032516	104001	

```

      CLR   R4              ;RESULT S / B = 177777
      COM   R4
      MOV   #MBUF0+6,R2   ;DEST ADDR = MBUF0+6
      MOV   #DMTA+2,R5   ;SOURCE ADDR = DMTA+2
      MOV   #MBUF0,R3    ;BASE DEST ADDR = MBUF0
      CLR   (R2)         ;MAKE DESTJ = 000000
      CCC   264            ;CLEAR FLAGS
      BCC   264            ;N:C = 0100

2$:   MOV   (R5),6(R3)    ;TEST THE MOV - SM1,DM6
      BPL   3$             ;N:C = 1000 ?
      BEQ   3$
      BVS   3$
      BCC   4$

3$:   ERROR 1              ;MOV FAILED TO ALTER CODES PROPERLY

4$:   CMP   R4,(R2)         ;RESULT CORRECT ??
      BEQ   TS461           ;BR IF YES

      CLR   R3              ;GET THE WAS DATA
      BIS   (R2),R3        ;MOV DELIVERED THE WRONG RESULT

5$:   ERROR 1              ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 461   MOV SM2,DM6 TEST - N:C = 0100
;*****
TS461: SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV   #461,R0         ;LOAD R0 WITH TEST NUMBER
      CLR   R4#25,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV   R4              ;RESULT S / B = 177777
      COM   R4
      MOV   #MBUF0+6,R2   ;DEST ADDR = MBUF0+6
      MOV   #DMTA+2,R5   ;SOURCE ADDR = DMTA+2
      MOV   #MBUF0,R3    ;BASE DEST ADDR = MBUF0
      CLR   (R2)         ;MAKE DESTJ = 000000
      CCC   264            ;CLEAR FLAGS
      BCC   264            ;N:C = 0100

2$:   MOV   (R5)+,6(R3)   ;TEST THE MOV - SM2,DM6
      BPL   3$             ;N:C = 1000 ?
      BEQ   3$
      BVS   3$
      BCC   4$

3$:   ERROR 1              ;MOV FAILED TO ALTER CODES PROPERLY

4$:   CMP   R4,(R2)         ;RESULT CORRECT ??
      BEQ   TS462           ;BR IF YES

      CLR   R3              ;GET THE WAS DATA
      BIS   (R2),R3        ;MOV DELIVERED THE WRONG RESULT

5$:   ERROR 1              ;MOV DELIVERED THE WRONG RESULT

```

```

9374
9375
9376
9377
9378 032520
9379 032520 000004
9380 032522 012700 000462
9381 032526 013701 032560
9382 032532 005004
9383 032536 015404
9384 032542 015705 063312
9385 032546 012703 063324
9386 032552 005012 063276
9387 032554 000257
9388 032556 000264
9389
9390 032560 011573 000010
9391
9392 032564 100003
9393 032566 001402
9394 032570 102401
9395 032572 103001
9396
9397 032574 104001
9398
9399 032576 020412
9400 032600 001403
9401
9402 032602 005003
9403 032604 051203
9404 032606 104001
9405
9406
9407
9408
9409
9410 032610
9411 032610 000004
9412 032612 012700 000463
9413 032616 013701 032650
9414 032622 005004
9415 032624 005104
9416 032626 012702 063312
9417 032632 012705 063324
9418 032636 005012 063276
9419 032642 005012
9420 032644 000257
9421 032646 000264
9422
9423 032650 011573 000010
9424
9425 032654 100003
9426 032656 001402
9427 032660 102401
9428 032662 103001
9429

```

```

;*****
;TEST 462 MOV SM1,DM7 TEST - N:C = 0100
;*****
;T462:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #462,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #DMTA+2,R5 ;SOURCE ADDR = DMTA+2
MOV #ATA,R3 ;BASE DEST ADDR = ATA
CLR (R2) ;MAKE [DEST] = 000000
CCC ;CLEAR FLAGS
264 ;SCOPE SYNC
;N:C = 0100

2$: MOV (R5),@10(R3) ;TEST THE MOV - SM1,DM7
BPL 3$ ;N:C = 1000 ?
BEQ 4$
BVS 3$
BCC 4$

3$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ T$463 ;BR IF YES

CLR R3 ;GET THE WAS DATA
BIS (R2),R3
5$: ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 463 MOV SM2,DM7 TEST - N:C = 0100
;*****
;T463:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #463,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #DMTA+2,R5 ;SOURCE ADDR = DMTA+2
MOV #ATA,R3 ;BASE DEST ADDR = ATA
CLR (R2) ;MAKE [DEST] = 000000
CCC ;CLEAR FLAGS
264 ;SCOPE SYNC
;N:C = 0100

2$: MOV (R5),@10(R3) ;TEST THE MOV - SM2,DM7
BPL 3$ ;N:C = 1000 ?
BEQ 4$
BVS 3$
BCC 4$

3$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ T$464 ;BR IF YES

CLR R3 ;GET THE WAS DATA
BIS (R2),R3
5$: ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 464 MOV SM0,DM1 TEST
;*****
;T464:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #464,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = TEST NUMBER
COM R4 ;R5 GETS DEST ADDR
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV R0,R4 ;RESULT S / B = TEST NUMBER
MOV R2,R5 ;R5 GETS DEST ADDR
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV R0,(R5) ;TEST THE MOV

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 465 MOV SM0,DM2 TEST
;*****
;T465:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #465,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;DEST ADDR = MBUF0
COM R4 ;RESULT S / B = TEST NUMBER
MOV #MBUF0,R2 ;R5 GETS DEST ADDR
MOV R0,R4 ;RESULT S / B = TEST NUMBER
MOV R2,R5 ;R5 GETS DEST ADDR
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV R0,(R5)+ ;TEST THE MOV

3$: CMP R4,(R2) ;RESULT CORRECT ?
BEQ T$466 ;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 466 MOV SM0,DM3 TEST
;*****
;T466:
SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

9430 032664 104001
9431
9432 032666 020412
9433 032670 001403
9434
9435 032672 005003
9436 032674 051203
9437 032676 104001
9438
9439
9440
9441
9442 032700
9443 032700 000004
9444 032702 012700 000464
9445 032706 013701 032726
9446 032712 012702 063312
9447 032716 010004
9448 032720 010205
9449 032722 005012
9450 032724 000257
9451
9452 032726 010015
9453
9454 032730 020412
9455 032732 001402
9456
9457 032734 011203
9458 032736 104001
9459
9460
9461
9462
9463 032740
9464 032740 000004
9465 032742 012700 000465
9466 032746 013701 032766
9467 032752 012703 063312
9468 032756 010004
9469 032760 010205
9470 032762 005012
9471 032764 000257
9472
9473 032766 010025
9474
9475 032770 020412
9476 032772 001402
9477
9478 032774 011203
9479 032776 104001
9480
9481
9482
9483
9484 033000
9485 033000 000004

```

```

3$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ T$464 ;BR IF YES

CLR R3 ;GET THE WAS DATA
BIS (R2),R3
5$: ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 464 MOV SM0,DM1 TEST
;*****
;T464:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #464,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = TEST NUMBER
COM R4 ;R5 GETS DEST ADDR
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV R0,R4 ;RESULT S / B = TEST NUMBER
MOV R2,R5 ;R5 GETS DEST ADDR
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV R0,(R5) ;TEST THE MOV

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 465 MOV SM0,DM2 TEST
;*****
;T465:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #465,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;DEST ADDR = MBUF0
COM R4 ;RESULT S / B = TEST NUMBER
MOV #MBUF0,R2 ;R5 GETS DEST ADDR
MOV R0,R4 ;RESULT S / B = TEST NUMBER
MOV R2,R5 ;R5 GETS DEST ADDR
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV R0,(R5)+ ;TEST THE MOV

3$: CMP R4,(R2) ;RESULT CORRECT ?
BEQ T$466 ;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 466 MOV SM0,DM3 TEST
;*****
;T466:
SCOPE ;CALL THE SCOPE LOOP UTILITY

```

9486 033002 012700 000466
 9487 033006 012701 033030
 9488 033016 012702 063312
 9489 033016 010004
 9490 033020 012705 063306
 9493 033024 005012
 9494 033026 000257
 9494 033030 010035
 9495 033032 020412
 9497 033034 001402
 9498 033036 011203
 9500 033040 104001
 9501
 9502
 9503
 9504
 9505 033042
 9506 033042 000004
 9507 033044 012700 000467
 9508 033050 013701 033072
 9509 033054 012702 063312
 9510 033060 010004
 9511 033062 012705 063314
 9512 033066 005012
 9513 033070 000257
 9514
 9515 033072 010045
 9516
 9517 033074 020412
 9518 033076 001402
 9519
 9520 033100 011203
 9521 033102 104001
 9522
 9523
 9524
 9525
 9526 033104
 9527 033106 000004
 9528 033106 012700 000470
 9529 033112 013701 033134
 9530 033116 012702 063312
 9531 033122 010004
 9532 033124 012705 063310
 9533 033130 005012
 9534 033132 000257
 9535
 9536 033134 010055
 9537
 9538 033136 020412
 9539 033140 001402
 9540
 9541 033142 011203

```

MOV #466,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV R0,R4 ;RESULT S / B = TEST NUMBER
MOV #ATA+10,R5 ;BASE DEST ADDR = ATA+10
CLR (R2) ;DESTJ = 000000
CCC ;SCOPE SYNC

2$: MOV R0,@(R5)+ ;TEST THE MOV
CMP R4,(R2) ;CORRECT RESULT
BEQ TS467 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 467 MOV SMO,DM4 TEST
;*****
TST467: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #467,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV R0,R4 ;RESULT S / B = TEST NUMBER
MOV #MBUF0+2,R5 ;R5 CONTAINS BASE DEST ADDR
CLR (R2) ;DESTJ = 000000
CCC ;SCOPE SYNC

2$: MOV R0,@-(R5) ;TEST THE MOV
CMP R4,(R2) ;CORRECT RESULT ?
BEQ TS470 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 470 MOV SMO,DM5 TEST
;*****
TST470: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #470,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV R0,R4 ;RESULT S / B = TEST NUMBER
MOV #ATA+12,R5 ;R5 CONTAINS BASE DEST ADDR
CLR (R2) ;DESTJ = 000000
CCC ;SCOPE SYNC

2$: MOV R0,@-(R5) ;TEST THE MOV
CMP R4,(R2) ;CORRECT RESULT ?
BEQ TS471 ;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA

```

9542 033144 104001
 9543
 9544
 9545
 9546
 9547
 9548 033146
 9549 033146 000004
 9550 033150 012700 000471
 9551 033160 012702 063316
 9552 033164 010004
 9553 033166 012705 063312
 9554 033172 005012
 9555 033174 000257
 9556
 9557 033176 010065 000004
 9558
 9559 033202 020412
 9560 033204 001402
 9561
 9562 033206 011203
 9563 033210 104001
 9564
 9565
 9566
 9567
 9568
 9569
 9570 033212
 9571 033212 000004
 9572 033214 012700 000472
 9573 033220 013701 033242
 9574 033224 012704 177652
 9575 033234 012705 000252
 9576 033236 005003
 9577 033240 000266
 9578
 9579 033242 110503
 9580
 9581 033244 100003
 9582 033246 001402
 9583 033250 102401
 9584 033252 103001
 9585
 9586 033254 104002
 9587
 9588 033256 020403
 9589 033260 001401
 9590
 9591 033262 104002
 9592
 9593
 9594
 9595
 9596 033264
 9597 033264 000004

```

3$: ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 471 MOV SMO,DM6 TEST
;*****
TST471: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #471,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF1,R2 ;DEST ADDR = MBUF1
MOV R0,R4 ;RESULT S / B = TEST NUMBER
MOV #MBUF0,R5 ;BASE DEST ADDR = MBUF0
CLR (R2) ;DESTJ = 000000
CCC ;SCOPE SYNC

2$: MOV R0,4(R5) ;TEST THE MOV
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TS472 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;TEST 472 MOV B TEST - SMO,DM0 - EXTEND 1'S
;*****
TST472: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #472,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177652,R4 ;RESULT S / B = 177652
MOV #252,R5 ;SOURCE OP = 252
CLR R3 ;DESTJ = 000000
CCC ;CLEAR FLAGS
266 ;N:C = 0110

2$: MOV B R5,R3 ;TEST THE MOV
BPL 3$ ;N:C = 1000 ?
BEQ 3$
BVS 3$
BCC 4$

3$: ERROR 2 ;MOV B FAILED TO ALTER CODES PROPERLY

4$: CMP R4,R3 ;RESULT CORRECT ?
BEQ TS473 ;BR IF YES

5$: ERROR 2 ;MOV B DELIVERED THE WRONG RESULT

;*****
;TEST 473 MOV B TEST - SMO,DM0 - EXTEND 0'S
;*****
TST473: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

9598 033266 012700 000473
9599 033272 013701 033314
9600 033276 005004
9601 033306 012705 177400
9602 033304 005003
9603 033306 005103
9604 033310 000257
9605 033312 000271
9606
9607 033314 110503
9608
9609 033316 100403
9610 033320 001002
9611 033322 102401
9612 033324 103401
9613
9614 033326 104002
9615
9616 033330 020403
9617 033332 001401
9618
9619 033334 104002
9620
9621
9622
9623
9624
9625 033336 000004
9626 033340 012700 000474
9627 033344 013701 033364
9628 033350 005004
9629 033352 012705 064630
9630 033356 005103
9631 033360 000257
9632 033362 000271
9633
9634 033364 111503
9635
9636 033366 020403
9637 033370 001401
9638
9639 033372 104002
9640
9641
9642
9643
9644 033374
9645 033374 000004
9646 033376 012700 000475
9647 033402 013701 033424
9648 033406 000125
9649 033412 012705 064633
9650 033416 012703 177400
9651 033422 000257
9652
9653 033424 111503

```

```

MOV #473,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #177400,R5 ;SOURCE DP = 177400
CLR R3 ;CDESTJ = 177777
COM
CCC
;CLEAR FLAGS
;N:C = 1001
2$: MOV B R5,R3 ;TEST THE MOV B
BMI 3$
BNE 3$ ;N:C = 0101 ?
BVS 3$
BCS 4$
3$: ERROR 2 ;MOV B FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;RESULT CORRECT ?
BEQ T$474 ;BR IF YES
5$: ERROR 2 ;MOV B DELIVERED THE WRONG RESULT
;*****
;TEST 474 MOV B TEST - SM1,DM0 - SOURCE ADDR EVEN
;*****
T$T474: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #474,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #DBTA,R5 ;SOURCE ADDR = DBTA
CLR R3 ;CDESTJ = 177777
COM
CCC ;SCOPE SYNC
2$: MOV B (R5),R3 ;TEST THE MOV B
CMP R4,R3 ;RESULT CORRECT ?
BEQ T$475 ;BR IF YES
3$: ERROR 2 ;MOV B DELIVERED THE WRONG RESULT
;*****
;TEST 475 MOV B TEST - SM1,DM0 - SOURCE ADDR ODD
;*****
T$T475: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #475,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 125
MOV #DBTA+3,R5 ;SOURCE ADDR = DBTA+3
MOV #177400,R3 ;CDESTJ = 177400
COM
CCC ;SCOPE SYNC
2$: MOV B (R5),R3 ;TEST THE MOV B

```

```

9654
9655 033426 020403
9656 033430 001401
9657
9658 033432 104002
9659
9660
9661
9662
9663 033434
9664 033434 000004
9665 033436 012700 000476
9666 033440 013701 033460
9667 033446 012704 177794
9668 033452 012705 064631
9669 033456 005003
9670 033460 000257
9671
9672 033462 112503
9673
9674 033464 020403
9675 033466 001401
9676
9677 033470 104002
9678
9679 033472 022705 064632
9680 033476 001401
9681
9682 033500 104005
9683
9684
9685
9686
9687 033502
9688 033503 000004
9689 033504 012700 000477
9690 033510 013701 033530
9691 033514 005004
9692 033516 012705 064630
9693 033522 012703 177400
9694 033526 000257
9695
9696 033530 112503
9697
9698 033532 020403
9699 033534 001401
9700
9701 033536 104002
9702
9703 033540 022705 064631
9704 033544 001401
9705
9706 033546 104005
9707
9708
9709

```

```

CMP R4,R3 ;RESULT CORRECT ?
BEQ T$476 ;BR IF YES
3$: ERROR 2 ;MOV B DELIVERED THE WRONG RESULT
;*****
;TEST 476 MOV B TEST - SM2,DM0 - SOURCE ADDR ODD
;*****
T$T476: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #476,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
CLR R3 ;CDESTJ = 000000
COM
CCC ;SCOPE SYNC
2$: MOV B (R5)+,R3 ;TEST THE MOV B
CMP R4,R3 ;RESULT CORRECT ?
BEQ 4$ ;BR IF YES
3$: ERROR 2 ;MOV B DELIVERED THE WRONG RESULT
4$: CMP #DBTA+2,R5 ;DID MOV B INCREMENT SRC REG ?
BEQ T$477 ;BR IF YES
5$: ERROR 5 ;MOV B FAILED TO UPDATE SRC REG
;*****
;TEST 477 MOV B TEST - SM2,DM0 - SOURCE ADDR EVEN
;*****
T$T477: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #477,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #DBTA,R5 ;SOURCE ADDR = DBTA
MOV #177400,R3 ;CDESTJ = 177400
COM
CCC ;SCOPE SYNC
2$: MOV B (R5)+,R3 ;TEST THE MOV B
CMP R4,R3 ;RESULT CORRECT ?
BEQ 4$ ;BR IF YES
3$: ERROR 2 ;MOV B DELIVERED THE WRONG RESULT
4$: CMP #DBTA+1,R5 ;DID MOV B INCREMENT SRC REG ?
BEQ T$500 ;BR IF YES
5$: ERROR 5 ;MOV B FAILED TO UPDATE SOURCE REG
;*****
;TEST 500 MOV B TEST - SM1,DM1 - SRC ADR ODD / DST ADR EVEN
;*****

```


9822 034064 000257
9823 034066 111553
9824 034070 022703 063306
9825 034074 001401
9826 034076 104005
9827 034100 020412
9828 034102 001402
9829 034104 011203
9830 034106 104001
9831
9832
9833
9834
9835
9836
9837
9838
9839
9840
9841 034110 000004
9842 034112 012700 000505
9843 034116 013701 034146
9844 034122 012702 063312
9845 034126 012704 000377
9846 034132 012705 064631
9847 034136 005012
9848 034140 012703 063320
9849 034144 000257
9850
9851
9852
9853 034146 111563 177772
9854 034152 020412
9855 034154 001402
9856
9857 034156 011203
9858 034160 104001
9859
9860
9861
9862 034162 000004
9863 034164 012700 000506
9864 034170 013701 034220
9865 034174 012702 063312
9866 034200 012704 000377
9867 034210 005012 064631
9868 034212 012703 063276
9869 034216 000257
9870
9871
9872 034220 111573 000010
9873
9874
9875 034224 020412
9876 034226 001402
9877

CCC ;CLEAR FLAGS - SCOPE SYNC
2\$: MOV B (R5),@-(R3) ;TEST THE MOV B
CMP #ATA+10,R3 ;DID MOV B DECREMENT DEST REG ?
BEQ 45 ;BR IF YES
3\$: ERROR 5 ;MOV B FAILED TO UPDATE DEST REG
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TS505 ;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED WRONG RESULT
;*****
;TEST 505 MOV B TEST - SM1,DM6 - SRC ADR ODD / DST ADR EVEN
;*****
TS505: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #505,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV #DATA+1,R5 ;SRC ADDR = DATA +1
CLR (R2) ;[DEST] = 000000
MOV #MRUF0+6,R3 ;BASE DEST ADDR = MBUF0+6
CCC ;CLEAR FLAGS - SCOPE SYNC
2\$: MOV B (R5),-6(R3) ;TEST THE MOV B
CMP R4,(R2) ;CORRECT RESULT ?
BEQ TS506 ;BR IF YES
3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED WRONG RESULT
;*****
;TEST 506 MOV B TEST - SM1,DM7 - SRC ADR ODD / DST ADR EVEN
;*****
TS506: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #506,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV #DATA+1,R5 ;SRC ADDR = DATA +1
CLR (R2) ;[DEST] = 000000
MOV #ATA,R3 ;BASE DEST ADDR = ATA
CCC ;CLEAR FLAGS - SCOPE SYNC
2\$: MOV B (R5),@10(R3) ;TEST THE MOV B
CMP R4,(R2) ;CORRECT RESULT ?
BEQ TS507 ;BR IF YES

9878 034230 011203
9879 034232 104001
9880
9881
9882
9883
9884 034234 000004
9885 034234 012700 000507
9886 034246 013701 034270
9887 034246 012702 063312
9888 034252 012704 000377
9889 034256 012703 177777
9890 034262 010205
9891 034264 005012
9892 034266 000257
9893
9894
9895 034270 110315
9896
9897 034272 020412
9898 034274 001402
9899
9900 034276 011203
9901 034300 104001
9902
9903
9904
9905
9906 034302 000004
9907 034302 012700 000510
9908 034304 013701 034336
9909 034310 012702 063312
9910 034314 012704 000377
9911 034324 012703 177777
9912 034330 010205
9913 034332 005012
9914 034334 000257
9915
9916
9917 034336 110325
9918
9919 034340 020412
9920 034342 001402
9921
9922 034344 011203
9923 034346 104001
9924
9925
9926
9927
9928 034350 000004
9929 034352 012700 000511
9930 034356 013701 034406
9931 034362 012702 063312
9932 034366 012704 000377
9933

3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED WRONG RESULT
;*****
;TEST 507 MOV B SMO,DM1 TEST
;*****
TS507: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #507,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV #-1,R3 ;R3 CONTAINS SOURCE OP
MOV R2,R5 ;R5 CONTAINS DEST ADDR
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2\$: MOV B R3,(R5)+ ;TEST THE MOV B
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TS510 ;BR IF YES
3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED THE WRONG RESULT
;*****
;TEST 510 MOV B SMO,DM2 TEST
;*****
TS510: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #510,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV #-1,R3 ;R3 CONTAINS SOURCE OP
MOV R2,R5 ;R5 CONTAINS DEST ADDR
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2\$: MOV B R3,(R5)+ ;TEST THE MOV B
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TS511 ;BR IF YES
3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED THE WRONG RESULT
;*****
;TEST 511 MOV B SMO,DM3 TEST
;*****
TS511: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #511,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377

```

9934 034372 012703 177777
9935 034376 012705 063306
9936 034402 000257
9937 034404 000257
9938
9939 034406 110335
9940
9941 034410 020412
9942 034412 001402
9943
9944 034414 011203
9945 034416 104001
9946
9947
9948
9949
9950 034420
9951 034420 000004
9952 034422 012700 000512
9953 034426 013701 034526
9954 034432 012702 063312
9955 034436 012704 177400
9956 034442 012703 177777
9957 034446 012705 063314
9958 034452 000512
9959 034454 000257
9960
9961 034456 110345
9962
9963 034460 020412
9964 034462 001402
9965
9966 034464 011203
9967 034466 104001
9968
9969
9970
9971
9972 034470
9973 034470 000004
9974 034472 012700 000513
9975 034476 013701 034526
9976 034506 012704 063312
9977 034512 012703 177777
9978 034516 012705 063314
9979 034522 000512
9980 034524 000257
9981
9982
9983 034526 110365 177776
9984
9985 034532 020412
9986 034534 001402
9987
9988 034536 011203
9989 034540 104001
    
```

```

MACY11 30A(1052) 15-NOV-78 15:26 PAGE 185
T511 MOVB SMO,DM3 TEST
MOV #-1,R3 ;SOURCE QP IN R3
MOV #ATA+10,R5 ;BASE DEST ADDR = ATA+10
CLR (R2) ;DEST1 = 000000
CCC ;SCOPE SYNC
2$: MOVB R3,@(R5)+ ;TEST THE MOVB
CMP R4,(R2) ;RESULT CORRECT ?
BEQ T512 ;;BR IF YES
3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOVb DELIVERED THE WRONG RESULT
;*****
;TEST 512 MOVb SMO,DM4 TEST
;*****
T512: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #512,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177400,R4 ;RESULT S / B = 177400
MOV #-1,R3 ;R3 CONTAINS SOURCE OP
MOV #MBUF0+2,R5 ;BASE DEST ADDR = MBUF0+2
CLR (R2) ;DEST1 = 000000
CCC ;SCOPE SYNC
2$: MOVB R3,-(R5) ;TEST THE MOVb
CMP R4,(R2) ;RESULT CORRECT ?
BEQ T513 ;;BR IF YES
3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOVb DELIVERED THE WRONG RESULT
;*****
;TEST 513 MOVb SMO,DM6 TEST
;*****
T513: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #513,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177,R4 ;RESULT S / B = 177
MOV #-1,R3 ;R3 CONTAINS SOURCE OP
MOV #MBUF0+2,R5 ;BASE DEST ADDR = MBUF0+2
CLR (R2) ;DEST1 = 000000
CCC ;SCOPE SYNC
2$: MOVB R3,-2(R5) ;TEST THE MOVb
CMP R4,(R2) ;RESULT CORRECT ?
BEQ T514 ;;BR IF YES
3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOVb DELIVERED THE WRONG RESULT
    
```

```

9990
9991
9992
9993
9994 034542
9995 034542 000004
9996 034544 012700 000514
9997 034550 013701 034526
9998 034554 012704 177777
9999 034560 012705 052525
10000 034570 012703
10001 034570 000257
10002 034572 000267
10003
10004 034574 050503
10005
10006 034576 100003
10007 034600 001402
10008 034602 102401
10009 034604 103401
10010
10011 034606 104002
10012
10013 034610 020403
10014 034612 001401
10015
10016 034614 104002
10017
10018
10019
10020
10021 034616
10022 034616 000004
10023 034620 012700 000515
10024 034624 013701 034654
10025
10026 034630 032737 002000 063234
10027 034636 001401
10028 034640 000000
10029 034642 005004
10030 034644 005005
10031 034646 005003
10032 034650 000257
10033 034652 000270
10034
10035 034654 050503
10036
10037 034656 100403
10038 034660 001002
10039 034662 102401
10040 034664 103001
10041
10042 034666 104002
10043
10044 034670 020403
10045 034672 001401
    
```

```

MACY11 30A(1052) 15-NOV-78 15:26 PAGE 186
T513 MOVb SMO,DM6 TEST
;*****
;TEST 514 BIS TEST - SMO,DM0 - N:C = 0111
;*****
T514: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #514,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S / B = 177777
MOV #52525,R5 ;SRC DPR = 52525
MOV #52525,R3 ;DEST1 = 52525
CCC ;CLEAR FLAGS
267 ;N:C = 0111
2$: BIS R5,R3 ;TEST THE BIS
BPL 3$ ;N:C = 1001 ?
BEQ 3$
BVS 3$
RCS 4$
3$: ERROR 2 ;BIS FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ T515 ;;BR IF YES
5$: ERROR 2 ;BIS DELIVERED THE WRONG RESULT
;*****
;TEST 515 BIS TEST - SMO,DM0 - N:C = 1000
;*****
T515: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #515,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;;LOAD R1 WITH TEST INSTRUCTION WORD
.SBTTL USER CONTROLLED BREAKPOINT -- BIT10
BIT #BIT10,@#BPTLOC ;BREAKPOINT HALT SET ??
HALT ;BR IF NOT
CLR +4 ;BREAK-DEPRESS CONTINUE TO CONTINUE
R4 ;RESULT S / B = 000000
R5 ;SRC DPR = 000000
R3 ;DEST1 = 000000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000
2$: BIS R5,R3 ;TEST THE BIS
BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$
3$: ERROR 2 ;BIS FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ T516 ;;BR IF YES
    
```

```

10046
10047 034674 104002 5$: ERROR 2 ;BIS DELIVERED THE WRONG RESULT
;*****
;TEST 516 BIC TEST - SMO,DMO - N:C = 0111
;*****
TST516:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #516,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100000,R4 ;RESULT S / B = 100000
MOV #17777,R5 ;SRC OPR = 77777
MOV #-1,R3 ;DESTJ = 77777
CCC ;CLEAR FLAGS
267 ;N:C = 0111

10062 034730 040503 2$: BIC R5,R3 ;TEST THE BIC
BPL 3$ ;N:C = 1001 ?
BEQ 3$
BVS 3$
BCS 4$

10069 034742 104002 3$: ERROR 2 ;BIC FAILED TO ALTER CODES PROPERLY

10071 034744 020403 4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TS1517 ;BR IF YES

10074 034750 104002 5$: ERROR 2 ;BIC DELIVERED THE WRONG RESULT
;*****
;TEST 517 BIC TEST - SMO,DMO - N:C = 1000
;*****
TST517:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #517,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100000,R4 ;RESULT S / B = 000000
MOV #00000,R5 ;SRC OPR = 00000
MOV #00000,R3 ;DESTJ = 00000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

10089 034776 040503 2$: BIC R5,R3 ;TEST THE BIC
BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$

```

```

10095
10096 035010 104002 3$: ERROR 2 ;BIC FAILED TO ALTER CODES PROPERLY
10097
10098 035012 020403 4$: CMP R4,R3 ;CORRECT RESULT ?
10099 035014 001401 BEQ TS1520 ;BR IF YES
10100
10101 035016 104002 5$: ERROR 2 ;BIC DELIVERED THE WRONG RESULT
;*****
;TEST 520 BIT TEST - SMO,DMO - N:C = 0111
;*****
TST520:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #520,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100000,R4 ;RESULT S / B = 100000
MOV #100000,R5 ;SRC OPR = 100000
MOV #100000,R3 ;DESTJ = 100000
CC ;CLEAR FLAGS
267 ;N:C = 0111

10115 035052 030503 2$: BIT R5,R3 ;TEST THE BIT
BPL 3$ ;N:C = 1001
BEQ 3$
BVS 3$
BCS 4$

10123 035064 104002 3$: ERROR 2 ;BIT FAILED TO ALTER CODES PROPERLY
10124
10125 035066 020403 4$: CMP R4,R3 ;CORRECT RESULT ?
10126 035070 001402 BEQ TS1521 ;BR IF YES
10127
10128 035072 011203 5$: MOV (R2),R3 ;GET THE WAS DATA
10129 035074 104002 ERROR 2 ;BIT DELIVERED A RESULT
;*****
;TEST 521 BIT TEST - SMO,DMO - N:C = 1000
;*****
TST521:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #521,R0 ;LOAD RO WITH TEST NUMBER
MOV @#25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #125252,R4 ;RESULT S / B = 125252
MOV #52525,R5 ;SRC OPR = 52525
MOV R4,R3 ;DESTJ = 125252
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

10144 035126 030503 2$: BIT R5,R3 ;TEST THE BIT
BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$

```

10151 035140 104002
10152
10153 035142 020403
10154 035144 001401
10155
10156 035146 104002
10157
10158
10159
10160
10161 035150 000004
10162 035150 012700 000522
10163 035152 012701 035200
10164 035152 012704 000001
10165 035166 005005
10166 035170 012703 000001
10167 035174 000257
10168 035176 000266
10169
10170
10171 035200 020503
10172
10173 035202 100003
10174 035204 001402
10175 035206 102401
10176 035210 103401
10177
10178 035212 104002
10179
10180 035214 020403
10181 035216 001401
10182
10183 035220 104002
10184
10185
10186
10187
10188 035222 000004
10189 035222 012700 000523
10190 035224 013701 035252
10191 035230 012704 177777
10192 035234 016403
10193 035244 000257
10194 035246 000272
10195 035250
10196
10197
10198 035252 020503
10199
10200 035254 100403
10201 035256 001002
10202 035260 102401
10203 035262 103001
10204
10205 035264 104002
10206

3\$: ERROR 2 ;BIT FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ T51522 ;;BR IF YES
5\$: ERROR 2 ;BIT DELIVERED A RESULT
;*****
;TEST 522 CMP TEST - SMO,DMO - N:C = 0110
;*****
T522: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #522,R0 ;LOAD R0 WITH TEST NUMBER
MOV #1,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S / B = +1
CLR R5 ;SRC DPR = 000000
MOV #+1,R3 ;[DEST] = +1
CCC ;CLEAR FLAGS
266 ;N:C = 0110
2\$: CMP R5,R3 ;TEST THE CMP
BPL 3\$;N:C = 1001
BEQ 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 2 ;CMP FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ T51523 ;;BR IF YES
5\$: ERROR 2 ;CMP DELIVERED A RESULT
;*****
;TEST 523 CMP TEST - SMO,DMO - N:C = 1010
;*****
T523: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #523,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S / B = 177777
MOV #1,R5 ;SRC DPR = 177777
MOV R4,R3 ;[DEST] = 177777
CCC ;CLEAR FLAGS
272 ;N:C = 1010
2\$: CMP R5,R3 ;TEST THE CMP
BMI 3\$;N:C = 0100
BNF 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 2 ;CMP FAILED TO ALTER CODES PROPERLY

10207 035266 020403
10208 035270 001401
10209
10210 035272 104002
10211
10212
10213
10214
10215 035274 000004
10216 035276 012700 000524
10217 035280 013701 035324
10218 035306 000001
10219 035306 012704 000001
10220 035312 012705 100000
10221 035316 012703 000001
10222 035322 000257
10223
10224 035324 020503
10225
10226 035326 100403
10227 035330 001402
10228 035330 102001
10229 035334 103001
10230
10231 035336 104002
10232
10233 035340 020403
10234 035342 001401
10235
10236 035344 104002
10237
10238
10239
10240
10241 035346 000004
10242 035346 012700 000525
10243 035350 013701 035404
10244 035354 012702 063312
10245 035360 012704 177777
10246 035364 012705 125252
10247 035370 012712 052525
10248 035374 000257
10249 035400 000267
10250 035402
10251
10252 035404 050512
10253
10254 035406 100003
10255 035410 001402
10256 035414 102401
10257 035414 103401
10258
10259 035416 104001
10260
10261 035420 020412
10262 035422 001402

4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ T51524 ;;BR IF YES
5\$: ERROR 2 ;CMP DELIVERED A RESULT
;*****
;TEST 524 CMP TEST - SMO,DMO - N:C = 0000
;*****
T524: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #524,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S / B = +1
MOV #100000,R5 ;SRC DPR = 100000
MOV #+1,R3 ;[DEST] = +1
CCC ;CLEAR FLAGS
2\$: CMP R5,R3 ;TEST THE CMP
BMI 3\$;N:C = 0010
BEQ 3\$
BVC 3\$
BCC 4\$
3\$: ERROR 2 ;CMP FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ T51525 ;;BR IF YES
5\$: ERROR 2 ;CMP DELIVERED A RESULT
;*****
;TEST 525 BIS TEST - SMO,DMI - N:C = 0111
;*****
T525: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #525,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #BUF0,R2 ;DEST ADDR = MBUF0
MOV #1,R4 ;RESULT S / B = 177777
MOV #125252,R5 ;SRC DPR = 125252
MOV #52525,(R2) ;[DEST] = 52525
CCC ;CLEAR FLAGS
267 ;N:C = 0111
2\$: BIS R5,(R2) ;TEST THE BIS
BPL 3\$;N:C = 1001
BEQ 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 1 ;BIS FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ T51526 ;;BR IF YES

```

10263
10264 035424 011203
10265 035426 104001
10266
10267
10268
10269
10270 035430 000004
10271 035430 012700 000526
10272 035432 013701 035460
10273 035436 013701 063312
10274 035442 012702
10275 035446 005004
10276 035450 005005
10277 035452 005014
10278 035454 000257
10279 035456 000270
10280
10281 035460 050512
10282
10283 035462 100403
10284 035464 001002
10285 035466 102401
10286 035470 103001
10287
10288 035472 104001
10289
10290 035474 020412
10291 035476 001402
10292
10293 035500 011203
10294 035502 104001
10295
10296
10297
10298
10299 035504 000004
10300 035504 012700 000527
10301 035512 013701 035542
10302 035516 012702 063312
10303 035522 012704 100000
10304 035526 012705 077777
10305 035532 012712 177777
10306 035536 000257
10307 035540 000267
10308
10309
10310 035542 040512
10311
10312 035544 100003
10313 035546 001402
10314 035550 102401
10315 035552 103401
10316
10317 035554 104001
10318

```

```

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BIS DELIVERED THE WRONG RESULT
;*****
;TEST 526 BIS TEST - SMO,DM1 - N:C = 1000
;*****
T526: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #526,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
CLR R5 ;SRC OPR = 000000
CLR (R2) ;EDESTJ = 000000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

2$: BIS R5,(R2) ;TEST THE BIS
;N:C = 0100

3$: ERROR 1 ;BIS FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ T527 ;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BIS DELIVERED THE WRONG RESULT
;*****
;TEST 527 BIC TEST - SMO,DM1 - N:C = 0111
;*****
T527: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #527,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #100000,R4 ;RESULT S / B = 100000
MOV #77777,R5 ;SRC OPR = 77777
CLR (R2) ;EDESTJ = 77777
CCC ;CLEAR FLAGS
SEN ;N:C = 0111

2$: BIC R5,(R2) ;TEST THE BIC
;N:C = 1001

3$: ERROR 1 ;BIC FAILED TO ALTER CODES PROPERLY

```

```

10319 035556 020412
10320 035560 001402
10321
10322 035562 011203
10323 035564 104001
10324
10325
10326
10327
10328 035566 000004
10329 035566 012700 000530
10330 035570 013701 035542
10331 035574 013701 063312
10332 035600 012702
10333 035604 005004
10334 035606 005005
10335 035610 005014
10336 035614 000257
10337 035614 000270
10338
10339 035616 040512
10340
10341 035620 100403
10342 035622 001002
10343 035624 102401
10344 035626 103001
10345
10346 035630 104001
10347
10348 035632 020412
10349 035634 001402
10350
10351 035636 011203
10352 035640 104001
10353
10354
10355
10356
10357 035642 000004
10358 035644 012700 000531
10359 035650 013701 035700
10360 035654 012702 063312
10361 035660 012704 100000
10362 035664 012705 100000
10363 035670 012712 100000
10364 035674 000257
10365 035676 000267
10366
10367 035700 030512
10368
10369
10370 035702 100003
10371 035704 001402
10372 035706 102401
10373 035710 103401
10374

```

```

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ T530 ;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BIC DELIVERED THE WRONG RESULT
;*****
;TEST 530 BIC TEST - SMO,DM1 - N:C = 1000
;*****
T530: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #530,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
CLR R5 ;SRC OPR = 000000
CLR (R2) ;EDESTJ = 000000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

2$: BIC R5,(R2) ;TEST THE BIC
;N:C = 0100

3$: ERROR 1 ;BIC FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ T531 ;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BIC DELIVERED THE WRONG RESULT
;*****
;TEST 531 BIT TEST - SMO,DM1 - N:C = 0111
;*****
T531: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #531,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #100000,R4 ;RESULT S / B = 100000
MOV #100000,R5 ;SRC OPR = 100000
MOV #100000,(R2) ;EDESTJ = 100000
CCC ;CLEAR FLAGS
SEN ;N:C = 0111

2$: BIT R5,(R2) ;TEST THE BIT
;N:C = 1001

```

```

10375 035712 104001
10376
10377 035714 020412
10378 035716 001402
10379
10380 035720 011203
10381 035722 104001
10382
10383
10384
10385
10386
10387 035724
10388 035726 000004 000532
10389 035728 012700 036044
10390 035732 013701 063312
10391 035736 012702 063312
10392 035742 012704 052525
10393 035746 012705 177777
10394 035752 012705 177777
10395 035756 000257 052525
10396 035760 000270
10397
10398 035762 030512
10399
10400 035764 100403
10401 035766 001002
10402 035770 102401
10403 035772 103001
10404
10405 035774 104001
10406 035776 020412
10407 036000 001402
10408
10409 036002 011203
10410 036004 104001
10411
10412
10413
10414 036006
10415 036008 000004 000533
10416 036010 012700 036044
10417 036014 013701 063312
10418 036020 012702 063312
10419 036024 012704 177777
10420 036030 012705 177777
10421 036034 012705 177777
10422 036040 006257
10423 036042 000272
10424
10425 036044 020512
10426
10427 036046 100403
10428 036050 001002
10429 036052 102401
10430 036054 103001

```

```

3$: ERROR 1 ;BIT FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST532 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BIT DELIVERED A RESULT
;*****
;TEST 532 BIT TEST - SMO,DM1 - N:C = 1000
;*****
TST532:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #532,R0 ;LOAD R0 WITH TEST NUMBER
MOV #22,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV MBUF0,R2 ;DEST ADDR = MBUF0
MOV #52525,R4 ;RESULT S / B = 52525
MOV #1,R5 ;SRC OPR = 175252
MOV #52525,(R2) ;DEST J = 52525
CCC ;CLEAR FLAGS
SEN ;N:C = 1000
2$: BIT R5,(R2) ;TEST THE BIT
BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$
3$: ERROR 1 ;BIT FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST533 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BIT DELIVERED A RESULT
;*****
;TEST 533 CMP TEST - SMO,DM1 - N:C = 1010
;*****
TST533:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #533,R0 ;LOAD R0 WITH TEST NUMBER
MOV #22,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV MBUF0,R2 ;DEST ADDR = MBUF0
MOV #-1,R4 ;RESULT S / B = -1
MOV #-1,R5 ;SRC OPR = 177777
MOV #-1,(R2) ;DEST J = 177777
CCC ;CLEAR FLAGS
272 ;N:C = 1010
2$: CMP R5,(R2) ;TEST THE CMP
BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$

```

```

10431
10432 036056 104001
10433
10434 036060 020412
10435 036062 001402
10436
10437 036064 011203
10438 036066 104001
10439
10440
10441
10442
10443 036070
10444 036072 000004 000534
10445 036074 012700 036124
10446 036076 013701 063312
10447 036102 012702 063312
10448 036106 012704 000001
10449 036112 005005 000001
10450 036114 012705
10451 036120 000257
10452 036122 000266
10453
10454 036124 020512
10455
10456 036126 100003
10457 036130 001402
10458 036132 102401
10459 036134 103401
10460
10461 036136 104001
10462
10463 036140 020412
10464 036142 001402
10465
10466 036144 011203
10467 036146 104001
10468
10469
10470
10471
10472 036150
10473 036152 000004 000535
10474 036154 012700 036204
10475 036156 013701 063312
10476 036162 012702 063312
10477 036166 012704 000001
10478 036172 012705 100000
10479 036176 012705 100000
10480 036202 000257 000001
10481
10482 036204 020512
10483
10484 036206 100403
10485 036208 001402
10486 036212 102001

```

```

3$: ERROR 1 ;CMP FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST534 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CMP DELIVERED A RESULT
;*****
;TEST 534 CMP TEST - SMO,DM1 - N:C = 0110
;*****
TST534:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #534,R0 ;LOAD R0 WITH TEST NUMBER
MOV #22,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV MBUF0,R2 ;DEST ADDR = MBUF0
MOV #+1,R4 ;RESULT S / B = +1
CLR R5 ;SRC OPR = 000000
MOV #+1,(R2) ;DEST J = +1
CCC ;CLEAR FLAGS
266 ;N:C = 0110
2$: CMP R5,(R2) ;TEST THE CMP
BPL 3$ ;N:C = 1001
REQ 3$
BVS 3$
BCS 4$
3$: ERROR 1 ;CMP FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST535 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CMP DELIVERED A RESULT
;*****
;TEST 535 CMP TEST - SMO,DM1 - N:C = 0000
;*****
TST535:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #535,R0 ;LOAD R0 WITH TEST NUMBER
MOV #22,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV MBUF0,R2 ;DEST ADDR = MBUF0
MOV #+1,R4 ;RESULT S / B = +1
MOV #100000,R5 ;SRC OPR = 100000
MOV #+1,(R2) ;DEST J = +1
CCC ;CLEAR FLAGS
2$: CMP R5,(R2) ;TEST THE CMP
BMI 3$ ;N:C = 0010
BEQ 3$
BVC 3$

```

```

10487 036214 103001
10488
10489 036216 104001
10490
10491 036220 020412
10492 036222 001402
10493
10494 036224 011203
10495 036226 104001
10496
10497
10498
10499
10500
10501 036230 000004
10502 036232 012700 000536
10503 036236 013701 036262
10504 036242 012704 177777
10505 036246 012705 063332
10506 036252 012703 052525
10507 036256 000270
10508 036260 000267
10509
10510 036262 051503
10511
10512 036264 100003
10513 036266 001402
10514 036270 102401
10515 036272 103401
10516
10517 036274 104002
10518
10519 036276 020403
10520 036300 001401
10521
10522 036302 104002
10523
10524
10525
10526
10527 036304 000004
10528 036304 012700 000537
10529 036306 013701 036332
10530 036312 013701
10531 036316 005004
10532 036320 012705 063322
10533 036324 005003
10534 036326 000257
10535 036330 000270
10536
10537 036332 051503
10538
10539 036334 100403
10540 036336 001002
10541 036340 102401
10542 036342 103001

```

```

BCC 4$
3$: ERROR 1 ;CMP FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TS536 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CMP DELIVERED A RESULT

*****
;TEST 536 BIS TEST - SM1,DM0 - N:C = 0111
*****
;TS536:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #536,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S / B = 177777
MOV #DWTA+10,R5 ;SRC ADDR = DWTA+10
MOV #52525,R3 ;CDESTJ = 52525
CC ;CLEAR FLAGS
267 ;N:C = 0111
2$: BIS (R5),R3 ;TEST THE BIS
BPL 3$ ;N:C = 1001
BEQ 3$
BVS 3$
BCS 4$
3$: ERROR 2 ;BIS FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TS537 ;;BR IF YES
5$: ERROR 2 ;BIS DELIVERED THE WRONG RESULT

*****
;TEST 537 BIS TEST - SM1,DM0 - N:C = 1000
*****
;TS537:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #537,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLP R4 ;RESULT S / B = 000000
MOV #DWTA,R5 ;SRC ADDR = DWTA
CLR R3 ;CDESTJ = 000000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000
2$: BIS (R5),R3 ;TEST THE BIS
BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$

```

```

10543
10544 036344 104002
10545
10546 036346 020403
10547 036350 001401
10548
10549 036352 104002
10550
10551
10552
10553
10554
10555 036354 000004
10556 036356 012700 000540
10557 036362 013701 036412
10558 036366 012704 100000
10559 036372 012705 063316
10560 036376 012703 177777
10561 036402 012715 077777
10562 036406 000257
10563 036410 000267
10564
10565 036412 041503
10566
10567 036414 100003
10568 036416 001402
10569 036420 102401
10570 036422 103401
10571
10572 036424 104002
10573
10574 036426 020403
10575 036430 001401
10576
10577 036432 104002
10578
10579
10580
10581
10582 036434 000004
10583 036436 012700 000541
10584 036442 013701 036462
10585 036446 005004
10586 036450 012705 063322
10587 036454 005003
10588 036456 000257
10589 036460 000270
10590
10591
10592 036462 041503
10593
10594 036464 100403
10595 036466 001002
10596 036470 102401
10597 036472 103001
10598

```

```

3$: ERROR 2 ;BIS FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TS540 ;;BR IF YES
5$: ERROR 2 ;BIS DELIVERED THE WRONG RESULT

*****
;TEST 540 BIC TEST - SM1,DM0 - N:C = 0111
*****
;TS540:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #540,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100000,R4 ;RESULT S / B = 100000
MOV #MBUF1,R5 ;SRC ADDR = MBUF1
MOV #1,R3 ;CDESTJ = 177777
MOV #77777,(R5) ;SRC OPR = 77777
CC ;CLEAR FLAGS
267 ;N:C = 0111
2$: BIC (R5),R3 ;TEST THE BIC
BPL 3$ ;N:C = 1001 ?
BEQ 3$
BVS 3$
BCS 4$
3$: ERROR 2 ;BIC FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TS541 ;;BR IF YES
5$: ERROR 2 ;BIC DELIVERED THE WRONG RESULT

*****
;TEST 541 BIC TEST - SM1,DM0 - N:C = 1000
*****
;TS541:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #541,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #DWTA,R5 ;SRC ADDR = DWTA
CLR R3 ;CDESTJ = 000000
CC ;CLEAR FLAGS
SEN ;N:C = 1000
2$: BIC (R5),R3 ;TEST THE BIC
BMI 3$ ;N:C = 0100
BNE 3$
BVS 3$
BCC 4$

```


10599 036474 104002
10600
10601 036476 020403
10602 036500 001401
10603
10604 036502 104002
10605
10606
10607
10608
10609 036504
10610 036504 000004
10611 036506 012700 000542
10612 036512 013701 036534
10613 036516 012704 100000
10614 036522 012705 063324
10615 036526 010403
10616 036530 000257
10617 036532 000267
10618
10619 036534 031503
10620
10621 036536 100003
10622 036540 001402
10623 036542 102401
10624 036544 103401
10625
10626 036546 104002
10627
10628 036550 020403
10629 036552 001401
10630
10631 036554 104002
10632
10633
10634
10635
10636 036556
10637 036556 000004
10638 036560 012700 000543
10639 036564 013701 036606
10640 036570 012704 052525
10641 036574 012705 063332
10642 036600 010403
10643 036602 000257
10644 036604 000270
10645
10646 036606 031503
10647
10648 036610 100403
10649 036612 001002
10650 036614 102401
10651 036616 103001
10652
10653 036620 104002
10654

3\$: ERROR 2 ;BIT FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
 BEQ T542 ;;BR IF YES
5\$: ERROR 2 ;BIT DELIVERED THE WRONG RESULT
;*****
;TEST 542 BIT TEST - SM1,DMO - N:C = 0111
;*****
T542: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #542,R0 ;;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #10000,R4 ;RESULT S / B = 10000
 MOV DWTA+2,R5 ;SRC ADDR = DWTA+2
 MOV R4,R3 ;DEST1 = 10000
 CCC ;CLEAR FLAGS
 267 ;N:C = 0111
2\$: BIT (R5),R3 ;TEST THE BIT
 BPL 3\$;N:C = 1001 ?
 BEQ 3\$
 BVS 3\$
 BCS 4\$
3\$: ERROR 2 ;BIT FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
 BEQ T543 ;;BR IF YES
5\$: ERROR 2 ;BIT DELIVERED A RESULT
;*****
;TEST 543 BIT TEST - SM1,DMO - N:C = 1000
;*****
T543: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #543,R0 ;;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #52525,R4 ;RESULT S / B = 52525
 MOV DWTA+10,R5 ;SRC ADDR = DWTA+10
 MOV R4,R3 ;DEST1 = 52525
 CCC ;CLEAR FLAGS
 SEN ;N:C = 1000
2\$: BIT (R5),R3 ;TEST THE BIT
 BMI 3\$;N:C = 0100
 BNE 3\$
 BVS 3\$
 BCC 4\$
3\$: ERROR 2 ;BIT FAILED TO ALTER CODES PROPERLY

10655 036622 020403
10656 036624 001401
10657
10658 036626 104002
10659
10660
10661
10662 036630
10663 036630 000004
10664 036632 012700 000544
10665 036636 013701 036660
10666 036642 012704 000001
10667 036646 012705 063322
10668 036652 010403
10669 036654 000257
10670 036656 000266
10671
10672 036660 021503
10673
10674 036662 100003
10675 036664 001402
10676 036666 102401
10677 036670 103401
10678
10679 036672 104002
10680
10681 036674 020403
10682 036676 001401
10683
10684 036700 104002
10685
10686
10687
10688
10689 036702
10690 036702 000004
10691 036704 012700 000545
10692 036710 013701 036732
10693 036714 012704 177777
10694 036720 012705 063324
10695 036724 010403
10696 036726 000257
10697 036730 000272
10698
10699 036732 021503
10700
10701 036734 100403
10702 036736 001002
10703 036740 102401
10704 036742 103001
10705
10706 036744 104002
10707
10708 036746 020403
10709 036750 001401
10710

4\$: CMP R4,R3 ;CORRECT RESULT ?
 BEQ T544 ;;BR IF YES
5\$: ERROR 2 ;BIT DELIVERED A RESULT
;*****
;TEST 544 CMP TEST - SM1,DMO - N:C = 0110
;*****
T544: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #544,R0 ;;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #1,R4 ;RESULT S / B = +1
 MOV DWTA+1,R5 ;SRC ADDR = DWTA
 MOV R4,R3 ;DEST1 = +1
 CCC ;CLEAR FLAGS
 266 ;N:C = 0110
2\$: CMP (R5),R3 ;TEST THE CMP
 BPL 3\$;N:C = 1001
 BEQ 3\$
 BVS 3\$
 BCS 4\$
3\$: ERROR 2 ;CMP FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
 BEQ T545 ;;BR IF YES
5\$: ERROR 2 ;CMP DELIVERED A RESULT
;*****
;TEST 545 CMP TEST - SM1,DMO - N:C = 1010
;*****
T545: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #545,R0 ;;LOAD R0 WITH TEST NUMBER
 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #1,R4 ;RESULT S / B = 177777
 MOV DWTA+2,R5 ;SRC ADDR = DWTA+2
 MOV R4,R3 ;DEST1 = 177777
 CCC ;CLEAR FLAGS
 272 ;N:C = 1010
2\$: CMP (R5),R3 ;TEST THE CMP
 BMI 3\$;N:C = 0100
 BNE 3\$
 BVS 3\$
 BCC 4\$
3\$: ERROR 2 ;CMP FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
 BEQ T546 ;;BR IF YES

0711 036752 104002
0712
0713
0714
0715 036754
0716 036754 000004
0717 036754 012700 000546
0718 036756 013701 037010
0719 036762 013701 037010
0720 036765 012704 000001
0721 036776 012705 063316
0722 037002 012715 100000
0723 037006 000257
0724
0725 037010 021503
0726
0727 037012 100403
0728 037014 001402
0729 037016 102001
0730 037020 103001
0731
0732 037022 104002
0733
0734 037024 020403
0735 037026 001401
0736
0737 037030 104002
0738
0739
0740
0741
0742
0743 037032
0744 037034 012700 000547
0745 037034 013701 037070
0746 037040 012702 063312
0747 037044 012704 177777
0748 037050 012704 063312
0749 037054 012705 052525
0750 037060 000257
0751 037064 000267
0752 037066
0753
0754 037070 051512
0755
0756 037072 100003
0757 037074 001402
0758 037076 102401
0759 037100 103401
0760
0761 037102 104001
0762
0763 037104 020412
0764 037106 001402
0765
0766 037110 011203

```
5$: ERROR 2 ;CMP DELIVERED A RESULT
;*****
;TEST 546 CMP TEST - SM1,DW0 - N:C = 0000
;*****
TST546:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #546,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #R4 ;RESULT S / B = +1
MOV #MBUF1,R5 ;SRC ADDR = MBUF1
MOV #+1,R3 ;CDESTJ = +1
MOV #100000,(R5) ;SRC OPR = 100000
CCC ;CLEAR FLAGS
;N:C = 0010

2$: CMP (R5),R3 ;TEST THE CMP
BMI 3$ ;N:C = 0010
BEQ 3$
BVC 3$
BCC 4$

3$: ERROR 2 ;CMP FAILED TO ALTER CODES PROPERLY

4$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST547 ;;BR IF YES

5$: ERROR 2 ;CMP DELIVERED A RESULT
;*****
;TEST 547 BIS SM1,DW1 TEST - N:C = 0111
;*****
TST547:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #547,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #-1,R4 ;RESULT S / B = 177777
MOV #DWTA+10,R5 ;SOURCE ADDR = DWTA+10
MOV #52525,(R2) ;CDESTJ = 052525
CCC ;CLEAR FLAGS
MOV 267 ;N:C = 0111

2$: BIS (R5),(R2) ;TEST THE BIS
BPL 3$ ;N:C = 1001?
BEQ 3$
BVS 3$
BCS 4$

3$: ERROR 1 ;BIS FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST550 ;;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
```

10767 037112 104001
10768
10769
10770
10771
10772 037114
10773 037116 012700 000550
10774 037116 013701 037146
10775 037122 012702 063312
10776 037126 012702 063312
10777 037132 005004
10778 037134 012705 063322
10779 037142 005012
10780 037142 000257
10781 037144 000270
10782
10783 037146 051512
10784
10785 037150 100403
10786 037152 001002
10787 037154 102401
10788 037156 103001
10789
10790 037160 104001
10791
10792 037162 020412
10793 037164 001402
10794
10795 037166 011203
10796 037170 104001
10797
10798
10799
10800
10801 037172
10802 037172 000004
10803 037174 012700 000551
10804 037200 014701 037234
10805 037204 012702 063312
10806 037210 012704 100000
10807 037214 012705 063316
10808 037220 012715 077777
10809 037224 012712 177777
10810 037230 000257
10811 037232 000267
10812
10813 037234 041512
10814
10815 037236 100003
10816 037240 001402
10817 037242 102401
10818 037244 103401
10819
10820 037246 104001
10821
10822 037250 020412

```
5$: ERROR 1 ;BIS DELIVERED THE WRONG RESULT
;*****
;TEST 550 BIS SM1,DW1 TEST - N:C = 1000
;*****
TST550:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #550,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
MOV #DWTA,R5 ;SOURCE ADDR = DWTA
CLR (R2) ;CDESTJ = 000000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

2$: BIS (R5),(R2) ;TEST THE BIS
BMI 3$ ;N:C = 0100 ?
BNE 3$
BVS 3$
BCC 4$

3$: ERROR 1 ;BIS FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST551 ;;BR IF YES

5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BIS DELIVERED THE WRONG RESULT
;*****
;TEST 551 BIC SM1,DW1 TEST - N:C = 0111
;*****
TST551:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #551,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #100000,R4 ;RESULT S / B = 100000
MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
MOV #177777,(R5) ;CDESTJ = 177777
MOV #-1,(R2) ;CDESTJ = 177777
CCC ;CLEAR FLAGS
MOV 267 ;N:C = 0111

2$: BIC (R5),(R2) ;TEST THE BIC
BPL 3$ ;N:C = 1001 ?
BEQ 3$
BVS 3$
BCS 4$

3$: ERROR 1 ;BIC FAILED TO ALTER CODES PROPERLY

4$: CMP R4,(R2) ;CORRECT RESULT ?
```

10823 037252 001402
 10824 037252 011203
 10825 037254 011203
 10826 037256 104001
 10827
 10828
 10829
 10830
 10831 037260
 10832 037260 000004
 10833 037260 012700 000552
 10834 037260 013701 037314
 10835 037260 013702 063312
 10836 037276 005004
 10837 037300 012705 063316
 10838 037304 005015
 10839 037306 005014
 10840 037310 000270
 10841 037312 000270
 10842
 10843 037314 041512
 10844
 10845 037316 100403
 10846 037320 011002
 10847 037322 102401
 10848 037324 103001
 10849
 10850 037326 104001
 10851
 10852 037330 020412
 10853 037332 001402
 10854
 10855 037334 011203
 10856 037336 104001
 10857
 10858
 10859
 10860
 10861 037340
 10862 037340 000004
 10863 037340 012700 000553
 10864 037346 013701 037402
 10865 037352 012702 063312
 10866 037356 012704 125252
 10867 037362 012705 063316
 10868 037366 012715 052525
 10869 037378 000254 125252
 10870 037400 000270
 10871
 10872
 10873 037402 031512
 10874
 10875 037404 100403
 10876 037406 001002
 10877 037410 102401
 10878 037412 103001

BEQ T552 ;BR IF YES
 5\$: MOV (R2),R3 ;GET THE WAS DATA
 ERROR 1 ;BIC DELIVERED THE WRONG RESULT
 ;*****
 ;TEST 552 BIC SM1,DM1 TEST - N:C = 1000
 ;*****
 T552: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #552,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #425,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
 R4 ;RESULT S / B = 000000
 CLR #MBUF1,R5 ;SOURCE ADDR = MBUF1
 (R5) ;[SOURCE] = 000000
 (R2) ;[DEST] = 000000
 CCC ;CLEAR FLAGS
 SEN ;N:C = 1000
 2\$: BIC (R5),(R2) ;TEST THE BIC
 BMI 3\$;N:C = 0100 ?
 BNE 3\$
 BVS 3\$
 BCC 4\$
 3\$: ERROR 1 ;BIC FAILED TO ALTER CODES PROPERLY
 4\$: CMP R4,(R2) ;CORRECT RESULT ?
 BEQ T553 ;BR IF YES
 5\$: MOV (R2),R3 ;GET THE WAS DATA
 ERROR 1 ;BIC DELIVERED THE WRONG RESULT
 ;*****
 ;TEST 553 BIT SM1,DM1 TEST - N:C = 1000
 ;*****
 T553: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #553,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #425,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
 MOV #125252,R4 ;RESULT S / B = 125252
 MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
 MOV #52525,(R5) ;[SOURCE] = 052525
 MOV #125252,(R2) ;[DEST] = 125252
 CCC ;CLEAR FLAGS
 SEN ;N:C = 1000
 2\$: BIT (R5),(R2) ;TEST THE BIT
 BMI 3\$;N:C = 0100 ?
 BNE 3\$
 BVS 3\$
 BCC 4\$

10879
 10880 037414 104001
 10881
 10882 037416 020412
 10883 037420 001402
 10884
 10885 037422 011203
 10886 037424 104001
 10887
 10888
 10889
 10890
 10891 037426
 10892 037426 000004
 10893 037430 013700 000554
 10894 037434 013701 037502
 10895
 10896 037440 032737 004000 063234
 10897 037446 001401
 10898 037450 000000
 10899 037452 012702 063312
 10900 037456 012704 100000
 10901 037462 012705 063316
 10902 037466 012715 100000
 10903 037472 012714 100000
 10904 037476 000257
 10905 037500 000267
 10906
 10907 037502 031512
 10908
 10909 037504 100003
 10910 037506 001402
 10911 037510 102401
 10912 037512 103401
 10913
 10914 037514 104001
 10915
 10916 037516 020412
 10917 037520 001402
 10918
 10919 037522 011203
 10920 037524 104001
 10921
 10922
 10923
 10924
 10925 037526
 10926 037526 000004
 10927 037526 013700 000555
 10928 037534 013701 037562
 10929 037540 012702 063312
 10930 037544 012704 177777
 10931 037550 012705 063316
 10932 037554 012715 177777
 10933 037558 010411
 10934 037562 000254

3\$: ERROR 1 ;BIT FAILED TO ALTER CODES PROPERLY
 4\$: CMP R4,(R2) ;CORRECT RESULT ?
 BEQ T554 ;BR IF YES
 5\$: MOV (R2),R3 ;GET THE WAS DATA
 ERROR 1 ;BIT DELIVERED A RESULT
 ;*****
 ;TEST 554 BIT SM1,DM1 TEST - N:C = 0111
 ;*****
 T554: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #554,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #425,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 .SBTTL USER CONTROLLED BREAKPOINT -- BIT11
 BIT #BIT11,#MBPTLOC ;BREAKPOINT HALT SET ??
 BEQ +*4 ;BR IF NOT
 HALT ;BREAK-DEPRESS CONTINUE TO CONTINUE
 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
 MOV #100000,R4 ;RESULT S / B = 100000
 MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
 MOV #100000,(R5) ;[SOURCE] = 100000
 MOV #100000,(R2) ;[DEST] = 100000
 CCC ;CLEAR FLAGS
 267 ;N:C = 0111
 2\$: BIT (R5),(R2) ;TEST THE BIT
 BPL 3\$;N:C = 1001 ?
 BEQ 3\$
 BVS 3\$
 BCS 4\$
 3\$: ERROR 1 ;BIT FAILED TO ALTER CODES PROPERLY
 4\$: CMP R4,(R2) ;CORRECT RESULT ?
 BEQ T555 ;BR IF YES
 5\$: MOV (R2),R3 ;GET THE WAS DATA
 ERROR 1 ;BIT DELIVERED A RESULT
 ;*****
 ;TEST 555 CMP SM1,DM1 TEST - N:C = 1010
 ;*****
 T555: SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #555,R0 ;LOAD R0 WITH TEST NUMBER
 MOV #425,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
 MOV #1,R4 ;RESULT S / B = 177777
 MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
 MOV #1,(R5) ;[SOURCE] = 177777
 MOV R4,(R2) ;[DEST] = 177777
 CCC ;CLEAR FLAGS

```

10935 037564 000272          272          ;N:C = 1010
10936 037566 021512          2$: CMP      (R5),(R2)    ;TEST THE CMP
10937 037570 100403          BMI      3$             ;N:C = 0100 ?
10938 037572 001002          BNE     3$
10939 037574 102401          BVS     3$
10940 037576 103001          BCC     4$
10941 037600 104001          3$: ERROR 1             ;CMP FAILED TO ALTER CODES PROPERLY
10942 037602 020412          4$: CMP      R4,(R2)     ;CORRECT RESULT ?
10943 037604 001402          BEQ     TS$556         ;;BR IF YES
10944 037606 011203          5$: MOV      (R2),R3    ;GET THE WAS DATA
10945 037610 104001          ERROR 1               ;CMP DELIVERED A RESULT
10946 *****
10947 ;*TEST 556 CMP SM1,DM1 TEST - N:C = 0110
10948 ;*****
10949 TS$556:
10950 SCOPE          ;CALL THE SCOPE LOOP UTILITY
10951 MOV #556,R0    ;LOAD R0 WITH TEST NUMBER
10952 MOV #25,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
10953 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
10954 MOV #1,R4      ;RESULT S / B = 000001
10955 MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
10956 CLR (R5)      ;SOURCE = 000000
10957 MOV #+1,(R2)  ;DESTJ = 000001
10958 CCC          ;CLEAR FLAGS
10959 266          ;N:C = 0110
10960 037612 000004          2$: CMP      (R5),(R2)    ;TEST THE CMP
10961 037614 012700 000556    BPL     3$             ;N:C = 1001 ?
10962 037620 013701 037652    BEQ     3$
10963 037624 012702 063312    BVS     3$
10964 037630 012704 000001    BCC     4$
10965 037634 012705 063316    CLR     (R5)
10966 037642 002577 000001    MOV     #+1,(R2)
10967 037646 000257          CCC
10968 037650 000266          266
10969 037652 021512          2$: CMP      (R5),(R2)    ;TEST THE CMP
10970 037654 100003          BPL     3$             ;N:C = 1001 ?
10971 037656 001402          BEQ     3$
10972 037660 102401          BVS     3$
10973 037662 103401          BCC     4$
10974 037664 104001          3$: ERROR 1             ;CMP FAILED TO ALTER CODES PROPERLY
10975 037666 020412          4$: CMP      R4,(R2)     ;CORRECT RESULT ?
10976 037670 001402          BEQ     TS$557         ;;BR IF YES
10977 037672 011203          5$: MOV      (R2),R3    ;GET THE WAS DATA
10978 037674 104001          ERROR 1               ;CMP DELIVERED A RESULT
10979 *****
10980 ;*TEST 557 CMP SM1,DM1 TEST - N:C = 0000
10981 ;*****
10982 TS$557:
10983 SCOPE          ;CALL THE SCOPE LOOP UTILITY
10984 MOV #557,R0    ;LOAD R0 WITH TEST NUMBER
10985 MOV #25,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
10986 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
10987 MOV #+1,R4      ;RESULT S / B = 000001
10988 037700 012700 000557    MOV     #MBUF1,R5
10989 037704 013701 037736    CLR     (R5)
10990 037710 012702 063312    MOV     #+1,R4
10991 037714 012704 000001    MOV     #+1,R4

```

```

10991 037720 012705 063316    MOV     #MBUF1,R5    ;SOURCE ADDR = MBUF1
10992 037724 012715 100000    MOV     #100000,(R5) ;SOURCE = 000000
10993 037730 012712 000001    MOV     #+1,(R2)    ;DESTJ = 000001
10994 037734 000257          CCC                ;CLEAR FLAGS
10995 037736 021512          2$: CMP      (R5),(R2)    ;TEST THE CMP
10996 037740 100403          BMI     3$             ;N:C = 0010 ?
10997 037742 001402          BEQ     3$
10998 037744 102001          BVC     3$
10999 037746 103001          BCC     4$
11000 037750 104001          3$: ERROR 1             ;CMP FAILED TO ALTER CODES PROPERLY
11001 037752 020412          4$: CMP      R4,(R2)     ;CORRECT RESULT ?
11002 037754 001402          BEQ     TS$560         ;;BR IF YES
11003 037756 011203          5$: MOV      (R2),R3    ;GET THE WAS DATA
11004 037760 104001          ERROR 1               ;CMP DELIVERED A RESULT
11005 *****
11006 ;*TEST 560 B1SB SM1,DM0 TEST - SOURCE ADDR ODD
11007 ;*****
11008 TS$560:
11009 SCOPE          ;CALL THE SCOPE LOOP UTILITY
11010 MOV #560,R0    ;LOAD R0 WITH TEST NUMBER
11011 MOV #25,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
11012 MOV #377,R4    ;RESULT S / B = 377
11013 MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
11014 MOV #R3        ;DESTJ = 000000
11015 CCC          ;SCOPE SYNC
11016 040010 151503          2$: B1SB     (R5),R3     ;TEST THE B1SB
11017 040012 020403          CMP      R4,R3        ;RESULT CORRECT ?
11018 040014 001401          BEQ     TS$561         ;;BR IF YES
11019 040016 104002          3$: ERROR 2             ;B1SB DELIVERED THE WRONG RESULT
11020 *****
11021 ;*TEST 561 B1SB SM1,DM1 TEST - SOURCE ADDR ODD
11022 ;*****
11023 TS$561:
11024 SCOPE          ;CALL THE SCOPE LOOP UTILITY
11025 MOV #561,R0    ;LOAD R0 WITH TEST NUMBER
11026 MOV #25,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
11027 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
11028 MOV #17,R4     ;RESULT S / B = 177
11029 MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
11030 MOV (R2)      ;DESTJ = 000000
11031 CCC          ;SCOPE SYNC
11032 040052 151512          2$: B1SB     (R5),(R2)    ;TEST THE B1SB
11033 040054 020412          CMP      R4,(R2)     ;CORRECT RESULT
11034 040056 001402          BEQ     TS$562         ;;BR IF YES

```

```

11047
11048 040060 011203
11049 040062 104001
11050
11051
11052
11053
11054 040064
11055 040064 000004
11056 040066 012700 000562
11057 040072 013701 040120
11058 040076 012702 063312
11059 040102 012704 000377
11060 040106 012705 064631
11061 040112 012703
11062 040114 010203
11063 040116 000257
11064
11065 040120 151523
11066
11067 040122 020412
11068 040124 001402
11069
11070 040126 011203
11071 040130 104001
11072
11073
11074
11075
11076 040132
11077 040134 000004
11078 040134 012700 000563
11079 040140 013701 040170
11080 040144 012702 063312
11081 040150 012704 000377
11082 040154 012705 064631
11083 040160 012703
11084 040162 012703 063306
11085 040166 000257
11086
11087 040170 151533
11088
11089 040172 020412
11090 040174 001402
11091
11092 040176 011203
11093 040200 104001
11094
11095
11096
11097
11098 040202
11099 040202 000004
11100 040204 012700 000564
11101 040210 013701 040240
11102 040214 012702 063312

```

```

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;B1SB DELIVERED THE WRONG RESULT
;*****
;TEST 562 B1SB SM1,DM2 TEST - SOURCE ADDR ODD
;*****
T562: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #562,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
CLR (R2) ;[DEST] = 000000
MOV R2,R3 ;DEST ADDR IN R3
CCC ;SCOPE SYNC

2$: B1SB (R5),(R3)+ ;TEST THE B1SB
CMP R4,(R2) ;CORRECT RESULT
BEQ T563 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;B1SB DELIVERED THE WRONG RESULT
;*****
;TEST 563 B1SB SM1,DM3 TEST - SOURCE ADDR ODD
;*****
T563: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #563,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
CLR (R2) ;[DEST] = 000000
MOV #ATA+10,R3 ;BASE DEST ADDR = ATA+10
CCC ;SCOPE SYNC

2$: B1SB (R5),@(R3)+ ;TEST THE B1SB
CMP R4,(R2) ;CORRECT RESULT
BEQ T564 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;B1SB DELIVERED THE WRONG RESULT
;*****
;TEST 564 B1SB SM1,DM4 TEST - SOURCE ADDR ODD
;*****
T564: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #564,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0

```

```

11103 040220 012704 177400
11104 040224 012705 064631
11105 040230 012703 063314
11106 040234 005012
11107 040236 000257
11108
11109 040240 151543
11110
11111 040242 020412
11112 040244 001402
11113
11114 040246 011203
11115 040250 104001
11116
11117
11118
11119
11120 040252
11121 040252 000004
11122 040254 012700 000565
11123 040260 013701 040310
11124 040264 012702 063312
11125 040270 012704 000377
11126 040274 012705 064631
11127 040300 012703 063310
11128 040304 005012
11129 040306 000257
11130
11131 040310 151553
11132
11133 040312 020412
11134 040314 001402
11135
11136 040316 011203
11137 040320 104001
11138
11139
11140
11141
11142
11143 040322
11144 040322 000004
11145 040324 012700 000566
11146 040330 013701 040360
11147 040334 012702 063312
11148 040340 012704 000377
11149 040344 012705 064631
11150 040350 012703 063320
11151 040354 005012
11152 040356 000257
11153
11154 040360 151563 177772
11155
11156 040364 020412
11157 040366 001402
11158 040370 011203

```

```

MOV #177400,R4 ;RESULT S / B = 177400
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
MOV #MBUF0+2,R3 ;BASE DEST ADDR = MBUF0+2
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: B1SB (R5),-(R3) ;TEST THE B1SB
CMP R4,(R2) ;CORRECT RESULT
BEQ T565 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;B1SB DELIVERED THE WRONG RESULT
;*****
;TEST 565 B1SB SM1,DM5 TEST - SOURCE ADDR ODD
;*****
T565: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #565,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
MOV #ATA+12,R3 ;BASE DEST ADDR = ATA+12
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: B1SB (R5),@(R3) ;TEST THE B1SB
CMP R4,(R2) ;CORRECT RESULT
BEQ T566 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;B1SB DELIVERED THE WRONG RESULT
;*****
;TEST 566 B1SB SM1,DM6 TEST - SOURCE ADDR ODD
;*****
T566: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #566,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
MOV #MBUF0+6,R3 ;BASE DEST ADDR = MBUF0+6
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC

2$: B1SB (R5),-6(R3) ;TEST THE B1SB
CMP R4,(R2) ;CORRECT RESULT
BEQ T567 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA

```

```

11159 040372 104001
11160
11161
11162
11163
11164 040374 000004
11165 040374 012700 000567
11166 040376 013701 040432
11167 040402 012702 063312
11168 040406 012704 000377
11169 040412 012704 064631
11170 040415 012705 063276
11171 040426 005012
11172 040430 000257
11173
11174
11175 040432 151573 000010
11176
11177 040436 020412
11178 040440 001402
11179
11180 040442 011203
11181 040444 104001
11182
11183
11184
11185
11186 040446 000004
11187 040446 012700 000570
11188 040450 013701 040576
11189 040454 012702 063312
11190 040460 012704 000377
11191 040464 012704
11192 040470 010203
11193 040472 005012
11194 040474 000257
11195
11196 040476 150423
11197
11198 040500 020412
11199 040502 001402
11200
11201 040504 011203
11202 040506 104001
11203
11204
11205
11206
11207 040510
11208 040510 000004
11209 040512 012700 000571
11210 040516 013701 040546
11211 040526 012704 063312
11212 040532 012704 177400
11213 040532 012705 000377
11214 040536 012703 063313

```

```

3$: ERROR 1 ;BISB DELIVERED THE WRONG RESULT
;*****
;TEST 567 BISB SM1,DM7 TEST - SOURCE ADDR ODD
;*****
TST567:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #567,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
MOV #ATA,R3 ;BASE DEST ADDR = ATA
CLR (R2) ;DEST = 000000
CCC ;SCOPE SYNC

2$: BISB (R5),@10(R3) ;TEST THE BISB

CMP R4,(R2) ;CORRECT RESULT
BEQ TST570 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BISB DELIVERED THE WRONG RESULT
;*****
;TEST 570 BISB SM0,DM2 TEST - DEST ADDR EVEN
;*****
TST570:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #570,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV R2,R3 ;DEST ADDR IN R3
CLR (R2) ;DEST = 000000
CCC ;SCOPE SYNC

2$: BISB R4,(R3)+ ;TEST THE BISB

CMP R4,(R2) ;CORRECT RESULT
BEQ TST571 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BISB DELIVERED THE WRONG RESULT
;*****
;TEST 571 BISB SM0,DM1 TEST - DEST ADDR ODD
;*****
TST571:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #571,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177400,R4 ;RESULT S / B = 177400
MOV #377,R5 ;CRS1=SOURCE OPR = 377
MOV #MBUF0+1,R3 ;ODD DEST ADDR IN R3

```

```

11215 040542 005012
11216 040544 000257
11217
11218 040546 150513
11219
11220 040550 020412
11221 040552 001402
11222
11223 040554 011203
11224 040556 104001
11225
11226
11227
11228
11229 040560
11230 040560 000004
11231 040562 012700 000572
11232 040566 013701 040610
11233 040572 012702 063312
11234 040576 012704 000377
11235 040602 010203
11236 040604 005012
11237 040606 000257
11238
11239 040610 150413
11240
11241 040612 020412
11242 040614 001402
11243
11244 040616 011203
11245 040620 104001
11246
11247
11248
11249
11250 040622
11251 040622 000004
11252 040624 012700 000573
11253 040630 013701 040660
11254 040634 012701 063312
11255 040640 012704 177400
11256 040644 012705 064631
11257 040650 012703 063313
11258 040654 005012
11259 040656 000257
11260
11261 040660 151513
11262
11263 040662 020412
11264 040664 001402
11265
11266 040666 011203
11267 040670 104001
11268
11269
11270

```

```

CLR (R2) ;DEST = 000000
CCC ;SCOPE SYNC

2$: BISB R5,(R3) ;TEST THE BISB

CMP R4,(R2) ;CORRECT RESULT
BEQ TST572 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BISB DELIVERED THE WRONG RESULT
;*****
;TEST 572 BISB SM0,DM1 TEST - DEST ADDR EVEN
;*****
TST572:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #572,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV R2,R3 ;DEST ADDR IN R3
CLR (R2) ;DEST = 000000
CCC ;SCOPE SYNC

2$: BISB R4,(R3) ;TEST THE BISB

CMP R4,(R2) ;CORRECT RESULT
BEQ TST573 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BISB DELIVERED THE WRONG RESULT
;*****
;TEST 573 BISB SM1,DM1 TEST - DEST ADDR ODD
;*****
TST573:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #573,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177400,R4 ;RESULT S / B = 177400
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
MOV #MBUF0+1,R3 ;ODD DEST ADDR IN R3
CLR (R2) ;DEST = 000000
CCC ;SCOPE SYNC

2$: BISB (R5),(R3) ;TEST THE BISB

CMP R4,(R2) ;CORRECT RESULT
BEQ TST574 ;BR IF YES

3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BISB DELIVERED THE WRONG RESULT
;*****
;TEST 574 JMP MODE 1 TEST, FLAGS = 1111
;*****

```


11383
11384
11385 041142 022702 041156
11386 041146 001404
11387 041150 104006

6\$: CMP #7\$+2,R2 ;DID JMP UPDATE R2?
BEQ TST601 ;BR IF YES
ERROR 6 ;JMP FAILED TO UPDATE REGISTER

11388 041152 000402
11389 041154 041130
11390 041156 104006

7\$: BR TST601 ;GO TO SCOPE EXIT
4\$;JMP3 CONTAINS JUMP ADDRESS
ERROR 6 ;ERROR CALL OCCURS IF MODE3 HAPPENS
;TO EXECUTE AS MODE 1 OR 2 AND
;4\$ IS LEGAL INSTRUCTION

;TEST 601 JMP TEST MODE 3; FLAGS = 0000

11391
11392
11393
11394
11395
11396
11397 041160
11398 041160 000004
11399 041162 012700 000601
11400 041166 013701 041200
11401 041172 012702 041222
11402 041176 000257
11403

TST601: SCOPE #601,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #2\$,R1 ;LOAD R0 WITH TEST NUMBER
MOV #6\$,R2 ;LOAD R1 WITH TEST INSTRUCTION WORD
CCC ;R2 CONTAINS ADDRESS OF JUMP ADDRESS
;MAKE N:C = 0000

11404 041200 000132
11405
11406 041202 104006
11407 041204 000410

2\$: JMP @(R2)+ ;TEST THE JMP - GO TO 4\$
3\$: ERROR 6 ;JMP FAILED TO LOAD THE PC
BR TST602 ;GO TO SCOPE EXIT

11408 041206 103403
11409 041210 102402
11410 041212 001401
11411 041214 100004
11412
11413

4\$: BCS 5\$;BR IF JMP SET "C"
BVS 5\$;BR IF JMP SET "V"
BEQ 5\$;BR IF JMP SET "Z"
BPL TST602 ;BR IF "N" STILL CLEAR

11414 041216 104006
11415 041220 000402
11416
11417 041222 041206
11418 041224 104006
11419

5\$: ERROR 6 ;JMP ALTERED CODES - SET
BR TST602 ;GO TO SCOPE EXIT
6\$: 4\$;JUMP ADDRESS IN 6\$
ERROR 6 ;JMP MODE 3 EXECUTED LIKE MODE 1 OR 2

;TEST 602 JMP TEST MODE 4; FLAGS = 1111

11420
11421
11422
11423 041226
11424 041226 000004
11425 041230 012700 000602
11426 041234 013701 041246
11427 041240 013701 041256
11428 041244 000277
11429

TST602: SCOPE #602,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #2\$,R1 ;LOAD R0 WITH TEST NUMBER
MOV #5\$,R2 ;R2] = JUMP ADDRESS PLUS 2
SCC ;MAKE N:C = 1111

11430 041246 000142
11431
11432 041250 104006
11433 041252 000414
11434

2\$: JMP -(R2) ;TEST THE JMP - GO TO 5\$ MINUS 2
3\$: ERROR 6 ;JMP FAILED TO LOAD PC
BR TST603 ;GO TO SCOPE EXIT

11435 041254 000402
11436 041256 104006
11437 041260 000411
11438

5\$: BR 4\$;GO TEST FLAGS - JMP LOADED PC OK
ERROR 6 ;JMP FAILED TO AUTO-DECREMENT R2
BR TST603 ;GO TO SCOPE EXIT

11439 041262 103003
11440 041264 102002
11441 041266 001001
11442 041270 100401
11443

4\$: BCC 7\$;BR IF JMP CLEARED "C"
BVC 7\$;BR IF JMP CLEARED "V"
BNE 7\$;BR IF JMP CLEARED "Z"
BMI 6\$;BR IF "N" STILL SET

11444 041272 104006
11445
11446 041274 022702 041254
11447 041300 001401
11448
11449 041302 104006
11450
11451
11452
11453
11454 041304
11455 041304 000004
11456 041306 012700 000603
11457 041306 013701 041334
11458 041312 012702 041334
11459 041322 000257
11460
11461 041324 000142
11462
11463 041326 104006
11464 041330 000405
11465
11466 041332 103403
11467 041334 102402
11468 041336 001401
11469 041340 100001
11470
11471 041342 104006
11472
11473
11474
11475
11476 041344
11477 041344 000004
11478 041346 012700 000604
11479 041352 013701 041364
11480 041352 012702 041420
11481 041362 000277
11482
11483 041364 000152
11484
11485 041366 104006
11486 041370 000414
11487
11488 041372 103003
11489 041374 102002
11490 041376 001001
11491 041400 100401
11492
11493 041402 104006
11494
11495 041404 022702 041416
11496 041410 001404
11497
11498 041412 104006
11499 041414 000402

7\$: ERROR 6 ;JMP ALTERED FLAGS
6\$: CMP #5\$,R2 ;DID JMP UPDATE R2 PROPERLY?
BEQ TST603 ;BR IF YES
9\$: ERROR 6 ;JMP FAILED TO UPDATE REGISTER
;*****
;TEST 603 JMP TEST MODE 4; FLAGS = 0000
;*****
TST603: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #603,R0 ;LOAD R0 WITH TEST NUMBER
MOV #4\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #4\$,R2 ;R2 = JUMP ADDRESS PLUS 2
CC ;MAKE N:C = 0000
2\$: JMP -(R2) ;TEST THE JMP - TO TO 4\$
3\$: ERROR 6 ;JMP FAILED TO LOAD PC
BR TST604 ;GO TO SCOPE EXIT
4\$: BCS 5\$;BR IF JMP SET "C"
BVS 5\$;BR IF JMP SET "V"
BEQ 5\$;BR IF JMP SET "Z"
BPL TST604 ;BR IF "N" STILL CLEAR
5\$: ERROR 6 ;JMP ALTERED CODES - SET
;*****
;TEST 604 JMP TEST MODE 5; FLAGS = 1111
;*****
TST604: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #604,R0 ;LOAD R0 WITH TEST NUMBER
MOV #4\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #JMP5,R2 ;JMP CONTAINS ADDR+2 OF JUMP ADDRESS
SCC
2\$: JMP @-(R2) ;TEST THE JMP - GO TO 4\$
3\$: ERROR 6 ;JMP FAILED TO LOAD PC
RR TST605 ;GO TO SCOPE EXIT
4\$: BCC 5\$;BR IF JMP CLEARED "C"
BVC 5\$
BNE 5\$
BMI 6\$
5\$: ERROR 6 ;JMP ALTERED CODES - CLEARED
6\$: CMP #JMP5-2,R2 ;DID R2 GET AUTO-DECREMENTED
BEQ TST605 ;BR IF YES
7\$: ERROR 6 ;JMP FAILED TO UPDATE REGISTER
BR TST605 ;GO TO SCOPE EXIT

11500 041416 041372
11501 041420 104006
11502
11503
11504
11505
11506 041422
11507 041422 000004
11508 041424 012700 000605
11509 041430 013701 041442
11510 041430 012702 041466
11511 041440 000257
11512
11513 041442 000152
11514
11515 041444 104006
11516 041446 000410
11517
11518 041450 103403
11519 041452 102402
11520 041454 001401
11521 041456 100004
11522
11523 041460 104006
11524 041462 000402
11525
11526 041464 041450
11527 041466 104006
11528
11529
11530
11531 041470
11532 041472 000004
11533 041474 012700 000606
11534 041476 013701 041510
11535 041502 012702 041534
11536 041506 000277
11537
11538 041510 000162 177764
11539
11540
11541 041514 104006
11542 041516 000407
11543
11544 041520 103003
11545 041522 102002
11546 041524 001001
11547 041526 100403
11548
11549 041530 104006
11550 041532 000401
11551
11552 041534 104006
11553
11554
11555

JMP5: ERROR 6 ;THIS LOCATION CONTAINS JMP ADDRESS
;JMP EXECUTED LIKE A MODE 1 OR 2
;*****
;TEST 605 JMP TEST MODE 5; FLAG = 0000
;*****
TST605: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #605,R0 ;LOAD R0 WITH TEST NUMBER
MOV #4\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #JMP5A,R2 ;R2 = ADDR +2 OF JUMP ADDRESS
CC ;SET N:C = 0000
2\$: JMP @-(R2) ;TEST THE JMP - GO TO 4\$
3\$: ERROR 6 ;JMP FAILED TO LOAD PC
BR TST606 ;GO TO SCOPE EXIT
4\$: BCS 5\$;BR IF JMP SET "C"
BVS 5\$;BR IF JMP SET "V"
BEQ 5\$;BR IF JMP SET "Z"
BPL TST606 ;BR IF "N" STILL CLEAR
5\$: ERROR 6 ;JMP ALTERED THE CODES - SET
BR TST606 ;GO TO SCOPE EXIT
JMP5A: 4\$ ERROR 6 ;THIS LOCATION CONTAINS JUMP ADDRESS
;JMP EXECUTED LIKE A MODE 1 OR 2
;*****
;TEST 606 JMP TEST MODE 6; FLAGS = 1111
;*****
TST606: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #606,R0 ;LOAD R0 WITH TEST NUMBER
MOV #4\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #7\$,R2 ;R2] = BASE ADDRESS TO BE INDEXED
SCC ;MAKE N:C = 1111
2\$: JMP 4\$-7\$(R2) ;TEST THE JMP - GO TO 4\$
3\$: ERROR 6 ;JMP FAILED TO LOAD THE PC
BR TST607 ;GO TO SCOPE EXIT
4\$: BCC 5\$;BR IF JMP CLEARED "C"
BCV 5\$
BNE 5\$
BMI TST607 ;BR IF "N" STILL SET
5\$: ERROR 6 ;JMP ALTERED CODES - CLEARED
BR TST607 ;GO TO SCOPE EXIT
7\$: ERROR 6 ;JMP EXECUTED LIKE A MODE 1 OR 2 OR
;FAILED TO INDEX [R2]
;*****

```

11556 ;*TEST 607 JMP TEST MODE 6; FLAGS = 0000
11557 ;*****
11558 TST607:
11559 SCOPE #607,R0 ;CALL THE SCOPE LOOP UTILITY
11560 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
11561 MOV #5,R2 ;CR2] = LOAD R1 WITH TEST INSTRUCTION WORD
11562 ;CR2] = BASE ADDRESS FOR JUMP
11563 CCC ;MAKE N:C = 0000
11564
11565 2$: JMP 45-75(R2) ;TEST THE JMP - GO TO 45
11566
11567 3$: ERROR 6 ;JMP FAILED TO LOAD PC
11568 BR TST610 ;GO TO SCOPE EXIT
11569
11570 4$: BCS 55 ;BR IF JMP SET "C"
11571 BVS 55 ;BR IF JMP SET "V"
11572 BEQ 72 ;BR IF JMP SET "Z"
11573 BPL TST610 ;BR IF "N" STILL CLEAR
11574
11575 5$: ERROR 6 ;JMP ALTERED CODES
11576 BR TST610 ;GO TO SCOPE EXIT
11577
11578 7$: ERROR 6 ;JMP EXECUTED LIKE A MODE 1 OR 2, OR
11579 ;FAILED TO INDEX [R2]
11580
11581 ;*****
11582 ;*TEST 610 JMP TEST MODE 7; FLAGS = 1111
11583 ;*****
11584 TST610:
11585 SCOPE #610,R0 ;CALL THE SCOPE LOOP UTILITY
11586 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
11587 MOV #5,R2 ;CR2] = LOAD R1 WITH TEST INSTRUCTION WORD
11588 ;CR2] = BASE ADDRESS
11589 CCC ;MAKE N:C = 1111
11590
11591 2$: JMP @R5-55(R2) ;TEST THE JMP - GO TO 45
11592
11593 3$: ERROR 6 ;JMP FAILED TO LOAD PC
11594 BR TST611 ;GO TO SCOPE EXIT
11595
11596 5$: ERROR 6 ;JMP FAILED TO INDEX OR ACTED LIKE MODE 1 OR 2
11597 BR TST611 ;GO TO SCOPE EXIT
11598
11599 4$: BCC 75 ;BR IF JMP CLEARED "C"
11600 BVC 72 ;BR IF JMP CLEARED "V"
11601 BNE 72 ;BR IF JMP CLEARED "Z"
11602 BMI TST611 ;BR IF "N" STILL SET
11603
11604 7$: ERROR 6 ;JMP ALTERED CODES - CLEARED
11605 BR TST611 ;GO TO SCOPE EXIT
11606
11607 8$: 4$ ;THIS LOCATION CONTAINS JMP ADDRESS
11608
11609 ERROR 6 ;JMP EXECUTED LIKE MODE 6
11610
11611 ;*****

```

```

11612 ;*TEST 611 JMP TEST MODE 7; FLAGS = 0000
11613 ;*****
11614 TST611:
11615 SCOPE #611,R0 ;CALL THE SCOPE LOOP UTILITY
11616 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
11617 MOV #5,R2 ;CR2] = LOAD R1 WITH TEST INSTRUCTION WORD
11618 ;CR2] = BASE ADDRESS
11619 CCC ;MAKE N:C = 0000
11620
11621 2$: JMP @R5-55(R2) ;TEST THE JMP - GO TO 45
11622
11623 3$: ERROR 6 ;JMP FAILED TO LOAD PC
11624 BR TST612 ;GO TO SCOPE EXIT
11625
11626 5$: ERROR 6 ;JMP FAILED TO INDEX
11627 BR TST612 ;GO TO SCOPE EXIT
11628
11629 4$: BCS 75 ;BR IF JMP SET "C"
11630 BVS 72 ;BR IF JMP SET "V"
11631 BEQ 72 ;BR IF JMP SET "Z"
11632 BPL TST612 ;BR IF "N" STILL CLEAR
11633
11634 7$: ERROR 6 ;JMP ALTERED CODES - SET
11635 BR TST612 ;GO TO SCOPE EXIT
11636
11637 9$: 4$ ;THIS LOCATION CONTAINS JMP ADDRESS
11638
11639 ERROR 6 ;JMP EXECUTED LIKE A MODE 6
11640
11641 ;*****
11642 ;*TEST 612 JSR MODE 1 TEST - LOAD PC / PUSH SP
11643 ;*****
11644 TST612:
11645 SCOPE #612,R0 ;CALL THE SCOPE LOOP UTILITY
11646 MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
11647 MOV #5,R2 ;CR2] = LOAD R1 WITH TEST INSTRUCTION WORD
11648 MOV SP,R5 ;SAVE THE SP
11649 PC,#SLPERR ;SET ERROR LOOP ADDRESS
11650 MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
11651 MOV #45,R2 ;DEST ADDR = 45
11652 CCC ;SCOPE SYNC
11653
11654 2$: JSR R4,(R2) ;TEST THE JSR - GO TO 45
11655
11656 3$: ERROR 6 ;JSR FAILED TO LOAD THE PC
11657
11658 4$: TST (SP)+ ;POP THE SP
11659 CMP SP,R5 ;DID JSR PUSH THE SP ?
11660 BEQ TST613 ;BR IF YES
11661
11662 TST -(SP) ;RESTORE ERROR SP
11663 MOV SP,R3 ;CR3] = WAS SP
11664 MOV R2,R4
11665 TST -(R4) ;CR4] = S/B SP
11666 ERROR 3 ;JSR FAILED TO PUSH THE SP
11667

```

```

11668 042010 010506
11669
11670
11671
11672 042012
11673 042012 000004
11674 042014 012700 000613
11675 042020 013701 042052
11676 042024 010605
11677 042026 010737
11678 042036 010506 001010
11679 042034 012702 042056
11680 042040 005066 177776
11681 042044 012704 125252
11682 042050 000257
11683
11684 042052 004412
11685
11686 042054 104006
11687
11688 042056 022726 125252
11689 042062 001401
11690
11691 042064 104005
11692
11693 042066 022704 042054
11694 042072 001401
11695
11696 042074 104005
11697
11698 042076 010506
11699
11700
11701
11702
11703 042100
11704 042100 000004
11705 042102 012700 000614
11706 042106 013701 042142
11707
11708 042112 032737 010000 063234
11709 042120 001401
11710 042122 000000
11711 042124 010605
11712 042134 010737
11713 042132 010506 001010
11714 042134 012702 042146
11715 042140 000257
11716
11717 042142 004412
11718
11719 042144 104006
11720
11721 042146 100403
11722 042150 001402
11723 042152 102401

```

```

*****
;*****
;TEST 613 JSR MODE 1 TEST - CHECK RN AND OLD PC
;*****
TST613:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #613,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #25,R1                            ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5                             ;SAVE THE SP
PC,@#SLPERR                          ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP                         ;RESTORE SP FOR ERROR LOOPING
    MOV #45,R2                       ;DEST ADDR = 45
    CLR -2(SP)                        ;INIT STACK LOC TO GET [R4]
    MOV #125252,R4                   ;INIT RN = 125252
    CCC                              ;SCOPE SYNC
2$: JSR R4,(R2)                      ;TEST THE JSR - GO TO 4$
3$: ERROR 6                          ;JSR FAILED TO LOAD THE PC
4$: CMP #125252,(SP)+                ;DID JSR SAVE REG ON STACK
    BEQ 8$                           ;BR IF IT DID
5$: ERROR 5                          ;JSR FAILED TO SAVE REG ON STACK
6$: MOV R5,SP                        ;RESTORE SP IN CASE ERROR SCREWED IT UP
*****
;*****
;TEST 614 JSR MODE 1 TEST - N:C = 0000
;*****
TST614:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #614,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #25,R1                            ;LOAD R1 WITH TEST INSTRUCTION WORD
-SBTTL U CONTROLLED BREAKPOINT -- BIT12
BIT #BIT12,@#BPTLOC                 ;BREAKPOINT HALT SET ??
BEQ +4                                ;BR IF NOT
HALT                                 ;BREAK-DEPRESS CONTINUE TO CONTINUE
MOV SP,R5                             ;SAVE THE SP
PC,@#SLPERR                          ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP                         ;RESTORE SP FOR ERROR LOOPING
    MOV #45,R2                       ;DEST ADDR = 45
    CCC                              ;N:C = 0000
2$: JSR R4,(R2)                      ;TEST THE JSR - GO TO 4$
3$: ERROR 6                          ;JSR FAILED TO LOAD THE PC
4$: BMI 5$                           ;N:C = 0000 ?
    BEQ 5$
    BVS 5$

```

```

11724 042154 103001
11725
11726 042156 104005
11727 042160 010506
11728
11729
11730
11731
11732 042162
11733 042162 000004
11734 042164 012700 000615
11735 042170 013701 042212
11736 042176 010737
11737 042176 010737 001010
11738 042202 010506
11739 042204 012702 042216
11740 042210 000277
11741
11742 042212 004412
11743
11744 042214 104006
11745
11746 042216 100003
11747 042220 001002
11748 042222 102001
11749 042224 103401
11750 042226 104005
11751
11752 042230 010506
11753
11754
11755
11756
11757 042232
11758 042234 000004
11759 042240 013701 000616
11760 042244 010737 042262
11761 042246 010605
11762 042246 010737 001010
11763 042252 010506
11764 042254 012702 042266
11765 042260 000257
11766
11767 042262 004422
11768
11769 042264 104006
11770
11771 042266 005726
11772 042270 020605
11773 042272 001406
11774
11775 042274 005746
11776 042276 010603
11777 042300 010504
11778 042302 005744
11779 042304 104003

```

```

BCC 6$
5$: ERROR 5                          ;JSR FAILED - ALTERED FLAGS
6$: MOV R5,SP                        ;RESET SP IN CASE OF ERROR
;*****
;TEST 615 JSR MODE 1 TEST - N:C = 1111
;*****
TST615:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #615,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #25,R1                            ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5                             ;SAVE THE SP
PC,@#SLPERR                          ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP                         ;RESTORE SP FOR ERROR LOOPING
    MOV #45,R2                       ;DEST ADDR = 45
    SCC                              ;N:C = 1111
2$: JSR R4,(R2)                      ;TEST THE JSR - GO TO 4$
3$: ERROR 6                          ;JSR FAILED TO LOAD THE PC
4$: BPL 5$                           ;N:C = 1111 ?
    BNE 5$
    BVC 5$
    RCS 6$
5$: ERROR 5                          ;JSR ALTERED FLAGS
6$: MOV R5,SP                        ;RESET SP IN CASE OF ERROR
;*****
;*****
;TEST 616 JSR MODE 2 TEST
;*****
TST616:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #616,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #25,R1                            ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5                             ;SAVE THE SP
PC,@#SLPERR                          ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP                         ;RESTORE SP FOR ERROR LOOPS
    MOV #45,R2                       ;DEST ADDR = 45
    CCC                              ;SCOPE SYNC
2$: JSR R4,(R2)+                      ;TEST THE JSR - GO TO 4$
3$: ERROR 6                          ;JSR FAILED TO LOAD THE PC
4$: TST (SP)+                         ;RESET SP
    CMP SP,R5                         ;DID JSR PUSH STACK ?
    BEQ TST617                       ;BR IF YES
TST -(SP)                             ;RESET SP TO ERROR VALUE
MOV SP,R3                             ;WAS SP
MOV R4,R4
TST -(R4)                             ;S/B SP
5$: ERROR 3                          ;JSR FAILED TO PUSH SP

```

```

11780
11781 042306 010506
11782
11783
11784
11785
11786 042310 000004
11787 042312 012700 000617
11788 042316 013701 042340
11789 042322 010605
11790 042324 010737 001010
11791 042330 010506
11792 042332 012702 042366
11793 042336 000257
11794
11795
11796 042340 004432
11797
11798 042342 104006
11799
11800 042344 005726
11801 042346 020605
11802 042350 001411
11803
11804 042352 005746
11805 042354 010603
11806 042356 010504
11807 042360 005744
11808 042362 104003
11809 042364 000402
11810
11811 042366 042344
11812 042370 104006
11813
11814 042372 010506
11815
11816
11817
11818
11819 042374
11820 042376 000004
11821 042378 012700 000620
11822 042402 013701 042424
11823 042406 010605
11824 042410 010737 001010
11825 042414 010506
11826 042416 012702 042432
11827 042422 000257
11828
11829 042424 004442
11830
11831 042426 104006
11832
11833 042430 000401
11834 042432 104005
11835

MOV R5,SP ;RESTORE SP JUST IN CASE
;*****
;TEST 617 JSR MODE 3 TEST
;*****
TST617:
SCOPE #617,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV R5,R5 ;SAVE THE SP
MOV PC,@SLPERR ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP ;RESET SP FOR ERROR LOOPS
MOV #7,R2 ;DEST ADDR = [7$]
CCC ;SCOPE SYNC

2$: JSR R4,@(R2)+ ;TEST THE JSR - GO TO 4$ VIA 7$

3$: ERROR 6 ;JSR FAILED TO LOAD THE PC

4$: TST (SP)+ ;RESET SP
CMP SP,R5 ;DID JSR PUSH STACK ?
BEQ T$T620 ;BR IF YES

5$: TST -(SP) ;RESET SP TO ERROR VALUE
MOV SP,R3 ;WAS SP
MOV R5,R4
TST -(R4) ;S/B SP
5$: ERROR 3 ;JSR FAILED
BR 6$ ;GO EXIT

7$: 4$ ;CONTAINS JUMP ADDR
ERROR 6 ;JSR EXECUTED LIKE A MODE 1 OR 2

6$: MOV R5,SP ;RESTORE SP JUST IN CASE
;*****
;TEST 620 JSR MODE 4 TEST
;*****
TST620:
SCOPE #620,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV R5,R5 ;SAVE THE SP
MOV PC,@SLPERR ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP ;RESET SP FOR ERROR LOOPS
MOV #5,R2 ;DEST ADDR = 4$+2
CCC ;SCOPE SYNC

2$: JSR R4,-(R2) ;TEST THE JSR - GO TO 4$

3$: ERROR 6 ;JSR FAILED TO LOAD THE PC

4$: BR 6$ ;JUMPED OK - GO CHECK SP
5$: ERROR 5 ;JSR FAILED TO DECREMENT DEST REG

```

```

11836 042434 005726
11837 042436 020605
11838 042440 001406
11839
11840 042442 005746
11841 042444 010603
11842 042446 010504
11843 042450 005744
11844 042452 104003
11845
11846 042454 010506
11847
11848
11849
11850
11851 042456
11852 042458 000004
11853 042460 012700 000621
11854 042464 013701 042506
11855 042470 010605
11856 042472 010737 001010
11857 042476 010506
11858 042500 012702 042536
11859 042504 000257
11860
11861 042506 004452
11862
11863 042510 104006
11864
11865 042512 005726
11866 042514 020605
11867 042516 001411
11868
11869 042520 005746
11870 042522 010603
11871 042524 010504
11872 042526 005744
11873 042530 104003
11874 042532 000402
11875
11876 042534 042512
11877 042536 104005
11878
11879 042540 010506
11880
11881
11882
11883
11884 042542
11885 042544 000004
11886 042546 012700 000622
11887 042550 013701 042572
11888 042554 010605
11889 042556 010737 001010
11890 042562 010506
11891 042564 012702 042576

TST (SP)+ ;RESET SP
CMP SP,R5 ;DID JSR PUSH STACK ?
BEQ T$T621 ;BR IF YES

TST -(SP) ;RESET SP TO ERROR VALUE
MOV SP,R3 ;WAS SP
MOV R5,R4
TST -(R4) ;S/B SP
7$: ERROR 3 ;JSR FAILED TO PUSH SP

8$: MOV R5,SP ;RESTORE SP JUST IN CASE
;*****
;TEST 621 JSR MODE 5 TEST
;*****
TST621:
SCOPE #621,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV R5,R5 ;SAVE THE SP
MOV PC,@SLPERR ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP ;RESET SP FOR ERROR LOOPS
MOV #7,R2 ;DEST ADDR = [7$ - 2]
CCC ;SCOPE SYNC

2$: JSR R4,@-(R2) ;TEST THE JSR - GO TO 4$

3$: ERROR 6 ;JSR FAILED TO LOAD THE PC

4$: TST (SP)+ ;RESET SP
CMP SP,R5 ;DID JSR PUSH STACK ?
BEQ T$T622 ;BR IF YES

5$: TST -(SP) ;RESET SP TO ERROR VALUE
MOV SP,R3 ;WAS SP
MOV R5,R4
TST -(R4) ;S/B SP
5$: ERROR 3 ;JSR FAILED TO PUSH SP
BR 6$ ;GO EXIT

7$: 4$ ;CONTAINS JUMP ADDRESS
ERROR 5 ;JSR EXECUTED LIKE A MODE 1 OR 2

6$: MOV R5,SP ;RESTORE SP JUST IN CASE
;*****
;TEST 622 JSR MODE 6 TEST
;*****
TST622:
SCOPE #622,R0 ;CALL THE SCOPE LOOP UTILITY
MOV #25,R1 ;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV R5,R5 ;SAVE THE SP
MOV PC,@SLPERR ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP ;RESET SP FOR ERROR LOOPS
MOV #3,R2 ;[R2] = BASE DEST ADDR

```

```

11892 042570 000257
11893
11894 042572 004462 000002
11895
11896 042576 104006
11897
11898 042600 005726
11899 042602 020605
11900 042604 001406
11901
11902 042606 005746
11903 042610 010603
11904 042612 010504
11905 042614 005744
11906 042616 104003
11907 042620 010506
11908
11909
11910
11911
11913 042622
11914 042624 000004
11915 042626 012700 000623
11916 042630 013701 042652
11917 042634 010605
11918 042636 010737 001010
11919 042642 010506
11920 042644 012702 042656
11921 042650 000257
11922
11923 042652 004472 000024
11924
11925 042656 104006
11926
11927 042660 005726
11928 042662 020605
11929 042664 001411
11930
11931 042666 005746
11932 042670 010603
11933 042672 010504
11934 042674 005744
11935 042676 104003
11936 042700 000402
11937
11938 042702 042660
11939 042704 104005
11940
11941 042706 010506
11942
11943
11944
11945
11946 042710 000004
11947 042710

```

```

CCC ;SCOPE SYNC
2$: JSR R4,4$-3$(R2) ;TEST THE JSR - GO TO 4$
3$: ERROR 6 ;JSR FAILED TO LOAD THE PC OR INDEX FAILED
4$: TST (SP)+ ;RESET SP
CMP SP,R5 ;DID JSR PUSH STACK ?
BEQ T$T623 ;BR IF YES
TST -(SP) ;RESET SP TO ERROR VALUE
MOV SP,R3 ;WAS SP
MOV R5,R4
TST -(R4) ;S/B SP
5$: ERROR 3 ;JSR FAILED TO PUSH STACK
MOV R5,SP ;RESET SP JUST IN CASE
;*****
;TEST 623 JSR MODE 7 TEST
;*****
T$T623: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #623,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2$ ,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV R5,#$LPERR ;SET ERROR LOOP ADDRESS
1$: MOV R5,R3 ;RESET SP FOR ERROR LOOPS
MOV #3$ ,R2 ;BASE DEST ADDR = 3$
CCC ;SCOPE SYNC
2$: JSR R4,@7$-3$(R2) ;TEST THE JSR - GO TO 4$ VIA 7$
3$: ERROR 6 ;JSR FAILED TO LOAD THE PC
;OR THE INDEX FAILED
4$: TST (SP)+ ;RESET SP
CMP SP,R5 ;DID JSR PUSH STACK ?
BEQ T$T624 ;BR IF YES
TST -(SP) ;RESET SP TO ERROR VALUE
MOV SP,R3 ;WAS SP
MOV R5,R4
TST -(R4) ;S/B SP
5$: ERROR 3 ;JSR FAILED TO PUSH STACK
BR 6$ ;SKIP TO EXIT
7$: 4$ ;CONTAINS JUMP ADDR
ERROR 5 ;JSR WORKED LIKE A MODE 1 OR 2
6$: MOV R5,SP ;RESTORE SP JUST IN CASE
;*****
;TEST 624 SOB TEST, CRJ = 1, NO BRANCH
;*****
T$T624: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

11948 042712 012700 000624
11949 042716 013701 042736
11950 042722 012702 000001
11951 042726 000402
11952
11954 042730 104006
11954 042732 000402
11955
11956 042734 000257
11957 042736 077204
11958
11960
11961
11962 042740
11963 042742 000004
11964 042744 012700 000625
11965 042746 013701 043000
11966 042752 012702 000005
11967 042756 012705 177773
11968 042762 000405
11969
11970 042764 000474
11971 042766 000240
11972 042770 000240
11973
11974 042772 005205
11975 042774 001406
11976
11977 042776 000257
11978 043000 077204
11979 043002 005702
11980 043004 001403
11981
11983 043006 104006
11983 043010 000401
11984 043012 104006
11985
11986
11987
11988
11989 043014
11990 043016 000004
11991 043018 012700 000626
11992 043022 013701 043034
11993 043026 012702 000001
11994 043032 000277
11995
11996 043034 077202
11997
11998 043036 103003
11999 043040 104001
12000 043042 001001
12001 043044 100401
12002
12003 043046 104006

```

```

MOV #624,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2$ ,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1$ ,R2 ;SET SOB COUNTER = 1
BR 2$-2 ;GO DO THE SOB
3$: ERROR 6 ;SOB SHOULDN'T HAVE BRANCHED HERE
BR T$T625 ;GO TO SCOPE CALL
2$: CCC ;SYNC INSTR.
SOB R2,3$ ;TEST THE SOB
;*****
;TEST 625 SOB TEST, CRJ = 5, BRANCH 4 TIMES
;*****
T$T625: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #625,R0 ;LOAD R0 WITH TEST NUMBER
MOV #5$ ,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #5$ ,R2 ;SET SOB COUNTER = 5
MOV #5$ ,R5 ;SET UP R5 TO COUNT 5 BRANCHES
BR SOB2-2 ;GO DO THE SOB
SOB1: BR SOB3 ;USED BY LAST SOB TEST TO TEST MAX OFFSET
NOP ;OFFSET ADJUSTMENT
SOB5: INC R5 ;COUNT ONE BRANCH
BEQ SOBERR ;BR IF TOO MANY LOOPS BY SOB
SOB2: CCC ;SCOPE SYNC
SOB R2,SOB5 ;TEST THE SOB
TST R2 ;R2 SHOULD CONTAIN 0
BEQ T$T626 ;BR IF IT DOES
ERROR 6 ;SOB COUNTER NOT ZERO
BR T$T626 ;GO TO SCOPE CALL
SOBERR: ERROR 6 ;SOB MADE TOO MANY BRANCHES
;*****
;TEST 626 SOB TEST, CRJ = 1, FLAGS = 1111
;*****
T$T626: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #626,R0 ;LOAD R0 WITH TEST NUMBER
MOV #2$ ,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1$ ,R2 ;SET SOB COUNTER = 1
SCC ;MAKE N:C = 1111
2$: SOB R2,2$-2 ;TEST THE SOB
BCC 3$ ;BR IF C = 0
BVC 3$ ;BR IF V = 0
BR 3$ ;BR IF Z = 0
RMT T$T627 ;BR IF N = 1
3$: ERROR 6 ;SOB ALTERED CODES - CLEARED ONE

```

```

12004
12005
12006
12007
12008 043050
12009 043050 000004
12010 043052 012700 000627
12011 043052 013701 043070
12012 043052 013701 000001
12013 043066 000257
12014
12015 043070 077202
12016
12017 043072 103403
12018 043074 102402
12019 043076 001401
12020 043100 100001
12021
12022 043102 104006
12023
12024
12025
12026
12027
12028 043104
12029 043104 000004
12030 043106 012700 000630
12031 043112 013701 043124
12032 043116 012702 000005
12033 043122 000277
12034
12035 043124 077201
12036
12037 043126 103003
12038 043130 102002
12039 043132 001001
12040 043134 100401
12041
12042 043136 104006
12043
12044
12045
12046
12047 043140
12048 043142 000004
12049 043146 013701 043160
12050 043152 012702 000005
12051 043156 000257
12052
12053 043160 077277
12054
12055 043162 103403
12056 043164 102402
12057 043166 001401
12058 043170 100001
12059

```

```

*****
;TEST 627 SOB TEST, [R] = 1, FLAGS = 0000
*****
TST627:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #627,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #25,R1                            ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R2                              ;SET SOB COUNTER = 1
CCC                                    ;MAKE N:C = 0000

2$: SOB R2,2$-2                      ;TEST THE SOB
BCS 3$                                ;BR IF C = 1
BVS 3$                                ;BR IF V = 1
BEQ 3$                                ;BR IF Z = 1
BPL T$T630                            ;BR IF N = 0

3$: ERROR 6                          ;SOB ALTERED CODES - SET ONE
*****
;TEST 630 SOB TEST, [R] = 5, FLAGS = 1111
*****
TST630:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #630,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #25,R1                            ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #5,R2                              ;SET SOB COUNTER = 5
SCC                                    ;MAKE N:C = 1111

2$: SOB R2,2$                        ;TEST THE SOB
BCC 3$                                ;BR IF C = 0
RVC 3$                                ;BR IF V = 0
BNE 3$                                ;BR IF Z = 0
BMI T$T631                            ;BR IF N = 1

3$: ERROR 6                          ;SOB ALTERED CODES - CLEARED ONE
*****
;TEST 631 SOB TEST, [R] = 5, FLAGS = 0000
*****
TST631:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #631,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #S0B4,R1                          ;GET COPY OF TEST INSTRUCTION WORD
MOV #5,R2                              ;SET SOB COUNTER = 5
CCC                                    ;MAKE N:C = 0000

SOB3: SOB R2,SOB1                    ;TEST THE SOB
SOB4: SOB R2,SOB1                    ;TEST THE SOB
BCS 3$                                ;BR IF C = 1
BVS 3$                                ;BR IF V = 1
BEQ 3$                                ;BR IF Z = 1
BPL T$T632                            ;BR IF N = 0

```

```

12060 043172 104006
12061
12062
12063
12064
12065 043174 000004
12066 043176 012700 000632
12067 043202 013701 043234
12068 043206 010609
12069 043210 010704 001010
12070 043214 012704 177777
12071 043220 010506
12072 043222 012703 043242
12073 043226 012746 177777
12074 043232 000257
12075
12076 043234 000203
12077
12078
12079 043236 104005
12080 043240 000415
12081
12082 043242 100403
12083 043244 001402
12084 043246 102401
12085 043250 103001
12086
12087 043252 104005
12088
12089 043254 020403
12090 043256 001401
12091
12092 043260 104002
12093
12094 043262 020506
12095 043264 001404
12096
12097 043266 010504
12098 043270 010603
12099 043272 104003
12100
12101 043274 100506
12102
12103
12104
12105
12106 043276 000004
12107 043276 012700 000633
12108 043300 013701 043346
12109 043310 012702 177776
12110 043314 010609
12111 043316 010704 001010
12112 043322 010506
12113 043324 012704 000340
12114 043330 012746 000340
12115

```

```

3$: ERROR 6                          ;SOB ALTERED CODES - SET ONE
*****
;TEST 632 RTS TEST - N:C = 0000
*****
TST632:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #632,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #25,R1                            ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R2                              ;SAVE THE SP
PC,@R2                                ;SET ERROR LOOP ADDRESS
MOV #5,SP                              ;R3 SHOULD GET 177777
MOV #4,R3                              ;RESET SP FOR ERROR LOOP
MOV #-1,-(SP)                         ;RTS SHOULD LOAD PC FROM [R3]
CCC                                    ;RTS SHOULD LOAD R3 WITH 177777
;N:C = 0000

2$: RTS R3                            ;TEST THE RTS - GO TO 4$

3$: ERROR 5                          ;RTS FAILED TO LOAD THE PC
BR 10$                                ;GO TO EXIT - SCHOOLS OUT

4$: BMI 5$                            ;N:C = 0000 ?
BEQ 5$
BVS 5$
BCC 6$

5$: ERROR 5                          ;RTS ALTERED CODES - CLEARED ONE

6$: CMP R4,R3                        ;DID R3 GET LOADED FROM STACK ?
BEQ 8$                                ;BR IF YES

7$: ERROR 2                          ;RTS FAILED TO LOAD REG

8$: CMP R5,SP                        ;DID RTS POP THE STACK POINTER ?
BEQ T$T633                            ;BR IF YES

9$: MOV R5,R4                        ;[R4] = S / B SP
MOV SP,R3                             ;[R3] = WAS SP
ERROR 3                                ;RTS FAILED TO POP SP

10$: MOV R5,SP                       ;FIX THE SP
*****
;TEST 633 RTT TEST - N:C = 1111
*****
TST633:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #633,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #25,R1                            ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #5,R2                              ;DEST=PSW FOR 5$ CALL
MOV #1,R3                              ;SAVE THE SP
PC,@R2                                ;SET ERROR LOOP ADDRESS
MOV #5,SP                              ;RESET SP FOR ERROR LOOP
MOV #340,R4                            ;[R4] = S / B PSW AT RTIS POINT
MOV #340,-(SP)                        ;NEW PSW S / B = 340

```

```

12116 043334 012746 043354
12117 043340 005037 177776
12118 043344 000277
12119
12120 043346 000006
12121
12122 043350 104005
12123 043352 000412
12124
12125 043354 013703 177776
12126 043360 020403
12127 043362 001401
12128
12129 043364 104001
12130
12131 043366 020506
12132 043370 001404
12133
12134 043372 010504
12135 043374 010603
12136 043376 104003
12137
12138 043400 010506
12139
12140
12141
12142
12143 043402
12144 043402 000004
12145 043404 012700 000634
12146 043410 013701 043454
12147 043414 012702 177776
12148 043420 010605
12149 043422 010737 001010
12150 043426 010506
12151 043430 012704 000017
12152 043434 012746 000017
12153 043440 012746 043462
12154 043444 012737 000340 177776
12155 043452 000257
12156
12157 043454 000006
12158
12159 043456 104005
12160 043460 000412
12161
12162 043462 013703 177776
12163 043466 020403
12164 043470 001401
12165
12166 043472 104001
12167
12168 043474 020506
12169 043476 001404
12170
12171 043500 010504

```

```

MOV #4$,-(SP) ;NEW PC S / B = 4$
CLR @#PSW ;CLEAR THE PSW
SCC @#PSW ;N:C = 1111
2$: RTT ;TEST THE RTT - GO TO 4$
3$: ERROR 5 ;RTT FAILED TO LOAD THE PC
BR 8$ ;GO TO EXIT - SCHOOL'S OUT
4$: MOV @#PSW,R3 ;SAVE THE PSW
CMP R4,R3 ;WAS PSW = 340 ?
BEQ 6$ ;BR IF IT WAS
5$: ERROR 1 ;RTT FAILED TO LOAD PSW PROPERLY
6$: CMP R5,SP ;DID RTT UPDATE THE SP ?
BEQ TST634 ;;BR IF YES
MOV R5,R4 ;[R4] = S / B SP
MOV SP,R3 ;[R3] = WAS SP
7$: ERROR 3 ;RTT FAILED TO UPDATE SP
8$: MOV R5,SP ;FIX THE SP
;*****
;TEST 634 RTT TEST - N:C = 0000
;*****
TST634: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #634,R0 ;LOAD R0 WITH TEST NUMBER
MOV @2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV @PSW,R2 ;DEST=PSW FOR 5$ CALL
MOV SP,R5 ;SAVE THE SP
MOV R5,@$SLPERR ;SET ERROR LOOP ADDRESS
1$: MOV R5,SP ;RESET SP FOR ERROR LOOP
MOV @1$,R4 ;[R4] = S / B PSW AT THIS POINT
MOV @017,-(SP) ;NEW PC S / B = 017
MOV #4$,-(SP) ;NEW PC S / B = 4$
MOV #340,@#PSW ;MAKE [PSW] = 340
CCC ;N:C = 0000
2$: RTT ;TEST THE RTT - GO TO 4$
3$: ERROR 5 ;RTT FAILED TO LOAD THE PC
BR 8$ ;GO TO EXIT - SCHOOL'S OUT
4$: MOV @#PSW,R3 ;SAVE THE PSW
CMP R4,R3 ;WAS PSW = 017 ?
BEQ 6$ ;BR IF IT WAS
5$: ERROR 1 ;RTT FAILED TO LOAD PSW PROPERLY
6$: CMP R5,SP ;DID RTT UPDATE THE SP ?
BEQ TST635 ;;BR IF YES
MOV R5,R4 ;[R4] = S / B SP

```

```

12172 043502 010603
12173 043504 104003
12174
12175 043506 010506
12176
12177
12178
12179
12180 043510
12181 043510 000004
12182 043512 012700 000635
12183 043516 013701 043542
12184 043522 010604
12185 043524 012704 125252
12186 043530 012705 043572
12187 043534 010437 043556
12188 043540 000257
12189
12190 043542 006405
12191
12192 043544 010637 001074
12193 043550 010206
12194 043552 104005
12195
12196 043554 000444
12197
12198 043556 125252
12199
12200 043560 010637 001074
12201 043564 010206
12202 043566 104005
12203
12204 043570 000436
12205
12206 043572 100403
12207 043574 001402
12208 043576 103401
12209 043600 103011
12210
12211 043602 013703 177776
12212 043606 010637 001074
12213 043612 010206
12214 043614 012704 177776
12215 043620 104004
12216 043622 000421
12217
12218 043624 020627 043560
12219 043630 001406
12220 043632 010603
12221 043634 012704 043560
12222 043640 010206
12223 043642 104003
12224
12225 043644 000410
12226
12227 043646 020504

```

```

7$: MOV SP,R3 ;[R3] = WAS SP
ERROR 3 ;RTT FAILED TO UPDATE SP
8$: MOV R5,SP ;FIX THE SP
;*****
;TEST 635 MARK INSTRUCTION TEST - N:C=0000
;*****
TST635: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #635,R0 ;LOAD R0 WITH TEST NUMBER
MOV @2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R4 ;SAVE SP
MOV #125252,R4 ;[R5] SHOULD BE 125252
MOV #4$,R5 ;MARK GOES TO 4$ VIA [R5]
MOV R4,@#6$ ;INITIALIZE WORD LOADED INTO R5
CCC ;N:C=0000
2$: MARK+5 ;TEST THE MARK
MOV SP,@#SREG5 ;SAVE BAD SP FOR PRINTING
MOV R2,SP ;RESET SP
3$: ERROR 5 ;MARK FAILED TO EXECUTE
BR TST636 ;;GO TO SCOPE EXIT
6$: 125252 ;THIS WORD SHOULD GET LOADED INTO R5
MOV SP,@#SREG5 ;SAVE BAD SP FOR PRINTING
MOV R2,SP ;RESET SP
5$: ERROR 5 ;MARK FAILED TO LOAD RC FROM [R5]
BR TST636 ;;GO TO SCOPE EXIT
4$: BMI 10$ ;N:C=0000?
BEQ 10$
BVS 10$
BCC 8$
10$: MOV @#PSW,R3 ;SAVE FLAGS IN R3
MOV SP,@#SREG5 ;SAVE BAD SP FOR PRINTING
MOV R2,SP ;RESET SP
7$: ERROR @#PSW,R2 ;DEST=PSW
BR TST636 ;;GO TO SCOPE EXIT
8$: CMP SP,#6$+2 ;DID MARK RESET SP?
BEQ 11$ ;BR IF YES
MOV SP,R3 ;PUT BAD SP IN R3
MOV #6$+2,R4 ;[R3] = SP
MOV R2,SP ;RESET SP
9$: ERROR 3 ;MARK FAILED TO RESET SP
BR TST636 ;;GO TO SCOPE EXIT
11$: CMP R5,R4 ;DID MARK RESTORE OLD R5

```

```

12228 043650 001405
12229
12230 043652 010637 001074
12231 043656 010503
12232 043660 010206
12233 043662 104004
12234
12235 043664 010206
12236
12237
12238
12239
12240 043666 000004
12241 043666 012700 000636
12242 043670 013701 043720
12243 043674 013701 043720
12244 043700 010602
12245 043702 012704 125252
12246 043706 012705 043750
12247 043712 010437 043734
12248 043716 000277
12249
12250 043720 006405
12251
12252 043722 010637 001074
12253 043722 010206
12254 043730 104005
12255
12256 043732 000444
12257 043734 125252
12258
12259 043736 010637 001074
12260 043742 010206
12261 043744 104005
12262
12263 043746 000436
12264
12265 043750 100003
12266 043752 001002
12267 043754 102001
12268 043756 103411
12269
12270 043760 013703 177776
12271 043764 010637 001074
12272 043770 010206
12273 043772 012702 177776
12274 043776 104007
12275 044000 000421
12276
12277
12278 044002 020627 043736
12279 044006 001406
12280 044010 010603
12281 044012 012704 043736
12282 044016 010206
12283 044020 104003

```

```

12284
12285 044022 000410
12286
12287 044024 020504
12288 044026 001405
12289
12290 044030 010637 001074
12291 044034 010503
12292 044036 010206
12293 044040 104004
12294
12295 044042 010206
12296
12297
12298
12299
12300 044044 000004
12301 044044 012700 000637
12302 044046 013701 044102
12303 044052 013701 044102
12304 044056 010605
12305 044060 012702 177546
12306 044064 010737 001010
12307
12308 044070 010506
12309 044072 012737 044106 000004
12310 044100 000257
12311
12312 044102 005712
12313
12314 044104 000404
12315
12316 044106 012737 061220 000004
12317 044114 104006
12318
12319 044116 010506
12320 044120 012737 061220 000004
12321
12322
12323
12324
12325 044126 000004
12326 044130 012700 000640
12327 044134 013701 044152
12328 044140 012702 177546
12329 044144 012704 000200
12330 044150 000257
12331
12332 044152 030412
12333
12334 044154 001002
12335
12336 044156 011203
12337 044160 104001
12338
12339

```



```

12340
12341
12342 044162
12343 044162 000004
12344 044164 012700 000641
12345 044170 013701 044206
12346 044174 012702 177546
12347 044204 010605 000200
12348 044204 000257
12349
12350 044206 032712 000100
12351
12352 044212 001402
12353
12354 044214 011203
12355 044216 104001
12356
12357
12358
12359
12360 044220
12361 044220 000004
12362 044222 012700 000642
12363 044226 013701 044276
12364 044232 010605
12365 044234 012702 177546
12366 044240 012704 000300
12367 044244 010737 001010
12368 044250 012737 044312 000100
12369 044256 012737 000340 000102
12370 044264 010606
12371 044266 012737 000340 177776
12372 044274 000257
12373
12374 044276 052712 000100
12375
12376 044302 020412
12377 044304 001402
12378
12379 044306 011203
12380 044310 104001
12381
12382 044312 042737 000102 000100
12383 044320 005037 000102
12384 044324 042712 000100
12385 044330 010506
12386
12387
12388
12389
12390 044332
12391 044332 000004
12392 044334 012700 000643
12393 044334 013701 044406
12394 044344 010605
12395 044346 012702 177546

```

```

*****
;TEST 641 LINE CLOCK TEST - LKCSR BIT 6 CLEAR
*****
TST641:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #641,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #256,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #LKCSR,R2                        ;R2 POINTS TO LKCSR
MOV #200,R4                          ;SCOPE SYNC / B = 200
CCC

2$: BIT #100,(R2)                    ;TEST BIT 6 IN LKCSR
BEQ TST642                          ;;BR IF CLEAR

3$: MOV (R2),R3                      ;GET WAS DATA
ERROR 1                              ;BIT 6 (INTR. ENAB.) IN LKCSR WAS SET

*****
;TEST 642 LINE CLOCK TEST - LKCSR BIT 6 SET
*****
TST642:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #642,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #256,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #LKCSR,R2                        ;SAVE SP
MOV #LKCSR,R2                        ;R2 POINTS TO LKCSR
MOV #300,R4                          ;[LKCSR]S / B = 300
MOV PC,@#SLPERR                      ;SET ERROR LOOP ADDRESS
MOV #436,@#100                       ;SET UP LCLK VECTOR IN CASE LOGIC
MOV #340,@#102                       ;FAULT CAUSES ALL INTERRUPT
MOV R5,SP                             ;RESET SP FOR ERROR LOOP
MOV #340,@#PSW                       ;SET PRIORITY TO LEVEL 7
CCC

2$: BIS #100,(R2)                    ;SET BIT 6 IN LKCSR
CMP R4,(R2)                          ;RESULT CORRECT?
BEQ 4$                               ;BR IF YES

3$: MOV (R2),R3                      ;GET WAS DATA
ERROR 1                              ;BIT 6 FAILED TO SET IN LKCSR

4$: BIC #102,@#100                   ;RESTORE TRAP CATCHER IN LINE CLOCK VECTOR
CLR #102                              ;SET SP FOR ERROR LOOP
BIC #100,(R2)                        ;TURN OFF LINE CLK INTR. ENAB.
MOV R5,SP                             ;RESET SP

*****
;TEST 643 LINE CLK BASIC INTERRUPT TEST
*****
TST643:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #643,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #256,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #LKCSR,R2                        ;R2 POINTS TO LKCSR

```

```

12396 044352 010737 001010
12397 044356 010506
12398 044360 005004
12399 044362 012737
12400 044370 012737 044424 000100
12401 044376 005012 000340 000102
12402 044400 005037 177776
12403 044404 000257
12404
12405 044406 052712 000100
12406
12407 044412 005304
12408 044414 001376
12409
12410 044416 042712 000100
12411 044422 104006
12412
12413 044424 042712 000100
12414 044430 012737 000102 000100
12415 044436 005037 000102
12416 044442 010506
12417 044444 005037 177776
12418
12419
12420
12421
12422 044450
12423 044450 000004
12424 044452 012700 000644
12425 044456 013701 044510
12426 044462 012737 000001 001110
12427 044470 012702 177564
12428 044474 012737 000340 177776
12429 044502 052712 000004
12430 044506 000277
12431
12432 044510 000005
12433
12434 044512 013705 177776
12435 044516 032712 000004
12436 044522 001403
12437
12438 044524 042712 000004
12439 044530 104006
12440
12441 044532 022705 000357
12442 044536 001406
12443
12444 044540 012704 000357
12445 044544 012503 177776
12446 044552 104001
12447
12448 044554 005037 177776
12449 044560 042737 000004 177564

```

```

*****
;TEST 643 LINE CLK BASIC INTERRUPT TEST
*****
T643:
MOV PC,@#SLPERR                      ;SET ERROR LOOP ADDRESS
MOV R5,SP                             ;RESET SP FOR ERROR LOOP
CLR R4                                ;INITIALIZE TIMER
MOV #4$,@#100                        ;SET UP LINE CLOCK VECTOR TO T
MOV #340,@#102                       ;R0 4$ WITH PROCESSOR PRIORITY = 7
CLR (R2)                             ;CLEAR LKCSR
CLR @#PSW                             ;SET PRIORITY TO LEVEL 000
CCC

2$: BIS #100,(R2)                    ;ENABLE LINE CLK INTERRUPT
DEC R4                                ;WAIT FOR INTR - REPORT ERROR IF
BNE -2                                ;R4 GOES TO 000000

3$: BIC #100,(R2)                    ;TURN OFF INTR. ENAB.
ERROR 6                              ;LINE CLK FAILED TO INTERRUPT

4$: BIC #100,(R2)                    ;TURN OFF INTR. ENAB.
MOV #102,@#100                       ;RESTORE TRAP CATCHER IN LINE CLK VECTOR
CLR #102                              ;SET SP FOR ERROR LOOP
MOV R5,SP                             ;RESET SP
CLR @#PSW                             ;RESET PRIORITY TO LEVEL 0

*****
;TEST 644 RESET TEST - N:IC = 1111
*****
TST644:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #644,R0                          ;LOAD R0 WITH TEST NUMBER
MOV #256,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,@#TIMES                       ;NO ITERATIONS ON THIS TEST
MOV #LKCSR,R2                        ;R2 POINTS TO DL11 XCSR
MOV #340,@#PSW                       ;MAKE PRTY. BITS ALL 1'S
BIS #4,(R2)                          ;SET THE DL11 MAINT. BIT
N:C = 1111

2$: RESET                            ;TEST THE RESET - IT SHOULD CLEAR THE DL11 MAINT BIT

MOV @#PSW,R5                          ;SAVE THE PSW
BIT #4,(R2)                          ;DID MAINT. BIT CLEAR ??
BEQ 4$                               ;BR IF YES

3$: BIC #4,(R2)                      ;MAKE SURE TO TURN OFF MAINT. BIT
ERROR 6                              ;RESET FAILED TO CLEAR MAINT BIT

4$: CMP #357,R5                      ;DID RESET ALTER THE PSW ??
BEQ 6$                               ;BR IF NOT

MOV #357,R4                          ;[R4] = S/B PSW
MOV R5,R3                             ;[R3] = WAS PSW
MOV #PSW,R2                          ;[R2] = PSW
BIC 1                                 ;RESET ALTERED THE PSW

5$: ERROR 1                          ;RESET ALTERED THE PSW

6$: CLR @#PSW                        ;CLEAR OUT THE PSW
BIC #4,@#XCSR                        ;MAKE SURE MAINT BIT IS OFF

```



```

12564 045166 005037 177776 5$: CLR @#PSW ;SET PSW PRTY BITS TO LEVEL 0
12565 045174 000257 ;N:C=0000
12566 045174 152712 000100 ;ENAB. DL11 INTR - N:C=1000
12567
12568 045200 000001 2$: WAIT ;TEST THE WAIT-GO TO 4$ ON INTR
12569
12570 045202 012737 000340 177776 MOV #340,@#PSW ;LOCK OUT INTR
12571 045210 005012 ;TURN OFF DL11 INTR ENAB
12572 104006 ;WAIT FAILED TO EXECUTE PROPERLY
12573 045214 000424 BR 8$ ;GO EXIT THIS TEST
12574
12575 045216 042712 000100 4$: BIC #100,(R2) ;TURN OFF DL11 INTR ENAB
12576 045222 022716 045202 CMP #2$+2,(SP) ;DID WAIT GET FETCHED ??
12577 045226 001402 BEQ 6$ ;BR IF YES
12578
12579 045230 104006 ERROR 6 ;WAIT NOT FETCHED PROPERLY
12580 045232 000415 BR 8$ ;GO EXIT THIS TEST
12581
12582 045234 022766 000010 000002 6$: CMP #010,2(SP) ;DID "WAIT" ALTER THE PSW ??
12583 045242 001411 BEQ 8$ ;BR IF NO
12584
12585 045244 012704 000010 MOV #010,R4 ;[R4] = S/R PSW
12586 045250 016603 000002 MOV 2(SP),R3 ;[R3] = WAS PSW
12587 045254 012702 177776 MOV #PSW,R2 ;DEST = PSW
12588 045260 104001 7$: ERROR 1 ;"WAIT" ALTERED THE PSW
12589 045262 000401 BR 8$ ;GOT TO EXIT TEST
12590
12591 045264 104006 9$: ERROR 6 ;DL11 FAILED TO SET READY ON TIME
12592
12593 045266 010506 8$: MOV R5,SP ;RESET THE SP
12594 045270 005037 177776 CLK @#PSW ;CLEAR OUT THE PSW
12595 045274 005012 ;TURN OFF DL11 INTR.
12596 045276 012737 000066 000064 MOV #6,@#64 ;RESTORE DL11 VECTOR WITH TRAPCATCHER
12597 045304 005037 000066 CLR @#66
12598
12599 *****
12600 ;TEST 650 BR PRIORITY ARBITRATION TEST - LEVEL 1 USING LINE CLK
12601 *****
12602 045310 000004 1$T650: SCOPE ;CALL THE SCOPE LOOP UTILITY
12603 045310 012700 000650 MOV #650,R0 ;LOAD R0 WITH TEST NUMBER
12604 045316 013701 045364 MOV #2$R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
12605 045322 010605 MOV SP,R5 ;SAVE THE SP
12606 045324 010737 001010 PC,@#SLPERR ;SET ERROR LOOP ADDRESS
12607 045330 012702 177546 MOV #LCKCR,R2 ;R2 POINTS TO LINE CLK CSR
12608 045334 012702 045402 000100 MOV #4$,@#100 ;IF INTR OCCURS - GO TO 4$
12609 045342 012737 000340 000102 MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
12610 045350 010506 MOV R5,SP ;RESET SP FOR ERROR LOOPING
12611 045352 005004 CLR R4 ;INITIALIZE R4 AS TIMER
12612 045354 012737 000040 177776 MOV #40,@#PSW ;SET CPU PRIORITY TO LEVEL 1
12613 045362 000257 CCC ;SCOPE SYNC
12614
12615 045364 052712 000100 2$: BIS #100,(R2) ;ENABLE LINE CLK INTERRUPTS
12616
12617 045370 005304 DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
12618 045372 001376 BNE -2 ;TIMER FROM GETTING BACK TO 000000
12619

```

```

12620
12621 045374 042712 000100 3$: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
12622 045400 104006 ERROR 6 ;LINE CLK FAILED TO INTR AT LEVEL 1
12623
12624 045402 042712 000100 4$: BIC #100,(R2) ;TURN OFF INTR. ENABLE

```

```

12625 045406 012737 000102 000100 MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
12626 045414 005037 000102 CLR #102 ;
12627 045420 010506 MOV #5,SP ;RESET THE SP
12628 045422 005037 177776 CLR #5,SP ;SET CPU PRIORITY BACK TO LEVEL 0
;*****
;TEST 651 BR PRIORITY ARBITRATION TEST - LEVEL 2 USING LINE CLK
;*****
TST651:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #651,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
PC,##SLPERR ;SET ERROR LOOP ADDRESS
1S: MOV #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
MOV #48,@#100 ;IF INTR OCCURS - GO TO 4S
MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
MOV R5,SP ;RESET SP FOR ERROR LOOPING
CLR R4 ;INITIALIZE R4 AS TIMER
MOV #100,@#PSW ;SET CPU PRIORITY TO LEVEL 2
CCC ;SCOPE SYNC
2S: BIS #100,(R2) ;ENABLE LINE CLK INTERRUPTS
DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
BNE -2 ;TIMER FROM GETTING BACK TO 000000
;TURN OFF THE INTERRUPT ENABLE
3S: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
ERROR 6 ;LINE CLK FAILED TO INTR AT LEVEL 2
;TURN OFF INTR. ENABLE
4S: BIC #100,(R2) ;TURN OFF INTR. ENABLE
MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
CLR #5,SP ;RESET THE SP
MOV #5,SP ;SET CPU PRIORITY BACK TO LEVEL 0
CCC
;*****
;TEST 652 BR PRIORITY ARBITRATION TEST - LEVEL 3 USING LINE CLK
;*****
TST652:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #652,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
PC,##SLPERR ;SET ERROR LOOP ADDRESS
1S: MOV #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
MOV #48,@#100 ;IF INTR OCCURS - GO TO 4S
MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
MOV R5,SP ;RESET SP FOR ERROR LOOPING
CLR R4 ;INITIALIZE R4 AS TIMER
MOV #140,@#PSW ;SET CPU PRIORITY TO LEVEL 3
CCC ;SCOPE SYNC
2S: BIS #100,(R2) ;ENABLE LINE CLK INTERRUPTS
DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT

```

```

12681 045626 001376 BNE -2 ;TIMER FROM GETTING BACK TO 000000
12682 045630 042712 000100 BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
12683 045634 104006 000100 3S: ERROR 6 ;LINE CLK FAILED TO INTR AT LEVEL 3
;TURN OFF INTR. ENABLE
12685 045636 042712 000100 4S: BIC #100,(R2) ;TURN OFF INTR. ENABLE
12686 045642 012737 000102 MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
12687 045650 005037 000102 CLR #5,SP ;RESET THE SP
12688 045654 010506 MOV #5,SP ;SET CPU PRIORITY BACK TO LEVEL 0
12689 045656 005037 177776 CLR #5,SP ;SET CPU PRIORITY BACK TO LEVEL 0
;*****
;TEST 653 BR PRIORITY ARBITRATION TEST - LEVEL 4 USING LINE CLK
;*****
TST653:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #653,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
PC,##SLPERR ;SET ERROR LOOP ADDRESS
1S: MOV #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
MOV #48,@#100 ;IF INTR OCCURS - GO TO 4S
MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
MOV R5,SP ;RESET SP FOR ERROR LOOPING
CLR R4 ;INITIALIZE R4 AS TIMER
MOV #200,@#PSW ;SET CPU PRIORITY TO LEVEL 4
CCC ;SCOPE SYNC
2S: BIS #100,(R2) ;ENABLE LINE CLK INTERRUPTS
DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
BNE -2 ;TIMER FROM GETTING BACK TO 000000
;TURN OFF THE INTERRUPT ENABLE
3S: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
ERROR 6 ;LINE CLK FAILED TO INTR AT LEVEL 4
;TURN OFF INTR. ENABLE
4S: BIC #100,(R2) ;TURN OFF INTR. ENABLE
MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
CLR #102 ;
MOV #5,SP ;RESET THE SP
MOV #5,SP ;SET CPU PRIORITY BACK TO LEVEL 0
CCC
;*****
;TEST 654 BR PRIORITY ARBITRATION TEST - LEVEL 5 USING LINE CLK
;*****
TST654:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #654,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
PC,##SLPERR ;SET ERROR LOOP ADDRESS
1S: MOV #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
MOV #48,@#100 ;IF INTR OCCURS - GO TO 4S
MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
MOV R5,SP ;RESET SP FOR ERROR LOOPING
CLR R4 ;INITIALIZE R4 AS TIMER

```

```

12737 046044 012737 000240 177776      MOV      #240,@#PSW      ;SET CPU PRIORITY TO LEVEL 5
12738 048052 000257          CCC                      ;SCOPE SYNC
12739
12740 046054 052712 000100      2$:     BIS      #100,(R2) ;ENABLE LINE CLK INTERRUPTS
12741
12742 046060 005304          DEC      R4              ;COUNT THE TIMER - LCLK SHOULD PREVENT
12743 048062 001376          BNE     #-2              ;TIMER FROM GETTING BACK TO 000000
12744
12745 046064 042712 000100      3$:     BIC      #100,(R2)   ;TURN OFF THE INTERRUPT ENABLE
12746 046070 104006          ERROR   6                ;LINE CLK FAILED TO INTR AT LEVEL 5
12747
12748 046072 042712 000100      4$:     BIC      #100,(R2)   ;TURN OFF INTR. ENABLE
12749 046076 012737 000102      MOV      #102,@#100     ;RESTORE TRAP CATCHER IN THE VECTOR
12750 046104 005037 000102      CLR      #102           ;
12751 046110 010506          MOV      R5,SP          ;RESET THE SP
12752 046112 005037 177776      CLR      @#PSW          ;SET CPU PRIORITY BACK TO LEVEL 0
12753
;*****
;TEST 655 BR PRIORITY ARBITRATION TEST - LEVEL 6 USING LINE CLK
;*****
12754 046116 000004          SCOPE                   ;CALL THE SCOPE LOOP UTILITY
12755 046120 012700          MOV      #655,R0        ;LOAD R0 WITH TEST NUMBER
12756 046124 013701          MOV      @R2,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
12757
12758 046130 032737 020000 063234 .SBTTL  USER CONTROLLED BREAKPOINT - BIT13
12759 046136 001401          BIT      #BIT13,@#BPTLOC ;BREAKPOINT HALT SET ??
12760 046140 000000          BRQ     .+4             ;BR IF NOT
12761 046142 010605          HALT                    ;BREAK-DEPRESS CONTINUE TO CONTINUE
12762 046144 010737          MOV      PC,@#SLPERR    ;SAVE THE SP
12763 046146 012702 001010      MOV      SP,R5          ;SET ERROR LOOP ADDRESS
12764 046148 177546          MOV      #LCKCSR,R2    ;R2 POINTS TO LINE CLK CSR
12765 046150 046216 000100      MOV      #45,@#100     ;IF INTR OCCURS - GO TO 4$
12766 046152 012737 000340 000102      MOV      #340,@#102    ;WITH CPU PRIORITY AT LEVEL 7
12767 046154 010506          MOV      R5,SP          ;RESET SP FOR ERROR LOOP
12768 046156 012737 000300 177776      CLR      R4              ;INITIALIZE R4 AS TIMER
12769 046158 012737 000300 177776      MOV      #300,@#PSW    ;SET CPU PRIORITY TO LEVEL 6
12770 046160 000257          CCC                      ;SCOPE SYNC
12771
12772 046204 052712 000100      2$:     BIS      #100,(R2)   ;ENABLE INTERRUPTS
12773
12774 046210 005304          DEC      R4              ;COUNT UNTIL [R4] = 000000 - THEN
12775 046212 001376          BNE     #-2              ;CONTINUE - NO INTERRUPT SHOULD OCCUR
12776 046214 000403          BR      6$              ;GO TO EXIT - ALL OK
12777
12778 046216 042712 000100      4$:     BIC      #100,(R2)   ;TURN OFF THE INTR ENABLE
12779 046222 104006          ERROR   6                ;INTR OCCURRED WITH CPU AT LEVEL 6
12780
12781 046224 042712 000100      6$:     BIC      #100,(R2)   ;TURN OFF INTR ENABLE
12782 046230 012737 000102      MOV      #102,@#100     ;RESET THE TRAP CATCHER IN THE VECTOR
12783 046236 005037 000102      CLR      #102           ;
12784 046242 010506          MOV      R5,SP          ;RESET SP JUST IN CASE
12785 046244 005037 177776      CLR      @#PSW          ;SET CPU PRIORITY BACK TO LEVEL 0
12786
;*****
;TEST 656 BR PRIORITY ARBITRATION TEST - LEVEL 7 USING DL11
;*****
12790
12791
12792

```

```

12793 046250 000004          SCOPE                   ;CALL THE SCOPE LOOP UTILITY
12794 046256 012700          MOV      #656,R0        ;LOAD R0 WITH TEST NUMBER
12795 046258 013701          MOV      @R2,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
12796 046262 010605          MOV      SP,R5          ;SAVE THE SP
12797 046264 010737          MOV      PC,@#SLPERR    ;SET ERROR LOOP ADDRESS
12798 046266 012702 001010      MOV      #LCKCSR,R2    ;R2 POINTS TO DL11 KCSR
12799 046270 177546          MOV      #45,@#64      ;IF INTR OCCURS - GO TO 4$
12800 046274 012737 000064 000066      MOV      #340,@#66     ;WITH CPU PRIORITY AT LEVEL 7
12801 046278 010506          MOV      R5,SP          ;RESET SP FOR ERROR LOOP
12802 046310 005004          CLR      R4              ;INITIALIZE R4 AS TIMER
12803 046312 005004          MOV      #340,@#PSW    ;SET CPU PRIORITY TO LEVEL 7
12804 046314 012737 000340 177776      MOV      @#PSW          ;SCOPE SYNC
12805 046322 000257          CCC                      ;
12806
12807 046324 052712 000100      2$:     BIS      #100,(R2)   ;ENABLE INTERRUPTS
12808
12809 046330 005304          DEC      R4              ;COUNT UNTIL [R4] = 000000 - THEN
12810 046332 001376          BNE     #-2              ;CONTINUE - NO INTERRUPT SHOULD OCCUR
12811 046334 000403          BR      6$              ;GO TO EXIT - ALL OK
12812
12813 046336 042712 000100      4$:     BIC      #100,(R2)   ;TURN OFF THE INTR ENABLE
12814 046342 104006          ERROR   6                ;INTR OCCURRED WITH CPU AT LEVEL 7
12815
12816 046344 042712 000100      6$:     BIC      #100,(R2)   ;TURN OFF INTR ENABLE
12817 046350 012737 000066 000064      MOV      #66,@#64      ;RESET THE TRAP CATCHER IN THE VECTOR
12818 046356 005037 000066      CLR      #66           ;
12819 046362 010506          MOV      R5,SP          ;RESET SP JUST IN CASE
12820 046364 005037 177776      CLR      @#PSW          ;SET CPU PRIORITY BACK TO LEVEL 0
12821
;*****
;TEST 657 "CLR @#PSW" ALLOWS IMMEDIATE BR-BG-INTR SEQUENCE
;THIS TEST VERIFIES THAT IF A "BR" REQUEST IS PENDING WHEN A "CLR @#PSW"
;IS EXECUTED TO LOWER THE CPU PRIORITY THE REQUEST IS GRANTED BEFORE
;EXECUTION OF THE INSTRUCTION FOLLOWING THE "CLR @#PSW"
;*****
12822
12823
12824
12825
12826
12827
12828 046370 000004          SCOPE                   ;CALL THE SCOPE LOOP UTILITY
12829 046376 012700          MOV      #657,R0        ;LOAD R0 WITH TEST NUMBER
12830 046378 013701          MOV      @R2,R1         ;LOAD R1 WITH TEST INSTRUCTION WORD
12831 046382 012702 001010      MOV      #LCKCSR,R2    ;R2 POINTS TO LINE CLK CSR
12832 046402 177546          MOV      SP,R5          ;SAVE THE SP
12833 046406 010605          MOV      PC,@#SLPERR    ;SET ERROR LOOP ADDRESS
12834 046410 010737 001010      MOV      #45,@#100     ;SET UP LCLK VECTOR TO GO TO 4$
12835 046414 012737 000300 000100      MOV      #300,@#102    ;SET UP LCLK VECTOR TO GO TO 4$
12836 046418 010506          MOV      R5,SP          ;SET UP LCLK VECTOR TO GO TO 4$
12837 046422 005004          CLR      R4              ;RESET THE SP FOR ERROR LOOPING
12838 046424 005003          CLR      R3              ;INITIALIZE TIMER FOR N
12839 046426 005003          CLR      R4              ;CLEAR SOFTWARE FLAG
12840 046430 012737 000340 177776      MOV      #340,@#PSW    ;LOCK OUT ALL INTRs
12841 046434 052712 000100      BIS      #100,(R2)     ;ENABLE LCLK INTRs
12842 046438 000200 000200      TSTB   (R2)            ;CHECK LINE CLOCK READY
12843 046442 105715          BMI     12$             ;LCLK READY TO INTR ??
12844 046446 100403          BMI     12$             ;BR IF YES
12845 046450 005304          DEC      R4              ;COUNT THE TIMER
12846 046454 000374          BNE     11$            ;BR IF NO TIMEOUT
12847 046458 000374          BR      11$            ;GO REPORT TIMEOUT
12848 046462 000257          CCC                      ;SCOPE SYNC

```

```

12849 046470 005037 177776 2$: CLR @#PSW ;ALLOW INTRs - LCLK SHOULD INTERRUPT
12850 ;BEFORE FETCHING NEXT INSTRUCTION
12851 COM R3 ;SHOULD NOT BE FETCHED
12852 046474 005103 4$: CLR (R2) ;DISABLE THE LCLK INTR
12853 046476 005012 ;DID SOFTWARE FLAG GET SET ??
12854 046500 005703 ;BR IF NOT - IT WORKED OK
12855 046502 001404 ;LCLK FAILED TO INTR ONTIME
12856 046504 104006 3$: BRQ 8$ ;GO EXIT
12857 046506 000402 ;
12858 046510 005012 6$: CLR (R2) ;DISABLE LCLK INTR
12859 046512 104006 5$: ERROR 6 ;LINE CLK TIMED OUT
12860 ;
12861 046514 010506 8$: MOV R5,SP ;RESET THE SP
12862 046516 012737 000102 000100 CLR #102,@#100 ;RESTORE THE LINE CLK TRAPCATCHER
12863 046524 005037 000102 CLR #102
;*****
;TEST 660 "BR6 VS BR4" PRIORITY ARBITRATION TEST
;THIS TEST VERIFIES THAT IF BOTH A "BR4" AND A "BR6" REQUEST ARE
;PENDING WHEN THE CPU PRIORITY IS LOWERED TO ALLOW INTRs. THAT "BR6"
;REQUEST IS GRANTED FIRST EVEN THOUGH THE "BR4" REQUEST MAY HAVE
;OCCURRED FIRST
;*****
12873 046530 000004 1$T660: SCOPE ;CALL THE SCOPE LOOP UTILITY
12874 046530 000004 MOV #660,R0 ;LOAD R0 WITH TEST NUMBER
12875 046532 012700 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
12876 046536 013701 MOV SP,R5 ;SAVE SP
12877 046542 010437 MOV PC,@SLPERR ;SET ERROR LOOP ADDRESS
12878 046544 012702 MOV #LCCSR,R2 ;R2 POINTS TO LINE CLK CSR
12879 046550 012702 MOV #XCSR,R3 ;R3 POINTS TO DL11 XCSR
12880 046554 012703 MOV #4$,@#100 ;SET UP THE LCLK VECTOR - GO TO 4$
12881 046560 012737 000300 000102 MOV #300,@#102
12882 046566 012737 046740 000064 MOV #206,@#66 ;SET UP THE DL11 VECTOR - GO TO 8$
12883 046574 012737 000200 MOV R5,S6
12884 046610 010506 MOV #340,@#PSW ;RESET SP FOR ERROR LOOPING
12885 046612 012737 000340 177776 CLR @#MBOF0 ;LOCK OUT ALL INTRs
12886 046620 005037 063312 CLR @#MBOF1 ;INIT TIMER
12887 046624 005037 063316 CLR R4 ;CLEAR DL11 INTR FLAG
12888 046630 005004 CLR R4 ;INIT TIMER
12889 046636 052713 000100 BIT #100,(R3) ;ENABLE DL11 XMIT INTR
12890 046642 105713 11$: TSTB (R3) ;XMIT READY SET ??
12891 046640 100403 BMI 12$ ;BR IF YES
12892 046642 005304 DEC R4 ;COUNT THE TIMER
12893 046644 001374 BNE 11$ ;BR IF NO TIMEOUT
12894 046646 000443 BR 5$ ;GO REPORT TIMEOUT FOR DL11
12895 ;
12896 046650 005004 12$: CLR R4 ;INIT THE TIMER AGAIN
12897 046652 052712 BIT #100,(R2) ;ENABLE LCLK INTRs
12898 046656 042712 000200 BIC #200,(R2) ;CLEAR THE LINE CLOCK READY BIT
12899 046666 105712 13$: TSTB (R2) ;LCLK READY TO INTR
12900 046674 100403 BMI 14$ ;BR IF YES
12901 046676 005304 DEC R4 ;COUNT THE TIMER
12902 046678 001374 BNE 13$ ;BR IF NO TIMEOUT
12903 046672 000436 BR 7$ ;GO REPORT LINE CLK TIMEOUT

```

```

12905 046674 000257 14$: CCC ;SCOPE SYNC
12906 ;
12907 046676 005037 177776 2$: CLR @#PSW ;ALLOW INTRs - KW SHOULD INTR FIRST
12908 ;
12909 046702 005137 063312 4$: COM @#MBOF0 ;SET SOFTWARE FLAG IF FETCHED
12910 046706 005013 CLR (R3) ;DISABLE BOTH INTERRUPTS
12911 046710 005012 CLR (R2)
12912 046712 005737 063312 TST @#MBOF0 ;DID SOFTWARE FLAG GET SET ??
12913 046716 001402 BRQ 6$ ;BR IF NOT
12914 ;
12915 046720 104006 3$: ERROR 6 ;LINE CLK INTR OCCURRED TOO LATE
12916 046722 000425 BR 9$ ;GO TO EXIT
12917 ;
12918 046724 005737 063316 6$: TST @#MBOF1 ;DID DL11 SOFTWARE FLAG SET ??
12919 046730 001422 BRQ 9$ ;BR IF NOT
12920 ;
12921 046732 010302 8$: MOV R3,R2 ;FOR CORRECT DESTINATION TYP0UT
12922 046734 104006 ERROR 6 ;DL11 INTERRUPTED THE KW11
12923 046736 000417 BR 9$ ;GO TO EXIT TEST
12924 ;
12925 046740 005137 063316 8$: COM @#MBOF1 ;FLAG THE DL11 INTR
12926 046744 005013 CLR (R3) ;DISABLE BOTH INTR ENABLES
12927 046746 005012 CLR (R2)
12928 046750 010302 MOV R3,R2 ;FOR CORRECT DESTINATION TYP0UT
12929 046752 104006 ERROR 6 ;DL11 SHOULD NOT HAVE INTERRUPTED
12930 046754 000410 BR 9$ ;GO EXIT TEST
12931 ;
12932 046756 005012 5$: CLR (R2) ;DISABLE THE INTR ENABLES
12933 046760 005013 CLR (R3)
12934 046762 010302 MOV R3,R2 ;FOR CORRECT DESTINATION TYP0UT
12935 046764 104006 ERROR 6 ;DL11 TIMEOUT
12936 046766 000403 BR 9$ ;GO TO EXIT
12937 ;
12938 046770 005012 7$: CLR (R2) ;DISABLE INTR ENABLES
12939 046772 005013 CLR (R3)
12940 046774 104006 ERROR 6 ;KW11 TIMEOUT
12941 ;
12942 046776 010506 9$: MOV R5,SP ;RESET THE SP
12943 047000 @#PSW ;RESET THE CPU PRIORITY
12944 047004 012737 000102 000100 MOV #102,@#100 ;RESTORE LCLK VECTOR
12945 047012 005037 000102 CLR #102
12946 047016 012737 000066 000064 MOV #66,@#64 ;RESTORE THE DL11 XMIT VECTOR
12947 047024 005037 000066 CLR #66
;*****
; ///////////////COMBINED INSTRUCTION EXERCISER SECTION ///////////////
;*****
;*****
;TEST 661 "BPT" TRAP LINKAGE TEST
;*****
12956 047030 000004 1$T661: SCOPE ;CALL THE SCOPE LOOP UTILITY
12957 047030 012700 MOV #661,R0 ;LOAD R0 WITH TEST NUMBER
12958 047032 013701 MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
12959 047036 010605 MOV SP,R5 ;SAVE THE SP

```

```

12961 047044 010737 001010
12962 047050 012737 047066 000014 1S: MOV PC,##SLPERR ;SET ERROR LOOP ADDRESS
12963 047056 010506 ;GO TO 4S ON "BPT" TRAP
12964 047060 000257 ;RESET THE SP FOR ERROR LOOPING
12965 ;SCOPE SYNC
12966 047062 000003 2S: BPT ;TEST THE "BPT" - GO TO 4S
12967 047064 104005 3S: ERROR 5 ;BPT FAILED TO TRAP
12968 047066 010506 4S: MOV R5,SP ;RESET THE SP
12969 047070 012737 000016 000014 MOV #16,@#14 ;RESTORE THE VECTOR
12970
12971
12972
12973
12974
12975
12976 047076 000004 ;*****
12977 047076 000004 ;TEST 662 RED ZONE OVERFLOW TEST - MOV R,-(SP)
12978 047100 012700 000662 ;*****
12979 047104 013701 047144 ;*****
12980 047110 010605 ;*****
12981 047112 013704 000004 ;*****
12982 047116 013703 000336 ;*****
12983 047122 012737 047162 000004 ;*****
12984 047130 012737 125252 000336 ;*****
12985 047136 012706 000340 ;*****
12986 047142 000257 ;*****
12987 ;*****
12988 047144 010046 2S: MOV R0,-(SP) ;FORCE RED ZONE TRAP - GO TO 4S
12989 ;*****
12990 047146 010437 000004 ;*****
12991 047152 010637 001074 ;*****
12992 047156 010506 ;*****
12993 047160 104005 3S: ERROR 5 ;MOV FAILED TO CAUSE TRAP
12994 ;*****
12995 047162 010437 000004 ;*****
12996 047166 022706 000000 4S: MOV R4,@#4 ;RESTORE T.O. VECTOR
12997 047172 001404 BEQ #0,SP ;[SP]=0?
12998 ;BE IF YES
12999 047174 010637 001074 ;*****
13000 047200 010506 ;*****
13001 047202 104005 5S: ERROR 5 ;SAVE BAD SP FOR PRINTING
13002 ;RESET SP FOR ERROR CALL
13003 ;SP NOT BEING JAMMED TO 4
13004 047204 022737 125252 000336 6S: CMP #125252,@#336 ;DID PUSH OCCUR IN YELLOW ZONE?
13005 047212 001404 BEQ #5 ;BR IF NOT
13006 047214 010637 001074 ;*****
13007 047220 010506 ;*****
13008 047222 104005 7S: ERROR 5 ;SAVE BAD SP FOR PRINTING
13009 ;RESET SP FOR ERROR CALL
13010 ;MOV PUSHED INTO YELLOW ZONE
13011 047224 010337 000336 8S: MOV R3,@#336 ;RESTORE VECTOR 336
13012 047230 010506 MOV R5,SP ;RESET SP
13013 ;*****
13014 ;TEST 663 YELLOW ZONE OVERFLOW TEST - MOV R,-(SP)
13015 ;*****
13016 047232 ;*****
13017
13018
13019
13020
13021
13022
13023
13024
13025
13026
13027
13028
13029
13030
13031
13032
13033
13034
13035
13036
13037
13038
13039
13040
13041
13042
13043
13044
13045
13046
13047
13048
13049
13050
13051
13052
13053
13054
13055
13056
13057
13058
13059
13060
13061
13062
13063
13064
13065
13066
13067
13068
13069
13070
13071
13072

```

```

13017 047232 000004 ;*****
13018 047234 012700 000663 ;*****
13019 047240 013701 047272 ;*****
13020 047244 010605 ;*****
13021 047246 012702 000376 ;*****
13022 047252 013704 000004 ;*****
13023 047256 012737 047310 000004 ;*****
13024 047264 012706 000400 ;*****
13025 047270 000257 ;*****
13026 ;*****
13027 047272 010046 2S: MOV R0,-(SP) ;FORCE STACK OVFLW - GO TO 4S
13028 ;*****
13029 047274 010437 000004 ;*****
13030 047300 010637 001074 ;*****
13031 047304 010506 ;*****
13032 047306 104005 3S: ERROR 5 ;STACK OVFLW FAILED TO TRAP
13033 ;*****
13034 047310 010437 000004 ;*****
13035 047314 020012 4S: MOV R4,@#4 ;RESTORE T.O. VECTOR
13036 047316 001404 BEQ #0,(R2) ;DID CR0 GET PUSHED?
13037 ;BR IF YES
13038 047320 010637 001074 ;*****
13039 047324 010506 ;*****
13040 047326 104005 5S: ERROR 5 ;SAVE BAD SP FOR PRINTING
13041 ;RESET SP FOR ERROR CALL
13042 ;MOV FAILED TO PUSH IN YELLOW ZONE
13043 047330 005706 6S: TST SP ;[SP]=0?
13044 047332 001004 BNE #0 ;BR IF NOT
13045 047334 010637 001074 ;*****
13046 047340 010506 ;*****
13047 047342 104005 7S: ERROR 5 ;SAVE BAD SP FOR PRINTING
13048 ;RESET SP FOR ERROR CALL
13049 ;RED ZONE INSTEAD OF YELLOW ZONE
13050 047344 010506 8S: MOV R5,SP ;RESET SP
13051 ;*****
13052 ;TEST 664 YELLOW ZONE OVERFLOW TEST - (CMP R0,-(SP))
13053 ;*****
13054 047346 000004 ;*****
13055 047346 000004 ;*****
13056 047350 012700 000664 ;*****
13057 047354 013701 047402 ;*****
13058 047360 010605 ;*****
13059 047362 013704 000004 ;*****
13060 047366 012737 047406 000004 ;*****
13061 047374 013706 000400 ;*****
13062 047400 000257 ;*****
13063 ;*****
13064 047402 020046 2S: CMP R0,-(SP) ;TEST THE CMP - NO TRAP SHOULD OCCUR
13065 ;*****
13066 047404 000406 BR #6 ;GO TO EXIT TEST
13067 ;*****
13068 047406 010437 000004 4S: MOV R4,@#4 ;RESTORE TRAP VECTOR
13069 047412 010637 001074 ;*****
13070 047416 010506 ;*****
13071 047420 104005 3S: ERROR 5 ;SAVE BAD SP FOR PRINTING
13072 ;RESET THE SP
13073 ;CMP CAUSED OVERFLOW TRAP

```

```

13073 047422 010437 000004 6$: MOV R4,@#4 ;RESTORE THE VECTOR
13074 047426 010506 ;R5,SP ;RESET THE SP
13075
13076
13077
13078
13079
13080 047430
13081 047430 000004 ;CALL THE SCOPE LOOP UTILITY
13082 047432 012700 ;LOAD R0 WITH TEST NUMBER
13083 047436 013701 000665 ;LOAD R1 WITH TEST INSTRUCTION WORD
13084 047438 013704 047464 ;SAVE THE SP
13085 047442 013704 000004 ;SAVE TRAP VECTOR
13086 047450 012737 047470 000004 ;GO TO 4$ IF TRAP SPRUNG
13087 047456 012706 000400 ;SET SP TO PUSH INTO "YELLOW ZONE"
13088 047462 000257 ;SCOPE SYNC
13089
13089 047464 030046 2$: BIT R0,-(SP) ;TEST THE BIT - NO TRAP SHOULD OCCUR
13090
13091 047466 000406 BR 6$ ;GO TO EXIT TEST
13092
13093 047470 010437 000004 4$: MOV R4,@#4 ;RESTORE TRAP VECTOR
13094 047474 010637 001074 ;SAVE BAD SP FOR PRINTING
13095 047500 010506 ;RESET THE SP
13096 047502 104006 3$: ERROR 5 ;BIT CAUSED OVERFLOW TRAP
13097
13098 047504 010437 000004 6$: MOV R4,@#4 ;RESTORE THE VECTOR
13099 047510 010506 ;R5,SP ;RESET THE SP
13100
13101
13102
13103
13104 047512
13105 047512 000004 ;CALL THE SCOPE LOOP UTILITY
13106 047514 012700 ;LOAD R0 WITH TEST NUMBER
13107 047520 013701 000666 ;LOAD R1 WITH TEST INSTRUCTION WORD
13108 047524 010605 047546 ;SAVE THE SP
13109 047526 013704 000004 ;SAVE TRAP VECTOR
13110 047532 012737 047552 000004 ;GO TO 4$ IF TRAP SPRUNG
13111 047540 012706 000400 ;SET SP TO PUSH INTO "YELLOW ZONE"
13112 047544 000257 ;SCOPE SYNC
13113
13114 047546 005746 2$: TST -(SP) ;TEST THE TST - NO TRAP SHOULD OCCUR
13115
13116 047550 000406 BR 6$ ;GO TO EXIT TEST
13117
13117 047552 010437 000004 4$: MOV R4,@#4 ;RESTORE TRAP VECTOR
13118 047556 010637 001074 ;SAVE BAD SP FOR PRINTING
13119 047562 010506 ;RESET THE SP
13120 047564 104006 3$: ERROR 6 ;TST CAUSED OVERFLOW TRAP
13121
13122 047566 010437 000004 6$: MOV R4,@#4 ;RESTORE THE VECTOR
13123 047572 010506 ;R5,SP ;RESET THE SP
13124
13125
13126
13127
13128

```

```

13129 047574
13130 047574 000004 ;CALL THE SCOPE LOOP UTILITY
13131 047576 012700 ;LOAD R0 WITH TEST NUMBER
13132 047602 013701 000667 ;LOAD R1 WITH TEST INSTRUCTION WORD
13133 047606 010605 ;SAVE SP
13134 047610 010737 001010 ;SET ERROR LOOP ADDRESS
13135 047614 013704 000004 ;SAVE T.O. VECTOR
13136 047620 012737 047646 000004 ;ADDR ERROR - GO TO 4$
13137 047624 012706 000001 ;RESET SP FOR ERROR LOOP
13138 047630 012706 ;R2 GETS ODD ADDRESS
13139 047634 000257 ;SCOPE SYNC
13140
13141 047636 160012 2$: SUB R0,(R2) ;FORCE ODD ADDR ERROR - GO TO 4$
13142
13143 047640 010437 000004 3$: MOV R4,@#4 ;RESTORE T.O. VECTOR
13144 047644 104006 ;ODD ADDR FAILED TO TRAP
13145
13146 047646 010437 000004 4$: MOV R4,@#4 ;RESTORE T.O. VECTOR
13147 047652 010506 ;R5,SP ;RESET SP
13148 047654 005037 000000 ;CLR LOC. 0 JUST IN CASE
13149
13150
13151
13152
13153 047660
13154 047660 000004 ;CALL THE SCOPE LOOP UTILITY
13155 047662 012700 ;LOAD R0 WITH TEST NUMBER
13156 047666 013701 000670 ;LOAD R1 WITH TEST INSTRUCTION WORD
13157 047672 012702 063317 ;DEST ADDR=MBUF1+1 (ODD)
13158 047676 012737 047760 000004 ;GO TO 4$ ON ODA TRAP
13159
13160 047704 010205 ;[R5] = DEST. ADDR
13161 047706 000257 ;SCOPE SYNC
13162
13163 047710 105435 2$: NEGB @(R5)+ ;TEST DM=3 TRAP
13164
13165 047712 104006 3$: ERROR 6 ;ODA TRAP NOT SPRUNG
13166
13167 047714 104006
13168 047714 012705 063321 ;[R5] = DEST. ADDR
13169 047720 013701 047726 ;[R1] = TEST INSTR
13170 047724 000257 ;SCOPE SYNC
13171
13172 047726 105455 20$: NEGB @-(R5) ;TEST DM=5 TRAP
13173
13174 047730 104006 5$: ERROR 6 ;ODA TRAP NOT SPRUNG
13175
13176 047732 010205 ;[R5] = DEST ADDR
13177 047734 013701 047742 ;[R1] = TEST INSTR
13178 047740 000257 ;SCOPE SYNC
13179
13180 047742 105475 000000 21$: NEGB @0(R5) ;TEST DM=7 TRAP
13181
13182 047746 104006 7$: ERROR 6 ;ODA TRAP NOT SPRUNG
13183 047750 012737 061220 000004 ;BERR,@#4 ;RESET T.O. VECTOR
13184 047756 000403 ;TST671 ;GO TO SCOPE EXIT

```



```
13185
13186 047760 062716 000002 4$: ADD #2,(SP) ;MOV RETURN PC AROUND ERROR CALL
13187 047764 000002 ;RETURN TO NEXT SUB-TEST
13188
13189 ;*****
13190 ;*TEST 671 TEST FOR ODD ADDR ERROR TRAP FOR SOURCE DEFERRED MODES
13191 ;*****
13192
13193 TST671: SCOPE ;CALL THE SCOPE LOOP UTILITY
13194 047766 000004 ;LOAD R0 WITH TEST NUMBER
13195 047770 012701 000671 MOV #671,R0 ;LOAD R1 WITH TEST INSTRUCTION WORD
13196 047774 013701 050016 MOV #205,R1 ;R2 = SOURCE ADDR. (ODD)
13197 050000 012702 063317 MOV #806+1,R2 ;R5 = SOURCE ADDR. (ODD)
13198 050004 012737 050066 MOV #45,#4 ;GO TO 4$ ON TRAP
13199
13200 MOV R2,R5 ;R5 = SOURCE ADDR.
13201 CCC ;SCOPE SYNC
13202
13203 050016 113504 2$: MOVB @R5+,R4 ;TEST SM=3
13204
13205 050020 104006 3$: ERROR 6 ;ODA TRAP NOT SPRUNG
13206
13207 050022 012705 063321 MOV #MBUF1+3,R5 ;R5 = SOURCE ADDR
13208 050026 013701 050034 MOV #205,R1 ;R1 = TEST INSTR
13209 050032 000257 CCC ;SCOPE SYNC
13210
13211 050034 115504 20$: MOVB @-(R5),R4 ;TEST SM=5
13212
13213 050036 104006 5$: ERROR 6 ;ODA TRAP NOT SPRUNG
13214 050040 010205 MOV R2,R5 ;R5 = SOURCE ADDR
13215 050042 013701 050050 MOV #215,R1 ;R1 = TEST INSTR
13216 050046 000257 CCC ;SCOPE SYNC
13217
13218 050050 117504 000000 21$: MOVB @0(R5),R4 ;TEST SM=7
13219
13220 050054 104006 7$: ERROR 6 ;ODA TRAP NOT SPRUNG
13221
13222 050056 012737 061220 000004 MOV #BERR,#4 ;RESET T.O. VECTOR
13223 050064 000403 BR TST672 ;GO TO SCOPE EXIT
13224
13225 050066 062716 000002 4$: ADD #2,(SP) ;MOVE RETURN PC AROUND ERROR CALL
13226 050072 000002 ;RETURN TO NEXT SUB-TEST
13227
13228 ;*****
13229 ;*TEST 672 TEST FOR ODD ADDR ERROR TRAP FOR JMP DEST DEFERRED MODES
13230 ;*****
13231
13232 TST672: SCOPE ;CALL THE SCOPE LOOP UTILITY
13233 050074 000004 ;LOAD R0 WITH TEST NUMBER
13234 050078 012700 000672 MOV #672,R0 ;LOAD R1 WITH TEST INSTRUCTION WORD
13235 050102 013701 050144 MOV #205,R1 ;DEST ADDR = 65+3 (ODD)
13236 050106 012702 050177 MOV #65+3,R2 ;R5 = SOURCE ADDR. (ODD)
13237 050112 012737 050202 MOV #45,#4 ;GO TO 4$ ON ODA TRAP
13238
13239 MOV R2,R5 ;R5 = DEST ADDR
13240 CCC ;SCOPE SYNC
13241
13242 050120 010205 2$: JMP @R5+ ;TEST JMP DM=3
13243
13244 050122 000257
13245
13246 050124 000135
```

```
13241
13242 050126 104006 3$: ERROR 6 ;ODA TRAP NOT SPRUNG IN ROM LOC 153
13243
13244 050130 012705 050177 MOV #65+3,R5 ;R5 = DEST ADDR
13245 050134 013701 050142 MOV #205,R1 ;R1 = TEST INSTR
13246 050140 000257 CCC ;SCOPE SYNC
13247
13248 050142 000155 20$: JMP @-(R5) ;TEST JMP DM=5
13249
13250 050144 104006 5$: ERROR 6 ;ODA TRAP NOT SPRUNG IN ROM LOC 155
13251
13252 050146 010205 MOV R2,R5 ;R5 = DEST ADDR
13253 050150 013701 050156 MOV #215,R1 ;R1 = TEST INSTR
13254 050154 000257 CCC ;SCOPE SYNC
13255
13256 050156 000175 000000 21$: JMP @0(R5) ;TEST JMP DM=7
13257
13258 050162 104006 7$: ERROR 6 ;ODA TRAP NOT SPRUNG
13259
13260 050164 012737 061220 000004 MOV #BERR,#4 ;RESET BUS T.O. VECTOR
13261 050172 000420 BR TST673 ;GO TO SCOPE EXIT
13262
13263 050174 000000 6$: HALT ;CATASTROPHIC ERROR - (CPC) QUESTIONABLE.
13264 050176 000605 HALT ;RESTART PROGRAM - DO NOT CONTINUE.
13265 050200 000000 HALT
13266
13267 050202 032716 000001 4$: BIT #1,(SP) ;TRAP DUE TO ODD PC?
13268 050206 001003 BNE B5 ;BR IF YES
13269 050210 062716 000002 ADD #2,(SP) ;MOV RETURN PC AROUND ERROR CALL
13270 050214 000002 RTI ;RETURN TO NEXT SUB TEST
13271
13272 050216 011603 0$: MOV (SP),R3 ;GET ODD PC OFF STACK INTO R3
13273 050220 062706 000004 ADD #4,SP ;FIX SP
13274
13275 050224 104007 9$: ERROR 7 ;PC TRAPPED WITH ODD ADDRESS
13276
13277 050226 012737 061220 000004 MOV #BERR,#4 ;RESET T.O. VECTOR
13278
13279 ;*****
13280 ;*TEST 673 TEST FOR STACK OFLW FOR DEST MODES 1,2,4, AND 6.
13281 ;*****
13282
13283 TST673: SCOPE ;CALL THE SCOPE LOOP UTILITY
13284 050234 000004 ;LOAD R0 WITH TEST NUMBER
13285 050238 012700 000673 MOV #673,R0 ;LOAD R1 WITH TEST INSTRUCTION WORD
13286 050242 013701 050266 MOV #205,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13287 050246 012737 050400 MOV #45,#4 ;GO TO 4$ ON OVLW TRAP
13288 050250 010605 SP,R5 ;SAVE SP
13289 050256 012702 000376 MOV #376,R2 ;USE R2 TO SET UP SP TO CAUSE TRAP
13290
13291 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
13292 CCC ;SCOPE SYNC
13293
13294 050266 005016 2$: CLR (SP) ;TEST DM1 - SHOULD SPRING TRAP
13295
13296 050270 010637 001074 MOV SP,#$REG5 ;SAVE BAD SP FOR PRINTING
13297 050274 010506 MOV R5,SP ;RESET SP
```

```

13297 050276 104006 3$: ERROR 6 ;DM1 FAILED TO CAUSE OVERFLOW TRAP
13298
13299 050300 013701 050310 MOV R0,#20$R1 ;ER11 = TEST INSTR.
13300 050304 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
13301 050306 000257 CCC ;SCOPE SYNC
13302
13303 050310 005026 20$: CLR (SP)+ ;TEST DM2 - SHOULD SPRING TRAP
13304
13305 050312 010637 001074 MOV SP,#$REG5 ;SAVE BAD SP FOR PRINTING
13306 050316 010506 MOV R5,SP ;RESET SP
13307 050320 104006 5$: ERROR 6 ;DM2 FAILED TO CAUSE OVERFLOW TRAP
13308
13309 050322 013701 050332 MOV R0,#21$R1 ;ER11 = TEST INSTR.
13310 050326 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
13311 050330 000257 CCC ;SCOPE SYNC
13312
13313 050332 005046 21$: CLR -(SP) ;TEST DM4 - SHOULD SPRING TRAP
13314
13315 050334 010637 001074 MOV SP,#$REG5 ;SAVE BAD SP FOR PRINTING
13316 050340 010506 MOV R5,SP ;RESET SP
13317 050342 104006 7$: ERROR 6 ;DM4 FAILED TO CAUSE OVERFLOW TRAP
13318
13319 050344 013701 050354 MOV R0,#22$R1 ;ER11 = TEST INSTR.
13320 050350 010206 MOV R2,SP ;SET UP SP TO CAUSE ERROR
13321 050352 000257 CCC ;SCOPE SYNC
13322
13323 050354 005066 000000 22$: CLR 0(SP) ;TEST DM6 - SHOULD SPRING TRAP
13324
13325 050360 010637 001074 MOV SP,#$REG5 ;SAVE BAD SP FOR PRINTING
13326 050364 010506 MOV R5,SP ;RESET SP
13327 050366 104006 9$: ERROR 6 ;DM6 FAILED TO CAUSE OVERFLOW TRAP
13328
13329 050370 012737 061220 000004 MOV #BERR,#4 ;RESET BUS T.O. VECTOR
13330 050376 000407 BR TST674 ;GO TO SCOPE EXIT
13331
13332 050400 011604 4$: MOV (SP),R4 ;GET RETURN PC OFF STACK
13333 050402 062704 ADD #10,R4 ;MOVE RETURN PC AROUND ERROR CALL
13334 050406 010506 MOV R5,SP ;RESET SP
13335 050410 005046 CLR -(SP) ;PUSH NEW PS ON STACK
13336 050412 010446 MOV R4,-(SP) ;PUSH RETURN PC ON STACK
13337 050414 000002 RTI ;RETURN TO NEXT SUB-TEST

```

```

;*****
;TEST 674 TEST FOR STACK OVLW FOR MOV DEST MODES 1,2,4, AND 6.
;*****
TST674:

```

```

13342 050416 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
13343 050420 012700 MOV #674,R0 ;LOAD R0 WITH TEST NUMBER
13344 050424 013701 050450 MOV R0,#2$R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13345 050430 012737 000004 MOV #4$R4 ;GO TO 4$ ON STACK OVLW TRAP
13346 050436 010605 MOV SP,R5 ;SAVE SP
13347 050440 012702 000376 MOV #376,R2 ;USE R2 TO SET UP SP TO CAUSE TRAP
13348
13349 050444 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
13350 050446 000257 CCC ;SCOPE SYNC
13351
13352

```

```

13353 050450 010016 2$: MOV R0,(SP) ;TEST MOV DM1 - SHOULD SPRING TRAP
13354
13355 050452 010637 001074 MOV SP,#$REG5 ;SAVE BAD SP FOR PRINTING
13356 050454 010506 MOV R5,SP ;RESET SP
13357 050460 104006 3$: ERROR 6 ;MOV DM1 FAILED TO SPRING TRAP
13358
13359 050462 013701 050472 MOV R0,#20$R1 ;ER11 = TEST INSTR.
13360 050466 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
13361 050470 000257 CCC ;SCOPE SYNC
13362
13363 050472 010026 20$: MOV R0,(SP)+ ;TEST MOV DM2 - SHOULD SPRING TRAP
13364
13365 050474 010637 001074 MOV SP,#$REG5 ;SAVE BAD SP FOR PRINTING
13366 050500 010506 MOV R5,SP ;RESET SP
13367 050502 104006 5$: ERROR 6 ;MOV DM2 FAILED TO SPRING TRAP
13368
13369 050504 013701 050514 MOV R0,#21$R1 ;ER11 = TEST INSTR.
13370 050510 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
13371 050512 000257 CCC ;SCOPE SYNC
13372
13373 050514 010046 21$: MOV R0,-(SP) ;TEST MOV DM4 - SHOULD SPRING TRAP
13374
13375 050516 010637 001074 MOV SP,#$REG5 ;SAVE BAD SP FOR PRINTING
13376 050522 010506 MOV R5,SP ;RESET SP
13377 050524 104006 7$: ERROR 6 ;MOV DM4 FAILED TO SPRING TRAP
13378
13379 050526 013701 050536 MOV R0,#22$R1 ;ER11 = TEST INSTR.
13380 050532 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
13381 050534 000257 CCC ;SCOPE SYNC
13382
13383 050536 010066 000000 22$: MOV R0,0(SP) ;TEST MOV DM6 - SHOULD SPRING TRAP
13384
13385 050542 010637 001074 MOV SP,#$REG5 ;SAVE BAD SP FOR PRINTING
13386 050546 010506 MOV R5,SP ;RESET SP
13387 050550 104006 9$: ERROR 6 ;MOV DM6 FAILED TO CAUSE OVLW TRAP
13388
13389 050552 012737 061220 000004 MOV #BERR,#4 ;RESET T.O. VECTOR
13390 050560 000407 BR TST675 ;GO TO SCOPE EXIT
13391
13392 050562 011604 4$: MOV (SP),R4 ;GET RETURN PC
13393 050564 062704 ADD #10,R4 ;MOVE RETURN PC AROUND ERROR CALL
13394 050570 010506 MOV R5,SP ;RESET SP
13395 050572 005046 CLR -(SP) ;PUSH NEW PSW
13396 050574 010446 MOV R4,-(SP) ;PUSH RETURN PC
13397 050576 000002 RTI ;RETURN TO NEXT SUB-TEST

```

```

;*****
;TEST 675 TEST THAT JSR CAN CAUSE OVERFLOW TRAP
;*****
TST675:

```

```

13403 050600 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
13404 050602 012700 MOV #675,R0 ;LOAD R0 WITH TEST NUMBER
13405 050606 013701 050630 MOV R0,#2$R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13406 050612 012737 000004 MOV #4$R4 ;GO TO 4$ ON OVERFLOW ERROR
13407 050620 010605 MOV SP,R5 ;SAVE SP
13408 050622 012706 MOV #400,SP ;SET THE SP TO CAUSE TRAP

```

```

13409 050626 000257          CCC          ;SCOPE SYNC
13410
13411 050630 004737 050656    2$: JSR      PC,@#6$ ;TEST JSR - SHOULD SPRING TRAP
13412
13413 050634 010637 001074    MOV      SP,@#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
13414 050640 010506          MOV      R5,SP      ;RESET SP
13415 050642 104005          ERROR    5          ;JSR PUSH DID NOT SPRING OVFL TRAP
13416
13417 050644 000410          BR       8$         ;GO TO SCOPE EXIT
13418
13419 050646 010637 001074    MOV      SP,@#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
13420 050652 010506          MOV      R5,SP      ;RESET SP
13421 050654 000404          BR       8$         ;GO EXIT TEST - ALL OK
13422
13423 050656 010637 001074    MOV      SP,@#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
13424 050662 010506          MOV      R5,SP      ;RESET SP
13425 050664 104005          ERROR    5          ;JSR PUSH FAILED TO SPRING OVFLW TRAP
13426
13427 050666 012737 061220 000004 8$: MOV      #BERR,@#4 ;RESET BUS T.O. VECTOR
13428
13429 *****
13430 ;TEST 676 TEST THAT 1ST PUSH IN TRAP MICROROUTINE CAUSES OVFLW TRAP
13431 ;*****
13432 ;TEST676:
13433 SCOPE          ;CALL THE SCOPE LOOP UTILITY
13434 MOV      #676,R0 ;LOAD R0 WITH TEST NUMBER
13435 MOV      @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13436 MOV      @#14,R4 ;SAVE BREAK POINT TRAP VECTOR
13437 MOV      SP,R5    ;SAVE SP
13438 MOV      @#4,@#4 ;GO TO 4$ ON OVFLW TRAP
13439 MOV      @#6,@#14 ;GO TO 6$ IF BPT SERVICED
13440 MOV      @#8,@#14 ;SET UP SP TO CAUSE OVFLW ON 1ST PUSH
13441 CCC          ;SCOPE SYNC
13442
13443 050736 000003          2$: BPT          ;TEST THE BPT - SHOULD CAUSE OVERFLOW TRAP
13444
13445 050740 010637 001074    MOV      SP,@#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
13446 050744 010506          MOV      R5,SP      ;RESET SP
13447 050746 104005          ERROR    5          ;BPT FAILED TO TRAP
13448
13449 050750 000406          BR       8$         ;GO TO SCOPE EXIT
13450
13451 050752 010506          MOV      R5,SP      ;RESET SP
13452 050754 000404          BR       8$         ;GO EXIT - ALL OK
13453
13454 050756 010637 001074    MOV      SP,@#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
13455 050762 010506          MOV      R5,SP      ;RESET SP
13456 050764 104005          ERROR    5          ;OVFLW TRAP FAILED TO BUMP BPT SERVICE
13457
13458 050766 012737 061220 000004 8$: MOV      #BERR,@#4 ;RESET VECTORS
13459 050774 010437 000014    MOV      R4,@#14
13460
13461 *****
13462 ;TEST 677 TEST THAT 2ND PUSH IN TRAP MICROROUTINE CAUSES OVFLW TRAP
13463 ;*****
13464 051000

```

```

13465 051000 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
13466 051002 012700 000676    MOV      #677,R0 ;LOAD R0 WITH TEST NUMBER
13467 051006 013701 051042    MOV      @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13468 051012 013704 000014    MOV      @#14,R4 ;SAVE BPT VECTOR
13469 051016 010605          MOV      SP,R5    ;SAVE SP
13470 051020 012737 051056 000004 4$: @#4 ;GO TO 4$ ON STACK OVFLOW
13471 051026 012737 051062 000014 6$: @#14 ;GO TO 6$ IF BPT SERVICED
13472 051034 012706 000402    MOV      @#6,@#14 ;SET SP TO CAUSE TRAP ON 2ND PUSH
13473 051040 000257          CCC          ;SCOPE SYNC
13474
13475 051042 000003          2$: BPT          ;TEST THE BPT - SHOULD CAUSE OVERFLOW TRAP
13476
13477 051044 010637 001074    MOV      SP,@#$REG5 ;SAVE BAD SP FOR PRINTING
13478 051050 010506          MOV      R5,SP      ;RESET SP
13479 051052 104005          ERROR    5          ;BPT FAILED TO TRAP
13480
13481 051054 000406          BR       8$         ;GO TO SCOPE EXIT
13482
13483 051056 010506          MOV      R5,SP      ;RESET SP
13484 051060 000404          BR       8$         ;GO EXIT - ALL OK
13485
13486 051062 010637 001074    MOV      SP,@#$REG5 ;SAVE BAD SP FOR PRINTING
13487 051070 010506          MOV      R5,SP      ;RESET SP
13488 051076 104005          ERROR    5          ;OVFLW TRAP FAILED TO BUMP BPT SERVICE
13489
13490 051072 012737 061220 000004 8$: MOV      #BERR,@#4 ;RESET VECTORS
13491 051100 010437 000014    MOV      R4,@#14
13492
13493 *****
13494 ;TEST 700 ILLEGAL INSTRUCTION TEST - JSR RN,R#
13495 ;*****
13496 ;TEST700:
13497 SCOPE          ;CALL THE SCOPE LOOP UTILITY
13498 MOV      #700,R0 ;LOAD R0 WITH TEST NUMBER
13499 MOV      @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13500 MOV      SP,R5    ;SAVE SP
13501 MOV      PC,@#4 ;SET ERROR LOOP ADDRESS
13502 MOV      @#4,R4 ;SAVE T.O. VECTOR
13503 MOV      @#4,@#4 ;ILLEGAL INSTR. TRAP GOES TO 4$
13504 MOV      R5,SP    ;RESET SP FOR ERROR LOOP
13505 MOV      #3$,R2  ;IN CASE JSR JUMPS TO ER2]
13506 CCC          ;SCOPE SYNC
13507
13508 051146 0 4302          2$: JSR      R3,R2    ;JSR MODE 0 FORCES TRAP - GO TO 4$
13509
13510 051150 010437 000004    MOV      R4,@#4    ;RESTORE T.O. VECTOR
13511 051154 104005          ERROR    5          ;JSR FAILED TO SPRING TRAP
13512
13513 051156 010437 000004    4$: MOV      R4,@#4  ;RESTORE VECTOR
13514 051162 010506          MOV      R5,SP      ;RESET SP
13515
13516 *****
13517 ;TEST 701 ILLEGAL INSTRUCTION TEST - JMP &R
13518 ;*****
13519 051164 000004          ;CALL THE SCOPE LOOP UTILITY
13520

```

```

13521 051166 012700 000701      MOV    #701,R0      ;;LOAD R0 WITH TEST NUMBER
13522 051172 013701 051226      MOV    @#25,R1     ;;LOAD R1 WITH TEST INSTRUCTION WORD
13523 051176 010605          MOV    SP,R5       ;;SAVE SP
13524 051200 010737 001010      MOV    PC,@#SLPERR ;;SET ERROR LOOP ADDRESS
13525 051204 013704 000004 1$:    MOV    @#4,R4      ;;SAVE VECTOR POINTER AT LOC. 4
13526 051210 012737 051236 000004 2$:    MOV    @#4,ON TRAP - GO TO 4$
13527 051216 010506          MOV    R5,SP       ;;RESET SP FOR ERROR LOOP
13528 051220 012702 051234          MOV    #3$,R2      ;;IN CASE IT JUMPS TO ADDR IN RN
13529 051224 000257          CCC               ;;SCOPE SYNC
13530
13531 051226 000102          2$:    JMP    R2          ;;JMP MODE 0 FORCES TRAP - GO TO 4$
13532
13533 051230 010437 000004          3$:    MOV    R4,@#4      ;;RESTORE VECTOR POINTER AT LOC. 4
13534 051234 104005          ERROR 5           ;;ILLEGAL INSTR TRAP FAILED
13535
13536 051236 010437 000004          4$:    MOV    R4,@#4      ;;RESTORE VECTOR POINTER AT LOC. 4
13537 051242 010506          MOV    R5,SP       ;;RESET SP
13538
13539
13540 ;;*****
13541 ;;TEST 702 BUS TIMEOUT TRAP TEST - TST (R)
13542 ;;*****
13543 TST702:
13544 SCOPE
13545 MOV    #702,R0     ;;CALL THE SCOPE LOOP UTILITY
13546 MOV    @#25,R1     ;;LOAD R0 WITH TEST NUMBER
13547 MOV    SP,R5       ;;LOAD R1 WITH TEST INSTRUCTION WORD
13548 MOV    PC,@#SLPERR ;;SAVE SP
13549 MOV    @#4,R4      ;;SET ERROR LOOP ADDRESS
13550 MOV    @#4,R4      ;;SAVE ORIGINAL T.O. VECTOR POINTER
13551 1$:    MOV    @#4,ON T.O. TRAP - GO TO 4$
13552 MOV    #160000,R2  ;;ADDRESS CAUSES T.O.
13553 MOV    R5,SP       ;;RESET SP FOR ERROR LOOP
13554 CCC               ;;SCOPE SYNC
13555
13556 051306 005712          2$:    TST    (R2)      ;;FORCE T.O. TRAP - GO TO 4$
13557
13558 051310 010437 000004          3$:    MOV    R4,@#4      ;;RESTORE T.O. VECTOR
13559 051314 104005          ERROR 5           ;;TIMEOUT TRAP FAILED
13560 051316 010437 000004          4$:    MOV    R4,@#4      ;;RESTORE T.O. VECTOR
13561 051322 010506          MOV    R5,SP       ;;RESET SP
13562
13563 ;;*****
13564 ;;TEST 703 "T" BIT TRAP TEST
13565 ;;*****
13566 TST703:
13567 SCOPE
13568 MOV    #703,R0     ;;CALL THE SCOPE LOOP UTILITY
13569 MOV    @#25,R1     ;;LOAD R0 WITH TEST NUMBER
13570 MOV    SP,R5       ;;LOAD R1 WITH TEST INSTRUCTION WORD
13571 MOV    PC,@#SLPERR ;;SAVE SP
13572 MOV    @#4,R4      ;;SET ERROR LOOP ADDRESS
13573 1$:    MOV    @#4,ON T.O. TRAP - GO TO 4$
13574 MOV    @#20,@#14  ;;GET T.O. WHEN "T" TRAP SPRUNG
13575 MOV    @#25,-(SP)  ;;SET "T" BIT ON STACK
13576 MOV    @#25,-(SP)  ;;SET UP NEW PC ON STACK
13577 CCC               ;;SCOPE SYNC
13578 RTT               ;;TURN ON "T" BIT - GO TO 2$

```

```

13577 051370 005700          2$:    TST    R0        ;;SPRING "T" BIT TRAP - GO TO 4$
13578
13579 051372 104005          3$:    ERROR 5         ;;NO "T" BIT TRAP OCCURRED
13580
13581 051374 000405          BR    6$         ;;GO EXIT
13582
13583 051376 032766 000020 000002 4$:    BIT    #20,2(SP)  ;;"T" BIT SET IN OLD PSW?
13584 051404 001001          BNE   6$         ;;BR IF YES
13585
13586 051406 104001          5$:    ERROR 1         ;;#T# BIT NOT SAVED ON STACK
13587
13588 051410 012737 000016 000014 6$:    MOV    #16,@#14   ;;RESTORE "T" BIT TRAP CATCHER
13589 051416 005037 000016          CLR    @#16       ;;RESET SP
13590 051422 010506          MOV    R5,SP
13591
13592 ;;*****
13593 ;;TEST 704 TEST PUSH INTO PSW WITH C$P1 = 000000
13594 ;;THESE NEXT TWO TESTS VERIFY THAT A "RED ZONE" TRAP OCCURS IF A
13595 ;;PUSH IS ATTEMPTED WITH THE C$P1 INITIALLY EQUAL TO 000000,177572,
13596 ;;*****
13597 TST704:
13598 SCOPE
13599 MOV    #704,R0     ;;CALL THE SCOPE LOOP UTILITY
13600 MOV    @#25,R1     ;;LOAD R0 WITH TEST NUMBER
13601 MOV    SP,R5       ;;LOAD R1 WITH TEST INSTRUCTION WORD
13602 MOV    PC,@#SLPERR ;;SAVE THE SP
13603 MOV    @#4,R4      ;;SAVE THE BUS ERROR VECTOR
13604 1$:    MOV    @#4,;"RED ZONE" TRAP GOES TO 4$
13605 CLR    SP          ;;MAKE SP = 000000
13606 CCC               ;;SCOPE SYNC
13607
13608 051456 012746 007777          2$:    MOV    #7777,-(SP) ;;ATTEMPT PUSH INTO PSW - SHOULD CAUSE
13609 ;;"RED ZONE" TRAP TO BE SPRUNG
13610
13611 051462 010437 000004          MOV    R4,@#4     ;;RESTORE BUS ERROR VECTOR
13612 051466 005004          CLR    R4         ;;CR4 = S / B SP
13613 051470 010603          MOV    SP,R3      ;;CR3 = WAS SP
13614 051474 010506          MOV    R5,SP      ;;RESET THE SP
13615 051476 000414          3$:    ERROR 3         ;;TRAP NOT SPRUNG
13616 BR    TST705     ;;GO TO SCOPE EXIT - SCHOOL'S OUT
13617
13618 051500 022706 000000          4$:    CMP    #0,SP     ;;WAS IT A RED ZONE TRAP ?
13619 051504 001406          BEQ   6$         ;;BR IF YES
13620
13621 051506 010437 000004          MOV    R4,@#4     ;;RESTORE BUS ERROR VECTOR
13622 051512 005004          CLR    R4         ;;CR4 = S / B SP
13623 051514 010603          MOV    SP,R3      ;;CR3 = WAS SP
13624 051516 010506          MOV    R5,SP      ;;RESET THE SP
13625 051520 104003          5$:    ERROR 3         ;;TRAP SPRUNG BUT NOT RED ZONE
13626
13627 051522 010506          6$:    MOV    R5,SP     ;;FIX UP THE SP
13628 051524 010437 000004          MOV    R4,@#4     ;;RESTORE BERR VECTOR
13629
13630 ;;*****
13631 ;;TEST 705 TEST PUSH INTO SP WITH C$P1 = 177572
13632 ;;*****
13633 TST705:

```

```

13633 051530 000004
13634 051532 012700 000705
13635 051536 013701 051564
13636 051542 010605
13637 051544 013704 000004
13638 051550 012737 051606 000004
13639 051556 012706 177572
13640 051562 000257
13641
13642
13643 051564 012746 177777 2$: MOV #-1,-(SP) ;ATTEMPT PUSH INTO SR - SHOULD CAUSE
; "RED ZONE" TRAP TO BE SPRUNG
13644
13645 051570 010437 000004
13646 051574 005004
13647 051674 013701 051672
13648 051600 010506
13649 051602 104003
13650 051604 000414 3$: BR TST706 ;TRAP NOT SPRUNG
; ;GO TO SCOPE EXIT - SCHOOL'S OUT
13651
13652 051606 022706 000000 4$: CMP #0,SP ;WAS IT A RED ZONE TRAP ?
13653 051612 001406 ;BR IF YES
13654
13655 051614 010437 000004
13656 051620 005004
13657 051624 010603
13658 051626 010506 5$: MOV R4,@#4 ;RESTORE BUS ERROR VECTOR
13659 051626 010506 ;CR4 = S / B SP
;CR3 = WAS SP
;RESET THE SP
;TRAP SPRUNG BUT NOT RED ZONE
13660
13661 051630 010506 000004 6$: MOV R5,SP ;FIX UP THE SP
13662 051632 010437 ;RESTORE BUS ERROR VECTOR
13663
13664
13665
13666
13667 051636 000004
13668 051640 012700 000706
13669 051640 013701 051672
13670 051650 010605
13671 051650 010605
13672 051652 013704 000004
13673 051656 012737 051714 000004
13674 051664 012706 177776
13675 051670 000257
13676
13677 051672 012746 000200 2$: MOV #200,-(SP) ;ATTEMPT PUSH INTO SLR - SHOULD CAUSE
; "RED ZONE" TRAP TO BE SPRUNG
13678
13679
13680 051676 010437 000004
13681 051704 005004
13682 051704 010603
13683 051706 010506
13684 051710 104003
13685 051712 000414 3$: BR TST707 ;TRAP NOT SPRUNG
; ;GO TO SCOPE EXIT - SCHOOL'S OUT
13686
13687 051714 022706 000000 4$: CMP #0,SP ;WAS IT A RED ZONE TRAP ?
13688 051720 001406 ;BR IF YES

```

```

13689
13690 051722 010437 000004
13691 051726 005004
13692 051730 010603
13693 051734 010506
13694 051734 104003
13695
13696 051736 010506 000004 6$: MOV R5,SP ;FIX UP THE SP
13697 051740 010437 ;RESTORE BUS ERROR VECTOR
13698
13699
13700
13701
13702 051744
13703 051744 000004
13704 051746 012700 000707
13705 051752 013695
13706 051754 012737 052012 000010 5$: MOV #45,R5 ;CALL THE SCOPE LOOP UTILITY
; ;LOAD R0 WITH TEST NUMBER
; ;SAVE THE SP
; ;SET UP RSVD INSTR. TRAP VECTOR
13707 051762 005037 000012
13708 051766 012701 000007
13709 051772 010737 001010
13710 051776 010506
13711 052000 010137 052006 1$: MOV R5,SP ;SET UP FIRST ONE IN GROUP
; ;ONLY LOOP ON BAD OP CODE
; ;RESET SP FOR ERROR LOOP AND NEW INSTR
13712 052004 000257 ;LOAD NEW INSTR
; ;SCOPE SYNC
13713
13714 052006 000007 2$: 000007 ;TEST THE RSVD INSTR - THIS LOCATION
; ;GETS CHANGED EACH PASS THROUGH
13715
13716 052010 104005 3$: ERROR 5 ;RSVD INSTR. IN R1 FAILED TO TRAP
13717
13718
13719 052012 005201
13720 052014 022701 000100 4$: INC R1 ;GENERATE NEW RSVD INSTR
; ;AT END OF THIS GROUP ??
; ;BR IF NOT
13721 052020 001366
13722
13723 052022 010506
13724 052024 012737 051752 001010 6$: MOV R5,SP ;MAKE SURE TO RESET THE SP
; ;LOOP FROM BEGINNING ON ERROR
13725
13726
13727
13728 052032
13729 052034 000004
13730 052034 012700 000710
13731 052040 010605
13732 052042 012737 052100 000010 5$: MOV #45,R5 ;CALL THE SCOPE LOOP UTILITY
; ;LOAD R0 WITH TEST NUMBER
; ;SAVE THE SP
; ;SET UP RSVD INSTR. TRAP VECTOR
13733 052050 005037 000012
13734 052054 012701 000210
13735 052060 010737 001010
13736 052064 010506
13737 052066 010137 052074 1$: MOV R2,0,R1 ;SET UP FIRST ONE IN GROUP
; ;SET ERROR LOOP ADDRESS
; ;RESET SP FOR ERROR LOOP AND NEW INSTR
13738 052072 000257 ;LOAD NEW INSTR
; ;SCOPE SYNC
13739
13740 052074 000210 2$: 000210 ;TEST THE RSVD INSTR - THIS LOCATION
; ;GETS CHANGED EACH PASS THROUGH
13741
13742
13743 052076 104005 3$: ERROR 5 ;RSVD INSTR. IN R1 FAILED TO TRAP
13744

```

13745 052100 005201
 13746 052102 022701 000240
 13747 052106 001366
 13748
 13749
 13750 052110 010506
 13751 052112 012737 052040 001010
 13752
 13753
 13754
 13755
 13756 052120
 13757 052122 012700 000711
 13758 052126 010605
 13759 052130 012737 052166 000010
 13760 052136 005037 000012
 13761 052142 012701 007000
 13762 052146 010737 001010
 13763 052152 010506
 13764 052154 010137 052162
 13765 052160 000257
 13766
 13767 052162 007000
 13768
 13769
 13770 052164 104005
 13771
 13772 052166 005201
 13773 052170 022701 010000
 13774 052174 001366
 13775
 13776 052176 010506
 13777 052200 012737 052126 001010
 13778
 13779
 13780
 13781
 13782 052206
 13783 052206 000004
 13784 052210 012700 000712
 13785 052214 010605
 13786 052216 012737 052256 000010
 13787 052224 005037 000012
 13788 052230 012701 075000
 13789 052234 012737 001010
 13790 052240 010506
 13791 052242 010137 052250
 13792 052246 000257
 13793
 13794 052250 075000
 13795
 13796
 13797 052252 000240
 13798 052254 104005
 13799
 13800 052256 005201

```

4$: INC R1 ;GENERATE NEW RSVD INSTR
    CMP #240,R1 ;AT END OF THIS GROUP ??
    BNE IS ;BR IF NOT

    MOV R5,SP ;MAKE SURE TO RESET THE SP
    MOV #5$,@#SLPERR ;LOOP FROM BEGINNING ON ERROR

;*****
;TEST 711 RSVD INSTRUCTION TEST - 007000 THRU 007777
;*****
TST711:
    SCOPE ;CALL THE SCOPE LOOP UTILITY
    MOV #711,R0 ;LOAD R0 WITH TEST NUMBER
    MOV SP,R5 ;SAVE THE SP
    MOV #4$,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
    CLR #1 ;
    MOV #1000,R1 ;SET UP FIRST ONE IN GROUP
    MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
    MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
    MOV R1,@#2$ ;LOAD NEW INSTR
    CCC ;SCOPE SYNC

2$: 007000 ;TEST THE RSVD INSTR - THIS LOCATION
    ;GETS CHANGED EACH PASS THROUGH

3$: ERROR 5 ;RSVD INSTR. IN R1 FAILED TO TRAP

4$: INC R1 ;GENERATE NEW RSVD INSTR
    CMP #10000,R1 ;AT END OF THIS GROUP ??
    BNE IS ;BR IF NOT

    MOV R5,SP ;MAKE SURE TO RESET THE SP
    MOV #5$,@#SLPERR ;LOOP FROM BEGINNING ON ERROR

;*****
;TEST 712 RSVD INSTRUCTION TEST - 075000 THRU 076777
;*****
TST712:
    SCOPE ;CALL THE SCOPE LOOP UTILITY
    MOV #712,R0 ;LOAD R0 WITH TEST NUMBER
    MOV SP,R5 ;SAVE THE SP
    MOV #4$,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
    CLR #1 ;
    MOV #75000,R1 ;SET UP FIRST ONE IN GROUP
    MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
    MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
    MOV R1,@#2$ ;LOAD NEW INSTR
    CCC ;SCOPE SYNC

2$: 75000 ;TEST THE RSVD INSTR - THIS LOCATION
    ;GETS CHANGED EACH PASS THROUGH

3$: ERROR 5 ;IN CASE NON TRAPPING INSTR IS TWO WORDS
    ;RSVD INSTR. IN R1 FAILED TO TRAP

4$: INC R1 ;GENERATE NEW RSVD INSTR

```

13801 052260 022701 076600
 13802 052264 001774
 13803 052266 022701 077000
 13804 052272 001362
 13805
 13806 052274 010506
 13807 052276 012737 052214 001010
 13808
 13809
 13810
 13811
 13812 052304
 13813 052306 000004
 13814 052310 012700 000713
 13815 052312 010605
 13816 052314 012737 052352 000010
 13817 052322 005037 000012
 13818 052326 012701 106400
 13819 052332 010737 001010
 13820 052336 010506
 13821 052340 010137 052346
 13822 052344 000257

```

    CMP #MED,R1 ;MED INSTRUCTION?
    BEQ 4$ ;BR IF YES--SKIP IT.
    CMP #077000,R1 ;AT END OF THIS GROUP ??
    BNE IS ;BR IF NOT

    MOV R5,SP ;MAKE SURE TO RESET THE SP
    MOV #5$,@#SLPERR ;LOOP FROM BEGINNING ON ERROR

;*****
;TEST 713 RSVD INSTRUCTION TEST - 106400 THRU 107777
;*****
TST713:
    SCOPE ;CALL THE SCOPE LOOP UTILITY
    MOV #713,R0 ;LOAD R0 WITH TEST NUMBER
    MOV SP,R5 ;SAVE THE SP
    MOV #4$,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
    CLR #1 ;
    MOV #106400,R1 ;SET UP FIRST ONE IN GROUP
    MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
    MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
    MOV R1,@#2$ ;LOAD NEW INSTR
    CCC ;SCOPE SYNC

```

```

13823
13824 052346 106400 2$: 106400 ;TEST THE RSVD INSTR - THIS LOCATION
13825 ;GETS CHANGED EACH PASS THROUGH
13826
13827 052350 104005 3$: ERROR 5 ;RSVD INSTR. IN R1 FAILED TO TRAP
13828
13829 052352 005201 4$: INC R1 ;GENERATE NEW RSVD INSTR
13830 052354 022701 106500 CMP #106500,R1 ;MFPD INSTRUCTION ??
13831 052360 001002 BNE 10$ ;BR IF NOT
13832 052362 012701 106700 MOV #106700,R1 ;SKIP MFPD AND MTPD INSTRUCTIONS
13833 052366 022701 110000 10$: CMP #110000,R1 ;AT END OF THIS GROUP ??
13834 052372 001361 BNE 1$ ;BR IF NOT
13835
13836 052374 010506 MOV R5,SP ;MAKE SURE TO RESET THE SP
13837 052376 012737 052312 001010 MOV #5,@#SLPERR ;LOOP FROM BEGINNING ON ERROR
13838 052404 012737 061122 000010 MOV #RSERR,@#10 ;RESTORE RSVD INSTR VECTOR
13839 052412 012737 000340 000012 MOV #340,@#12
13840 052420 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
13841
13842 ;THIS NEXT GROUP OF SEQUENTIAL TESTS VERIFIES THAT A "T" BIT
13843 ;TRAP CAN BE SERVICED IN EACH MICROWORD THAT DOES A "BUT SERVICE"
13844 ;EACH ROUTINE ENTERS THE TRAP MICROROUTINE WHEN THE TRAP IS SPRUNG
13845
13846 052422 012737 061070 000014 TSET: MOV #TBSER,@#14 ;SET UP THE "T" BIT TRAP VECTOR
13847 052430 012737 000340 000016 MOV #340,@#16 ;PRIORITY 7
13848
13849 ;*****
13850 ;*TEST 714 BUT SERVICE -- ONE WORD INSTRUCTIONS--ALL MODES -- FROM TABLE
13851 ;*INSTAB* (INSTRUCTION TABLE) CONTAINS ALL ONE WORD INSTRUCTIONS
13852 ;*THAT TEST A "BUT SERVICE" IN A UNIQUE ROM LOCATION. THE TABLE MUST
13853 ;*BE TERMINATED WITH A 0 ENTRY.
13854 ;*****
13855 ;*T714:
13856 052436 012700 000714 MOV #714,R0 ;LOAD R0 WITH TEST NUMBER
13857 052438 010605 SP,R5 ;SAVE THE SP
13858 052444 012704 063636 6$: MOV #INSTAB,R4 ;PUT POINTER TO TABLE IN R4
13859 052450 012401 4$: MOV (R4)+,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13860 052452 001422 BEQ 5$ ;EXIT TEST IF END OF TABLE
13861 052454 010737 001010 MOV PC,@#SLPERR ;LOOP ON FAILING INSTRUCTION ONLY
13862 052460 010437 052512 1$: MOV R1,@#2$ ;STORE TEST INSTRUCTION TO BE EXECUTED
13863 052464 012702 024312 MOV #MBUF1,R3 ;IN CASE DM1 DEST--(R2)
13864 052470 012703 063316 MOV #MBUF1,R3 ;IN CASE SM1--(R1)
13865 052474 010506 MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
13866 052476 012746 000020 MOV #2,-(SP) ;SET "T" BIT IN THE NEW PSW
13867 052502 012746 052512 MOV #2,-(SP) ;MAKE NEW PC = 2$
13868 052506 000257 CCC ;SCOPE SYNC
13869 052510 000006 RTT ;SET "T" BIT - GO TO 2$
13870
13871 052512 000240 2$: NOP ;INSTRUCTION FROM TABLE IS STORED HERE AND
13872 ;SHOULD SPRING TRAP
13873
13874 052514 104005 3$: ERROR 5 ;BUT SERVICE FAILED
13875
13876 052516 000754 BR 4$ ;GET NEXT INSTRUCTION FOR BUT SERVICE TEST
13877 052520 012737 052442 001010 5$: MOV #6$,@#SLPERR ;LOOP FROM BEGINNING ON ERROR
13878

```

```

13879
13880
13881 052526 000004 ;*****
13882 052526 000004 ;*TEST 715 BUT SERVICE TEST - (RTI)
13883 052530 012700 000715 ;*****
13884 052534 013701 052552 T$T715: SCOPE ;CALL THE SCOPE LOOP UTILITY
13885 052534 013701 052552 MOV #715,R0 ;LOAD R0 WITH TEST NUMBER
13886 052544 012746 000020 MOV #2$,-(SP) ;LOAD R1 WITH TEST INSTRUCTION WORD
13887 052550 000257 052554 MOV #3$,-(SP) ;SET "T" BIT IN THE NEW PSW
13888 052550 000257 CCC ;MAKE NEW PC = 3$
13889 052550 000257 ;SCOPE SYNC
13890
13891 052552 000002 2$: RTI ;INSTRUCTION SHOULD SPRING TRAP
13892
13893 052554 104005 3$: ERROR 5 ;BUT SERVICE IN XXX FAILED
13894
13895 ;*****
13896 ;*TEST 716 BUT SERVICE TEST - (JSR R,R)
13897 ;*****
13898 052556 000004 T$T716: SCOPE ;CALL THE SCOPE LOOP UTILITY
13899 052560 012700 000716 MOV #716,R0 ;LOAD R0 WITH TEST NUMBER
13900 052564 013701 052634 MOV #2$,-(SP) ;LOAD R1 WITH TEST INSTRUCTION WORD
13901 ;SBTTL USER CONTROLLED BREAKPOINT -- BIT14
13902 052570 032737 040000 063234 BIT #BIT14,@#RPTLOC ;BREAKPOINT HALT SET ??
13903 052576 001401 BEQ -+4 ;BR IF NOT
13904 052600 000000 HALT ;BREAK-DEPRESS CONTINUE TO CONTINUE
13905 052602 010605 MOV SP,R5 ;SAVE THE SP
13906 052604 010737 001010 MOV PC,@#SLPERR ;FOR PROPER SP RESETTING ON ERROR LOOP
13907 052610 010506 MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
13908 052612 012737 052640 063316 MOV #3,@#MBUF1 ;SET UP POINTER--DEST ADDR = 3$ FOR JSR
13909 052640 012746 000020 MOV #2,-(SP) ;SET "T" BIT IN THE NEW PSW
13910 052644 012746 052634 MOV #2,-(SP) ;MAKE NEW PC = 2$
13911 052630 000257 CCC ;SCOPE SYNC
13912 052632 000006 RTT ;SET "T" BIT - GO TO 2$
13913
13914 052634 004777 010456 2$: JSR PC,@#MBUF1 ;INSTRUCTION SHOULD SPRING TRAP
13915
13916 052640 104005 3$: ERROR 5 ;BUT SERVICE IN XXX FAILED
13917
13918 052642 000506 MOV R5,SP ;RESTORE SP IF ALL OK OR NOT LOOPING
13919
13920 ;*****
13921 ;*TEST 717 BUT SERVICE TEST - (JMP A)
13922 ;*****
13923 052644 000004 T$T717: SCOPE ;CALL THE SCOPE LOOP UTILITY
13924 052644 012700 000717 MOV #717,R0 ;LOAD R0 WITH TEST NUMBER
13925 052652 013701 052672 MOV #2$,-(SP) ;LOAD R1 WITH TEST INSTRUCTION WORD
13926 052656 012746 000020 MOV #2$,-(SP) ;SET "T" BIT IN THE NEW PSW
13927 052660 000257 052672 MOV #2$,-(SP) ;MAKE NEW PC = 2$
13928 052666 000257 CCC ;SCOPE SYNC
13929 052670 000006 RTT ;SET "T" BIT - GO TO 2$
13930
13931 052672 000167 000000 2$: JMP 3$ ;JMP INSTRUCTION SHOULD SPRING TRAP
13932
13933 052676 104005 3$: ERROR 5 ;BUT SERVICE IN XXX FAILED
13934

```

```

13935
13936
13937
13938
13939 052700
13940 052702 000004
13941 052706 012700 000720
13942 052712 012737 052734 063312
13943 052720 012746 000020
13944 052724 012746 052734
13945 052730 000257
13946 052732 000006
13947
13948 052734 000177 010352
13949
13950 052740 104005
13951
13952
13953
13954
13955 052742
13956 052744 000004
13957 052744 012700 000721
13958 052750 013701 053004
13959 052754 010605
13960 052756 010737 001010
13961 052762 010506
13962 052764 012746 053006
13963 052770 012746 000020
13964 052774 012746 053004
13965 053000 000257
13966 053002 000006
13967
13968 053004 000207
13969
13970 053006 104005
13971
13972
13973
13974
13975
13976
13977
13978
13979
13980
13981
13982
13983
13984
13985
13986
13987
13988
13989
13990

```

```

;*****
;TEST 720 BUT SERVICE TEST - (JMP @A)
;*****
TST720:
SCOPE
MOV #720,R0 ;CALL THE SCOPE LOOP UTILITY
;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV @#R0,R2 ;SET UP POINTER--DEST ADDR = 3$ FOR JMP
MOV #20,@#R2 ;SET "T" BIT IN THE NEW PSW
MOV #25,-(SP) ;MAKE NEW PC = 2$
CCC ;SCOPE SYNC
RTT ;SET "T" BIT - GO TO 2$

2$: JMP @R2 ;JMP INSTRUCTION SHOULD SPRING TRAP
3$: ERROR 5 ;BUT SERVICE IN XXX FAILED

;*****
;TEST 721 BUT SERVICE TEST - (RTS PC)
;*****
TST721:
SCOPE
MOV #721,R0 ;CALL THE SCOPE LOOP UTILITY
;LOAD R0 WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R2 ;SAVE THE SP
MOV @#R0,R3 ;FOR PROPER SP RESETTING ON ERROR LOOP
MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
1$: MOV #35,-(SP) ;RTS WILL LOAD PC WITH 3$
MOV #20,-(SP) ;SET "T" BIT IN THE NEW PSW
MOV #25,-(SP) ;MAKE NEW PC = 2$
CCC ;SCOPE SYNC
RTT ;SET "T" BIT - GO TO 2$

2$: RTS PC ;RTS INSTRUCTION SHOULD SPRING TRAP
3$: ERROR 5 ;BUT SERVICE IN XXX FAILED

;*****
;TEST 722 ALU ADD FUNCTION TEST
;THIS TEST VERIFIES THAT THE ALU ADD FUNCTION CAN RESPOND CORRECTLY
;TO THE 8 POSSIBLE COMBINATIONS THAT COULD OCCUR AT THE INPUTS OF
;EACH OF THE 16 BIT POSITIONS AS DESCRIBED BELOW:
;
; AIN BIN CIN
;
; 0 0 0
; 0 0 1
; 0 1 0
; 0 1 1
; 1 0 0
; 1 0 1
; 1 1 0
; 1 1 1
;
;THE TEST NO.S ALONG WITH THE CORRECT ANSWERS ARE STORED IN A TABLE
;TAGGED "ALUADD" AS SHOWN BELOW:

```

```

13991
13992
13993
13994
13995
13996
13997
13998
13999
14000
14001
14002
14003
14004
14005 053010
14006 053012 000004
14007 053016 012700 000722
14008 053022 010737 063340
14009 053026 024545 001010
14010
14011 053030 005725
14012 053032 012705 063416
14013 053036 011413
14014 053040 012501
14015 053042 012503
14016 053044 000257
14017
14018 053046 060103
14019
14020 053050 021503
14021 053052 001766
14022
14023 053054 011504
14024 053056 014503
14025 053060 104010
14026
14027 053062 005725
14028 053064 000761
14029
14030 053066 012737 053016 001010
14031
14032
14033
14034
14035
14036
14037
14038
14039
14040
14041
14042
14043
14044
14045
14046

```

```

;ALUADD: NULL
SRC OP1
DST OP1
SUM1
SRC OP2
DST OP2
SUM2
ETC.

;AFTER REPORTING THE ERROR THE ROUTINE WILL LOCK ON THE FAILING PAIR
;OF NO.S IF SW09=1 OR GO ON TO THE NEXT PAIR IF SW09=0.
;*****
TST722:
SCOPE
MOV #722,R0 ;CALL THE SCOPE LOOP UTILITY
;LOAD R0 WITH TEST NUMBER
1$: MOV #ALUADD+4,R5 ;R5 POINTS TO TABLE OF NO.S
MOV PC,@#R5 ;LOOP ONLY ON FAILING PAIR OF #'S
CMP -(R5),-(R5) ;RESET R5 TO POINT TO BAD GUYS
; (OR NULL ENTRY FIRST TIME THROUGH)
4$: TST (R5)+ ;POINT TO A SRC OP
CMP #ALUADD+62,R5 ;DONE ALL NO.S IN TABLE ?
BEQ 5$ ;BR IF YES
MOV (R5)+,R1 ;LOAD SRC OP
MOV (R5)+,R3 ;LOAD DEST OP
CCC ;SCOPE SYNC

2$: ADD R1,R3 ;TEST THE ADD FUNCTION
CMP (R5),R3 ;CORRECT SUM ?
BEQ 4$ ;GO ADD NEXT PAIR IF YES

3$: MOV (R5),R4 ;GET S / B SUM
MOV -(R5),R2 ;GET DEST OP
ERROR 10 ;ALU ADD OPERATION FAILED

5$: TST (R5)+ ;CORRECT R5 POINTER
BR 4$ ;GO DO NEXT PAIR

5$: MOV #1$,@#R5 ;LOOP FROM BEGINNING ON ERROR

;*****
;TEST 723 ALU SUB FUNCTION TEST
;THIS TEST VERIFIES THAT THE ALU ADD FUNCTION CAN RESPOND CORRECTLY
;TO THE 8 POSSIBLE COMBINATIONS THAT COULD OCCUR AT THE INPUTS OF
;EACH OF THE 16 BIT POSITIONS AS DESCRIBED BELOW:
;
; AIN BIN CIN
;
; 0 0 0
; 0 0 1
; 0 1 0
; 0 1 1
; 1 0 0
; 1 0 1
; 1 1 0
; 1 1 1
;

```


14047
14048
14049
14050
14051
14052
14053
14054
14055
14056
14057
14058
14059
14060
14061
14062
14063
14064
14065
14066
14067
14068
14069
14070
14071
14072
14073
14074
14075
14076
14077
14078
14079
14080
14081
14082
14083
14084
14085
14086
14087
14088
14089
14090
14091
14092
14093
14094
14095
14096
14097
14098
14099
14100
14101
14102

053074 000004
053074 012700
053102 012705 000723
053106 010737 063560
053112 024545 001010
053114 005725
053116 022705 063636
053122 001413
053124 012501
053126 012503
053130 000257
053132 160103
053134 021503
053136 001766
053140 011504
053142 014502
053144 104010
053146 005725
053150 000761
053152 012737 053102 001010

```

; THE TEST NO.S ALONG WITH THE CORRECT ANSWERS ARE STORED IN A TABLE
; TAGGED "ALUADD" AS SHOWN BELOW:
;
;ALUSUB:      NULL
;             SRC OP1
;             DST OP1
;             DIFF1
;             SRC OP2
;             DST OP2
;             DIFF2
;             ETC.
;AFTER REPORTING THE ERROR THE ROUTINE WILL LOCK ON THE FAILING PAIR
; OF NO.S IF SW09=1 OR GO ON TO THE NEXT PAIR IF SW09=0
;*****
TST723:
SCOPE
MOV #723,R0 ;CALL THE SCOPE LOOP UTILITY
;LOAD R0 WITH TEST NUMBER
MOV #ALUSUB+4,R5 ;R5 POINTS TO TABLE OF NO.S
PC,#$LPERR ;R5 POINTS TO TABLE OF NO.S
CMP -(R5),-(R5) ;LOOP ONLY ON FAILING PAIR OF #'S
;RESET R5 TO POINT TO BAD GUYS
;OR NULL ENTRY FIRST TIME THROUGH)
;POINT TO A SRC OP
;DONE ALL NO.S IN TABLE ?
;BR IF YES
MOV (R5)+,R1 ;LOAD SRC OP
MOV (R5)+,R3 ;LOAD DEST OP
CCC ;SCOPE SYNC
2S: SUB R1,R3 ;TEST THE SUB FUNCTION
CMP (R5),R3 ;CORRECT DIFF?
BEQ 4S ;GO SUB NEXT PAIR IF YES
3S: MOV (R5),R4 ;GET S / B DIFF
MOV -(R5),R2 ;GET DEST OP
ERROR 10 ;ALU SUB OPERATION FAILED
TST (R5)+ ;CORRECT R5 POINTER
BR 4S ;GO DO NEXT PAIR
5S: MOV #1$,@$LPERR ;LOOP FROM BEGINNING ON ERROR
;*****
;TEST 724 ALU "AND" FUNCTION TEST USING BIC INSTRUCTION
;THIS TEST VERIFIES THAT THE ALU "AND" FUNCTION RESPONDS CORRECTLY
;TO ALL POSSIBLE COMBINATIONS FOR EACH OF THE 16 BIT POSITIONS
;IT EXECUTES THE BIC INSTRUCTION FOR THE FOLLOWING PAIRS OF
;OPERANDS AND TESTS FOR THE INDICATED RESULT:
;SOURCE OP DEST. OP RESULT
;000000 000000 000000
;177777 177777 000000

```

14103
14104
14105
14106
14107
14108
14109
14110
14111
14112
14113
14114
14115
14116
14117
14118
14119
14120
14121
14122
14123
14124
14125
14126
14127
14128
14129
14130
14131
14132
14133
14134
14135
14136
14137
14138
14139
14140
14141
14142
14143
14144
14145
14146
14147
14148
14149
14150
14151
14152
14153
14154
14155
14156
14157
14158

053160 000004
053160 012700
053166 012705 000724
053172 010737 063420
053176 024545 001010
053200 005725
053202 022705 063476
053206 001413
053210 012501
053212 012503
053214 000257
053216 040103
053220 020315
053222 001766
053224 011504
053226 014502
053230 104010
053232 005725
053234 000761
053236 012737 053166 001010

```

;*****
;THE 8 PAIRS OF NO.S AND THE ANSWERS ARE STORED IN A TABLE TAGGED
;"ANDTAB" IN THE FOLLOWING PATTERN:
;ANDTAB:      NULL
;             SRC OP1
;             DST OP1
;             ANS1
;             SRC OP2
;             DST OP2
;             ANS2
;             ETC.
;AFTER REPORTING THE ERROR THE ROUTINE WILL LOCK ON THE FAILING
; PAIR OF NO.S IF SW09=1 OR GO ON TO TEST THE NEXT PAIR IF SW09=0
;*****
TST724:
SCOPE
MOV #724,R0 ;CALL THE SCOPE LOOP UTILITY
;LOAD R0 WITH TEST NUMBER
MOV #ANDTAB+4,R5 ;R5 POINTS TO TABLE OF TEST NO.S
PC,#$LPERR ;R5 POINTS TO TABLE OF TEST NO.S
CMP -(R5),-(R5) ;LOOP ONLY ON FAILING PAIR OF #'S
;RESET R5 TO POINT TO BAD GUYS
;OR NULL ENTRY FIRST TIME THROUGH)
;POINT TO A SOURCE OPR
;DONE ALL COMBINATIONS ?
;BR IF YES
MOV (R5)+,R1 ;LOAD THE SRC OP
MOV (R5)+,R3 ;LOAD THE DEST OP
CCC ;SCOPE SYNC
2S: BIC R1,R3 ;TEST THE "AND"
CMP R3,(R5) ;RESULT CORRECT ?
BEQ 4S ;BR IF YES - GET THE NEXT PAIR
3S: MOV (R5),R4 ;GET THE S / B DATA
MOV -(R5),R2 ;GET DEST OP
ERROR 10 ;ALU "AND" FAILED
TST (R5)+ ;CORRECT R5 POINTER
BR 4S ;GO GET NEXT PAIR
5S: MOV #1$,@$LPERR ;LOOP FROM BEGINNING ON ERROR
;*****
;TEST 725 ALU "OR" FUNCTION TEST USING BIS INSTRUCTION
;THIS TEST VERIFIES THAT THE ALU "OR" FUNCTION RESPONDS CORRECTLY
;TO ALL POSSIBLE COMBINATIONS FOR EACH OF THE 16 BIT POSITIONS
;IT EXECUTES THE BIS INSTRUCTION FOR THE FOLLOWING PAIRS OF
;OPERANDS AND TESTS FOR THE INDICATED RESULT:

```

```

14159 ;SOURCE OP DEST. OP RESULT
14160 ;000000 000000 000000
14161 ;177777 177777 177777
14162 ;000000 177777 177777
14163 ;177777 000000 177777
14164 ;125252 052525 125252
14165 ;052525 125252 052525
14166 ;125252 052525 177777
14167 ;052525 125252 177777
14168 ;THE 8 PAIRS OF NO.S AND THE ANSWERS ARE STORED IN A TABLE TAGGED
14169 ;"ORTAB" IN THE FOLLOWING PATTERN:
14170 ;ORTAB: NULL
14171 ;
14172 ; SRC OP1
14173 ; DST OP1
14174 ; ANS1
14175 ; SRC OP2
14176 ; DST OP2
14177 ; ANS2
14178 ; ETC.
14179 ;
14180 ;AFTER REPORTING THE ERROR THE ROUTINE WILL LOCK ON THE FAILING
14181 ;PAIR OF NO.S IF SW09=1 OR GO ON TO TEST THE NEXT PAIR IF SW09=0
14182 ;*****
14183 ;*****
14184 ;T725:
14185 SCOPE ;CALL THE SCOPE LOOP UTILITY
14186 MOV #725,R0 ;LOAD R0 WITH TEST NUMBER
14187 MOV ORTAB+4,R5 ;R5 POINTS TO TABLE OF TEST NO.S
14188 MOV PC,@#SLPERR ;LOOP ONLY ON FAILING PAIR OF #S
14189 CMP -(R5),-(R5) ;RESET R5 TO POINT TO BAD GUYS
14190 ;OR NULL ENTRY FIRST TIME THROUGH)
14191 TST (R5)+ ;POINT TO A SOURCE OPR
14192 CMP ORTAB+62,R5 ;DONE ALL COMBINATIONS ?
14193 BEO 5 ;BR IF YES
14194 MOV (R5)+,R1 ;LOAD THE SRC OP
14195 MOV (R5)+,R3 ;LOAD THE DEST OP
14196 CCC ;SCOPE SYNC
14197 ;
14198 BIS R1,R3 ;TEST THE "OR"
14199 ;
14200 CMP R3,(R5) ;RESULT CORRECT ?
14201 BEQ 4$ ;BR IF YES - GET THE NEXT PAIR
14202 ;
14203 MOV (R5),R4 ;GET THE S / B DATA
14204 MOV -(R5),R2 ;GET DEST OP
14205 ERROR 10 ;ALU "OR" FAILED
14206 ;
14207 TST (R5)+ ;CORRECT R5 POINTER
14208 BR 4$ ;GO GET NEXT PAIR
14209 ;
14210 MOV #1$,@#SLPERR ;LOOP FROM BEGINNING ON ERROR
14211 ;*****
14212 ;*TEST 726 INC / DEC / ADD TEST - CYCLE NO.S 000000-077777
14213 ;THIS TEST COMBINES THE INC / DEC / ADD INSTRUCTIONS IN THE FOLLOWING
14214

```

```

14215 ;TEST SEQUENCE:
14216 ;1. BOTH SOURCE AND DEST OPS ARE ZEROED
14217 ;2. THE TWO NO.S ARE ADDED AND THE RESULT COMPARED WITH 000000
14218 ;3. THE SOURCE OP IS INCREMENTED
14219 ;4. THE DEST OP IS DECREMENTED
14220 ;5. STEPS 2,3, AND 4 ARE REPEATED UNTIL THE SOURCE OP GOES
14221 ; NEGATIVE
14222 ;
14223 ;ON DETECTION OF A NON-ZERO RESULT THE ERROR IS REPORTED AND THEN IF:
14224 ;
14225 ; 1. SW09=0 THE TEST IS EXITED
14226 ; 2. SW09=1 THE ROUTINE LOCKS ON THE FAILING PAIR OF OPERANDS
14227 ;*****
14228 ;*****
14229 ;T726:
14230 SCOPE ;CALL THE SCOPE LOOP UTILITY
14231 MOV #726,R0 ;LOAD R0 WITH TEST NUMBER
14232 CLR R1 ;INITIALIZE REGS TO 000000
14233 CLR R2
14234 CLR R3
14235 CLR R4
14236 MOV PC,@#SLPERR ;LOOP ONLY ON FAILING PAIR OF #S
14237 MOV R2,R3 ;LOAD DEST OPERAND
14238 CCC ;SCOPE SYNC
14239 ;
14240 ADD R1,R3 ;ADD THE TWO TEST NO.S
14241 ;RESULT S / B = 000000
14242 ;
14243 CMP R4,R3 ;RESULT = 000000 ?
14244 BEQ 4$ ;BR IF YES
14245 ;
14246 ERROR 10 ;INCORRECT RESULT IN R3
14247 BR TST727 ;;EXIT TO NEXT TEST
14248 ;
14249 INC R1 ;ADD 1 TO SOURCE OP
14250 BMI 5$ ;GET OUT IF IT WENT NEGATIVE
14251 DEC R2 ;SUB 1 FROM THE DEST OP
14252 BR 1$ ;GO ADD THE TWO NO.S
14253 ;
14254 MOV #10$,@#SLPERR ;LOOP FROM BEGINNING ON ERROR
14255 ;*****
14256 ;*TEST 727 INC / DEC / ADD TEST - CYCLE NO.S 077777-000000
14257 ;THIS TEST COMBINES THE INC / DEC / ADD INSTRUCTIONS IN THE FOLLOWING
14258 ;TEST SEQUENCE:
14259 ;1. BOTH SOURCE AND DEST OPS ARE ZEROED
14260 ;2. THE TWO NO.S ARE ADDED AND THE RESULT COMPARED WITH 000000
14261 ;3. THE SOURCE OP IS INCREMENTED
14262 ;4. THE DEST OP IS DECREMENTED
14263 ;5. STEPS 2,3, AND 4 ARE REPEATED UNTIL THE DEST. OP GOES
14264 ; NEGATIVE
14265 ;
14266 ;ON DETECTION OF A NON-ZERO RESULT THE ERROR IS REPORTED AND THEN IF:
14267
14268
14269

```

```

14271 ; 1- SW09=0 THE TEST IS FAILED
14272 ; SW09=1 THE ROUTINE LOCKS ON THE FAILING PAIR OF OPERANDS
14273 ;*****
14274 ;TST727: SCOPE ;CALL THE SCOPE LOOP UTILITY
14275 MOV #727,R0 ;LOAD R0 WITH TEST NUMBER
14276 CLR R1 ;GET TEST INSTRUCTION WORD
14277 CLR R2 ;S/B RESULT IN R2
14278 CLR R3 ;S/B RESULT IN R3
14279 CLR R4 ;INITIALIZE REG
14280 MOV PC,#SLPERR ;LOOP ONLY ON FAILING PAIR OF #'S
14281 MOV R2,R3 ;LOAD DEST OPERAND
14282 CCC ;SCOPE SYNC
14283 053430 060103 2$: ADD R1,R3 ;ADD THE TWO TEST NO.S
14284 ;RESULT S / B = 000000
14285 053432 020403 CMP R4,R3 ;RESULT = 000000 ?
14286 053434 001402 BEQ 4$ ;BR IF YES
14287 053436 104010 3$: ERROR 10 ;INCORRECT RESULT IN R3
14288 053440 000407 BR TST730 ;GO TO SCOPE EXIT
14289 053442 005202 4$: INC R2 ;ADD 1 TO DEST. OP
14290 053444 100402 BMI 5$ ;GET OUT IF IT WENT NEGATIVE
14291 053446 005301 DEC R1 ;SUB 1 FROM THE SOURCE OP
14292 053450 000765 BR 1$ ;GO ADD THE TWO NO.S
14293 053452 012737 053412 001010 5$: MOV #10$,#SLPERR ;LOOP FROM BEGINNING ON ERROR
14294 ;*****
14295 ;TST730: MUL RA,RB TEST ; N:C = 1111
14296 ;*****
14297 ;TST730: SCOPE ;CALL THE SCOPE LOOP UTILITY
14298 MOV #730,R0 ;LOAD R0 WITH TEST NUMBER
14299 MOV #2$,#STMPO ;GET TEST INSTRUCTION WORD
14300 CLR R1 ;S/B RESULT IN R2
14301 CLR R2 ;S/B RESULT IN R3
14302 MOV #6,R4 ;INITIALIZE REG
14303 MOV #2,R2 ;INITIALIZE REG + 1
14304 CLR R3 ;INITIALIZE SRC
14305 MOV #3,R5 ;SCOPE SYNC
14306 SCC ;SCOPE SYNC
14307 053516 070205 2$: MUL R5,R2 ;TEST THE MUL
14308 ;N:C=0000?
14309 053520 100403 BMI 3$
14310 053522 001402 BEQ 3$
14311 053524 102401 BVS 3$
14312 053526 103001 BCC 4$
14313 053530 104044 3$: ERROR 44 ;COND CODES SET IMPROPERLY
14314 053532 020304 4$: CMP R3,R4 ;REG+1 CORRECT?
14315 053534 001002 BNE 5$ ;BR IF NOT
14316 053536 020102 CMP R1,R2 ;REG CORRECT?
14317
14318
14319
14320
14321
14322
14323
14324
14325
14326

```

```

14327 053540 001401 BEQ TST731 ;BR IF YES
14328 053542 104045 5$: ERROR 45 ;MUL DELIVERED WRONG RESULT
14329 ;*****
14330 ;TST731: MUL (RA),RB TEST ; N:C = 0000-SET C
14331 ;*****
14332 ;TST731: SCOPE ;CALL THE SCOPE LOOP UTILITY
14333 MOV #731,R0 ;LOAD R0 WITH TEST NUMBER
14334 MOV #2$,#STMPO ;GET TEST INSTRUCTION WORD
14335 CLR R1 ;S/B RESULT IN R2
14336 CLR R2 ;S/B RESULT IN R3
14337 MOV #12345,R4 ;INITIALIZE REG
14338 MOV #12345,R2 ;INITIALIZE REG + 1
14339 CLR R3 ;SET UP POINTER TO SRC
14340 MOV #10,(R5) ;INITIALIZE SRC
14341 CCC ;SCOPE SYNC
14342 053604 000257 2$: MUL (R5),R2 ;TEST THE MUL
14343 ;N:C=0001?
14344 053610 100403 BMI 3$
14345 053612 001402 BEQ 3$
14346 053614 102401 BVS 3$
14347 053616 103401 BCS 4$
14348 053620 104044 3$: ERROR 44 ;COND CODES SET IMPROPERLY
14349 053622 020304 4$: CMP R3,R4 ;REG+1 CORRECT?
14350 053624 001002 BNE 5$ ;BR IF NOT
14351 053626 020102 CMP R1,R2 ;REG CORRECT?
14352 053630 001401 BEQ TST732 ;BR IF YES
14353 053632 104045 5$: ERROR 45 ;MUL DELIVERED WRONG RESULT
14354 ;*****
14355 ;TST732: MUL (RA)+,RB TEST ; N:C = 0000-SET Z
14356 ;*****
14357 ;TST732: SCOPE ;CALL THE SCOPE LOOP UTILITY
14358 MOV #732,R0 ;LOAD R0 WITH TEST NUMBER
14359 MOV #2$,#STMPO ;GET TEST INSTRUCTION WORD
14360 CLR R1 ;S/B RESULT IN R2
14361 CLR R2 ;S/B RESULT IN R3
14362 MOV #-1,R3 ;INITIALIZE REG
14363 MOV #MBUFO,R5 ;SET UP POINTER TO SRC
14364 MOV #10,(R5) ;INITIALIZE SRC
14365 CCC ;SCOPE SYNC
14366 053674 070225 2$: MUL (R5)+,R2 ;TEST THE MUL
14367 ;N:C=0100?
14368 053676 100403 BMI 3$
14369 053678 001002 BNE 3$
14370 053680 102401 BVS 3$
14371 053682 103001 BCC 4$
14372 053684 000004 000731 001076 000732 001076
14373 053686 012700 053674 000732 001076
14374 053688 013737 053674 000732 001076
14375 053690 005001 053674 000732 001076
14376 053692 005004 053674 000732 001076
14377 053694 005002 053674 000732 001076
14378 053696 012703 177777 053674 000732 001076
14379 053698 012705 063312 053674 000732 001076
14380 053700 012715 000010 053674 000732 001076
14381 053702 000257 053674 000732 001076
14382 053704 070225 053674 000732 001076

```

```

14383 053706 104044 3$: ERROR 44 ;COND CODES SET IMPROPERLY
14384
14385
14386 053710 020304 4$: CMP R3,R4 ;REG+1 CORRECT?
14387 053712 010102 ;BNE IF NOT
14388 053714 020102 ;CMP R1,R2 ;REG CORRECT?
14389 053716 001401 ;BEQ G5 ;BR IF YES
14390
14391 053720 104045 5$: ERROR 45 ;MUL DELIVERED WRONG RESULT
14392
14393 053722 022705 063314 6$: CMP #MBUF0+2,R5 ;DID R5 GET AUTO-INCREMENTED?
14394 053726 001401 ;BEQ TST733 ;BR IF YES
14395
14396 053730 104046 ERROR 46 ;AUTO INCREMENT DID NOT OCCUR
14397
;*****
;TEST 733 MUL @-(RA)+,RB TEST ; N:C = 0000-SET N ; SRC,DST = -,+
;*****
TST733: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #733,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,@#STMP0 ;GET TEST INSTRUCTION WORD
MOV #-1,R1 ;S/B RESULT IN R2
MOV #-10,R4 ;S/B RESULT IN R3
MOV #-1,R2 ;INITIALIZE REG
CLR R3 ;INITIALIZE REG + 1
MOV #ATA+10,R5 ;SET UP POINTER TO POINTER TO MBUF0
MOV #-10,@#MBUF0 ;INITIALIZE SRC
CCC ;SCOPE SYNC
2$: MUL @-(R5)+,R2 ;TEST THE MUL
BPL 3$ ;N:C=1000?
BEQ 3$
BVS 3$
BCC 4$
14402 053732 000004 3$: ERROR 44 ;COND CODES SET IMPROPERLY
14403 053734 012700 000733 001076 MOV #733,R0 ;LOAD R0 WITH TEST NUMBER
14404 053736 013737 054000 MOV #25,@#STMP0 ;GET TEST INSTRUCTION WORD
14405 053746 012701 177777 MOV #-1,R1 ;S/B RESULT IN R2
14406 053752 012704 177777 MOV #-10,R4 ;S/B RESULT IN R3
14407 053756 012702 000001 MOV #-1,R2 ;INITIALIZE REG
14408 053762 005003 CLR R3 ;INITIALIZE REG + 1
14409 053764 012705 063306 MOV #ATA+10,R5 ;SET UP POINTER TO POINTER TO MBUF0
14410 053770 012702 177770 MOV #-10,@#MBUF0 ;INITIALIZE SRC
14411 053776 000257 CCC ;SCOPE SYNC
14412
14413 054000 070235 2$: MUL @-(R5)+,R2 ;TEST THE MUL
14414
14415 054002 100003 BPL 3$ ;N:C=1000?
14416 054004 01402 BEQ 3$
14417 054006 10401 BVS 3$
14418 054010 103001 BCC 4$
14419
14420 054012 104044 3$: ERROR 44 ;COND CODES SET IMPROPERLY
14421
14422 054014 020304 4$: CMP R3,R4 ;REG+1 CORRECT?
14423 054016 001002 ;BNE IF NOT
14424 054020 020102 ;CMP R1,R2 ;REG CORRECT?
14425 054022 001401 ;BEQ G5 ;BR IF YES
14426
14427 054024 104045 5$: ERROR 45 ;MUL DELIVERED WRONG RESULT
14428
14429 054026 022705 063310 6$: CMP #ATA+12,R5 ;DID R5 GET AUTO-INCREMENTED?
14430 054032 001401 ;BEQ TST734 ;BR IF YES
14431
14432 054034 104046 ERROR 46 ;AUTO INCREMENT DID NOT OCCUR
14433
;*****
;TEST 734 MUL -(RA),RB TEST ; N:C = 1111-CLR ALL BUT N ; SRC,DSK = +,-
;*****
TST734: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

14439 054040 012700 000734 MOV #734,R0 ;LOAD R0 WITH TEST NUMBER
14440 054044 013737 054104 MOV #25,@#STMP0 ;GET TEST INSTRUCTION WORD
14441 054052 012701 177777 MOV #-1,R1 ;S/B RESULT IN R2
14442 054056 012704 177777 MOV #-10,R4 ;S/B RESULT IN R3
14443 054062 012702 177777 MOV #-1,R2 ;INITIALIZE REG
14444 054066 005003 CLR R3 ;INITIALIZE REG + 1
14445 054070 012705 063314 MOV #MBUF0+2,R5 ;SET UP POINTER TO SRC
14446 054074 012702 000010 MOV #-10,@#MBUF0 ;INITIALIZE SRC
14447 054102 000277 SCC ;SCOPE SYNC
14448
14449 054104 070245 2$: MUL -(R5),R2 ;TEST THE MUL
14450
14451 054106 100003 BPL 3$ ;N:C=1000?
14452 054110 01402 BEQ 3$
14453 054112 102401 BVS 3$
14454 054114 103001 BCC 4$
14455
14456 054116 104044 3$: ERROR 44 ;COND CODES SET IMPROPERLY
14457
14458 054120 020304 4$: CMP R3,R4 ;REG+1 CORRECT?
14459 054122 001002 ;BNE IF NOT
14460 054124 020102 ;CMP R1,R2 ;REG CORRECT?
14461 054126 001401 ;BEQ G5 ;BR IF YES
14462
14463 054130 104045 5$: ERROR 45 ;MUL DELIVERED WRONG RESULT
14464
14465 054132 022705 063312 6$: CMP #MBUF0,R5 ;DID SRC REG GET AUTO-DECREMENTED?
14466 054136 001401 ;BEQ TST735 ;BR IF YES
14467
14468 054140 104046 ERROR 46 ;AUTO DECREMENT DID NOT OCCUR
14469
;*****
;TEST 735 MUL @-(RA),RB TEST ; N:C = 1111-CLR ALL BUT C ; SRC,DST = -,-
;*****
TST735: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #735,R0 ;LOAD R0 WITH TEST NUMBER
MOV #25,@#STMP0 ;GET TEST INSTRUCTION WORD
CLR R1 ;S/B RESULT IN R2
MOV #106420,R4 ;S/B RESULT IN R3
MOV #-2,R2 ;INITIALIZE REG
MOV #-1,R3 ;INITIALIZE REG + 1
MOV #ATA+12,R5 ;SET UP POINTER TO POINTER TO MBUF0
MOV #-43210,@#MBUF0 ;INITIALIZE SRC
CCC ;SCOPE SYNC
2$: MUL @-(R5),R2 ;TEST THE MUL
BMI 3$ ;N:C=0001?
BEQ 3$
BVS 3$
BCS 4$
14474 054142 000004 3$: ERROR 44 ;COND CODES SET IMPROPERLY
14475 054144 012700 000735 001076 MOV #735,R0 ;LOAD R0 WITH TEST NUMBER
14476 054150 013737 054210 MOV #25,@#STMP0 ;GET TEST INSTRUCTION WORD
14477 054156 005001 CLR R1 ;S/B RESULT IN R2
14478 054160 012704 106420 ;S/B RESULT IN R3
14479 054164 012702 177776 MOV #-2,R2 ;INITIALIZE REG
14480 054170 012703 177777 MOV #-1,R3 ;INITIALIZE REG + 1
14481 054174 012705 063310 MOV #ATA+12,R5 ;SET UP POINTER TO POINTER TO MBUF0
14482 054178 012702 134570 MOV #-43210,@#MBUF0 ;INITIALIZE SRC
14483 054206 000277 SCC ;SCOPE SYNC
14484
14485 054210 070255 2$: MUL @-(R5),R2 ;TEST THE MUL
14486
14487 054212 100403 BMI 3$ ;N:C=0001?
14488 054214 001402 BEQ 3$
14489 054216 102401 BVS 3$
14490 054220 103401 BCS 4$
14491
14492 054222 104044 3$: ERROR 44 ;COND CODES SET IMPROPERLY
14493
14494 054224 020304 4$: CMP R3,R4 ;REG+1 CORRECT?

```

```

14495 054226 001002
14496 054230 020102
14497 054232 001401
14498
14499 054234 104045
14500
14501 054236 022705 063306
14502 054242 001401
14503
14504 054244 104046
14505
14506
14507
14508
14509
14510 054246 000004
14511 054250 012700 000736
14512 054254 013737 054310 001076
14513 054252 005001
14514 054264 005004
14515 054266 012702 012345
14516 054272 012703 177777
14517 054276 012705 063312
14518 054302 005065 000002
14519 054306 000277
14520
14521 054310 070265 000002
14522
14523
14524 054314 100403
14525 054316 001002
14526 054320 102401
14527 054322 103001
14528
14529 054324 104044
14530
14531 054326 020304
14532 054330 001002
14533 054332 020102
14534 054334 001401
14535
14536 054336 104045
14537
14538
14539
14540 054340 000004
14541 054342 012700 000737
14542 054344 013737 054404 001076
14543 054354 005001
14544 054356 012704 000100
14545 054362 012703 000010
14546 054366 012705
14547 054370 012705 063276
14548 054374 012737 000010 063312
14549 054402 000257
14550

```

```

BNE 55 ;BR IF NOT
CMP R1,R2 ;REG CORRECT?
BEQ 65 ;BR IF YES
5$: ERROR 45 ;MUL DELIVERED WRONG RESULT
6$: CMP #ATA+10,R5 ;DID R5 GET AUTO-DECREMENTED?
BEQ TST736 ;BR IF YES
ERROR 46 ;AUTO INCREMENT DID NOT OCCUR
;*****
;TEST 736 MUL X(RA),RB TEST ; N:C = 1111 TO 0100
;*****
TST736: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #736,R0 ;LOAD R0 WITH TEST NUMBER
CLR #25,@#STMP0 ;GET TEST INSTRUCTION WORD
R1 ;S/B RESULT IN R2
R4 ;S/B RESULT IN R3
MOV #012345,R2 ;INITIALIZE REG
R3 ;INITIALIZE REG + 1
MBOF0,R5 ;SET UP POINTER TO SRC
CLR 2(R5) ;INITIALIZE SRC
SCC ;SCOPE SYNC
2$: MUL 2(R5),R2 ;TEST THE MUL
BNE 35 ;N:C=0100?
BVS 35
BCC 45
3$: ERROR 44 ;COND CODES SET IMPROPERLY
4$: CMP R3,R4 ;REG+1 CORRECT?
BNE 55 ;BR IF NOT
CMP R1,R2 ;REG CORRECT?
BEQ TST737 ;BR IF YES
5$: ERROR 45 ;MUL DELIVERED WRONG RESULT
;*****
;TEST 737 MUL @X(RA),RB TEST
;*****
TST737: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #737,R0 ;LOAD R0 WITH TEST NUMBER
CLR #25,@#STMP0 ;GET TEST INSTRUCTION WORD
R1 ;S/B RESULT IN R2
R4 ;S/B RESULT IN R3
MOV #10,R2 ;INITIALIZE REG
R2 ;INITIALIZE REG + 1
ATA,R5 ;SET UP POINTER TO TABLE OF POINTERS
MOV #10,@#MBOF0 ;INITIALIZE SRC
CCC ;SCOPE SYNC

```

```

14551
14552 054404 070275 000010
14553
14554 054410 020304
14555 054412 001002
14556 054414 020102
14557 054416 001401
14558
14559 054420 104045
14560
14561
14562
14563 054422 000004
14564 054424 012700 000740
14565 054430 013737 054456 001076
14566 054436 012701 010000
14567 054442 012704 000001
14568 054446 012703
14569 054456 005002 020001
14570 054454 000277
14571
14572 054456 071227 000002
14573
14574
14575 054462 100403
14576 054464 001402
14577 054466 102401
14578 054470 103001
14579
14580 054472 104044
14581
14582
14583 054474 020304
14584 054476 001002
14585 054500 020102
14586 054502 001401
14587
14588 054504 104045
14589
14590
14591
14592 054506 000004
14593 054508 012700 000741
14594 054514 013737 054544 001076
14595 054522 012701 177775
14596 054526 012704 177775
14597 054532 012703 177775
14598 054536 012705 177762
14599 054542 000257
14600
14601 054544 071227 000004
14602
14603
14604 054550 100003
14605 054552 001402
14606

```

```

2$: MUL @10(R5),R2 ;TEST THE MUL
CMP R3,R4 ;REG+1 CORRECT?
BNE 35 ;BR IF NOT
CMP R1,R2 ;REG CORRECT?
BEQ TST740 ;BR IF YES
3$: ERROR 45 ;MUL DELIVERED WRONG RESULT
;*****
;TEST 740 DIV #N,RA TEST ; N:C = 1111
;*****
TST740: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #740,R0 ;LOAD R0 WITH TEST NUMBER
CLR #25,@#STMP0 ;GET COPY OF TEST INSTRUCTION
R1 ;S/B RES IN R2
R4 ;S/B RES IN R3
MOV #010000,R1 ;S/B RES IN R2
R2 ;S/B RES IN R3
CLR R2 ;SET UP REG OPERAND
MOV #020001,R3 ;SET UP REG+1 OP
SCC ;SCOPE SYNC
2$: DIV #2,R2 ;TEST DIV
BNE 35 ;N:C=0000?
BVS 35
BCC 45
3$: ERROR 44 ;COND CODES SET IMPROPERLY
4$: CMP R3,R4 ;CORRECT RESULT IN REG+1?
BNE 55 ;BR IF NOT
CMP R1,R2 ;CORRECT RESULT IN REG?
BEQ TST741 ;BR IF YES
5$: ERROR 45 ;DIV DELIVERED WRONG RESULT
;*****
;TEST 741 DIV #N,RA TEST ; RA NEGATIVE ; N:C = 0000
;*****
TST741: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #741,R0 ;LOAD R0 WITH TEST NUMBER
CLR #25,@#STMP0 ;GET COPY OF TEST INSTRUCTION
R1 ;S/B RES IN R2
R3 ;S/B RES IN R2
R4 ;S/B RES IN R3
MOV #1,R2 ;SET UP REG OPERAND
R2 ;SET UP REG+1 OP
MOV #14,R3 ;SET UP REG+1 OP
SCC ;SCOPE SYNC
2$: DIV #4,R2 ;TEST DIV
BNE 35 ;N:C=1000?
BVS 35
BCC 45

```

```

14607 054554 102401      BVS      3$
14608 054556 103001      BCC      4$
14609
14610 054560 104044      3$:      ERROR      44      ;COND CODES SET IMPROPERLY
14611
14612 054562 020304      4$:      CMP      R3,R4      ;CORRECT RESULT IN REG+1?
14613 054564 001002      BNE      5$      ;BR IF NOT
14614 054566 020102      CMP      R1,R2      ;CORRECT RESULT IN REG?
14615 054570 001401      BEQ      T$T742      ;BR IF YES
14616
14617 054572 104045      5$:      ERROR      45      ;DIV DELIVERED WRONG RESULT
14618
14619
14620 *****
14621 ;*TEST 742 DIV #N,RA TEST ; N:C = 0000 TO 0100
14622 ;*****
14623 T$T742:
14624 SCOPE      ;CALL THE SCOPE LOOP UTILITY
14625 MOV      #742,R0      ;LOAD R0 WITH TEST NUMBER
14626 MOV      #2$,@$TMP0    ;GET COPY OF TEST INSTRUCTION
14627 CLR      R1      ;S/B RES IN R2
14628 MOV      #1,R4      ;S/B RES IN R3
14629 CLR      R2      ;SET UP REG OPERAND
14630 MOV      #1,R3      ;SET UP REG+1 OP
14631 CCC      ;SCOPE SYNC
14632
14633 2$:      DIV      #2,R2      ;TEST DIV
14634
14635 BMI      3$      ;N:C=0100?
14636 BNE      3$
14637 BVS      4$
14638 BCC      4$
14639
14640 3$:      ERROR      44      ;COND CODES SET IMPROPERLY
14641
14642 4$:      CMP      R3,R4      ;CORRECT RESULT IN REG+1?
14643 BNE      5$      ;BR IF NOT
14644 CMP      R1,R2      ;CORRECT RESULT IN REG?
14645 BEQ      T$T743      ;BR IF YES
14646
14647 5$:      ERROR      45      ;DIV DELIVERED WRONG RESULT
14648
14649 *****
14650 ;*TEST 743 DIV #N,RA TEST ; RA POS
14651 ;*****
14652 T$T743:
14653 SCOPE      ;CALL THE SCOPE LOOP UTILITY
14654 MOV      #743,R0      ;LOAD R0 WITH TEST NUMBER
14655 MOV      #2$,@$TMP0    ;GET COPY OF TEST INSTRUCTION
14656 CLR      R1      ;S/B RES IN R2
14657 MOV      #2,R4      ;S/B RES IN R3
14658 CLR      R2      ;SET UP REG OPERAND
14659 MOV      #14,R3      ;SET UP REG+1 OP
14660 CCC      ;SCOPE SYNC
14661
14662 2$:      DIV      #-4,R2      ;TEST DIV

```

```

14663 054716 020304      CMP      R3,R4      ;CORRECT RESULT IN REG+1?
14664 054720 001002      BNE      3$      ;BR IF NOT
14665 054722 020102      CMP      R1,R2      ;CORRECT RESULT IN REG?
14666 054724 001401      BEQ      T$T744      ;BR IF YES
14667
14668 054726 104045      3$:      ERROR      45      ;DIV DELIVERED WRONG RESULT
14669
14670 *****
14671 ;*TEST 744 DIV TEST - V BIT GETS SET
14672 ;* THIS TEST TESTS THAT THE V BIT CAN BE SET IN ALL THE
14673 ;* POSSIBLE WAYS. SINCE THE INSTRUCTION SHOULD BE ABORTED, THE
14674 ;* RESULTS CANNOT BE GUARANTEED. FOR THIS REASON, ONLY
14675 ;* THE CONDITION CODES ARE CHECKED.
14676 ;*****
14677 T$T744:
14678 SCOPE      ;CALL THE SCOPE LOOP UTILITY
14679 MOV      #744,R0      ;LOAD R0 WITH TEST NUMBER
14680 MOV      #2$,@$TMP0    ;LOAD R1 WITH TEST INSTRUCTION WORD
14681 MOV      #2,R4      ;S/B PSW
14682 CLR      #0,PSW      ;CLEAR OUT OTHER PSW BITS
14683 MOV      #50,R2      ;SET UP REG OP
14684
14685 2$:      DIV      #5,R2      ;TEST DIV -- SHOULD ABORT
14686
14687 BMI      3$      ;N:C=0010?
14688 BEQ      3$
14689 BVC      3$
14690 BCS      3$
14691
14692 MOV      #-1,R2      ;INITIALIZE REG OP
14693 CLR      R3      ;INITIALIZE REG+1 OP
14694
14695 DIV      #-2,R2      ;TEST DIV -- SHOULD ABORT
14696
14697 BMI      3$      ;N:C=0010?
14698 BEQ      3$
14699 BVC      3$
14700 BCS      3$
14701
14702 MOV      #3,R4      ;S/B PSW
14703
14704 DIV      #0,R2      ;TEST DIV BY 0 -- SHOULD ABORT
14705
14706 BMI      3$      ;N:C=0010?
14707 BNO      3$
14708 BVC      3$
14709 BCS      T$T745      ;IF ALL OK, THEN EXIT TEST
14710
14711 3$:      MOV      @PSW,R3      ;GET WAS PSW
14712 MOV      #PSW,R2      ;DESTINATION IS PSW
14713
14714 ERROR      1      ;CONDITION CODES SET WRONG
14715
14716 *****
14717 ;*TEST 745 ASH #N,RA TEST ; SHIFT LEFT ; N:C = 0000 TO 1010
14718 ;*****

```

```

14719 055046
14720 055046 000004
14721 055050 012700 000745
14722 055054 013701 055074
14723 055060 012704 123456
14724 055064 012703 112345
14725 055070 000257
14726
14727 055072 072327 000003
14728
14729 055076 100003
14730 055100 001402
14731 055102 102001
14732 055104 103001
14733
14734 055106 104002
14735
14736 055110 020304
14737 055112 001401
14738 055114 104002
14739
14740
14741
14742
14743
14744 055116 000004
14745 055120 012700 000746
14746 055124 013701 055142
14747 055128 015004
14748 055132 012703 000004
14749 055136 000257
14750 055140 000270
14751
14752 055142 072327 177775
14753
14754 055146 100403
14755 055150 001002
14756 055152 102401
14757 055154 103401
14758
14759 055156 104002
14760
14761 055160 020304
14762 055162 001401
14763 055164 104002
14764
14765
14766
14767
14768 055166 000004
14769 055166 012700 000747
14770 055174 013701 055212
14771 055200 012704 172345
14772 055204 012703 123432
14773 055210 000277
14774

```

```

TST745: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #745,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;GET TEST INSTRUCTION WORD
MOV #123450,R4 ;S/B RESULT
MOV #112345,R3 ;INITIAL REG
CCC ;SCOPE SYNC

2$: ASH #3,R3 ;TEST THE ASH

BPL 3$ ;N:C=1010?
BEQ 3$
BVC 3$
BCC 4$

3$: ERROR 2 ;INCORRECT CONDITION CODES

4$: CMP R3,R4 ;CORRECT RESULT?
BEQ TST746 ;BR IF YES
ERROR 2 ;ASH DELIVERED WRONG RESULT

;*****
;TEST 746 ASH #N,RA TEST ; SHIFT RIGHT ; N:C = 1000 TO 0101
;*****
TST746: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #746,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #4,R3 ;S/B RESULT
MOV #4,R3 ;INITIAL REG
CCC ;SCOPE SYNC
SEN ;CODES = 1000

2$: ASH #-3,R3 ;TEST THE ASH

BMI 3$ ;N:C=0101?
BNE 3$
BVS 3$
BCS 4$

3$: ERROR 2 ;INCORRECT CONDITION CODES

4$: CMP R3,R4 ;CORRECT RESULT?
BEQ TST747 ;BR IF YES
ERROR 2 ;ASH DELIVERED WRONG RESULT

;*****
;TEST 747 ASH #N,RA TEST ; SHIFT LEFT ; N:C = 1111 TO 1000
;*****
TST747: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #747,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #17234,R4 ;S/B RESULT
MOV #123432,R3 ;INITIAL REG
SCC ;SCOPE SYNC

```

```

14775
14776 055212 072327 177772
14777
14778 055216 100003
14779 055220 001402
14780 055222 102401
14781 055224 103001
14782
14783 055226 104002
14784
14785 055230 020304
14786 055232 001401
14787 055234 104002
14788
14789
14790
14791
14792 055236 000004
14793 055236 012700 000750
14794 055240 013701 055274 001076
14795 055244 013701 123456
14796 055252 012701 076530
14797 055256 012704 112345
14798 055262 012703 147653
14799 055266 000257
14800 055272 000270
14801
14802 055274 073227 000003
14803
14804 055300 100003
14805 055302 001402
14806 055304 102001
14807 055306 103001
14808
14809 055310 104044
14810
14811 055312 020102
14812 055314 001002
14813 055316 020403
14814 055320 001401
14815 055322 104045
14816
14817
14818
14819
14820 055324 000004
14821 055324 012700 000751
14822 055345 013701 055356 001076
14823 055340 005001
14824 055342 005004
14825 055344 005002
14826 055346 012703 000005
14827 055352 000257
14828 055354 000270
14830

```

```

2$: ASH #-6,R3 ;TEST THE ASH

BPL 3$ ;N:C=1000?
BEQ 3$
BVS 3$
BCC 4$

3$: ERROR 2 ;INCORRECT CONDITION CODES

4$: CMP R3,R4 ;CORRECT RESULT?
BEQ TST750 ;BR IF YES
ERROR 2 ;ASH DELIVERED WRONG RESULT

;*****
;TEST 750 ASHC #N,RA TEST ; SHIFT LEFT ; N:C = 0000 TO 1010
;*****
TST750: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #750,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,0($STMP0) ;GET TEST INSTRUCTION WORD
MOV #123456,R1 ;S/B RES IN R2
MOV #076530,R4 ;S/B RES IN R3
MOV #112345,R2 ;INITIALIZE COMBINED
MOV #147653,R3 ;REGISTERS
CCC ;SCOPE SYNC

2$: ASHC #3,R2 ;TEST ASHC

BPL 3$ ;N:C=1010?
BEQ 3$
BVC 3$
BCC 4$

3$: ERROR 44 ;COND CODES WRONG

4$: CMP R1,R2 ;TOP HALF OF RESULT CORRECT?
BNE 5$ ;BR IF NOT
CMP R4,R3 ;LOWER HALF OF RESULT CORRECT?
BEQ TST751 ;BR IF YES
ERROR 45 ;ASHC DELIVERED WRONG RES

;*****
;TEST 751 ASHC #N,RA TEST ; SHIFT RIGHT ; N:C = 1000 TO 0101
;*****
TST751: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #751,R0 ;LOAD RO WITH TEST NUMBER
MOV #25,0($STMP0) ;GET TEST INSTRUCTION WORD
MOV #1,R1 ;S/B RES IN R2
MOV #4,R4 ;S/B RES IN R3
MOV #5,R3 ;INITIALIZE COMBINED
MOV #5,R3 ;REGISTERS
CCC ;SCOPE SYNC
SEN ;CODES = 1000

```

14831 055356 073227 177775
14832
14833 055362 100403
14834 055368 012703
14835 055374 102401
14836 055370 103401
14837
14838 055372 104044
14839
14840 055374 020102
14841 055376 001002
14842 055400 020403
14843 055402 001401
14844 055404 104045
14845
14846
14847
14848
14849 055406
14850 055406 000004
14851 055410 012700 000752
14852
14853 055414 032737 100000 063234
14854 055422 001401
14855 055424 000000
14856 055426 013737 055456 001076
14857 055434 012704 177234
14858 055440 012704 135275
14859 055444 012702 123456
14860 055450 012703 127542
14861 055454 000257
14862
14863 055456 073227 177772
14864
14865 055462 100003
14866 055464 001402
14867 055466 102401
14868 055470 103401
14869
14870 055472 104044
14871
14872 055474 020102
14873 055476 001002
14874 055500 020403
14875 055502 001401
14876 055504 104045
14877
14878
14879
14880
14881
14882
14883
14884
14885
14886

2S: ASHC #-3,R2 ;TEST ASHC
BMI 3S ;N:C=0101?
BNE 3S
BVS 3S
BCS 4S
3S: ERROR 44 ;COND CODES WRONG
4S: CMP R1,R2 ;TOP HALF OF RESULT CORRECT?
BNE 5S ;BR IF NOT
CMP R4,R3 ;LOWER HALF OF RESULT CORRECT?
BEQ 5S ;BR IF YES
5S: ERROR 45 ;ASHC DELIVERED WRONG RES

;TEST 752 ASHC #N,RA TEST ; SHIFT RIGHT ; N:C = 1111 TO 1000

TST752: SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #752,R0 ;LOAD R0 WITH TEST NUMBER
;SBTTL USER CONTROLLED BREAKPOINT -- BIT15
BIT #BIT15,@#BPTLOC ;BREAKPOINT HALT SET ??
BEQ +4 ;BR IF NOT
HALT ;BREAK-DEPRESS CONTINUE TO CONTINUE
MOV @#26,@#STMP0 ;GET TEST INSTRUCTION WORD
MOV #135275,R1 ;S/B RES IN R1
MOV #123456,R4 ;S/B RES IN R4
MOV #123456,R2 ;INITIALIZE COMBINED
MOV #127542,R3 ; REGISTERS
CCC ;SCOPE SYNC
2S: ASHC #-6,R2 ;TEST ASHC
BPL 3S ;N:C=1000?
BEQ 3S
BVS 3S
RCS 4S
3S: ERROR 44 ;COND CODES WRONG
4S: CMP R1,R2 ;TOP HALF OF RESULT CORRECT?
BNE 5S ;BR IF NOT
CMP R4,R3 ;LOWER HALF OF RESULT CORRECT?
BEQ 5S ;BR IF YES
5S: ERROR 45 ;ASHC DELIVERED WRONG RES

; THIS SECTION OF THE MED TESTS EXERCISES CERTAIN SCRATCH
; PAD REGISTERS USING MED READS AND WRITES. THEIR ORIGINAL
; CONTENTS ARE RESTORED BUT:
***** IMPORTANT NOTE *****
* THE CONSOLE MUST NOT III BE USED DURING THESE MED *
* TESTS. NO INTERRUPTS OR TRAPS CAN BE ALLOWED EITHER*

14887
14888
14889
14890
14891
14892
14893
14894
14895
14896
14897
14898 055506
14899 055506 012700 000752
14900 055512 000004
14901 055514 012737 000304 177770
14902 055522 012737 140000 177776
14903 055530 012706 001000
14904 055534 012737 055566
14905 055532 012737 055566 000010
14906 055550 012701 177799
14907 055554 005000
14908 055556 076600
14909 055560 000041
14910 055562 104012
14911 055564 104044
14912 055566 005700
14913 055570 001401
14914 055572 104013
14915
14916 055574 022626
14917 055576 012737 061220 000004
14918 055604 012737 061122 000010
14919
14920 055612 005037 177776
14921 055616 076600
14922 055620 000041
14923 055622 103403
14924 055624 102402
14925 055626 100401
14926 055630 001001
14927 055632 104014
14928
14929
14930
14931
14932
14933
14934
14935
14936
14937
14938
14939
14940
14941 055634
14942 055634 012700 000753

;TEST 753 CHECK MED IS ILLEGAL IN USER - EXECUTES IN KERNAL
; THE NEXT TEST BELOW CHECKS TO SEE THAT THE "MED"
; (MAINTENANCE, EXAM, AND DEPOSIT) INSTRUCTION WILL EXECUTE
; WHEN IN KERNEL MODE WITHOUT AFFECTING THE PSW AND
; THAT IT IS ILLEGAL IN USER MODE

TST753: MOV #753,R0 ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
;CALL THE SCOPE LOOP UTILITY
SCOPE ;SET SCOPE SYNC FOR MED INSTR
MOV #304,@#UBREAK ;GO TO USER MODE
MOV #140000,@#PSW ;SETUP USER STACK PTR
MOV #2,@#ERRVEC ;SET ERROR TRAP VECTOR TO 2S BELOW
MOV #2,@#RESVEC ;LOAD RESERVED INST. TRAP VECTOR
MOV #-1,R1 ;LOAD R1 WITH A -1
CLR R0 ;CLEAR R0
MED ;TRY TO DO MAINT. EXAMINE
;WORD 041 ;MED READ CODE FOR R1
ERRR 4S ;ERROR - MED INST. NOT ILLEGAL IN USER
BR 4S
2S: TST R0 ;IS R0 UNCHANGED?
BEQ 3S ;BRANCH IF YES
ERRR 13 ;ERROR - MED INSTRUCTION WAS EXECUTED
;BEFORE TRAPPING
;CLEAN UP STACK
3S: CMP (SP)+,(SP)+ ;RESTORE ERROR TRAP VECTOR
4S: MOV #ERRR,@#ERRVEC ;RESTORE RESERVED INST. TRAP VECTOR
MOV #RESERR,@#RESVEC ;RESTORE RESERVED INST. TRAP VECTOR
MED0: CLR @#PSW ;GO TO KERNEL MODE,CLEAR COND. CODES
MED ;DO MAINT. EXAMINE OF R1
;WORD 041 ;MED READ CODE FOR R1
RCS MEDHLT ;MED READ CODE FOR R1
BVS MEDHLT
BMI MEDHLT
BNE +4
MEDHLT: ERRR 14 ;ERROR CC-BITS IN PSW AFFECTED BY MED

;TEST 754 MED TEST - R/W DATA PATTERNS TO REGS
; THIS PARTICULAR MED TEST WRITES DATA PATTERNS
; TO THOSE INTERNAL REGS. WHICH CAN BE WRITTEN
; AND READ WITHOUT SPECIAL CONSIDERATIONS. REGISTERS
; REQUIRING SPECIAL TESTS ARE TESTED IN LATER
; MED TESTS.
; TABLE II CONTAINS THE REGISTER ADDRESSES.
; A MAX. OF 3 ERRORS ARE REPORTED FOR EACH LOC.

TST754: MOV #753,R0 ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE


```

14943 055640 000004          SCOPE
14944 055642 012737 000340 177776 MEDT1: MOV #340,@PSW ;CALL THE SCOPE LOOP UTILITY
14945 055650 012701 064166 MOV #BL3,R1 ;KERNEL MODE-PRIORITY 7
14946 055654 012737 152522 001102 1$: MOV (R1),@#13S ;INITIALIZE ADDRESS POINTER
14947 055662 112137 055752 MOVB (R1),@#13S ;PUT WRITE CODE BY "WRITE-MED" S
14948 055666 112137 055752 MOVB (R1),@#13S ;AND POINT R1 TO READ CODE
14949 055672 111137 055710 MOVB (R1),@#10S ;PUT READ CODE BY "READ-MED" S
14950 055676 112137 055736 MOVB (R1),@#12S ;R1 NOW POINTS TO NEXT REG.
14951 055672 005037 001106 CLR @#STMP4 ;CLEAR ERROR COUNTER
14952 055670 000000 2$: MED ;MED-READ THE INTERNAL REG.
14953 055700 000000 10$: .WORD 0 ;MED-READ CODE
14954 055712 010037 001076 MOV RO,@#STMP0 ;SAVE ITS ORIGINAL CONTENTS
14955 055716 010137 001100 MOV R1,@#STMP1 ;SAVE ADDR. PTR. VALUE
14956 055722 013700 001102 MOV @#STMP2,RO ;LOAD RO WITH DATA TO BE WRITTEN
14957 055726 076600 MED ;MED-WRITE THE TEST DATA
14958 055730 000000 11$: CLR RO ;MED-WRITE CODE
14959 055732 005000 MOV RO ;CLEAR RO
14960 055734 076600 MED ;MED-READ THE DATA BACK
14961 055736 000000 12$: .WORD 0 ;MED-READ CODE
14962 055740 010037 001104 MOV RO,@#STMP3 ;SAVE DATA READ FOR COMPARISON
14963 055744 013700 001076 MOV @#STMP0,RO ;LOAD ORIGINAL DATA IN RO
14964 055750 076600 MED ;MED-WRITE ORG. DATA TO REG.
14965 055752 000000 13$: .WORD 0 ;MED-WRITE CODE
14966 055754 023737 001102 001104 CMP @#STMP2,@#STMP3 ;DID DATA READ=DATA WRITTEN?
14967 055762 001412 001100 BEQ 3$ ;BRANCH IF YES
14968 055764 013737 055736 001100 MOV @#12,@#STMP1 ;SAVE MED-CODE FOR ERROR
14969 055766 005401 000003 001106 BNE 14$ ;MAX. ERROR REPORTS YET?
14970 056000 052401 14$: BR 14$ ;BRANCH IF YES
14971 056002 104022 22: ERROR ;INT. REG. READ BACK WRONG DATA
14972 056004 005237 001106 14$: INC @#STMP4 ;INCREMENT ERROR COUNTER
14973 056010 005137 001102 3$: COM @#STMP2 ;CHANGE DATA PATTERN
14974 056014 013701 001100 MOV @#STMP1,R1 ;RESTORE ADDR. POINTER

```

```

14975 056020 022737 125252 001102 CMP #125252,@#STMP2 ;BOTH DATA PATTERNS BEEN USED?
14976 056026 061327 BNE 2$ ;BRANCH IF NO
14977 056030 005711 TST (R1) ;END OF ADDR. TABLE?
14978 056032 001310 BNE 1$ ;BRANCH IF NO
14979
14980 ;*****
14981 ;* TEST 755 MED TEST - VERIFY NOPS; READ R7 IN A & B SP
14982 ;*
14983 ;* THIS TEST CHECKS ALL OF THE "NOP" OPERATION CODES
14984 ;* TO ENSURE THEY WILL EXECUTE AS NOP'S AND
14985 ;* NOT RESULT IN A PROCESSOR HANG. THE "NOPS"
14986 ;* TABLE (TABLE III) HOLDS THESE CODES.
14987 ;* THIS TEST ALSO READS THE PROGRAM COUNTER (R7) VALUES
14988 ;* STORED IN A & B SCRATCH PADS TO SEE THAT THEY
14989 ;* READ PROPERLY. THE R7 ADDRESSES ARE IN TABLE IV.
14990 ;*****
14991 ;* TEST 755:
14992 ;*
14993 056034 012700 000754 MOV #754,RO ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
14994 056040 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
14995 056042 012701 064404 MEDT3: MOV #BL3,R1 ;INITIALIZE NOP TABLE PTR. (R1)
14996 056046 112137 056054 1$: MOVB (R1),@#10$ ;PLACE FIRST "NOP-CODE" AFTER MED
14997 ;AND POINT R1 TO LAST CODE IN GROUP
14998 056052 076600 5$: MED ;EXECUTE MED WITH NOP OP-CODE
14999 056054 000000 10$: .WORD 0
15000 056056 123711 056054 CMPB @#10$(,R1) ;HAVE ALL NOPS IN THAT GROUP
15001 ;BEEN TESTED?
15002 056062 103003 BHIS 6$ ;BRANCH IF YES
15003 056064 005477 INC @#10$ ;NEXT NOP IN GROUP
15004 056070 000770 BR 5$
15005 056072 105721 6$: TSTB (R1)+ ;POINT R1 TO NEXT NOP GROUP
15006 056074 005711 TST (R1) ;HAVE ALL GROUPS BEEN TESTED
15007 056076 001363 BNE 1$ ;BRANCH IF NO
15008
15009 056100 113737 064425 056112 MEDT4: MOVB @#R7A+1,@#5$ ;LOAD R7A READ CODE AFTER MED
15010 056106 005000 4$: CLR RO ;CLEAR RO
15011 056110 076600 MED ;MED READ R7 IN THE ASP
15012 056112 000000 5$: .WORD 0 ;READ CODE FOR R7A
15013 056114 020027 056114 CMP RO,#5$+2 ;DID R7A READ CORRECTLY?
15014 056120 001411 BEQ 6$ ;BRANCH IF YES
15015 056122 013737 056112 001100 MOV @#5$,@#STMP1 ;SAVE MED-CODE FOR ERROR
15016 056124 012737 056114 001102 MOV #5$+2,@#STMP2 ;SAVE DATA EXPECTED
15017 056136 010037 001104 MOV RO,@#STMP3 ;SAVE DATA RECEIVED
15018 056142 104022 22: ERROR ;R7A DID NOT READ THE RIGHT VALUE
15019 056144 023727 056112 000047 6$: CMP @#5$,#47 ;HAS R7B BEEN CHECKED?
15020 056152 001404 BEQ 8$ ;BRANCH IF YES
15021 056154 113737 064431 056112 MOVB @#R7B+1,@#5$ ;LOAD R7B READ CODE AFTER MED
15022 056164 000751 4$: BR 4$ ;TEST R7 BSP
15023
15024 ;*****
15025 ;* TEST 756 MED TEST - CSP CONSTANTS CHECK
15026 ;*
15027 ;* THIS TEST CHECKS THE CONSTANT VALUES LOCATED
15028 ;* IN THE C SCRATCH PAD. THE CONSTANTS ARE READ
15029
15030

```

15031
15032
15033
15034
15035
15036 056164 012700 000755
15037 056164 000004
15038 056170
15039
15040 056172 076600
15041 056174 000144
15042 056176 052700 004000
15043 056202 076600
15044 056204 000344
15045 056206 170000
15046
15047 056210 012701 064536
15048 056214 012167 000006
15049 056220 001414
15050 056222 005000
15051 056224 076600
15052 056226 000000
15053 056230 020021
15054 056232 001770
15055 056234 013737 056226 001100
15056 056242 016137 177776 001102
15057 056250
15058 056252
15059
15060
15061
15062
15063
15064
15065
15066
15067
15068
15069
15070
15071
15072
15073
15074
15075
15076
15077
15078 056252 012700 000756
15079 056252 000004
15080 056252 000004 000071 177770
15081 056252 012737 061104 000004
15082 056266 012737 000340 000006
15083 056274 012737 000340 000006
15084 056302 005037 061112
15085 056306 076600
15086 056310 000022

```

** WITH A MED INSTRUCTION AND COMPARED TO THEIR
** EXPECTED VALUE. THE ADDRESSES OF THESE CONSTANTS
** AND THE VALUES EXPECTED ARE IN TABLE VII.
*****
**ST756:
MOV #755,R0 ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;CALL THE SCOPE LOOP UTILITY
MED
RDFLAG
BIS #BIT11,R0 ;SET THE "CSP INVALID BIT" IN FLAG REG.
MED
WRFLAG
MEDT10: CFCC ;EXECUTE FLT. PT INST. SO FLT. PT.
CONSTANTS ARE LOADED INTO CSP
MOV #TBL7,R1 ;SETUP TABLE POINTER
MOV (R1),R1 ;LOAD MED READ CODE AT 15
BEQ R1 ;BR IF END OF TABLE
CLR R0
MED
WORD 0 ;READ INTERNAL CONTENTS INTO R0
CMP R0,(R1)+ ;WAS THE CONSTANT READ THE ONE EXPECTED
BEQ R1 ;BRANCH IF YES
MOV @R1,@#STMP1 ;SAVE MEDCODE FOR ERROR
MOV -2(R1),@#STMP2 ;SAVE CONSTANT VALUE EXPECTED
ERROR 21 ;CSP LOCATION HELD WRONG VALUE

```

```

*****
**TEST 757 MED TEST - MICROBK CHECK OF MICRO-POINTS
**
** THIS TEST USES THE MICROBREAK REGISTER AND THE
** INFORMATION IN TABLE V TO CHECK THAT THE
** CORRECT MED-FLOW IS ENTERED WHEN EACH
** REGISTER IS ACCESSED BY A MED INSTRUCTION.
** THE MICROBREAK REG. IS SETUP TO CAUSE A TRAP TO
** LOC 4 WHEN ITS CONTENTS EQUAL THE ADDRESS
** OF THE MICROWORD BEING EXECUTED.
**
** NOTE: THE MICRO BREAK - TRAP-TO-4 CAPABILITY
** IS TRIED AT THE BEGINNING OF THE TEST.
** IF IT DOESN'T WORK, AN ERROR IS PRINTED
** AND THE TEST IS SKIPPED
*****

```

```

**ST757:
MOV #756,R0 ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;CALL THE SCOPE LOOP UTILITY
MEDT11: #SWB01,@#UBREAK ;LOAD MICROBK. REG. WITH AN MICRO ADDR.
MOV #RKROUT,@#4 ;LOAD ADDR. OF MICROBK. ROUTINE IN 4
MOV #340,@#6 ;LOAD KERNEL PSW - PRIORITY 7 IN 6
CLR @#BKFLAG ;CLEAR MICROBK. TRAP FLAG
MED ;GET WHAMI INTO R0
RDWHAMI

```

15087 056312 052700 001000
15088 056316 076600
15089 056320 000222
15090 056322 076600
15091 056324 000144
15092 056326 052700 100000
15093 056328 076600
15094 056334 000344
15095 056336 000300
15096 056340 005737 061112
15097 056344 001007
15098 056346 005037 001076
15099 056354 016737 121513 001100
15100 056360 104015
15101 056362 000453
15102
15103 056364 012701 000710
15104 056370 076600
15105 056372 000103
15106 056374 042700 100007
15107 056400 020001
15108 056402 001401
15109 056404 104025
15110 056406 012701 064436
15111 056412 012701 064464
15112 056416 010737 001010
15113 056422 111137 056460
15114 056426 001431
15115 056430 011237 177770
15116 056434 005037 061112
15117 056440 076600
15118 056442 000144
15119 056444 052700 100000
15120 056450 076600
15121 056452 000344
15122 056454 005000
15123
15124 056456 076600
15125 056460 000000
15126 056462 005737 061112
15127 056466 001006
15128 056470 013737 056460 001076
15129 056476 013737 001100
15130 056502 104015
15131
15132 056504 105721
15133 056506 005722
15134 056510 000744
15135
15136 056512 076600
15137 056514 000022
15138 056516 042700 001000
15139 056524 076600
15140 056526 000344
15141 056526 076600
15142 056530 000222

```

10$: RIS #BIT9,R0 ;SET BIT 9
WRD ;MED-WRITE THE WHAMI REG TO
RDWHAMI ;ENABLE MICROBK-TRAP-TO-4
MED ;GET FLAG REGISTER
11$: BIS #BIT15,R0 ;SET BIT 15 IN R0
WRFLAG ;MED-WRITE THE FLAG REG TO
SWAB R0 ;ENABLE MICROBK TRAPPING
TST @#BKFLAG ;MICROBK TRAP SHOULD OCCUR ON SWAB
BNE R1 ;DID TRAP TO 4 OCCUR?
CLR @#STMP0 ;BRANCH IF YES
ERROR 15 ;SAVE EXPECTED UBREAK ADDR
BR 50$ ;MICROBREAK TRAP DIDN'T WORK
15$: MOV #SWB01*10,R1 ;GET CORRECT U-ADDR
MED ;GET LOG CUA REG
RDLCUA
BIC #100007,R0 ;GET RID OF IRRELEVANT BITS
CMP R0,R1 ;WAS CORRECT UADDR LOGGED?
BEQ R1 ;BR IF YES
ERROR 35 ;CUA CONTAINS INCORRECT U-ADDR
MOV #TBL5,R1 ;INITIALIZE TABLE PTR. (R1)
MOV #TBL6,R2
MOV PC,@#SLPERR ;SET ERROR LOOP RETURN TO 2$
MOV (R1),@#12$ ;LOAD WRITE CODE AFTER MED
REQ 50$ ;BR IF END OF TABLE
MOV (R2),@#UBREAK ;LOAD MICROBK REG. WITH MICROADDR.
CLR @#BKFLAG ;CLEAR MICROBK TRAP-TO-4 FLAG
MED ;GET FLAG REGISTER
15$: BIS #BIT15,R0 ;SET BIT 15 IN R0
WRFLAG ;MED WRITE TO FLAG REG TO
CLR R0 ;ENABLE MICROBK TRAPPING
;IN CASE U-BREAK TRAP DOESN'T OCCUR
;USUALLY BETTER TO WRITE 0'S
12$: .WORD 0
TST @#BKFLAG ;DID WE TRAP-TO-4? (FLAG NOT = 0)
BNE R1 ;BRANCH IF YES TO NEXT ENTRY
MOV @#12$,@#STMP0 ;SAVE MED-CODE FOR ERROR
MOV (R1),@#STMP1 ;SAVE EXPECTED U-ADDR FOR ERROR
ERROR 15 ;MICROBK. TRAP-TO-4 DID NOT OCCUR
20$: TSTB (R1)+ ;INCREMENT TO NEXT TABLE
TST (R2)+ ;ENTRIES AND
BR 2$ ;CONTINUE
50$: RDWHAMI ;GET WHAMI INTO R0
BIC #BIT9,R0 ;CLEAR THE FLAG REG. TO
MED ;DISABLE MICROBK. TRAPPING
WRFLAG ;CLEAR THE WHAMI REG. TO
MED ;DISABLE MICROBK. TRAP-TO-4
RDWHAMI
13$:
14$:

```

```

15143 056532 012737 056260 001010
15144 056540 012737 061720 001004
15145 056546 012737 000304 177770
15146
15147
15148
15149
15150
15151
15152
15153
15154
15155
15156
15157 056554 012700 000757
15158 056560 000004
15159 056562 012737 056622 000004 15:
15160 056570 012737 000340 000006
15161 056576 012700 100001
15162 056602 076600
15163 056604 000222
15164 056606 012702 056563
15165 056612 005767 177745
15166
15167 056616 104023
15168 056620 000441
15169 056622 022626
15170 056624 012737 061220 000004 25:
15171 056632 076600
15172 056634 000100
15173 056638 012700 177766
15174 056642 032701 000100
15175
15176 056646 001001
15177 056650 104024
15178
15179
15180 056652 032700 100004 35:
15181 056656 001001
15182 056660 104024
15183
15184
15185 056662 005005 45:
15186 056664 076600
15187 056666 000102
15188 056670 010003
15189 056672 020002
15190
15191 056674 001401
15192 056676 005205
15193 056700 076600 55:
15194 056702 000101
15195 056704 000300
15196 056706 042700 177774
15197 056712 001002
15198 056714 005705

```

```

MOV #MEDT11, @#SLPERR ;RESET LOOP ON ERROR POINTER
MOV #ERR, @#4 ;RESTORE NORMAL ERROR ROUTINE
MOV #30A, @#BREAK ;GENERATE SYNC PULSE ON MED INSTR

;*****
;*TEST 760 PHYSICAL ADDRESS & ODD ADDRESS ERROR LOGGING
;*THIS TEST CHECKS THAT THE PROPER PHYSICAL ADDRESS BITS
;* <17:00> ARE LOGGED UPON ERROR. THE ERROR IS CAUSED BY
;* FORCING AN ODD ADDRESS TRAP. THE ERROR LOG MODE USED
;* IS "LOG FIRST". ALSO, THE ODD ADDRESS ERROR BITS IN
;* THE LOG JAM AND CPU ERROR REGISTER ARE CHECKED.
;*****
45T760:
MOV #757,R0 ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #2S, @#4 ;SETUP PC FOR ODD ADDR SERVICE
MOV #340, @#6
MED #BIT15+BIT0, R0 ;SETUP "LOG FIRST" MODE
WRWHAMI
MOV #1S+1, R2 ;SAVE ADDRESS OF ODD ADDR. INSTRUCTION
TST 1S+1 ;DO ODD ADDRESS INSTRUCTION TO FORCE
; A JUMPUP & TRAP TO 4
;*** ODD ADDRESS ERROR
ERROR 23 ;EXIT TEST
BR 10S ;RESTORE STACK
CMP (SP)+, (SP)+ ;RESTORE OLD PC & PSW
MOV #BERR, @#4
MED
RDLJAM
MOV @#CPUERR, R1
BIT #BIT6, R1 ;WAS ODD ADDR. ERROR RECORDED BY
; THE CPU ERROR REGISTER?
RNE 3S ;BRANCH IF YES
RNROR 24 ;*** CPU ERROR REG. DID NOT
; REPORT ODD ADDRESS ERROR
; READ THE LOG JAM REGISTER
BIT #BIT15+BIT2, R0 ;WAS ODD ADDR. ERROR LOGGED BY LOG JAM
RNE 4S ;BRANCH IF YES
RNROR 24 ;*** LOG JAM REG. DID NOT LOG
; ODD ADDRESS ERROR CORRECTLY
45:
CLR R5 ;CLR ERROR FLAG
MED ;READ THE LOG PBA REGISTER
RDLPRBA
MOV R0, R3 ;SAVE RECEIVED PHYS ADDR <15:0>
CMP R0, R2 ;WERE BITS <15:00> OF THE PHYSICAL
; BUS ADDR. LOGGED CORRECTLY?
RNE 5S ;BRANCH IF YES
INC R5 ;SET ERROR FLAG
MED ;READ THE LOG SERVICE REGISTER
RDLSERVICE
SWAB R0 ;GET "PBA 17&16" DCWN TO BIT POSITION 0&1
RNE #177774, R0 ;BR IF PHYS ADDR BITS <17:16> LOGGED CORRECTLY
TST R5 ;PREVIOUS ERROR?

```

```

15199 056716 001402
15200 056720 005001
15201 056722 104026
15202
15203
15204 056724 005000
15205 056726 076600
15206 056730 000222
15207
15208
15209
15210
15211
15212
15213
15214
15215
15216
15217
15218 056732 012700 000760
15219 056736 000004
15220 056740 012701 064046
15221 056744 005711 177746
15222 056746 012737 000100 177746
15223 056748 012711 125252
15224 056754 012737 000001 177746
15225 056760 012737 000001 177746
15226
15227 056766 012737 057026 000114
15228 056774 012737 000340 000116
15229 057002 005000
15230 057004 076600
15231 057006 000302
15232 057010 076600
15233 057012 000306
15234 057014 076600
15235 057016 000307
15236 057020 005767 005022
15237 057024 000406
15238 057026 012700 000200 15:
15239 057032 076600
15240 057034 000322
15241 057036 022626
15242 057040 104030
15243
15244
15245
15246 057042 012700 000200 25:
15247 057046 076600 352
15248 057048 000322 125252
15249 057052 012711 000116 000114
15250 057056 012737 000116
15251 057064 005037 000116
15252 057070 005005
15253 057072 076600
15254 057074 000102

```

```

;*****
;*TEST 761 CHECK DISABLE PARITY ERROR TRAP
;*THIS TEST CHECKS THAT PARITY ERROR TRAPS TO LOCATION 114
;* ARE DISABLED WHEN BIT0 OF THE CACHE CONTROL REGISTER IS
;* SET (=1). A TRAP TO 114 SHOULD NOT OCCUR AND ERROR
;* INFORMATION SHOULD NOT BE LOGGED IN THE LOG PBA, LOG
;* CACHE DATA, OR LOG TAG DATA REGISTERS. WRONG PARITY IS
;* WRITTEN INTO A TEST LOCATION TO CAUSE THE PARITY ERROR
;* NEEDED IN THIS TEST.
;*****
45T761:
MOV #760, R0 ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #TLOC1, R1 ;GET POINTER TO TEST LOCATION
TST (R1) ;MAKE IT A HIT
MOV #WWP, @#CCR ;SET WRITE WRONG PARITY BIT
MOV #125252, (R1) ;WRITE TO TEST LOC. WITH WRONG PARITY
MOV #OPTRP, @#CCR ;DISABLE PARITY ERROR TRAPS
; AND CLEAR WWP
MOV #1S, @#114 ;SETUP PARITY ERROR VECTOR
MOV #340, @#116
CLR R0
WRWPBA ;CLEAR LOG PBA REGISTER
MED ;CLEAR LOG CACHE DATA REGISTER
WRLDATA
WRRTAG ;CLEAR LOG CACHE TAG REGISTER
TST TLOC1 ;READ TEST LOC TO FORCE PARITY ERROR
BR 2S ;BRANCH IF NO TRAP OCCURS
MOV #200, R0 ;CLEAN UP THE CACHE
; INITIALIZATION CODE
MED ;CLEAN UP STACK
CMP (SP)+, (SP)+ ;*** PARITY TRAP TO 114 OCCURRED
RNROR 30 ;WHEN IT SHOULD HAVE BEEN DISABLED
25:
MOV #200, P0 ;CLEAN UP THE CACHE
; INITIALIZATION CODE
MOV #125252, (R1) ;WRITE BAK GOOD PARITY IN TST LOC.
MOV #116, @#114 ;RESTORE ORIGINAL PARITY HANDLER & PSW
CLR @#116 ;CLEAR ERROR FLAG
CLR R5 ;READ LOG PBA REGISTER
RDLPRBA

```

```

15255 057076 010003
15256
15257 057100 001401
15258 057104 012701
15259 057104 076600
15260 057106 000106
15261 057110 010001
15262
15263 057112 001401
15264 057114 012705
15265 057116 076600
15266 057120 000107
15267 057122 010002
15268
15269 057124 001401
15270 057124 012705
15271 057130 005705
15272 057132 001401
15273 057134 104027
15274
15275
15276
15277 057136 005037 177746
15278
15279
15280
15281
15282
15283
15284
15285
15286 057142 012700 000761
15287 057142 000004
15288 057142 000004
15289 057150 012701 064046
15290 057154 005711
15291 057156 012737 000100 177746
15292 057164 012711 125252
15293 057170 042737 000100 177746
15294 057176 012737 057232 000114
15295 057204 012737 000340 000116
15296 057212 005737 064046
15297 057216 012700 000200
15298 057222 076600
15299 05724 000357
15300 05726 104037
15301 057230 000405
15302 057237 012700 000200
15303 057236 076600
15304 057240 000352
15305 057242 022739
15306 057244 000340 177744
15307
15308
15309 057252 001403
15310 057254 013700 177744

```

```

15311 057260 104032
15312
15313 057262 012737 000116 000114
15314 057270 005037 000116
15315
15316
15317
15318
15319
15320
15321
15322
15323
15324
15325
15326
15327 057274 012700 000762
15328 057300 000004
15329 057302 012737 057324 000004
15330 057310 012737 000340 000006
15331 057316 005737 160000
15332
15333
15334 057322 000461
15335 057324 022426
15336 057326 012737 061220 000004
15337 057334 076600
15338 057336 000100
15339 057340 013701 177766
15340 057344 022701 000020
15341
15342 057350 001401
15343 057352 104033
15344
15345
15346
15347 057354 022700 020200
15348 057360 001401
15349 057362 104033
15350
15351
15352
15353 057364 076600
15354 057366 000102
15355 057370 020027 160000
15356 057374 001403
15357 057376 012701 160000
15358 057402 104020
15359
15360 057404 012737 057426 000004
15361 057412 012737 000340 000006
15362 057420 005767 177741
15363
15364 057424 000420
15365 057430 012737 061220 000004
15366 057436 076600

```

15367 057440 000100 177766
15368 057440 013701 000100
15369 057446 022701 000100
15370 057452 001401
15371 057454 104024
15372
15373
15374
15375
15376
15377 057456 032700 000004
15378 057462 001001
15379 057464 104024
15380
15381
15382
15383 057466 076600
15384 057470 000104
15385 057472 170277
15386 057476 001401 000004
15387 057500 104036
15388
15389 057502
15390
15391
15392
15393
15394
15395
15396
15397
15398
15399
15400
15401
15402 057502
15403 057502 012700 000763
15404 057506 000004
15405 057510 012737 057540 000004
15406 057516 012737 000340 000006
15407 057524 005037 177746
15408 057530 012707 177746
15409 057534 104034
15410
15411 057536 000420
15412 057540 022626
15413 057542 012737 061220 000004
15414 057550 076600
15415 057552 000100
15416 057554 013701 177766
15417 057560 032701 000001
15418
15419 057564 001001
15420 057566 104035
15421
15422

RDLJAM
MOV @#CPUERR,R1
CMP #BIT5,R1 ;ODD ADDR. BUT SET 3
BEQ #5
ERROR 24 ;ODD ADDRESS BIT WAS
;NOT SET IN THE CPU
;ERROR REGISTER IN LOG
;CONTINUOUS MODE THE
;LAST ERROR SHOULD
;BE LOGGED
7\$: BIT #BIT2,R0 ;ODD ADDR. BIT SET IN
BNE #65 ;LOG JAM?
ERROR 24 ;ODD ADDRESS BIT WAS
;NOT SET IN THE LOG
;JAM REGISTER ON
;ODD ADDRESS ERROR
6\$: MED ;CHECK IF LAST INTERRUPT VECTOR
RDLFGINT ;WAS LOGGED?
CMPB R0,#4
BEQ #65
ERROR 36 ;LAST ERROR VECTOPR WS NOT LOGGED
8\$:
;*****
;TEST 764 CHECK ILLEGAL INTERNAL ADDRESS TRAP
;THIS TEST CHECKS THAT A TRAP OCCURS UPON REFERENCING AN
;ILLEGAL INTERNAL ADDRESS AND THAT "ILLEGAL INTERNAL ADDRESS"
;BIT (BIT0) OF THE CPU ERROR REGISTER AND BITS OF LOG JAM
;REGISTER GET SET. IT ALSO CHECKS IF THE INTERRUPT VECTOR
;(4) IS SAVED AS THE "LAST INTERRUPT VECTOR" IN THE LOG
;FLAG/INTERRUPT REG.
;*****
TST764: MOV #763,R0 ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #15,@#4 ;SETUP NEW HANDLER PC & PSW
MOV #340,@#6
CLR @#CCR
MOV @#CCR,PC ;ILLEGAL INTERNAL ADDRESS TRAP SHOULD OCCUR
ERROR 34 ;*** ILLEGAL INTERNAL ADDRESS
;DID NOT RESULT IN A TRAP
1\$: BR 3\$;BRANCH TO EXIT IF NO TRAP
CMP (SP)+,(SP)+ ;RESTORE STACK
MOV #BERR,@#4 ;RESTORE OLD HANDLER PC & PSW
MED
RDLJAM
MOV @#CPUERR,R1
BIT #BIT0,R1 ;DID "ILLEGAL INTERNAL ADDRESS" BIT (0)
BNE #25 ;IN CPU ERROR REGISTER GET SET?
ERROR 25 ;BRANCH IF YES
;*** ILLEGAL INTERNAL ADDRESS
;BIT DID NOT SET IN CPU ERROR REG.
;READ THE LOG JAM REG.

15423 057570 032700 000040
15424
15425 057574 001001
15426 057576 104035
15427
15428 057600
15429
15430
15431
15432
15433
15434
15435
15436
15437
15438
15439
15440
15441
15442
15443 057600
15444 057600 012700 000764
15445 057604 000004
15446
15447 057606 012737 000201 177746
15448 057614 005037 001062
15449 057620 012701 064046
15450 057624 005711
15451 057626 052717 000100 177746
15452 057634 012711 125252
15453 057640 042737 000100 177746
15454 057646 012700 100001
15455 057654 076600
15456 057654 000222
15457 057656 042737 000001 177746
15458 057664 012737 057712 000114
15459 057672 016737 004150 001062
15460 057700 012700 000200
15461 057704 076600
15462 057706 000352
15463 057710 104031
15464
15465
15466
15467
15468
15469 057712 012700 000200 PTRP1: MOV #200,R0
15470 057716 076600
15471 057720 000352
15472 057724 012737
15473 057730 000001 177746
15474 057736 005037 000116 000114
15475 057742 022626
15476 057744 005737 001062
15477
15478 057750 001401

2\$: BIT #BIT5,R0 ;DID "ILLEGAL INTERNAL ADDRESS" BIT (5)
BNE #3\$;IN LOG JAM REG. GET SET
ERROR 35 ;BRANCH IF YES
;*** ILLEGAL INTERNAL ADDRESS BIT
;DID NOT SET IN LOG JAM REG.
3\$:
;*****
;TEST 765 CHECK LOG SERVICE & MEMERR LOGS LD-HI BYTE & TAG IN CACHE ABORT MODE
;TEST CHECKS THAT LOG BYTE PARITY AND "TAG PARITY"
;BITS CAN SET IN "LOG SERVICE" REGISTERS. IT IS ALSO
;CHECKED THAT THE PROPER TAG AND DATA BITS GET STORED
;IN THE "LOG CACHE DATA", "LOG CACHE TAG/CPU" AND THE
;"MEMORY ADDRESS REGISTER" WHEN A PARITY ERROR IS
;FORCED.
;IT IS CHECKED IF THE INSTRUCTION WAS ABORTED AND THE
;LOG FLAG/INTERRUPT REGISTER LOGGED THE LAST INTERRUPT
;VECTOR.
;*****
TST765: MOV #764,R0 ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #DPTRP+PABORT,@#CCR ;DISABLE PARITY TRAPS (CACHE)
CLR @#SREG0
MOV #L0C1,R1 ;GET POINTER TO TEST LOC.
TST (R1) ;WAS IT A HIT
BTS #WPP,@#CCR ;WRITE WRONG PARITY SET
MOV #125252,(R1) ;WRITE TEST LOCATION WITH WRONG PARITY
BIC #WPP,@#CCR ;CLEAR WPP
MOV #BIT15+BIT0,R0
MED
WRHAM1 ;ENABLE "LOG FIRST" MODE, AND
;ERROR LOGGING
BIC #DPTRP,@#CCR ;ENABLE CACHE PARITY TRAPS
MOV #PTRP1,@#114 ;NEW PARITY TRAP SERVICE
CLR #L0C1,@#SREG0 ;READ TEST LOC, FORCE PARITY ERROR
MOV #200,R0
MED
MED ;CLEAN UP THE CACHE
352 ;INITIALIZATION CODE
ERROR 31 ;*** CACHE PARITY ERROR TRAP
;DID NOT OCCUR WHEN
;TEST LOC WITH BAD PARITY
;WAS READ
;ENTER HERE IF PARITY TRAP OCCURRED
;CLEAN UP THE CACHE
;INITIALIZATION CODE
PTRP1: MOV #DPTRP,@#CCR ;DISABLE CACHE PARITY ERROR TRAPS
CLR @#116,@#114 ;REESTABLISH OLD SERVICE VECTORS
MOV (SP)+,(SP)+
CMP @#SREG0
TST
BEQ #1\$;WAS THE INSTRUCTION ABORTED ON
;CACHE PARITY ERROR (ABORT MODE)?
;YES

```

15479 057752 104041          ERROR 41          ;INSTRUCTION HAVING CACHE PARITY
15480                                ;ERROR WAS NOT ABORTED, IN THE
15481                                ;CACHE ABORT MODE
15482                                ;READ THE "LOG SERVICE" REGISTER
15483 057754 076600          1$: MED
15484 057756 000101          RDLSERVICE
15485 057760 010004          MOV R0,R4          ;COPY
15486 057762 027704 177435  BIC #CLO+HI+TAG+BIT15,R4 ;MASK ALL BUT LO HI TAG BITS
15487 057766 027704 000342  CMP #342,R4        ;LO HI TAG, CACHE PARITY BITS SET? IN "SERVICE"
15488 057774 104042          BRD
15489                                ;*** "LO BYTE" PARITY ERROR
15490                                ;AND "TAG" PARITY ERROR BITS
15491                                ;WERE NOT LOGGED CORRECTLY IN "LOG
15492                                ;SERVICE" REGISTER WHEN PARITY
15493                                ;ERROR TRAP WAS FORCED.
15494                                ;CLEAR BITS ARE ACTIVE.
15495 057776 013700 177744  2$: MOV @MEMERR,R0    ;GET MEM ERR REG
15496 060002 022700 100340  CMP #HI+LO+TAG+BIT15,R0 ;DID "LO BYTE" "HI BYTE" AND "TAG"
15497                                ;PARITY ERROR BITS SET IN
15498                                ;THE MEMORY ERROR REGISTER?
15499                                ;YES
15500                                ;*** "LO BYTE" "HI BYTE" AND "TAG" PARITY
15501                                ;ERROR BITS DID NOT SET
15502                                ;CORRECTLY IN THE MEMORY
15503                                ;ERROR REGISTER
15504 060006 001401          3$: MED
15505 060010 104043          RDLPBA
15506                                ;READ "LOG PBA" REGISTER
15507                                ;DID "LOG PBA" CONTAIN CORRECT
15508                                ;PHYSICAL BUS ADDRESS-WHERE
15509                                ;THE PARITY ERROR OCCURRED?
15510                                ;YES
15511                                ;EXPECTED PBA
15512                                ;** PHYSICAL BUS ADDRESS
15513                                ;(WHERE PARITY ERROR OCCURRED)
15514                                ;WAS NOT LOGGED CORRECTLY
15515                                ;WHEN CACHE PARITY ERROR WAS FORCED
15516 060012 076600          4$: MED
15517 060014 000107          RDLTAG
15518 060036 000300          SWAB R0
15519 060040 012701 064046  MOV #TLOC1,R1
15520 060044 000301          SWAB R1
15521 060046 106201          ASRB R1
15522 060050 106201          ASRB R1
15523 060054 067701 000200  BIS #BIT7,R1
15524 060060 120100          CMPB #1,R0
15525 060062 001401          BEQ 5$
15526 060064 104017          ERROR 17
15527                                ;TAG BITS WERE NOT LOGGED
15528                                ;CORRECTLY WHEN CACHE
15529                                ;PARITY ERROR WAS FORCED
15530                                ;READ CACHE DATA
15531 060066 076600          5$: MED
15532 060070 000106          RDLDATA
15533 060072 020027 125252  CMP R0,#125252    ;CACHE DATA LOGGED CORRECTLY?
15534 060076 001403          BEQ 6$
15535 060100 012701 125252  MOV #125252,R1    ;EXPECTED DATA
15536 060104 104016          ERROR 16

```

```

15535 060106 012700 000001  6$: MOV #BIT0,R0      ;SET UP LOG CONTINUOUS
15536 060112 076600          MED
15537 060114 000222          WRWHAMI
15538 060116 012737 060130 000004  MOV #76, #4
15539 060144 057737 160000  TST #160000
15540 060130 012737 061220 000004  7$: CDB #SP+,(SP)+
15541 060132 012737 061220 000004  MOV #BERR,#4
15542 060140 076600          MED
15543 060142 000104          RDLFCINT
15544 060144 120027 000114  CMPB R0,#114
15545 060150 014903 001062  BEQ 8$
15546 060156 104036          MOV R0,@#SREGO
15547                                ;LAST INTERRUPT VECTOR WAS NOT
15548                                ;LOGGED CORRECTLY IN FLAG REGISTER
15549                                ;WHEN A CACHE PARITY ERROR WAS
15550                                ;FORCED.
15551 060160          8$:
15552                                ;*****
15553                                ;*TEST 766 CHECK "LOG FIRST" MODE OF ERROR LOGGING
15554                                ;*THIS TEST CHECKS THE "LOG FIRST" MODE OF ERROR LOGGING.
15555                                ;*THE "LOG FIRST" MODE IS ENABLED. THEN A TIME-OUT TRAP
15556                                ;*IS FORCED, BIT 4 OF CPU ERROR REGISTER SHOULD BE SET.
15557                                ;*THEN AN ODD ADDRESS TRAP IS FORCED HOWEVER THIS
15558                                ;*TIME THE ERROR SHOULD NOT BE LOGGED, BIT 6 (ODD
15559                                ;*ADDRESS) SHOULD NOT BE SET BECAUSE THE ERROR LOG
15560                                ;*IS LOCKED UP AFTER THE FIRST ERROR.
15561                                ;*THEN THE ERROR LOG IS ENABLED (BY SETTING BIT 0 OF
15562                                ;*WHAM) AN ODD ADDRESS ERROR IS FORCED AGAIN AND IT IS
15563                                ;*CHECKED THAT THIS TIME THE ERROR IS LOGGED, (BIT 6-ODD
15564                                ;*ADDRESS SHOULD BE SET IN CPU ERROR REGISTER).
15565                                ;*****
15566                                ;*ST766:
15567 060160 012700 000765  MOV #765,R0
15568 060164 000004          SCOPE
15569                                ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
15570                                ;CALL THE SCOPE LOOP UTILITY
15571 060166 012700 100001  MOV #BIT15+BIT0,R0 ;SET UP "LOG FIRST MODE"
15572 060172 076600          MED
15573 060174 000234          WRWHAMI
15574 060176 012737 060220 000004  MOV #15, #4
15575 060204 012737 000340 000006  MOV #340, #6
15576 060212 057737 160000  TST #160000
15577 060216 000462          BR 5$
15578                                ;SETUP NEW PC & PSW FOR
15579                                ;TIMEOUT
15580                                ;FORCE A TIMEOUT
15581                                ;SKIP TEST IF NO TIMEOUT
15582 060220 022626          1$: CMP (SP)+,(SP)+
15583                                ;RESTORE STACK
15584                                ;BIT 4 OF CPU ERROR REGISTER
15585                                ;SHOULD HAVE SET
15586 060222 012737 060236 000004  MOV #25, #4
15587 060230 057767 177765  TST 15+1
15588 060234 000453          BR 5$
15589                                ;SETUP NEW PC FOR ODD ADDRESS
15590                                ;FORCE ODD ADDRESS TRAP
15591                                ;SKIP TEST IF NO ODD ADDRESS TRAP
15592 060236 022626          2$: CMP (SP)+,(SP)+
15593 060240 012737 061220 000004  MOV #BERR,#4
15594 060246 076600          MED
15595 060250 000100          RDLJAM

```

```
15591 060252 013701 177766 MOV @#CPUERR,R1 ;"TIMEOUT" BIT SHOULD BE STILL
15592 060256 022701 000020 CMP #BIT4,R1 ;SET, CHECK?
15593
15594 060262 001402 BEQ 3$ ;*** SECOND ERROR (ODD ADDRESS)
15595 060264 104033 ERROR 3$ ;UPDATED THE ERROR LOG IN
;THE LOG FIRST MODE. BIT 4
;(UNIBUS TIMEOUT) SHOULD BE
;STILL SET FROM THE FIRST
;ERROR
15600 060266 000436 BR 5$ ;SKIP THE REST
15601 060270 032700 100004 3$: BIT #BIT2+BIT15,R0 ;CHECK THAT ODD ADRES ERROR BITS NOT
15602 060274 001401 BEQ 6$ ;SET IN LOG JAM. NOTE LOG FIRST
;MODE SHOULD INHIBIT FURTHER
;ERROR LOGGING
15603 060276 104037 ERROR 37 ;ODD ADDRESS ERROR BITS GOT SET IN LOG JAM
;THEY SHOULD NOT BE SINCE LOG FIRST MODE
;INHIBITS ERROR LOGGING AFTER THE FIRST ERROR
;ENABLE ERROR LOG AGAIN IN
;LOG FIRST MODE
15604
15605 060300 012700 100001 6$: MOV #BIT15+BIT0,R0
15606
15607 060304 076600 MED WRNHAMI
15608 060306 000272 MOV #4$,@#4 ;SET UP NEW PC & PSW FOR
15609 060310 012737 060332 000004 MOV #340,@#6 ;ODD ADDRESS ERROR
15610 060316 012737 000340 000006 TST #5+1 ;FORCE ODD ADDRESS TRAP
15611 060324 005767 177741 RR 5$ ;SKIP IF NO TRAP
15612 060330 000415 4$: CMP (SP)+,(SP)+ ;RESTORE STACK
15613 060332 022626 MOV #BERR,@#4 ;RESTORE OLD PC(4), PSW(6)
15614
15615 060334 012737 061220 000004 MOV #BERR,@#4
15616 060342 022737 000100 177766 CMP #BIT6,@#CPUERR ;THE ERROR LOG FROM PREVIOUS
;ERROR SHOULD BE OVER WRITTEN.
;ODD ADDRESS BIT SHOULD
;BE SET BECAUSE THE ERROR
;LOG WAS ENABLED.
;OK, IF YES
15617
15618 060350 001405 BEQ 5$
15619
15620 060352 076600 MED RDLJAM
15621 060354 000100 MOV #CPUERR,R1
15622 060356 013701 177765 ERROR 40 ;THE ERROR LOG WAS NOT UPDATED
;UPON AN ODD ADDRESS ERROR)
;AFTER THE LOG WAS ENABLED.
;AT THIS FORMAT BIT 6 OF
;CPU ERROR REGISTER SHOULD
;BE SET. IT WAS NOT.
;RESTORE OLD PC(4), PSW(6)
15623
15624 060364 012737 061220 000004 5$: MOV #BERR,@#4
15625 060372 012700 000001 MED WRNHAMI ;PUT THE LOGGING BACK INTO
;"CONTINUOUS" MODE
15626 060376 076600
15627 060384 000222
15628
15629 *****
15630 ;*TEST 76: CHECK LAST INTRRUPT VECTOR IS LOGGED IN FLAG REG.
15631 ;*****
15632
15633 060402 012700 000766 T767: MOV #766,R0 ;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
15634 060406 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
15635
15640
15641
15642
15643
15644
15645
15646
```

```
15647 060410 012737 060420 000030 MOV #1$,@#30 ;LOAD EMT VECTOR WITH 1$
15648 060416 104000 EMT ;FIRST INTERRUPT -- EMT
15649 060420 022626 CMP (SP)+,(SP)+ ;CLEAN UP STACK
15650 060422 012737 061620 000030 MOV #ERROR,@#30 ;RESTORE VECTOR
15651 060430 012737 060442 000004 MOV #2$,@#4 ;SET UP CPU VECTOR
15652 060436 005737 160000 TST #160000 ;FORCE TIMEOUT
15653 060440 012737 2$: CMP (SP)+,(SP)+ ;CLEAN UP STACK
15654 060444 012737 MOV #BERR,@#4 ;RESTORE BUS ERROR VECTOR
15655 060452 076600 MED ;CHECK FLAG
15656 060454 000104 RDLFGINT
15657 060456 120027 000030 CMPB R0,#30 ;EMT VECTOR LAST LOGGED?
15658 060462 001401 BEQ 3$ ;BR IF YES
15659 060464 104036 ERROR 3$ ;LOG FLAG/INT REG DID NOT LOG VECTOR
;LOW BYTE OF LOG FLAG/INT REG S/B=30
15660
15661
15662 060466 012737 060476 000020 3$: MOV #4$,@#20 ;LOAD IOT VECTOR WITH 4$
15663 060474 000004 IOT ;SECOND INTERRUPT-SHOULD LOAD LOG FLAG REG
15664 060476 022626 CMP (SP)+,(SP)+ ;CLEANUP STACK
15665 060500 012737 061260 000020 4$: MOV #SCOPE,@#20 ;RESTORE IOT VECTOR
15666 060506 012737 060520 000004 MOV #5$,@#4 ;SET UP CPU VECTOR
15667 060514 005737 160000 TST #160000 ;FORCE TIMEOUT
15668 060520 022626 CMP (SP)+,(SP)+ ;CLEAN UP STACK
15669 060522 012737 061220 000004 5$: MOV #BERR,@#4 ;RESTORE BUS ERROR VECTOR
15670 060530 076600 MED ;CHECK FLAG
15671 060532 000104 RDLFGINT
15672 060534 120027 000020 CMPB R0,#20 ;IOT VECTOR LAST LOGGED?
15673 060540 001401 BEQ 6$ ;BR IF YES
15674 060542 104036 ERROR 3$ ;LOG FLAG/INT REG DID NOT LOG VECTOR
;LOW BYTE S/B = 20
15675
15676 060544 012700 000767 6$: MOV #STN-1,R0 ;SET UP FOR MISSED TEST CHECK AND
;FULL WORD TEST NUMBER FOR APT
15677
15678
15679
15680
15681
15682
15683
15684
15685
15686
15687
15688
15689
15690
15691 060550 000004 SEOP: ;*****
15692 060552 005037 001110 CLR SPASS ;INCREMENT THE PASS NUMBER ($PASS)
15693 060554 005237 001126 INC SPASS ;IF THERE'S A MONITOR GO TO IT
15694 060556 042737 100000 001126 BIT #100000,$PASS ;IF THERE ISN'T JUMP TO INIT
15695 060558 005377 DEC (PC)+ ;LOOP?
15696 060572 000001 SEOPCT: .WORD 1
15697 060574 003027 BGT $DOAGN ;YES
15698 060576 012737 MOV (PC)+,@(PC)+ ;RESTORE COUNTER
15699 060600 000071 SENDCT: .WORD 1
15700 060602 060577 TYPE ;TYPE "END PASS #"
15701 060604 104401 EOP1 ;SAVE $PASS FOR TYP0UT
15702 060610 013746 MOV $PASS,-(SP)
```



```

15776
15777
15778
15779
15780
15781 061070 062716 000002 000002
15782 061074 047766 000020 000002
15783 061102 000006
15784
15785
15786
15787
15788
15789
15790 061104 005237 061112
15791
15792 061110 000002
15793 061112 000000
15794
15795
15796
15797
15798
15799
15800
15801
15802
15803
15804
15805
15806
15807
15808
15809
15810
15811
15812
15813
15814
15815
15816
15817
15818
15819
15820
15821
15822
15823
15824
15825
15826
15827
15828
15829
15830
15831

```

```

; *****
; .SBTTL "T" BIT SERVICE ROUTINE
; *****
TBSEB: ADD #2(SP) ;MOVE RETURN PC AROUND ERROR CALL
        BIC #20,2(SP) ;TURN OFF THE "T" BIT
        RTI ;RETURN TO THE CALLING TEST

.SBTTL MICROBREAK TRAP SERVICE ROUTINE
; *****
; THIS ROUTINE MERELY SETS A FLAG
; WHEN THE ROUTINE HAS BEEN ENTERED
; *
BKROUT: INC BKFLAG ;SET MICROBREAK FLAG TO
        ;INDICATE TRAP TO 4 OCCURRED
        RTI ;RETURN FROM TRAP
        BKFLAG: .WORD 0 ;MICROBREAK TRAP FLAG
; *****
; .SBTTL RSVD INSTRUCTION TRAP SERVICE ROUTINE
; *****
; THIS ROUTINE SERVICES UNEXPECTED RESERVED INSTRUCTION TRAP ERRORS
; IT RESULTS IN PRINTING THE ERROR MESSAGE: "TRAPPED TO 10 PC=XXXXXX"
; WHERE XXXXXX IS THE ADDRESS CONTAINING THE INSTRUCTION WORD THAT
; SPRUNG THE TRAP. AFTER PRINTING THE ERROR MESSAGE AN ATTEMPT IS
; MADE TO RESTART THE PROGRAM AT THE BEGINNING.
; IF THE TRAP IS SPRUNG WHILE IN THE PROCESS OF TRYING TO SERVICE A
; PREVIOUS RSVD INSTRUCTION TRAP OR AN UNEXPECTED BUS ERROR THE PROGRAM
; WILL HALT. AFTER THE HALT THE STACK WILL CONTAIN INFORMATION RELATIVE
; TO THE TWO SUCCESSIVE TRAPS AS SHOWN BELOW:
;
;[SP] PC+2 OF 2ND TRAP
;[SP]+2 PSW "
;[SP]+4 PC+2 OF 1ST TRAP
;[SP]+6 PSW "
;
;LOCATION "CATERR" CAN BE EXAMINED TO OBTAIN THE FOLLOWING
;INFORMATION:
;
;[CATERR]=401 RSVD INSTR TRAP COMBINED WITH A BUS ERROR
;TRAP (PC AT TIME OF ERROR HALT INDICATES
;WHICH OCCURRED FIRST)
;[CATERR]=2 TWO SUCCESSIVE BUS ERROR TRAPS
;[CATERR]=1000 TWO SUCCESSIVE RSVD INSTR TRAPS
;
;THE CONTENTS OF RO AT THE TIME OF THE
;HALT PROVIDES FURTHER INFORMATION AS TO THE LAST TEST BEING EXECUTED
;WHEN THE TRAPS OCCURRED.
;
;THESE TWO INSTRUCTIONS ARE USED BY THE BASIC INSTRUCTION
;TESTS TO VERIFY THE RSVD INSTR TRAP MECHANISM PRIOR TO ACTIVATING THE SERVICE
;ROUTINE

```

```

15832 061114 005137 063246
15833 061120 000002
15834
15835 061122 005737 063252
15836 061126 001025
15837 061130 005237 063253 117676
15838 061134 032777 010000
15839 061142 001015
15840 061144 104401
15841 061146 065232
15842 061150 011646
15843 061152 104402
15844 061154 104401
15845 061156 001115
15846 061160 005237 001012
15847 061164 032777 100000 117646
15848 061172 001401
15849 061174 000000
15850 061176 000137 003262
15851 061202 105237 063253
15852 061206 000000
15853 061210 000772
15854
15855
15856
15857
15858
15859
15860
15861
15862
15863
15864
15865
15866
15867
15868
15869
15870
15871
15872
15873
15874
15875
15876
15877
15878
15879
15880
15881
15882
15883
15884
15885
15886
15887

```

```

RSVST: COM RSVFLG ;SET RSVD INSTR TRAP TEST FLAG
        RTI ;RETURN TO BASIC TEST

RSERR: TST #CATERR ;ANY PENDING CATASTROPHIC ERRORS
        BNE INCRSV ;BE IF YES
        INCB #1,CATERR ;SET RSVD INSTR FLAG
        BIT #SW12,@SWR ;INHIBIT ERROR PRINT ?
        BNE RESTAR ;BR IF YES
        TYPE ;GO TYPE "TRAPPED TO 10 PC="
        RSMMSG

RSBERT: MOV (SP),-(SP) ;GET ERROR PC ON STACK FOR PRINTING
        TYP0C ;TYPE THE ERROR PC
        TYPE ;OUTPUT CR / LF
        INCB #1,CATERR ;COUNT THE ERROR
        INCB #1,CATERR ;HALT ON ERROR?
        BEQ RESTAR ;BR IF NOT
        HALT ;HALT ON ERROR--PRESS CONTINUE TO RESTART
        JMP #INIT ;GO ATTEMPT RESTART
        INCRSV: INCB #1,CATERR ;INCREMENT RSVD INSTR FLAG
        HALT ;CATASTROPHIC ERROR HALT
        BR RESTAR ;DEPRESSING CONTINUE WILL CAUSE
        ;ATTEMPT TO RESTART.
; *****
; .SBTTL BUS ERROR TRAP SERVICE ROUTINE
; *****
; THIS ROUTINE SERVICES UNEXPECTED BUS ERROR TRAPS (BUS TIMEOUT, ODD ADDRESS
; ERRORS, STACK OVERFLOW, AND ILLEGAL INSTRUCTIONS). IT RESULTS IN PRINTING THE
; ERROR MESSAGE: "TRAPPED TO 4 PC =XXXXXX" WHERE XXXXXX IS THE
; CONTENTS OF THE PC WHEN THE TRAP WAS SPRUNG. AFTER PRINTING THE
; ERROR MESSAGE AN ATTEMPT IS MADE TO RESTART THE PROGRAM AT
; THE BEGINNING.
; IF THE TRAP IS SPRUNG WHILE IN THE PROCESS OF TRYING TO SERVICE A PREVIOUS
; RSVD INSTR TRAP OR A PREVIOUS BUS ERROR, THE PROGRAM WILL HALT.
; AFTER THE HALT THE STACK WILL CONTAIN INFORMATION RELATIVE TO THE
; TWO SUCCESSIVE TRAPS AS SHOWN BELOW:
;
;[SP] PC+2 OF 2ND TRAP
;[SP]+2 PSW "
;[SP]+4 PC+2 OF 1ST TRAP
;[SP]+6 PSW "
;
;LOCATION "CATERR" CAN BE EXAMINED TO OBTAIN THE FOLLOWING
;INFORMATION:
;
;[CATERR]=401 RSVD INSTR TRAP COMBINED WITH A BUS ERROR
;TRAP (PC AT TIME OF ERROR HALT
;INDICATES WHICH OCCURRED FIRST)
;[CATERR]=2 TWO SUCCESSIVE BUS ERRORS
;[CATERR]=1000 TWO SUCCESSIVE RSVD INSTR TRAPS
;
;THE CONTENTS OF RO AT THE TIME OF
;THE HALT PROVIDED FURTHER INFORMATION AS TO THE TEST IN PROGRESS

```

```

15888
15889
15890
15891
15892
15893
15894
15895
15896
15897
15898
15899
15900 061212 005137 063250
15901 061216 000002
15902
15903 061220 005737 063252
15904 061224 001011
15905 061226 105237 063252 117600
15906 061230 032777 010000
15907 061240 001356
15908 061242 104401
15909 061244 065205
15910 061246 000740
15911
15912 061250 105237 063252
15913 061254 000000
15914 061256 000747
15915
15916
15917
15918
15919
15920
15921
15922
15923
15924
15925
15926
15927
15928
15929
15930
15931 061260
15932 061260 020037 001124
15933 061264 001406
15934 061266 012737 061276 001112
15935 061274 104011
15936 061276 005737 001112
15937 061302 110037 001002
15938 061306 032777 002000 117524
15939 061314 001411
15940 061316 017337 117516 063242
15941 061324 042737 177000 063242
15942 061332 020037 063242
15943 061336 001510

```

```

;WHEN THE TRAPS OCCURRED.
;THE CONTENTS OF THE SP CAN BE USED TO INDICATE IF STACK OVERFLOW CAUSED
;THE BUSS ERROR TRAP(S) AS SHOWN BELOW:
;400[SP]>336 YELLOW ZONE
;[SP]=0 RED ZONE
;THESE TWO INSTRUCTIONS ARE USED BY THE BASIC INSTRUCTION TESTS TO
;VERIFY THAT THE BUS ERROR TRAP MECHANISM WORKS PRIOR TO ACTIVATING
;THE SERVICE ROUTINE
BETST: COM BERFLG ;SET BUS ERROR TRAP TEST FLAG
RTI ;RETURN TO BASIC TEST
BERR: TST @CATERR ;ANY CATASTROPHIC ERRORS PENDING?
BNE ZS ;BR IF YES
INCB @CATERR ;SET CATASTROPHIC ERROR FLAG
BIT #SW1,@SWR ;INHIBIT ERROR PRINT
BNE RESTAR ;BR IF YES
TYPE ;PRINT "TRAP TO 4" MESSAGE
BEMSG ;TYPE REST OF BUS ERROR MESSAGE
BR RSBERT
2$: INCB @CATERR ;SET CATASTROPHIC ERROR FLAG
HALT ;CATASTROPHIC ERROR HALT-SCHOOLS OUT
BR RESTAR ;DEPRESS CONTINUE TO ATTEMPT RESTART

```

.SBTTL SCOPE HANDLER ROUTINE

```

;*****
;THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
;AND LOAD THE TEST NUMBER(STSTNM) INTO THE DISPLAY REG.(DISPLAY<7>:0)
;AND LOAD THE ERROR FLAG (SERFLG) INTO DISPLAY<15:08>
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;SW4=1 LOOP ON TEST
;SW11=1 INHIBIT ITERATIONS
;SW09=1 LOOP ON ERROR
;CALL SCOPE ;;SCOPE=IOT
SCOPE:
CMP R0,@#STESTN ;ANY MISSED TESTS ?
BREQ 10$ ;BR IF NOT
MOV #125,@#$ESCAPE ;NO ERROR LOOPING
ERROR 11 ;MISSED TESTS ERROR CALL
12$: @#$ESCAPE ;NORMAL ERROR LOOPING
10$: MOV R0,@#STSTNM ;INSURE TSTNUM IS CORRECT
BIT #SW10,@SWR ;LOOP ON SELECTED TEST?
BREQ 11$ ;BR IF NO
MOV @SWR,@#SELTST ;GET CONTENTS OF SWITCHES
BIT #1700,@#SELTST ;MASK OUT SW<15:9>
CMP R0,@#SELTST ;IS THIS THE SELECTED TEST?
BREQ $OVER ;BR IF YES

```

```

15944 061340 032777 040000 117472 11$: BIT #BIT14,@SWR ;LOOP ON PRESENT TEST?
15945 061340 012737 061376 ;YES IF SW14=1
15946 061346 001104 ;SERFLG###
15947
15948 061350 000416 ;###START OF CODE FOR THE XOR TESTER###
15949
15950 061352 013746 000004 MOV @ERRVEC,-(SP) ;SAVE THE CONTENTS OF THE ERROR VECTOR
15951 061356 012737 061376 ;#5,@ERRVEC ;SET FOR TIMEOUT
15952 061364 005737 177060 ;#17000 ;TIME OUT ON XOR?
15953 061366 000004 TST @ERRVEC ;GO TO THE ERROR VECTOR
15954 061374 000453 ;SVLAD,@ERRVEC ;GO TO THE NEXT TEST
15955 061376 022626 5$: CMP (SP)+,(SP)+ ;CLEAR THE STACK AFTER A TIME OUT
15956 061400 012637 000004 MOV @ERRVEC ;RESTORE THE ERROR VECTOR
15957 061404 000413 BR 7$ ;LOOP ON THE PRESENT TEST
15958
15959 061406 6$:###END OF CODE FOR THE XOR TESTER###
15960 061412 001421 2$: BREQ 3$ ;HAS AN ERROR OCCURRED?
15961 061414 123737 001015 001003 CMPB $ERMAX,SERFLG ;MAX. ERRORS FOR THIS TEST OCCURRED?
15962 061422 101015 BHI 3$ ;BR IF NO
15963 061424 032777 001000 117406 BIT #BIT09,@SWR ;LOOP ON ERROR?
15964 061432 001404 BREQ 4$ ;BR IF NO
15965 061434 013737 001010 001006 7$: MOV $PERR,$LPADR ;SET LOOP ADDRESS TO LAST SCOPE
15966 061442 000446 BR $OVER
15967 061444 105037 001003 4$: CLRB SERFLG ;ZERO THE ERROR FLAG
15968 061450 005037 001110 CLR $TIMES ;CLEAR THE NUMBER OF ITERATIONS TO MAKE
15969 061454 000415 BR 1$ ;ESCAPE TO THE NEXT TEST
15970 061456 032777 004000 117354 3$: BIT #BIT11,@SWR ;INHIBIT ITERATIONS?
15971 061464 001011 BNE 1$ ;BR IF YES
15972 061466 005737 001126 TST $PASS ;IF FIRST PASS OF PROGRAM
15973 061472 001406 BREQ 1$ ;INHIBIT ITERATIONS
15974 061474 005237 001004 INC $ICNT ;INCREMENT ITERATION COUNT
15975 061500 023737 001110 001004 CMP $TIMES,$ICNT ;CHECK THE NUMBER OF ITERATIONS MADE
15976 061506 002024 BGE $OVER ;BR IF MORE ITERATION REQUIRED
15977 061510 012737 000001 001004 1$: MOV #1,$ICNT ;REINITIALIZE THE ITERATION COUNTER
15978 061516 013737 061610 001110 MOV $MXCNT,$TIMES ;SET NUMBER OF ITERATIONS TO DO
15979 061524 105237 001002 ;COUNT TEST NUMBERS
15980 061530 113737 001002 001124 $SVLAD: INCB $STSTNM ;SET TEST NUMBER IN APT MAILBOX
15981 061536 011637 001006 MOV (SP),$LPADR ;SAVE SCOPE LOOP ADDRESS
15982 061542 011637 001010 MOV (SP),$PERR ;SAVE ERROR LOOP ADDRESS
15983 061546 005037 001112 CLRB $ESCAPE ;CLEAR THE ESCAPE FROM ERROR ADDRESS
15984 061552 112737 000001 001015 MOVB #1,$ERMAX ;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
15985 061560 013777 001002 117254 $OVER: MOV $STSTNM,@DISPLAY ;DISPLAY TEST NUMBER
15986 061566 013716 001006 MOV $LPADR,(SP) ;FUDGE RETURN ADDRESS
15987 061572 120037 001002 CMPB R0,@#STSTNM ;WAS $STSTNM INCREMENTED?
15988 061576 001401 BREQ 10$ ;BR IF NOT
15989 061600 005200 INC R0 ;INCREMENT TEST NUMBER
15990 061602 010037 001124 10$: MOV R0,@#STESTN ;FIX $STESTN TO BE WORD COUNT, NOT BYTE
15991 061606 000002 RTI
15992 061610 000200 $MXCNT: 200 ;MAX. NUMBER OF ITERATIONS
15993
15994 061612 005137 063244 SCOPEA: COM @#SCOFLG ;THESE TWO INSTRUCTIONS ARE
15995 061616 000002 RTI ;USED IN THE BASIC TESTS TO
;VERIFY THE IOT LINKAGE

```

```

.SBTTL ERROR HANDLER ROUTINE
;*****
;THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
;SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
;AND GO TO SERRTB ON ERROR
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;*SW15=1 HALT ON ERROR
;*SW13=1 INHIBIT ERROR TYPEOUTS
;*SW09=1 LOOP ON ERROR
;*CALL ERROR N ;;ERROR=EMT AND N=ERROR ITEM NUMBER

ERROR:
MOV R5,-(SP) ;SAVE R5 ON STACK
MOV #SREGAD,R5 ;GET POINTER
MOV 4(SP),(R5)+ ;SAVE ERROR PSM IN $REGAD FOR TYP0UT
MOV R0,(R5)+ ;SAVE R0 FOR TYPEOUTS
MOV R1,(R5)+ ;SAVE R1 IN SREG1
MOV R2,(R5)+ ;SAVE R2 IN $REG2, ETC.
MOV R3,(R5)+
MOV R4,(R5)+
CMP #1,(R5) ;IS SP ALREADY STORED IN $REG5?
BNE 10$ ;BR IF YES
;PUT SP IN $REG5 FOR TYP0UT
MOV SP,(R5)
RESTORE R5
10$: INCB SERRFLC ;SET THE ERROR FLAG
;DON'T LET THE FLAG GO TO ZERO
BEQ 75$ ;SKIP IF ZERO
MOV STSNM,@DISPLAY ;DISPLAY TEST NUMBER AND ERROR FLAG
INC 10$ ;INC THE ERROR COUNT
MOV (SP),SERRPC ;GET ADDRESS OF ERROR INSTRUCTION
SUB #2,SERRPC
MOV SERRPC,ITEMB ;STRIP AND SAVE THE ERROR ITEM CODE
BIT #BIT13,@SWR ;SKIP TYPEOUT IF SET
BNE 20$ ;SKIP TYPEOUTS
JCS SERRTYP ;GO TO USER ERROR ROUTINE
TYPE ;SERRFLC

20$: CMPE #APTENV,SENV ;RUNNING IN APT MODE
BNE 25$ ;NO SKIP APT ERROR REPORT
MOV SITEMB,21$ ;NO SKIP ITEM NUMBER AS ERROR NUMBER
JCS $,SATY4 ;REPORT FATAL ERROR TO APT

21$: .BYTE 0
.BYTE 0
22$: BR ;APT ERROR LOOP
25$: TST @SWR ;HALT ON ERROR
;SKIP IF CONTINUE
BAL ;HALT ON ERROR
3$: BIT #BIT09,@SWR ;LOOP ON ERROR SWITCH SET?
BEQ 4$ ;BR IF NO
MOV SLERR,(SP) ;FUJDE RETURN FOR LOOPING
TST ;CHECK FOR AN ESCAPE ADDRESS
BEQ 5$ ;BR IF NONE
MOV $ESCAPE,(SP) ;FUJDE RETURN ADDRESS FOR ESCAPE

5$: MOV #-1,@SREG5 ;FLAG CURRENT STACK POINTER TO BE TYPED

```

```

ERRA: COM @ERRFLG ;THESE TWO INSTRUCTIONS ARE USED
RTI ;IN THE BASIC TESTS TO VERIFY THE EMT

.SBTTL ERROR MESSAGE TYPEOUT ROUTINE
;*****
;THIS ROUTINE USES THE "ITEM CONTROL BYTE" (SITEMB) TO DETERMINE WHICH
;ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" (SERRTB),
;AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

SERRTYP:
TYPE ;SERRFLC ;"CARRIAGE RETURN" & "LINE FEED"
MOV R0,-(SP) ;SAVE R0
CLR R0 ;PICKUP THE ITEM INDEX
RPSB @SITEMB,R0 ;IF ITEM NUMBER IS ZERO, JUST
1$: BNE 1$ ;TYPE THE PC OF THE ERROR
MOV SERRPC,-(SP) ;SAVE SERRPC FOR TYPEOUT
;ERROR ADDRESS
;GO TYPE--OCTAL ASCII(ALL DIGITS)
BR 6$ ;GET OUT
DEC R0 ;ADJUST THE INDEX SO THAT IT WILL
ASL R0 ;WORK FOR THE ERROR TABLE
ASL R0
ADD #SERRTB,R0 ;FORM TABLE POINTER
MOV (R0)+,25$ ;PICKUP "ERROR MESSAGE" POINTER
BEQ 3$ ;SKIP TYPEOUT IF NO POINTER
TYPE ;TYPE THE "ERROR MESSAGE"
; "CARRIAGE RETURN" & "LINE FEED"
2$: WORD 0 ;"ERROR MESSAGE" POINTER GOES HERE
TYPE ;"CARRIAGE RETURN" & "LINE FEED"
3$: MOV (R0)+,45$ ;PICKUP "DATA HEADER" POINTER
BEQ 5$ ;SKIP TYPEOUT IF 0
TYPE ;TYPE THE "DATA HEADER"
4$: WORD 0 ;"CARRIAGE RETURN" & "LINE FEED"
TYPE ;"CARRIAGE RETURN" & "LINE FEED"
5$: MOV (R0),R0 ;PICKUP "DATA TABLE" POINTER
BEQ 6$ ;GO TYPE THE DATA
MOV (SP)+,R0 ;RESTORE R0
TYPE ;"CARRIAGE RETURN" & "LINE FEED"
RTS ;RETURN
7$: MOV @R0+,-(SP) ;SAVE @R0+ FOR TYPEOUT
;GO TYPE--OCTAL ASCII(ALL DIGITS)
;IS THERE ANOTHER NUMBER?
TST (R0)
BNE 8$ ;BR IF NO
TYPE 6$ ;TYPE TWO(2) SPACES
BR 4$ ;LOOP
8$: .ASCIZ / / ;TWO(2) SPACES
;*****

```

```

16112 ;*****SBTTL PRINT ROUTINES*****
16113 ;
16114 ;
16115 062202 005137 063236 PRINA: COM    @#PRIFLG    ;THESE TWO INSTRUCTIONS ARE
16116 062206 000002          RTI          ;USED BY THE BASIC TESTS TO VERIFY
16117          ;THE TRAP INSTRUCTION
16118
16119 .SBTTL TYPE ROUTINE
16120 ;*****
16121 ;ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
16122 ;THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
16123 ;NOTE1:  $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
16124 ;NOTE2:  $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
16125 ;NOTE3:  $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
16126 ;
16127 ;*CALL:
16128 ;*1) USING A TRAP INSTRUCTION
16129 ;*   TYPE    ,MESADR          ;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
16130 ;*OR
16131 ;*   TYPE    MESADR
16132 ;*
16133 ;*
16134 ;*
16135 ;*
16136 062210 105737 001057 STYPE:  TSTR  $TPFLG    ;IS THERE A TERMINAL?
16137 062214 000000          BPL  IS          ;BR IF YES
16138 062216 000000          BR   ;BRLT HERE IF NO TERMINAL
16139 062220 000430          BR   ;LEAVE
16140 062222 010046          MOV  RO,-(SP)    ;SAVE RO
16141 062224 017600 000002  MOV  @2(SP),RO    ;GET ADDRESS OF ASCIZ STRING
16142 062230 122737 000001  CMPB #APTENV,$ENV  ;RUNNING IN APT MODE
16143 062236 001011          BNE  ;NO, GO CHECK FOR APT CONSOLE
16144 062240 132737 000100 001141 BITB #APTSPOOL,$ENVM ;SPOOL MESSAGE TO APT CONSOLE
16145 062246 001405          BEQ  ;NO, GO CHECK FOR CONSOLE
16146 062250 010037 062260  MOV  RO,61S      ;SETUP MESSAGE ADDRESS FOR APT
16147 062254 004737 062726  JSR  PC,$ATV3    ;SPOOL MESSAGE TO APT
16148 062260 000000          ;MESSAGE ADDRESS
16149 062270 001003 000040 001141 BITB #APTCSUP,$ENVM ;APT CONSOLE SUPPRESSED
16150 062272 112046          BNE  ;YES, SKIP TYPE OUT
16151 062274 001005          MOVB (RO)+,-(SP) ;PUSH CHARACTER TO BE TYPED ONTO STACK
16152 062276 005726          BNE  ;BR IF IT ISN'T THE TERMINATOR
16153 062278 005726          TST  (SP)+       ;IF TERMINATOR POP IT OFF THE STACK
16154 062300 012600          MOV  (SP)+,RO    ;RESTORE RO
16155 062302 062716 000002  JSR  #2,(SP)     ;ADJUST RETURN PC
16156 062306 000002          RTI          ;RETURN
16157 062310 122716 000011  4$:  CMPB #HT,(SP)  ;BRANCH IF <HT>
16158 062314 001430          BEQ  ;BRANCH IF NOT <CRLF>
16159 062316 122716 000200  CMPB #CRLF,(SP) ;BRANCH IF NOT <CRLF>
16160 062322 001006          BNE  ;
16161 062324 007374          JSR  (SP)+       ;POP <CR><LF> EQUIV
16162 062326 004401          TYPE ;TYPE A CR AND LF
16163 062330 001115          SCRLF
16164 062332 105037 062466  CLR  $CHARCNT    ;CLEAR CHARACTER COUNT
16165 062336 000755          BR   ;GET NEXT CHARACTER
16166 062340 004737 062422  JSR  PC,$TYPEPC  ;GO TYPE THIS CHARACTER
16167 062344 122726 001056  6$:  CMPB $FILLC,(SP)+ ;IS IT TIME FOR FILLER CHARS.?

```

```

16168 062350 001350          BNE  ;IF NO GO GET NEXT CHAR.
16169 062352 013746 001054  MOV  $NULL,-(SP) ;GET # OF FILLER CHARS. NEEDED
16170          ;AND THE NULL CHAR.
16171 062356 105366 000001  7$:  DECB 1(SP)     ;DOES A NULL NEED TO BE TYPED?
16172 062362 002770          JSR  PC,$TYPEPC  ;BR IF NO--GO POP THE NULL OFF OF STACK
16173 062364 004737 062422  JSR  PC,$TYPEPC  ;GO TYPE A NULL
16174 062370 105337 062466  DECB  $CHARCNT   ;DO NOT COUNT AS A COUNT
16175 062374 000770          BR   ;LOOP
16176
16177 ;HORIZONTAL TAB PROCESSOR
16178
16179 062376 112716 000040  8$:  MOVB #-,(SP)    ;REPLACE TAB WITH SPACE
16180 062402 004737 062422  JSR  PC,$TYPEPC  ;TYPE A SPACE
16181 062406 132737 000007 062466 BITB #7,$CHARCNT ;BRANCH IF NOT AT
16182 062414 001372          BNE  ;TAB STOP
16183 062416 005726          TST  (SP)+       ;POP SPACE OFF STACK
16184 062420 000774          JSR  PC,$TYPEPC  ;GET NEXT CHARACTER
16185 062422 105777          BR   ;WAIT UNTIL PRINTER IS READY
16186 062426 100375          ;
16187 062430 116677 000002 116414  MOV  @2(SP),RSTPB ;LOAD CHAR TO BE TYPED INTO DATA REG.
16188 062436 122766 000015 000002  CMPB #CR,2(SP)   ;IS CHARACTER A CARRIAGE RETURN?
16189 062444 001003          BNE  ;BRANCH IF NO
16190 062446 105037 062466  CLR  $CHARCNT    ;YES--CLEAR CHARACTER COUNT
16191 062452 000406          BR   ;EXIT
16192 062454 122766 000012 1$:  CMPB #LF,2(SP)  ;IS CHARACTER A LINE FEED?
16193 062462 001404          BEQ  ;BRANCH IF YES
16194 062464 105227          INCB (PC)+      ;COUNT THE CHARACTER
16195 062466 000000          $CHARCNT,#WORD ;CHARACTER COUNT STORAGE
16196 062470 000207          $TYPEPC:RTS    PC
16197
16198 .SBTTL BINARY TO OCTAL (ASCII) AND TYPE
16199 ;*****
16200 ;THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
16201 ;OCTAL (ASCII) NUMBER AND TYPE IT
16202 ;*STYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
16203 ;*CALL:
16204 ;*   MOV    NUM,-(SP)          ;NUMBER TO BE TYPED
16205 ;*   TYPEPC ;CALL FOR TYPEOUT
16206 ;*   .BYTE  N                  ;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
16207 ;*   .BYTE  M                  ;M=1 OR 0
16208 ;*                                     ;1=TYPE LEADING ZEROS
16209 ;*                                     ;0=SUPPRESS LEADING ZEROS
16210 ;*
16211 ;*
16212 ;*
16213 ;*STYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
16214 ;*STYPOS OR $TYPEPC
16215 ;*CALL:
16216 ;*   MOV    NUM,-(SP)          ;NUMBER TO BE TYPED
16217 ;*   TYPON  ;CALL FOR TYPEOUT
16218 ;*
16219 ;*
16220 ;*STYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
16221 ;*CALL:
16222 ;*   MOV    NUM,-(SP)          ;NUMBER TO BE TYPED
16223 ;*   TYPOC  ;CALL FOR TYPEOUT

```

```

16224 062472 017646 000000
16225 062476 116637 000001 062715
16226 062504 112637 062717
16227 062510 062716 000002
16228 062516 112737
16229 062516 112737 000001 062715
16230 062524 112737 000006 062717
16231 062532 112737 000005 062714
16232 062540 010346
16233 062542 010446
16234 062542 010446
16235 062546 113704 062717
16236 062552 005404
16237 062554 062704 000006
16238 062560 110437 062716
16239 062564 113704 062716
16240 062570 016605 000012
16241 062574 005003
16242 062576 006105
16243 062600 000404
16244 062602 006105
16245 062604 006105
16246 062606 006105
16247 062610 010503
16248 062612 006103
16249 062614 105337 062716
16250 062620 100716
16251 062620 042710 177770
16252 062626 001002
16253 062630 005704
16254 062632 001403
16255 062634 005204
16256 062636 027703 000060
16257 062640 000040
16258 062646 110337 062712
16259 062652 104401 062712
16260 062656 105337 062714
16261 062662 003347
16262 062664 005402
16263 062666 005402
16264 062670 000744
16265 062672 012605
16266 062674 012604
16267 062676 012603
16268 062678 166656 000002 000004
16269 062680 012616
16270 062710 000002
16271 062712 000
16272 062713 000
16273 062714 000
16274 062715 000
16275 062716 000000

```

```

STYPOS: MOV 0(SP),-(SP) ;;PICKUP THE MODE
MOV 1(SP),SOFILL ;;LOAD ZERO FILL SWITCH
MOV 1(SP),SOMODE+1 ;;NUMBER OF DIGITS TO TYPE
ADD #4,0(SP) ;;ADJUST RETURN ADDRESS
STYPOC: MOV #1,SOFILL ;;SET THE ZERO FILL SWITCH
MOV #6,SOMODE+1 ;;SET FOR SIX(6) DIGITS
STYPON: MOV #5,SOCNT ;;SET THE ITERATION COUNT
MOV #3,-(SP) ;;SAVE R3
MOV #4,-(SP) ;;SAVE R4
MOV #5,-(SP) ;;SAVE R5
MOV #4,SOMODE+1,R4 ;;GET THE NUMBER OF DIGITS TO TYPE
NEG R4
ADD #6,R4 ;;SUBTRACT IT FOR MAX. ALLOWED
MOV #4,SOMODE ;;SAVE IT FOR USE
MOV #7,0(R4) ;;GET THE ZERO FILL SWITCH
MOV 12(SP),R5 ;;PICKUP THE INPUT NUMBER
CLR R3 ;;CLEAR THE OUTPUT WORD
ROL R5 ;;ROTATE MSB INTO "CH"
BR 35 ;;GO DO MSB
ROR R5 ;;FORM THIS DIGIT
ROR R5,R3
MOV R5,R3
R3
DECB SOMODE ;;GET LSB OF THIS DIGIT
BIC #177770,R3 ;;TYPE THIS DIGIT?
BNE 45 ;;GET RFD OF JUNK
TST R4 ;;TEST FOR 0
BEQ 55 ;;SUPPRESS THIS 0?
BR 55 ;;BR IF YES
INC #0,R3 ;;DON'T SUPPRESS ANYMORE 0'S
BIS #0,R3 ;;MAKE ASCII IF NOT ALREADY
MOV #3,6$ ;;SAVE FOR TYPING
TYPE 6$ ;;GO TYPE THIS DIGIT
DECB SOCNT ;;COUNT BY 1
BGT 25 ;;BR IF MORE TO DO
BLT 64 ;;BR IF DONE
INC 64 ;;INSURE LAST DIGIT ISN'T A BLANK
BR 25 ;;GO DO THE LAST DIGIT
MOV (SP)+,R5 ;;RESTORE R5
MOV (SP)+,R4 ;;RESTORE R4
MOV (SP)+,R3 ;;RESTORE R3
MOV 2(SP),4(SP) ;;SET THE STACK FOR RETURNING
MOV (SP)+,4(SP)
RTI ;;RETURN
-BYTE 0 ;;STORAGE FOR ASCII DIGIT
-BYTE 0 ;;TERMINATOR FOR TYPE ROUTINE
SOMODE: -BYTE 0 ;;OCTAL DIGIT COUNTER
SOFILL: -BYTE 0 ;;ZERO FILL SWITCH
-MWORD 0 ;;NUMBER OF DIGITS TO TYPE

```

.SBTTL APT COMMUNICATIONS ROUTINE
;;*****

```

16280 062720 112737 000001 063164
16281 062726 112737 000001 063162
16282 062734 000403
16283 062736 112737 000001 063164
16284 062744
16285 062744 010046
16286 062746 010146
16287 062750 105746 063162
16288 062754 001450
16289 062756 122737 000001 001140
16290 062764 001031
16291 062766 132737 000100 001141
16292 062774 001425
16293 062776 017600 000004
16294 063002 062766 000002 000004
16295 063010 005737 001120
16296 063014 001375
16297 063016 010037 001134
16298 063022 105720
16299 063024 001376
16300 063026 163700 001134
16301 063032 006200
16302 063034 010037 001136
16303 063040 012737 000004 001120
16304 063046 000413
16305 063050 017637 000004 063074
16306 063056 062766 000002 000004
16307 063064 013746 177776
16308 063070 064737 062210
16309 063074 000000
16310 063076
16311 063076 063164
16312 063102 001416
16313 063104 005737 001140
16314 063110 001413
16315 063112 005737 001120
16316 063116 001375
16317 063120 017637 000004 001122
16318 063126 062766 000002 000004
16319 063134 005237 001120
16320 063140 105037 063164
16321 063144 105037 063163
16322 063150 105037 063162
16323 063154 012601
16324 063156 012600
16325 063160 000207
16326 063164 000
16327 063164 000
16328 063164 000
16329 063166
16330 000200
16331 000001
16332 000010
16333 000040
16334
16335

```

```

$ATV1: MOV #1,$FFLG ;;TO REPORT FATAL ERROR
$ATV3: MOV #1,$MFLG ;;TO TYPE A MESSAGE
BR $ATVC
$ATV4: MOV #1,$FFLG ;;TO ONLY REPORT FATAL ERROR
$ATVC:
MOV R0,-(SP) ;;PUSH R0 ON STACK
MOV R1,(SP) ;;PUSH R1 ON STACK
TSTB $MFLG ;;SHOULD TYPE A MESSAGE?
BEQ 55 ;;IF NOT: BR
MPTENV,$ENV ;;OPERATING UNDER APT?
BNE 35 ;;IF NOT: BR
BITB $MPTSPool,$ENVM ;;SHOULD SPOOL MESSAGES?
MOV #4,(SP),R0 ;;GET MESSAGE ADDR.
ADD #2,4(SP) ;;BUMP RETURN ADDR.
TST $MSGTYPE ;;SEE IF DONE W/ LAST XMISSION?
BNE 15 ;;IF NOT: WAIT
MOV R0,$MSGAD ;;PUT ADDR IN MAILBOX
TSTR (R0)+ ;;FIND END OF MESSAGE
BNE 15
SUP $MSGAD,R0 ;;SUB START OF MESSAGE
ASR R0 ;;GET MESSAGE LNCTH IN WORDS
MOV R0,$MSGLCGT ;;PUT LENGTH IN MAILBOX
BR #4,$MSGTYPE ;;TELL APT TO TAKE MSG.
MOV #4,(SP),4$ ;;PUT MSG ADDR IN JSR LINKAGE
ADD #2,4(SP) ;;BUMP RETURN ADDRESS
JSR 177776,-(SP) ;;PUSH 177776 ON STACK
PC,$TYPE ;;CALL TYPE MACRO
-MWORD 0
5$:
10$: TSTB $FFLG ;;SHOULD REPORT FATAL ERROR?
BEQ 125 ;;IF NOT: BR
TST $ENV ;;RUNNING UNDER APT?
BEQ 125 ;;IF NOT: BR
TST $MSGTYPE ;;FINISHED LAST MESSAGE?
BNE 115 ;;IF NOT: WAIT
MOV #4,(SP),$FATAL ;;GET ERROR #
ADD #2,4(SP) ;;BUMP RETURN ADDR.
INC $MSGTYPE ;;TELL APT TO TAKE ERROR
CLRBB $FFLG ;;CLEAR FATAL FLAG
CLRBB $PLG ;;CLEAR LOG FLAG
CLRBB $MFLG ;;CLEAR MESSAGE FLAG
MOV (SP)+,R1 ;;POP STACK INTO R1
MOV (SP)+,R0 ;;POP STACK INTO R0
RTS PC ;;RETURN
-MESSG FLAG
-MLOG FLAG
-MFATAL FLAG
-EVEN
APTSIZE=200
APTENV=001
APTSPool=00
APTCSUP=040

```

.SBTTL TRAP DECODER

16336
16337
16338
16339
16340
16341
16342
16343 063166 010046
16344 063170 016600 000002
16345 063174 005740
16346 063176 111000
16347 063200 006300
16348 063202 016000 063222
16349 063206 000200
16350
16351
16352
16353
16354 063210 011646
16355 063212 016666 000004 000002
16356 063220 000002
16357
16358
16359
16360
16361
16362
16363
16364
16365 063222 063210
16366 063224 062210
16367 063226 062516
16368 063230 062472
16369 063232 062532
16370
16371
16372
16373
16374
16375 063234 000000
16376
16377 063236 000000
16378 063240 000000
16379 063242 000000
16380 063244 000000
16381 063246 000000
16382 063250 000000
16383 063252 000000
16384
16385 063254 000000
16386
16387
16388 063256 177400
16389 063260 177400
16390 063262 177400
16391 063264 177400

```
*****  
; THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION  
; AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS  
; OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL  
; GO TO THAT ROUTINE.
```

```
STRAP: MOV R0,-(SP) ;SAVE R0  
MOV 2(SP),R0 ;GET TRAP ADDRESS  
TST -(R0),R0 ;BACKUP BY 2  
MOV R0,R0 ;GET RIGHT BYTE OF TRAP  
ASL R0 ;POSITION FOR INDEXING  
MOV STRPAD(R0),R0 ;INDEX TO TABLE  
RTS R0 ;GO TO ROUTINE
```

```
;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
```

```
STRAP2: MOV (SP),-(SP) ;MOVE THE PC DOWN  
MOV 4(SP),2(SP) ;MOVE THE PSW DOWN  
RTI ;RESTORE THE PSW
```

```
.SBTTL TRAP TABLE
```

```
;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES CALLED  
;*BY THE "TRAP" INSTRUCTION.
```

```
ROUTINE  
-----  
STRPAD: WORD STRAP2 TRAP+1(104401) TTY TIMEOUT ROUTINE  
TYPE ;CALL=TYPE TRAP+2(104402) TYPE OCTAL NUMBER (WITH LEADING ZEROS)  
STYPOC ;CALL=TYPOC TRAP+3(104403) TYPE OCTAL NUMBER (NO LEADING ZEROS)  
STYPOS ;CALL=TYPOS TRAP+4(104404) TYPE OCTAL NUMBER (AS PER LAST CALL)  
STYPON ;CALL=TYPON
```

```
;FLAGS, CONSTANTS, AND VARIABLES
```

```
BPTLOC: 0 ;STORES 16 USER DEFINED MAINTENANCE  
;BREAKPOINTS  
PRIFLG: 0 ;FLAG USED BY BASIC TESTS FOR TRAP TEST  
ERRFLG: 0 ;FLAG USED BY BASIC TESTS FOR EMT TEST  
SELST: 0 ;STORES SR<8:0> FOR LOOP ON SELECTED TEST  
SCDFLG: 0 ;USED BY BASIC TESTS FOR IOT TEST  
RSWFLG: 0 ;FLAG USED BY BASIC TEST OF RSVD INSTR TRAP  
BERFLG: 0 ;FLAG USED BY BASIC TEST OF BUS ERROR TRAPS  
CATERR: 0 ;FLAGS USED BY BUS ERROR AND RSVD INSTR TRAP  
;SERVICE ROUTINES  
ONCE: 0 ;FLAGS PROGRAM TITLE HAS BEEN PRINTED  
;COMMON DATA STRUCTURES AND MISCELLANEOUS TABLES  
OBUF: 177400 ;DL11 OUTPUT TEST BUFFER  
177400  
177400  
177400
```

16392
16393 063266 000004
16394
16395 063276 063322
16396 063300 064032
16397 063302 064630
16398 063304 064634
16399 063306 063312
16400 063310 063316
16401
16402 063312 000000
16403 063314 000000
16404 063316 000000
16405 063320 000000
16406 063322 000000
16407 063324 177777
16408 063326 177400
16409 063330 000377
16410 063332 125252
16411 063334 052525
16412
16413
16414
16415
16416 063336 000000
16417 063340 000000
16418 063342 000000
16419 063344 177777
16420 063346 177777
16421 063350 177776
16422 063352 125252
16423 063354 063625
16424 063356 177777
16425 063360 052525
16426 063362 125252
16427 063364 177777
16428 063366 125252
16429 063370 125252
16430 063372 052525
16431 063374 052525
16432 063376 052525
16433 063400 125252
16434 063402 052525
16435 063404 125252
16436 063406 000000
16437 063410 125252
16438 063412 052525
16439 063414 000000
16440
16441
16442
16443
16444 063416 000000
16445 063420 000000
16446 063422 000000
16447 063424 177777

```
IBUF: .BLKW 4 ;DL11 INPUT TEST BUFFER  
ATA: DWTA  
DWTB  
DBTA  
DBTB  
MBOF0  
MBOF1
```

```
MBOF0: 0  
0  
MBOF1: 0  
0  
DWTA: 0  
-1  
177400  
377
```

```
ALUADD: 125252 ;ALSO SERVES AS NULL ENTRY FOR ALUADD  
052525
```

```
;*THIS TABLE OF 8 ENTRIES IS USED BY THE ALU ADD TEST IN THE  
;*COMBINED INSTRUCTION TESTS
```

```
000000 ;SRC OP1  
000000 ;DST OP1  
000000 ;ANS1  
077777 ;SRC OP2  
177777 ;DST OP2  
177776 ;ANS2  
125252 ;SRC OP3  
052525 ;DST OP3  
177777 ;ANS3  
052525 ;SRC OP4  
125252 ;DST OP4  
177777 ;ANS4  
125252 ;SRC OP5  
052525 ;DST OP5  
052525 ;ANS5  
052525 ;SRC OP6  
052525 ;DST OP6  
125252 ;ANS6  
052525 ;SRC OP7  
125252 ;DST OP7  
000000 ;ANS7  
125252 ;SRC OP8  
052525 ;DST OP8  
ANDTAB: 000000 ;ANS8 -- ALSO NULL ENTRY FOR ANDTAB
```

```
;*THIS TABLE OF 8 ENTRIES IS USED BY THE ALU "AND" TESTS IN THE  
;*COMBINED INSTRUCTION EXERCISER TESTS
```

```
000000 ;SRC OP1  
000000 ;DST OP1  
000000 ;ANS1  
177777 ;SRC OP2
```

16448	063426	177777	177777	};DST OP2
16449	063430	000000	000000	};ANS2
16450	063432	000000	000000	};SRC OP3
16451	063434	177777	177777	};DST OP3
16452	063436	177777	177777	};ANS3
16453	063440	177777	177777	};SRC OP4
16454	063442	000000	000000	};DST OP4
16455	063444	000000	000000	};ANS4
16456	063446	125252	125252	};SRC OP5
16457	063450	125252	125252	};DST OP5

16458	063452	000000	000000	};ANS5
16459	063454	052525	052525	};SRC OP6
16460	063456	052525	052525	};DST OP6
16461	063460	000000	000000	};ANS6
16462	063462	052525	052525	};SRC OP7
16463	063464	052525	052525	};DST OP7
16464	063466	052525	052525	};ANS7
16465	063470	052525	052525	};SRC OP8
16466	063472	125252	125252	};DST OP8
16467	063474	125252	125252	};ANS8 -- ALSO NULL ENTRY FOR ORTAB
16468				
16469				
16470				
16471				
16472	063476	000000	000000	};SRC OP1
16473	063500	000000	000000	};DST OP1
16474	063502	000000	000000	};ANS1
16475	063504	177777	177777	};SRC OP2
16476	063506	177777	177777	};DST OP2
16477	063510	177777	177777	};ANS2
16478	063512	000000	000000	};SRC OP3
16479	063514	177777	177777	};DST OP3
16480	063516	177777	177777	};ANS3
16481	063520	177777	177777	};SRC OP4
16482	063522	000000	000000	};DST OP4
16483	063524	177777	177777	};ANS4
16484	063526	125252	125252	};SRC OP5
16485	063530	125252	125252	};DST OP5
16486	063532	125252	125252	};ANS5
16487	063534	052525	052525	};SRC OP6
16488	063536	052525	052525	};DST OP6
16489	063540	052525	052525	};ANS6
16490	063542	125252	125252	};SRC OP7
16491	063544	125252	125252	};DST OP7
16492	063546	177777	177777	};ANS7
16493	063550	052525	052525	};SRC OP8
16494	063552	125252	125252	};DST OP8
16495	063554	177777	177777	};ANS8 -- ALSO NULL ENTRY FOR ALUSUB
16496				
16497				
16498				
16499				
16500	063556	000000	000000	};SRC OP1
16501	063560	000000	000000	};DST OP1
16502	063562	000000	000000	};ANS1
16503	063564	177777	177777	};SRC OP2
16504	063566	177777	177777	};DST OP2
16505	063570	000000	000000	};ANS2
16506	063572	125252	125252	};SRC OP3
16507	063574	052525	052525	};DST OP3
16508	063576	125252	125252	};ANS3
16509	063600	052525	052525	};SRC OP4
16510	063602	125252	125252	};DST OP4
16511	063604	052525	052525	};ANS4
16512	063606	125252	125252	};SRC OP5
16513	063610	125252	125252	};DST OP5

```

16514 063612 000000
16515 063614 054252
16516 063616 000000
16517 063618 052525
16518 063622 125253
16519 063624 052526
16520 063626 125253
16521 063630 054252
16522 063634 125253
16523 063636 005702
16524 063640 005002
16525 063644 005102
16526 063648 005303
16527 063652 005502
16528 063656 005602
16529 063660 005502
16530 063664 105202
16531 063668 105302
16532 063672 105502
16533 063676 105502
16534 063680 105202
16535 063684 105302
16536 063688 105202
16537 063692 105302
16538 063696 105502
16539 063700 105202
16540 063704 105302
16541 063708 105703
16542 063712 106202
16543 063716 106302
16544 063720 151302
16545 063724 074302
16546 063728 141302
16547 063732 141302
16548 063736 141302
16549 063740 111302
16550 063744 021102
16551 063748 014303
16552 063752 051302
16553 063756 006702
16554 063760 005402
16555 063764 161302
16556 063768 023112
16557 063772 033112
16558 063776 120312
16559 063780 131302
16560 063784 005712
16561 063788 105712
16562 063792 031312
16563 063796 031312
16564 063800 121312
16565 063804 131312
16566 063808 061302
16567 063812 000302
16568 063816 160302
16569 063820 160302
    
```

```

000000 ;ANS5
054252 ;SRC DP6
052525 ;DST DP6
000000 ;ANS6
052525 ;SRC DP7
125253 ;DST DP7
052526 ;ANS7
125253 ;SRC DP8
125253 ;DST DP8
125253 ;ANS8
    
```

```

INSTAB: TST R2 ;BEGINNING OF INSTRUCTION TABLE OF INSTRUCTIONS
          CLR R2 ;THAT TEST BUT SERVICE IN VARIOUS ROM LOCATIONS
          CUM R2
          INC R2
          DEC R2
          ADC R2
          SBC R2
          ASL R2
          ASL R2
          CLRB R2
          CUMB R2
          INCB R2
          DECB R2
          ADCB R2
          ADCB R2
          SFTB R2
          ASRB R2
          ASLB R2
          BISB (R3),R2
          XOR (R3),R2
          CMPB (R3),R2
          BITB (R3),R2
          BICB (R3),R2
          MOVB (R3),R2
          CMP (R3),R2
          BIT (R3),R2
          BIC (R3),R2
          BIS (R3),R2
          SXT R2
          NEG (R3),R2
          SUB (R3),R2
          CWT (R3),R2
          BIT (R3),R2
          CMPB (R3),R2
          BITB (R3),R2
          TST (R2)
          TSTB (R2)
          CWT (R3),R2
          BIT (R3),R2
          CMPB (R3),R2
          BITB (R3),R2
          ADD (R3),R2
          SWAB R2,R2
          SUB R3,R2
    
```

```

16570 063770 060302
16571 063772 010302
16572 063774 011302
16573 063776 110302
16574 064000 006102
16575 064004 102402
16576 064008 102402
16577 064012 102400
16578 064016 102000
16579 064020 000005
16580 064024 020302
16581 064028 030302
16582 064032 040303
16583 064036 120302
16584 064040 130302
16585 064044 140302
16586 064048 150302
16587 064032 000000
16588 064034 000001
16589 064036 000400
16590 064038 177401
16591 064040 052526
16592 064042 125253
16593 064044 125253
16594 064046 000000
16595 064048 000000
16596 064050 000000
16597 064052 000000
16598 064054 000000
16599 064056 000000
16600 064058 000040
16601 064156 000000
16602 064160 000000
16603 064162 000000
16604 064164 000000
16605 064166 000000
16606 064168 000000
16607 064170 000000
16608 064172 000000
16609 064174 000000
16610 064176 000000
16611 064178 000000
16612 064180 000000
16613 064182 000000
16614 064184 000000
16615 064186 000000
16616 064188 000000
16617 064190 000000
16618 064192 000000
16619 064194 000000
16620 064196 000000
16621 064198 001001
16622 064200 002002
16623 064202 003003
16624 064204 004004
    
```

```

          ADD R3,R2
          MOV R3,R2
          MOV (R3),R2
          MOVB R3,R2
          ROL R2
          ROLB R2
          NEGB R2
          BVS +2
          BVC +2
          RESET
          CMP R3,R2
          BIT R3,R2
          BIC R3,R2
          CMPB R3,R2
          BITB R3,R2
          BICB R3,R2
          BISB R3,R2

DWTB: 0 ;ALSO SERVES AS INSTAB TABLE TERMINATOR
      1
      400
      177401
      052526
      125253
; * MED TEST TABLES

TLOC1: .WORD 0
PSWHOL: .WORD 0
TABEG: .WORD 0
TABEND: .WORD 0
STCBLK: .BLKN 40
VADR: .WORD 0
PA1716: .WORD 0
PA1500: .WORD 0
TLOC2: .WORD 0

; *
; * TABLE II
; * FOLLOWING IS A TABLE OF INTERNAL REGISTER OPERATION CODES
; * USED FOR TESTING THE MED INSTRUCTION. LABELS CORRESPOND
; * TO REGISTER NAMES, THE HIGH BYTE IS THE READ OPERATION
; * CODE, THE LOW BYTE THE WRITE CODE.
; * NOTE: WHEN ADDING OR DELETING
; * ENTRIES IN THIS TABLE, CHECK DUAL
; * ADDRESSING TEST TO SEE THAT THE "SCRATCH
; * PAD LIMITS" ARE MAINTAINED.
; *

TBL2:
ASP1:
R1A: .BYTE 201,001 ;A SCRATCH PAD - LO
R2A: .BYTE 202,002 ;LOBYTE, HIBYTE=WRITE CODE, READ CODE
R3A: .BYTE 203,003
R4A: .BYTE 204,004
    
```


16625	064176	205	005
16627	064200	206	006
16628	064202	210	010
16629	064204	211	011
16630	064206	212	012
16631	064210	213	013
16632	064212	214	014
16633	064214	215	015
16634	064216	216	016
16635	064220	217	017
16636	064222	220	020
16637	064224	221	021
16638	064226	223	023
16639	064228	224	024
16640	064232	226	026
16641	064234	227	027
16642	064236	230	030
16643	064240	231	031
16644	064242	231	031
16645	064244	233	033
16646	064246	234	034
16647	064250	235	035
16648	064252	236	036
16649	064254	237	037
16650			
16651	064256		
16652	064256	241	041
16653	064260	242	042
16654	064262	243	043
16655	064264	245	045
16656	064266	245	045
16657	064270	246	046
16658	064272	250	050
16659	064274	251	051
16660	064276	252	052
16661	064278	253	053
16662	064302	254	054
16663	064304	255	055
16664	064306	256	056
16665	064310	257	057
16666	064312	260	060
16667	064314	261	061
16668	064316	262	062
16669	064320	263	063
16670	064322	266	066
16671	064324	270	070
16672	064326	272	072
16673	064330	273	073
16674	064332	274	074
16675	064334	275	075
16676	064336	276	076
16677	064340	277	077
16678			
16679			
16680	064342	300	100
16681	064344	301	101

RSA:	.BYTE	205,005	
R6A:	.BYTE	206,006	
FAC3.0:	.BYTE	210,010	
FAC3.1:	.BYTE	211,011	
FAC3.2:	.BYTE	212,012	
FAC3.3:	.BYTE	213,013	
FAC3.4:	.BYTE	214,014	
FAC3.5:	.BYTE	215,015	
UR6A:	.BYTE	216,016	
FDST3:	.BYTE	217,017	
WCSA.0:	.BYTE	220,020	
WCSA.1:	.BYTE	221,021	
GNMHS:	.BYTE	223,023	
CNSTS:	.BYTE	224,024	
CNSSM:	.BYTE	226,026	
CNSCDR:	.BYTE	227,027	
FAC1.0:	.BYTE	230,030	
FAC1.1:	.BYTE	231,031	
FAC1.4:	.BYTE	233,033	
FAC1.4:	.BYTE	234,034	
FAC1.5:	.BYTE	235,035	
PPSHI:	.BYTE	236,036	
ASP2:	.BYTE	237,037	
BSP1:			
R1B:	.BYTE	241,041	
R2B:	.BYTE	242,042	
R3B:	.BYTE	243,043	
R4B:	.BYTE	245,045	
R5B:	.BYTE	245,045	
R6B:	.BYTE	246,046	
FAC2.0:	.BYTE	250,050	
FAC2.1:	.BYTE	251,051	
FAC2.2:	.BYTE	252,052	
FAC2.3:	.BYTE	253,053	
FAC2.4:	.BYTE	254,054	
FAC2.5:	.BYTE	255,055	
UR6B:	.BYTE	256,056	
FDST2:	.BYTE	257,057	
WCSP.0:	.BYTE	260,060	
WCSP.1:	.BYTE	261,061	
WCSP.2:	.BYTE	262,062	
RZERRD:	.BYTE	263,063	
RVECT:	.BYTE	266,066	
FACO.0:	.BYTE	270,070	
FACO.1:	.BYTE	272,072	
FACO.4:	.BYTE	273,073	
FACO.4:	.BYTE	274,074	
FACO.5:	.BYTE	275,075	
FEA:	.BYTE	276,076	
BSP2:	.BYTE	277,077	
CSPL:			
LJAN:	.BYTE	300,100	
LSERV:	.BYTE	301,101	

;A SCRATCH PAD-HI

;B SCRATCH PAD - LO

;R SCRATCH PAD - HI

;C SCRATCH PAD

16682	064346	302	102
16683	064350	303	103
16684	064352	304	104
16685	064354	304	104
16686	064356	307	107
16687	064360	310	110
16688	064362	311	111
16689	064364	312	112
16690	064366	313	113
16691	064370	316	116
16692	064372	324	024
16693	064374	325	025
16694	064376	364	064
16695	064400	365	065
16696	064402	000000	065
16697			
16698			
16699			
16700			
16701			
16702			
16703			
16704			
16705	064404		
16706	064404	120	137
16707	064406	145	145
16708	064410	150	151
16709	064412	150	151
16710	064414	320	343
16711	064416	353	357
16712	064420	000000	
16713			
16714			
16715			
16716			
16717			
16718			
16719			
16720			
16721			
16722	064422		
16723	064422	200	000
16724	064424	207	007
16725	064426	240	040
16726	064430	247	047
16727	064432	314	114
16728	064434	317	117
16729			
16730			
16731	064436		
16732			
16733	064436	306	
16734	064437	106	
16735	064440	315	
16736	064441	115	
16737	064442	267	

LPBA:	.BYTE	302,102	
LCUA:	.BYTE	303,103	
LFGIM:	.BYTE	304,104	
LWJAM:	.BYTE	307,107	
LNAM:	.BYTE	310,110	
CNSCO:	.BYTE	311,111	
CNSC1:	.BYTE	312,112	
CNSC2:	.BYTE	313,113	
CNSC00:	.BYTE	316,116	
CNSTI:	.BYTE	324,024	
RT1A:	.BYTE	325,025	
RT2A:	.BYTE	364,064	
RT1B:	.BYTE	365,065	
RT2B:	.BYTE	0	
	.WORD	0	
**			
** TABLE III			
** THE FOLLOWING IS A LIST OF "NOP" OPERATION CODES			
** THAT WILL BE USED WITH A MED IN MED TEST 3 TO			
** ENSURE THAT A MED WITH THESE CODES WILL NOT HANG.			
**			
TBL3:			
NOPS:	.BYTE	120,137	;GROUP A
	.BYTE	145,145	;GROUP B
	.BYTE	150,151	;GROUP C
	.BYTE	150,151	;GROUP D
	.BYTE	320,343	;GROUP E
	.BYTE	353,357	;GROUP G
	.WORD	0	;A 0 TERMINATES TABLE
**			
** TABLE IV			
** THE LIST BELOW CONTAINS THOSE OPERATION CODES			
** CORRESPONDING TO THE INTERNAL REGISTERS WHICH MUST			
** BE TESTED SEPERATELY BECAUSE THEY ARE READ-ONLY			
** WRITE-ONLY, OR USED IN MACRO CODE EXECUTION, ETC. . .			
**			
TBL4:			
ROA:	.BYTE	200,000	;LOBYTE, HYBYTE - WRITE CODE, READ CODE
R7A:	.BYTE	207,007	;O REPLACES ANY NON EXISTENT CODES
ROB:	.BYTE	240,040	;EXCEPT IN THE CASE OF ROA
R7B:	.BYTE	247,047	
CNSTI2:	.BYTE	314,114	
CNSTI:	.BYTE	317,117	
**			
** TABLE V			
**			
TBL5:			
LCDTA:	.BYTE	306	;THIS TABLE CONTAINS THE OPERATION
	.BYTE	106	;CODES OF THOSE INTERNAL REGISTERS
MD:	.BYTE	315	;WHICH MUST BE TESTED USING THE
	.BYTE	115	;MICROBREAK REGISTER. THEIR
CNSCTL:	.BYTE	267	;ASSOCIATED MICRO-ADDRESSES ARE IN

16738 064443 067
16739 064444 140
16740 064445 143
16741 064446 143
16742 064447 143
16743 064450 344
16744 064451 144
16745 064452 345
16746 064453 146
16747 064454 349
16748 064455 149
16749 064456 347
16750 064457 351
16751 064460 152
16752 064461 152
16753 064462 153
16754 064463 000
16755
16756
16757
16758
16759
16760

JAM: .BYTE 067 ;THE NEXT TABLE
SERV: .BYTE 140
PBA: .BYTE 143
CLAG: .BYTE 143
FLAG: .BYTE 344
DREG: .BYTE 144
REV: .BYTE 345
SRG: .BYTE 146
COUNT: .BYTE 349
NUA: .BYTE 347
RES: .BYTE 351
DCS0: .BYTE 152
DCS1: .BYTE 153 ;INIT REG
.EVEN ;TABLE TERMINATOR

16761 064464 003330
16762 064466 003150
16763 064470 003375
16764 064472 003270
16765 064474 003240
16766 064476 003524
16767 064500 003160
16768 064502 003161
16769 064504 003170
16770 064506 003171
16771 064510 003344
16772 064512 003320
16773 064514 003345
16774 064516 003340
16775 064520 003350
16776 064522 003341
16777 064524 003351
16778 064526 003355
16779 064530 003720
16780 064532 003724
16781 064534 003721
16782
16783
16784
16785
16786
16787
16788
16789
16790
16791
16792
16793

;* TABLE VI
;*
TBL6:
ULCDTA: .WORD 3330 ;THIS TABLE CONTAINS THE MICRO-ADDRESSES
.WORD 3150 ;WHICH ARE LOADED INTO THE MICROBREAK
UMD: .WORD 3375 ;REG. TO TEST THE OPERATION CODES
.WORD 3240 ;CONTAINED IN THE PRECEDING TABLE.
UCNSCTL: .WORD 3224
.WORD 3160
UJAM: .WORD 3161
USERV: .WORD 3170
UPBA: .WORD 3171
UCUA: .WORD 3344
UFLAG: .WORD 3320
UDREG: .WORD 3345
UREV: .WORD 3340
USREG: .WORD 3350
UCOUNT: .WORD 3351
UNUA: .WORD 3355
URES: .WORD 3720
UDCS0: .WORD 3724
UINIT: .WORD 3721
UDCS1: .WORD 3721

16786 064536
16789 064536 000100 077600
16790 064542 000101 000010
16791 064546 000102 020000
16792 064552 000103 000004
16793 064556 000104 050000

;* TABLE VII
;*
;* THIS TABLE HOLDS THE OPERATION CODES AND THE CONSTANT
;* VALUE EXPECTED FOR CERTAIN INTERNAL REGISTERS.
TBL7:
CLJAM: .WORD 100,77600
CLSERV: .WORD 101,10
CLPBA: .WORD 102,20000
CLCUA: .WORD 103,4
CLFGIN: .WORD 104,50000

16794 064562 000105 054000
16795 064566 000107 024000
16796 064572 000110 177400
16797 064576 000111 177600
16798 064602 000112 100000
16799 064606 000113 000200
16800 064612 000114 000002
16801 064616 000116 000000
16802 064622 000117 000001
16803 064626 000000
16804
16805
16806
16807
16808
16809
16810
16811
16812
16813
16814
16815
16816
16817
16818
16819
16820
16821
16822
16823
16824
16825
16826
16827
16828
16829
16830
16831
16832
16833
16834
16835
16836
16837
16838
16839
16840
16841
16842
16843
16844
16845
16846
16847
16848
16849

CLWHAM: .WORD 105,54000
CLTAG: .WORD 107,54000
CCNSCO: .WORD 110,177400
CCNSCI: .WORD 111,177600
CCNSC2: .WORD 112,100000
CCST20: .WORD 113,400
CCNS2: .WORD 114,200
CCNST0: .WORD 116,0
CCNST1: .WORD 117,1
.WORD 0
DBTA: .EVEN
.BYTE 000,377,252,125
DBTB: .EVEN
.BYTE 000,001,120,253
;MESSAGE TABLES
EM1: .ASCII 'S/B DST '
EM2: .ASCII 'S/B DST '
EM4: .ASCII 'S/B DST '
EM7: .ASCII 'WAS DST '
EM6: .ASCII ' DEST<HT>
EM5: .ASCII '(IR)<HT> TEST<HT> (PC)<HT> (SP)<HT>(PSW)
EM10: .ASCIZ 'S/B RES WAS RES DST OP SRC OP TEST<HT> (PC)<HT> (SP)<HT>(PSW)
EM3: .ASCIZ 'S/B SP<HT>WAS SP<HT> (IR)<HT> TEST<HT> (PC)<HT>(PSW)
DH2: .ASCIZ <HT><HT> IS R3'
DH4: .ASCIZ <HT><HT> IS R5'
FOP1: .ASCIZ <15><12>'END PASS # '

064630 000 377 252
064633 125
064634 000 001 120
064637 253
064640
064640
064640 027523 020102 051504
064646 020124
064650 040527 020123 051504
064656 020124
064660 042040 051505 004524
064666 024040 051111 004451
064674 052040 051505 004524
064702 024040 041520 004451
064710 024040 050123 004451
064716 050050 053523 000051
064724 027523 020102 042522
064732 020123 040527 020123
064740 042522 020123 051504
064746 020124 050117 020040
064754 051123 050103 050117
064762 020040 042524 052123
064770 020011 050050 024503
064776 020011 051450 024520
065004 024011 051520 024527
065013 000
065013 123 041057 051440
065020 004520 040527 020123
065026 050123 020011 044450
065034 024522 020011 042524
065042 020011 050117 050050
065050 024503 024011 051520
065056 024527 000
065061 05 011 020011 051511
065066 051040 000063
065072 004411 044440 020123
065078 032510 000
065103 015 042412 042116
065110 050040 051501 020123

16850	065116	020043	000				
16851	065117	01	051105	047522	EDP2:	.ASCIZ <HT>"ERROR COUNT = "	
16852	065118	020122	047503	047125			
16853	065134	020124	020075	000			
16854	065141	015	041412	045521	IDENT1:	.ASCIZ <15><12>"CQKDAC KD11-K BASIC LOGIC TESTS"<15><12>	
16855	065146	040504	004503	042113			
16856	065154	030461	045455	041040			
16857	065170	051101	041511	045040			
16858	065170	043517	041511	052040			
16859	065176	051505	051524	005015			
16860	065204	000					
16861	065205	015	052012	040522	BEMSG:	.ASCIZ <CR><LF>"TRAPPED TO 4 PC = "	
16862	065210	050170	041105	052040			
16863	065216	020127	020064	041520			
16864	065226	036440	000040				
16865	065232	005015	051124	050101	RSMSG:	.ASCIZ <CR><LF>"TRAPPED TO 10 PC = "	
16866	065240	042520	020104	047524			
16867	065246	030440	020060	041520			
16868	065256	042524	052123		EM11:	.ASCIZ "TESTS SKIPPED"	
16869	065266	045523	050111	042520			
16871	065274	000104					
16872	065276	020040	041520	042411	DH11:	.ASCIZ " PC"<HT>"EXPTD"<HT>"ACTUAL"<HT>"(TEST #'S)"	
16873	065304	050130	052103	004504			
16874	065312	041501	051524	046101			
16875	065320	024011	042524	052123			
16876	065326	021440	051447	000051			
16877	065334	042515	020104	044504	EM12:	.ASCIZ /MED DID NOT ABORT IN USER MODE/	
16878	065342	020104	047516	020124			
16879	065350	041101	051117	020124			
16880	065356	041101	051117	042524			
16881	065364	020122	047515	042504			
16882	065372	000					
16883	065373	115	042105	042440	EM13:	.ASCIZ /MED EXECUTED IN USER MODE/	
16884	065400	042530	052503	042524			
16885	065406	020104	047111	052440			
16886	065411	042527	020123	047515			
16887	065422	042504	000				
16888	065425	115	042105	041440	EM14:	.ASCIZ /MED CHANGED PSW/	
16889	065432	040510	043516	042105			
16890	065440	050040	053523	000			
16891	065452	051115	040505	047522	EM15:	.ASCIZ /MICROBREAK TRAP-TO-4 DID NOT OCCUR/	
16892	065452	051115	040505	020113			
16893	065460	051124	050101	052055			
16894	065466	026517	020064	044504			
16895	065474	020104	047516	020124			
16896	065502	047517	052503	000122	EM17:	.ASCIZ /LOGCUA LOGGED WRONG/	
16897	065510	047514	043527	049507			
16898	065510	046040	043527	049507			
16899	065524	020104	051127	047117			
16900	065532	000107					
16901	065534	051503	020120	047503	EM21:	.ASCIZ /CSP CONSTANT WRONG/	
16902	065542	051516	040524	052116			
16903	065550	053440	047522	043516			
16904	065557	000					
16905	065557	102	042101	042040	EM22:	.ASCIZ /BAD DATA READ BY A MED/	

16906	065564	052101	020101	042522			
16907	065572	042101	041040	020131			
16908	065600	020101	042515	000104	EM23:	.ASCIZ /NO ODD PC TRAP/	
16909	065606	047440	047440	051124			
16910	065614	050040	020103	051124			
16911	065622	050101	000				
16912	065625	117	042104	040440	EM24:	.ASCIZ /ODD ADR. BIT NOT SET IN CPU ERR REG OR LOG JAM/	
16913	065632	051104	020056	044502			
16914	065640	020124	047516	020124			
16915	065646	020124	047514	047117			
16916	065654	041440	052520	042440			
16917	065662	051122	051040	043505			
16918	065670	047440	020122	047514			
16919	065676	020107	040512	000115			
16920	065704	044120	051531	041040	EM26:	.ASCIZ /PHYS BA LOGGED WRONG/	
16921	065712	020101	047514	043507			
16922	065720	042105	053440	047522			
16923	065726	043516	000				
16924	065731	103	041501	042510	EM27:	.ASCIZ /CACHE PARITY ERROR LOGGED IN BAKUP MODE/	
16925	065736	050040	051101	052111			
16926	065744	020131	041505	047509			
16927	065752	020124	044514	043509			
16928	065760	042105	044440	020116			
16929	065766	040502	052513	020120			
16930	065774	047515	042504	000			
16931	066001	050103	041501	042510	EM30:	.ASCIZ /CACHE PARITY TRAPPED WHEN DISABLED/	
16932	066006	050040	051101	052101			
16933	066014	020131	050124	050101			
16934	066022	042520	020104	044127			
16935	066030	047105	042040	051511			
16936	066036	041101	042514	000104			
16937	066044	047111	052123	027122	EM41:	.ASCIZ /INSTR. NOT ABORTED IN CACHE ABORT MODE/	
16938	066052	047043	052124	040440			
16939	066060	047502	052122	041055			
16940	066066	044440	020116	040503			
16941	066074	044103	020105	041101			
16942	066102	051117	020124	047515			
16943	066110	042504	000				
16944	066120	020115	046505	051117	EM32:	.ASCIZ /MEMORY ERR REG INCORRECT/	
16945	066126	020115	051105	020122			
16946	066126	042522	020107	047111			
16947	066134	047503	051122	041505			
16948	066142	000124					
16949	066144	044524	042515	052517	EM33:	.ASCIZ /TIMEOUT BIT NOT SET IN CPU ERR REG OR LOG JAM/	
16950	066152	044524	042515	052517			
16951	066160	047516	020124	042523			
16952	066166	020124	047111	041440			
16953	066174	052520	042440	051122			
16954	066202	051040	043505	047440			
16955	066210	020122	047514	020107			
16956	066222	040512	000				
16957	066222	047516	044440	046114	EM34:	.ASCIZ /NO ILLEGAL INTERNAL ADR TRAP/	
16958	066230	043505	046101	044440			
16959	066236	052116	051105	040516			
16960	066244	020114	042101	020122			
16961	066252	051124	050101	000			

16962	066257	111	052116	047122
16963	066264	046101	040440	051104
16964	066272	042440	051122	043040
16965	066300	051111	047040	042117
16966	066306	051440	052105	044440
16967	066314	020116	050103	020125
16968	066322	051105	020122	042522
16969	066330	020107	051117	046040
16970	066346	043517	045040	046501
16971	066344	000		
16972	066345	114	051501	020124
16973	066352	047111	051124	052057
16974	066360	040522	020120	042526
16975	066366	052103	051117	047040
16976	066374	051117	046040	043517
16977	066402	042507	020104	047111
16978	066410	043040	040514	020107
16979	066416	042522	000107	
16980	066422	047514	020107	044506
16981	066430	051522	020124	047515
16982	066436	047504	042040	042111
16983	066444	047040	052117	044440
16984	066452	044116	041111	052111
16985	066460	042440	051122	051117
16986	066466	046040	043517	040440
16987	066474	052106	052103	043040
16988	066502	051111	051103	042440
16989	066510	051122	051117	000
16990	066515	105	051122	051117
16991	066522	046040	043517	053440
16992	066530	051501	047040	052117
16993	066536	051040	043505	045116
16994	066544	046107	043505	020054
16995	066552	042117	020104	042101
16996	066560	020122	044502	020124
16997	066566	046103	020122	047111
16998	066574	041440	052520	051105
16999	066592	00612		
17000	066604	047516	041440	041501
17001	066612	042510	050040	051101
17002	066620	052111	020131	051124
17003	066626	050101	000	
17004	066631	114	020111	020046
17005	066636	044040	041040	052111
17006	066644	020105	020046	040524
17007	066652	020107	040520	044522
17008	066660	054524	041040	052111
17009	066666	020123	047516	020124
17010	066674	042523	020124	047111
17011	066702	046040	043040	051111
17012	066710	051105	044526	042503
17013	066716	000		
17014	066717	114	020117	020046
17015	066724	044510	041040	052131
17016	066732	020105	020046	040524
17017	066740	020107	040520	044522

EM35: .ASCIZ /INTRNAL ADR ERR BIT NOT SET IN CPU ERR REG OR LOG JAM/

EM36: .ASCIZ "LAST INTR/TRAP VECTOR NOT LOGGED IN FLAG REG"

EM37: .ASCIZ /LOG FIRST MODE DID NOT INHIBIT ERROR LOG AFTER FIRST ERROR/

EM40: .ASCIZ /ERROR LOG WAS NOT REENABLED, ODD ADR BIT CLR IN CPUERR/

EM31: .ASCIZ /NO CACHE PARITY TRAP/

EM42: .ASCIZ /LO & HI BYTE & TAG PARITY BITS NOT SET IN LOG SERVICE/

EM43: .ASCIZ /LO & HI BYTE & TAG PARITY BITS NOT SET IN MEM ERR REG/

17018	066746	054524	041040	052111
17019	066754	020123	047516	020124
17020	066762	042523	020124	047111
17021	066770	046440	046505	042440
17022	066776	051042	051040	043505
17023	067004	047040		
17024	067005	103	041501	042510
17025	067012	052040	043501	046040
17026	067020	043517	042507	020104
17027	067026	051127	047117	000107
17028	067034	040503	044103	020105
17029	067042	046040	046040	046040
17030	067050	042507	042507	040104
17031	067056	051127	047117	000107
17032	067064	044505	020123	042523
17033	067072	020124	047503	042116
17034	067100	041440	042117	051505
17035	067106	053440	047522	043516
17036	067114	000		
17037	067115	105	051511	043440
17038	067122	053101	020105	051127
17039	067130	047117	020107	042522
17040	067136	052523	052114	026517
17041	067143	101	052124	000
17042	067150	047110	051103	046505
17043	067156	047105	020124	042050
17044	067164	041505	042522	052115
17045	067172	020051	044504	020104
17046	067200	047516	020124	047117
17047	067206	052503	020125	047111
17048	067214	042440	051511	000
17049	067221	040	051520	004527
17050	067226	042522	026507	040527
17051	067234	026523	042524	026507
17052	067242	044461	042524	026507
17053	067250	042523	026502	042522
17054	067256	025507	004461	
17055	067262	020040	041520	020011
17056	067270	044450	024522	020011
17057	067276	042524	052123	000
17058	067303	040	050040	004503
17059	067310	042515	041500	042117
17060	067316	020105	044515	051103
17061	067324	041117	020113	042522
17062	067332	027107	000	
17063	067335	040	050040	004503
17064	067342	042515	041505	042520
17065	067350	020105	054105	042520
17066	067356	052103	020104	042522
17067	067364	042503	053111	000104
17068	067372	020040	041520	000
17069	067377	040	050040	004503
17070	067400	050103	042525	051127
17071	067412	046040	043517	040512
17072	067420	000115		
17073	067422	020040	041520	043011

EM45: .ASCIZ /CACHE TAG LOGGED WRONG/

EM16: .ASCIZ /CACHE DATA LOGGED WRONG/

EHEIS1: .ASCIZ "EIS SET COND CODES WRONG"

EHEIS2: .ASCIZ "EIS GAVE WRONG RESULT"

EM46: .ASCIZ "AUTO-INCREMENT (DECREMT) DID NOT OCCUR IN EIS"

DHEIS1: .ASCII "PSW<HT>"REG-WAS-REG+1"<HT>"REG-S/B-REG+1"<HT>

DH46: .ASCIZ " PC<HT>" (IR)<HT>" TEST"

DH15: .ASCIZ / PC/<HT>/MEDCODE MICROBK REG./

DH17: .ASCIZ / PC/<HT>/MEDCODE EXPECTD RECEIVD/

DH23: .ASCIZ / PC/

DH24: .ASCIZ / PC/<HT>/CPUERR/<HT>/LOGJAM/

DH25: .ASCIZ / PC/<HT>/FLGREG/

	11080	11102	11105	11124	11146	11149	11168	11190	11211	11214	11233	11254	11257
	12887*	12809*	12912	13863	13942*	13948	14342	14373	14393	14410*	14446	14446*	14485
MBUF1	063316	14517	14546*	16399	16402#								
	2157	2716	2714	2719*	2932	3313	3338	3363	3388	7846*	7850	7875*	7879
	7904*	7908	7933*	7937	7962*	7966	7999*	7995	8020*	8024	8049*	8053	8657
	8685	8715	9522	10559	10721	10807	10867	10867	10901	10931	10961	10991	12888*
MD	= 064440	12925*	13157	13167	13196	13206	13864	13908*	13914	16400	16404#		
MED	= 076600	12155#	13801	14908	14921	14952	14964	14964	14998	15011	15040	15043	15051
		15085	15088	15090	15093	15104	15117	15120	15124	15136	15139	15141	15162
		15186	15193	15205	15232	15234	15236	15241	15247	15253	15259	15265	15298
		15337	15352	15366	15383	15414	15455	15461	15470	15482	15503	15514	15528
		15443	15443	15443	14925	14927#	15611	15637	15655				
MEDHLT	055632												
MEDT1	055642												
MEDT10	0556206												
MEDT11	0556260												
MEDT3	0556042	15143											
MEDT4	0556100												
MED0	055612												
MED1	055514												
MEMERR	= 177744	15306	15310	15494									
NOPS	064404												
NJA	064456												
OBUP	063256												
ONCE	063254	4603	16388#		4634*	4636*	16385#						
ORTAB	063474	14187	14192	16467#									
PABORT	= 000200	1223	15447										
PA1500	064162												
PA1716	064160												
PBA	= 067772												
PIRO	= 000240												
PIROVE	= 063236	1930	4401*	4410*	16115*	16377#							
PRINA	= 062202	4403	16115#										
PRO	= 000000	752											
PR1	= 000040	754											
PR2	= 000100	755											
PR3	= 000140	756											
PR4	= 000200	757											
PR5	= 000240	758											
PR6	= 000300	759											
PS	= 002240	732											
PSW	= 177776	733	733	1446	1467	1485	1496	1505	1514	1563*	1577	1601	1605
		1432	1468	1685*	1706	1707*	1727	1728*	1791	1807	1997*	2017*	2026
		1631	1684	4238*	4262	4266*	4280*	4288	4292*	4299*	4306*	4341*	4349
		4234	4238*	4360*	4382	4388*	4639*	4681	4687	4701	4709	4723	4729
		4360*	4367*	4380*	4385	4388*	4639*	4681	4687	4701	4709	4723	4729
		12417*	12448*	12444	12444	12449*	12461*	12462	12470	12470	12474	12474*	12474*
		12564	12570*	12587	12594*	12613*	12628*	12644*	12659*	12675*	12690*	12706*	12711*
		12752	12772*	12788*	12804*	12820*	12840*	12850*	12886*	12907*	12943*	14682*	14712
		12800	14920*	14944*	15774*	12820*	12840*	12850*	12886*	12907*	12943*	14682*	14712
PSWHQL	064050												
PTRP1	057712												
PWRUP	061054	15458	15469#										
		15764	15773#										

PWRVEC	= 000024	824#	15728*	15729*	15738*	15744*	15759*	15760*
RCSR	= 177560	1260	4492	4578				
RDBR	= 177562	1261						
RDFLAG	= 000144	1236		15091	15118			
RDLUA	= 000103	1237						
RDLDA	= 000106	1250		15529				
RDLFCI	= 000104	1246		15543				
RDLJAM	= 000100	1238		15656	15671			
RDLPA	= 000102	1240		15367	15415	15590	15627	
RDLSPR	= 000101	1240		15353	15504			
RDLTAC	= 000107	1252		15315				
RDLWHA	= 000105	1248						
RDWHAM	= 000022	1234		15086	15137			
RES	= 064457	16750						
RESTAR	= 061176	15839	15848	15850#	15853	15907	15914	
RESVEC	= 000010	16719	14905*	14918*				
REV	= 064453	16719						
RSBERT	= 061150	15842						
RSERR	= 061122	4456						
RSMSG	= 065232	15841		14918	15835#			
RSVFLG	= 063246	4444*		15832*	16381#			
RSVTST	= 061114	4441						
RT1A	= 064372	16692						
RT1B	= 064376	16694						
RT2A	= 064374	16693						
RT2B	= 064400	16695						
RVECT	= 064322	16670						
RZERO	= 064320	16669						
RO1	= 064422	16722						
ROB	= 064426	16725						
R1A	= 064166	16622						
R1B	= 064256	16652						
R2A	= 064170	16623						
R2B	= 064260	16654						
R3A	= 064172	16654						
R3B	= 064262	16654						
R4A	= 064174	16625						
R4B	= 064264	16655						
R5A	= 064176	16626						
R5B	= 064266	16656						
R6A	= 064178	16627						
R6B	= 064270	16657						
R7A	= 064424	15009	16724#					
R7B	= 064430	15024*	16726#					
SCCPLG	= 063244	4318		15994*	16380#			
SCOPE	= 061022	4318						
SELTST	= 063242	15940*		15942	16379#			
SERV	= 064445	16740#						
SOBERR	= 043012	11975						
S0B1	= 042764	11970#						
S0B2	= 043000	11960		11978#				
S0B3	= 043056	11960						
S0B4	= 043160	12049						
S0B5	= 042772	11974#						
SRECC	= 064454	16747#						
STACK	= 001000	723	1431	1918	1934	14903	15773	

TST164	012342	4334#	
TST165	012464	4369#	4374#
TST169	012540	4377#	
TST169	012520	4311#	4419#
TST177	004054	2184	2192#
TST170	012664	4431	4438#
TST171	012746	4462#	
TST173	013016	4475#	4483#
TST173	013116	4513#	
TST174	013156	4521#	4532#
TST175	013210	4541	4550#
TST176	013244	4559	4576#
TST177	013556	4648#	
TST177	003350	1961	1969#
TST170	004110	2700	2709#
TST200	013600	4652	4672#
TST201	013620	4669	4676#
TST202	013660	4689	4696#
TST203	013722	4711	4718#
TST204	013760	4731	4738#
TST205	014020	4753	4760#
TST206	014042	4767	4774#
TST207	014064	4782	4789#
TST211	004154	2224	2232#
TST210	014106	4797	4804#
TST211	014130	4817	4819#
TST211	014130	4817	4819#
TST214	014176	4841	4848#
TST214	014220	4856	4863#
TST215	014242	4871	4878#
TST216	014264	4886	4893#
TST217	014332	4912	4919#
TST220	004374	2247	2255#
TST221	014374	4933	4940#
TST221	014442	4959	4966#
TST222	014474	4976	4983#
TST222	014622	5029	5035#
TST222	014622	5029	5035#
TST225	014926	5071	5078#
TST226	014764	5089	5096#
TST227	015034	5115	5122#
TST231	004250	2264	2272#
TST230	015104	5141	5148#
TST233	015134	5163	5170#
TST233	015210	5220	5227#
TST233	015354	5246	5253#
TST234	015424	5272	5279#
TST235	015472	5298	5305#
TST236	015542	5325	5332#
TST237	015610	5351	5358#
TST240	015672	5379	5386#
TST241	015744	5406	5413#
TST242	016014	5432	5439#
TST243	016062	5458	5465#
TST244	016132	5484	5491#

TST245	016200	5510	5517#
TST246	016250	5536	5543#
TST247	016320	5562	5569#
TST250	004360	2312	2320#
TST250	016376	5588	5596#
TST251	016452	5616	5624#
TST252	016530	5644	5652#
TST253	016604	5672	5680#
TST254	016662	5700	5708#
TST255	016740	5728	5736#
TST256	017020	5759	5766#
TST257	017072	5784	5792#
TST258	004434	2341	2350#
TST260	017170	5819	5827#
TST261	017244	5845	5853#
TST262	017330	5876	5884#
TST263	017404	5902	5910#
TST264	017464	5926	5934#
TST265	017534	5957	5965#
TST266	017600	5984	5991#
TST267	017646	6011	6018#
TST270	004470	2364	2369#
TST270	017714	6037	6044#
TST271	017760	6062	6069#
TST272	020030	6088	6095#
TST273	020076	6114	6121#
TST274	020144	6140	6147#
TST275	020214	6166	6173#
TST276	020322	6192	6199#
TST277	020384	6218	6225#
TST303	003364	1974	1981#
TST300	004526	2379	2387#
TST300	020376	6243	6250#
TST301	020444	6269	6276#
TST303	020514	6295	6302#
TST304	020580	6321	6328#
TST304	020630	6347	6354#
TST305	020700	6372	6379#
TST306	020744	6398	6405#
TST307	021014	6424	6431#
TST311	004552	2395	2403#
TST310	021062	6450	6457#
TST311	021132	6476	6483#
TST312	021200	6502	6509#
TST313	021250	6528	6535#
TST314	021320	6554	6561#
TST315	021366	6580	6587#
TST316	021440	6607	6615#
TST317	021516	6636	6644#
TST320	004606	2413	2421#
TST320	021604	6668	6676#
TST323	021666	6702	6709#
TST323	021744	6739	6747#
TST324	022020	6754	6762#
TST324	022074	6781	6789#
TST325	022152	6808	6816#

TST3326	022226	6835	6843#
TST3327	022304	6862	6870#
TST3328	004646	2482	2440#
TST3329	024430	6882	6890#
TST3330	024430	6915	6923#
TST3331	022510	6942	6950#
TST3332	022562	6969	6977#
TST3333	022562	6996	7004#
TST3334	024440	7026	7034#
TST3335	024446	7050	7058#
TST3336	024710	7077	7085#
TST3337	023046	7077	7085#
TST3338	004702	2453	2461#
TST3339	023122	7104	7112#
TST3340	023200	7131	7139#
TST3341	023200	7158	7166#
TST3342	023332	7185	7193#
TST3343	023332	7212	7220#
TST3344	023410	7239	7247#
TST3345	023464	7266	7274#
TST3346	023534	7293	7301#
TST3347	023604	7320	7328#
TST3348	024736	7347	7355#
TST3349	023654	7374	7382#
TST3350	023724	7401	7409#
TST3351	023774	7428	7436#
TST3352	024054	7455	7463#
TST3353	024124	7482	7490#
TST3354	024200	7509	7517#
TST3355	024200	7536	7544#
TST3356	024244	7563	7571#
TST3357	024324	7590	7598#
TST3358	004772	2494	2502#
TST3359	024406	7621	7629#
TST3360	024446	7648	7656#
TST3361	024446	7675	7683#
TST3362	024526	7702	7710#
TST3363	024600	7729	7737#
TST3364	024654	7756	7764#
TST3365	024722	7783	7791#
TST3366	024770	7810	7818#
TST3367	025004	7837	7845#
TST3368	005026	7864	7872#
TST3369	025112	7891	7899#
TST3370	025166	7918	7926#
TST3371	025222	7945	7953#
TST3372	025222	7972	7980#
TST3373	025242	7999	8007#
TST3374	025400	8026	8034#
TST3375	025400	8053	8061#
TST3376	025470	8080	8088#
TST3377	025556	8107	8115#
TST400	003402	1988	1996#
TST401	005066	4910	4918#
TST402	025636	7939	7947#
TST403	026024	7968	7976#
TST404	026112	7997	8005#
TST405	026202	8026	8034#
TST406	026272	8055	8063#
TST407	026344	8084	8092#

TST407	026416	8098	8106#
TST411	005132	2560	2568#
TST410	026466	8120	8128#
TST411	026536	8142	8150#
TST412	026606	8169	8177#
TST413	026656	8196	8204#
TST414	026656	8223	8231#
TST415	027002	8250	8258#
TST416	027056	8278	8286#
TST417	027136	8307	8315#
TST420	005174	2583	2591#
TST421	027336	8330	8338#
TST422	027336	8357	8365#
TST423	027434	8384	8392#
TST424	027510	8411	8419#
TST425	027564	8438	8446#
TST426	027636	8465	8473#
TST427	027720	8492	8500#
TST433	005236	2606	2614#
TST430	030000	8562	8570#
TST431	030060	8590	8598#
TST432	030136	8618	8626#
TST433	030136	8645	8653#
TST434	030260	8672	8680#
TST435	030344	8701	8709#
TST436	030432	8732	8740#
TST437	030506	8759	8767#
TST438	030574	8786	8794#
TST440	030664	8813	8821#
TST441	030640	8841	8849#
TST442	030716	8868	8876#
TST443	030766	8895	8903#
TST444	031036	8922	8930#
TST445	031106	8949	8957#
TST446	031172	8976	8984#
TST447	031272	9003	9011#
TST450	005322	2644	2652#
TST451	031356	9030	9038#
TST452	031432	9057	9065#
TST453	031506	9084	9092#
TST454	031644	9111	9119#
TST454	031742	9138	9146#
TST455	031836	9165	9173#
TST456	031932	9192	9200#
TST457	032036	9219	9227#
TST458	032132	9246	9254#
TST459	032236	9273	9281#
TST460	032336	9300	9308#
TST461	032430	9327	9335#
TST462	032520	9354	9362#
TST463	032610	9381	9389#
TST464	032710	9408	9416#
TST465	032800	9435	9443#
TST466	032900	9462	9470#
TST467	033042	9489	9497#
TST47	005420	2681	2689#

TEST470	033104	9518	9526#
TEST471	033146	9539	9548#
TEST472	033212	9561	9569#
TEST473	033264	9589	9598#
TEST474	033336	9617	9624#
TEST475	033374	9637	9644#
TEST476	033454	9656	9663#
TEST477	033502	9680	9688#
TEST50	003424	2002	2010#
TEST50	005462	2704	2712#
TEST500	033550	9704	9711#
TEST501	033644	9724	9732#
TEST503	033752	9752	9760#
TEST504	034030	9805	9813#
TEST505	034110	9832	9840#
TEST506	034162	9854	9862#
TEST507	034234	9876	9884#
TEST510	005520	2721	2729#
TEST510	034360	9898	9906#
TEST511	034350	9920	9928#
TEST512	034420	9942	9950#
TEST513	034470	9964	9972#
TEST514	034542	9986	9994#
TEST515	034616	10014	10021#
TEST516	034676	10045	10052#
TEST517	034752	10072	10079#
TEST52	005544	2744#	
TEST520	035020	10099	10106#
TEST521	035076	10126	10133#
TEST523	035150	10154	10161#
TEST524	035222	10181	10188#
TEST524	035274	10208	10215#
TEST525	035346	10234	10241#
TEST526	035430	10262	10269#
TEST527	035504	10290	10297#
TEST530	005600	2754	2761#
TEST530	035666	10320	10328#
TEST531	035642	10349	10357#
TEST532	035724	10378	10386#
TEST533	036006	10407	10414#
TEST534	036070	10435	10443#
TEST535	036150	10464	10472#
TEST536	036230	10492	10500#
TEST537	036304	10520	10527#
TEST54	005632	2771	2779#
TEST540	036354	10547	10555#
TEST543	036434	10575	10583#
TEST543	036504	10602	10610#
TEST545	036556	10629	10636#
TEST544	036630	10656	10664#
TEST545	036702	10682	10690#
TEST546	036754	10709	10717#
TEST547	037032	10736	10744#
TEST55	005700	2796	2804#
TEST550	037114	10764	10772#

TEST551	037172	10793	10801#
TEST552	037260	10823	10831#
TEST553	037340	10853	10861#
TEST554	037426	10883	10891#
TEST555	037520	10917	10925#
TEST556	037612	10947	10955#
TEST557	037676	10977	10985#
TEST56	005750	2821	2829#
TEST560	037752	11006	11014#
TEST561	040050	11036	11044#
TEST562	040064	11046	11054#
TEST563	040132	11068	11076#
TEST564	040202	11090	11098#
TEST565	040252	11112	11120#
TEST566	040322	11134	11142#
TEST567	040374	11156	11164#
TEST570	040446	11178	11186#
TEST571	040510	11199	11207#
TEST572	040560	11221	11229#
TEST573	040622	11242	11250#
TEST574	040672	11264	11272#
TEST575	040732	11283	11291#
TEST576	040772	11304	11312#
TEST577	041042	11326	11334#
TEST60	003444	2016	2024#
TEST600	006062	2853	2861#
TEST600	041102	11355	11363#
TEST601	041160	11375	11383#
TEST602	041226	11407	11415#
TEST603	041304	11433	11441#
TEST604	041344	11464	11472#
TEST605	041422	11496	11504#
TEST606	041470	11516	11524#
TEST607	041536	11542	11550#
TEST61	006124	2887	2895#
TEST610	041604	11568	11576#
TEST611	041660	11594	11602#
TEST612	041734	11624	11632#
TEST613	041812	11660	11668#
TEST614	042100	11703	11711#
TEST615	042162	11732	11740#
TEST616	042232	11757	11765#
TEST617	042310	11773	11781#
TEST620	006356	2903	2911#
TEST620	042456	11803	11811#
TEST621	042456	11838	11846#
TEST622	042542	11867	11875#
TEST623	042628	11900	11908#
TEST624	042710	11934	11942#
TEST625	042740	11954	11962#
TEST626	043014	11980	11988#
TEST627	043050	12001	12009#
TEST63	006216	2921	2929#
TEST630	043104	12020	12027#
TEST631	043140	12039	12046#

10801#			
10831#			
10861#			
10891#			
10925#			
10955#			
10985#			
10985#			
11014#			
11036#			
11046#			
11054#			
11076#			
11098#			
11120#			
11142#			
11164#			
11186#			
11207#			
11229#			
11250#			
11272#			
11291#			
11312#			
11334#			
11334#			
11363#			
11383#			
11415#			
11423#			
11441#			
11447#			
11476#			
11506#			
11532#			
11550#			
11550#			
11576#			
11584#			
11602#			
11632#			
11632#			
11668#			
11672#			
11786#			
11811#			
11846#			
11851#			
11884#			
11912#			
11954#			
11988#			
11988#			
12008#			
12008#			
12027#			
12027#			
12046#			
11294#			
11316#			
11343#			
11365#			
11397#			
11423#			
11454#			
11476#			
11506#			
11532#			
11558#			
11576#			
11584#			
11614#			
11644#			
11989#			

2418#	2477#	2458#	2478#	2499#	2520#	2541#	2566#	2589#	2612#	2633#	2649#	2668#
2489#	2479#	2476#	2471#	2475#	2476#	2477#	2478#	2479#	2480#	2481#	2482#	2483#
2495#	2980#	2998#	3016#	3034#	3051#	3075#	3099#	3109#	3128#	3146#	3166#	3186#
2522#	3308#	3333#	3355#	3383#	3409#	3435#	3461#	3487#	3513#	3539#	3565#	3591#
2571#	3371#	3366#	3688#	3707#	3723#	3751#	3766#	3779#	3793#	3800#	3813#	3826#
2630#	3648#	3822#	3897#	3916#	3927#	3944#	3956#	3971#	3986#	4000#	4013#	4030#
2652#	3867#	4075#	4105#	4116#	4136#	4172#	4188#	4217#	4248#	4270#	4310#	4331#
2677#	4045#	4309#	4315#	4325#	4359#	4408#	4426#	4544#	4565#	4566#	4620#	4630#
2700#	4371#	4693#	4715#	4733#	4757#	4771#	4783#	4816#	4831#	4845#	4860#	4875#
2723#	4673#	4916#	4937#	4963#	4980#	5032#	5049#	5075#	5093#	5110#	5144#	5177#
2748#	4890#	5250#	5276#	5302#	5329#	5356#	5383#	5410#	5436#	5462#	5488#	5514#
2773#	5250#	5539#	5565#	5592#	5619#	5646#	5673#	5700#	5727#	5754#	5781#	5808#
2800#	5539#	5828#	6045#	6071#	6098#	6125#	6152#	6179#	6206#	6233#	6260#	6287#
2827#	5828#	6315#	6341#	6368#	6395#	6422#	6449#	6476#	6503#	6530#	6557#	6584#
2854#	6315#	6641#	6673#	6705#	6733#	6759#	6786#	6813#	6840#	6867#	6894#	6921#
2881#	7001#	7028#	7055#	7082#	7109#	7136#	7163#	7190#	7217#	7244#	7271#	7298#
2908#	7348#	7374#	7401#	7428#	7455#	7482#	7509#	7536#	7563#	7590#	7617#	7644#
2935#	7677#	7697#	7721#	7745#	7769#	7793#	7817#	7841#	7865#	7889#	7913#	7937#
2962#	8004#	8030#	8059#	8088#	8113#	8138#	8163#	8188#	8213#	8238#	8263#	8288#
2989#	8045#	8374#	8401#	8428#	8455#	8482#	8509#	8536#	8563#	8590#	8617#	8644#
3016#	8706#	8717#	8763#	8788#	8815#	8841#	8868#	8895#	8922#	8949#	8976#	9003#
3043#	9088#	9122#	9159#	9199#	9233#	9274#	9311#	9348#	9385#	9422#	9459#	9496#
3070#	9502#	9523#	9545#	9566#	9593#	9621#	9641#	9660#	9684#	9708#	9732#	9756#
3097#	9810#	9837#	9859#	9881#	9903#	9926#	9949#	9969#	9991#	10018#	10045#	10072#
3124#	10163#	10195#	10215#	10233#	10252#	10271#	10290#	10309#	10328#	10347#	10366#	10385#
3151#	10499#	10524#	10551#	10578#	10606#	10633#	10660#	10688#	10715#	10742#	10769#	10796#
3178#	10859#	10888#	10922#	10955#	10988#	11011#	11030#	11051#	11073#	11095#	11117#	11139#
3205#	11183#	11204#	11226#	11248#	11269#	11291#	11313#	11334#	11356#	11377#	11400#	11421#
3232#	11503#	11529#	11555#	11581#	11611#	11641#	11665#	11689#	11700#	11729#	11754#	11783#
3259#	11597#	11601#	11638#	11659#	11686#	11709#	11732#	11754#	11777#	11800#	11823#	11846#
3286#	12011#	12031#	12051#	12071#	12091#	12111#	12131#	12151#	12171#	12191#	12211#	12231#
3313#	12123#	12154#	12189#	12224#	12259#	12294#	12329#	12364#	12399#	12434#	12469#	12504#
3340#	12723#	12754#	12790#	12825#	12860#	12895#	12930#	12965#	12999#	13033#	13068#	13103#
3367#	13126#	13150#	13189#	13227#	13279#	13339#	13399#	13429#	13461#	13499#	13531#	13576#
3394#	13592#	13594#	13629#	13666#	13699#	13725#	13755#	13779#	13809#	13849#	13881#	13914#
3421#	13919#	13935#	13952#	13974#	13997#	14022#	14034#	14092#	14094#	14152#	14154#	14212#
3448#	14253#	14279#	14311#	14343#	14375#	14398#	14425#	14478#	14506#	14532#	14559#	14591#
3475#	14648#	14670#	14692#	14716#	14740#	14765#	14789#	14817#	14850#	14891#	14929#	14961#
3502#	14988#	14982#	15026#	15061#	15063#	15063#	15146#	15150#	15150#	15209#	15281#	15316#
3529#	15318#	15392#	15394#	15431#	15433#	15553#	15555#	15640#	15209#	15211#	15281#	15316#
3556#	16231#	16260#	16273#	16298#	16340#	16388#	16438#	16488#	16538#	16588#	16638#	16688#
3583#	16226#	16230#	16235#	16240#	16245#	16250#	16255#	16260#	16265#	16270#	16275#	16280#
3610#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3637#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3664#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3691#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3718#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3745#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3772#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3799#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3826#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3853#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3880#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3907#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3934#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3961#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
3988#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4015#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4042#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4069#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4096#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4123#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4150#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4177#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4204#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4231#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4258#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4285#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4312#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4339#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4366#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4393#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4420#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4447#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4474#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4501#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4528#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4555#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4582#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4609#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4636#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4663#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4690#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4717#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#	16973#	16978#	16983#	16988#
4744#	16943#	16938#	16943#	16948#	16953#	16958#	16963#	16968#				

Table with columns for macro names (COMMEN, ENDDERR, ENDDPAS, ENDDSC, ERROR) and a grid of numerical values. Includes an ESCAPE 1# 830# label at the bottom left.

Table with columns for macro names (GETPRI, GETSWR, MSG, MSGJ, MSGM1, MSGM10, MSGM11, MSGM3, MULT, NEWTST, NOINST, NOSCOP) and a grid of numerical values.

Table with columns for macro names (POP, PRENEW, PRESCO, PUSH, REPORT, SCOPE, SETPRI, SETTRA, SETUP, SKIP) and numerical values. The table is organized into two main sections, each with a list of macro names on the left and a grid of numbers on the right.

Table with columns for macro names (SLASH, SPACE, STARS) and numerical values. Similar to the first table, it consists of two main sections with macro names on the left and a grid of numbers on the right.

Table with 12 columns of macro names and their corresponding addresses. The first column lists macro names like 9660, 9662, 9684, etc. The subsequent columns list addresses for each macro.

Table with 12 columns of macro names and their corresponding addresses. The first column lists macro names like 12666, 12668, 12670, etc. The subsequent columns list addresses for each macro.

Table with 12 columns of macro names and their corresponding addresses. The first column lists macro names like 12978, 13018, 13056, etc. The subsequent columns list addresses for each macro.

Table with 12 columns of macro names and their corresponding addresses. The first column lists macro names like TRMTPP, TYPB, TYPD, etc. The subsequent columns list addresses for each macro.

7470	7497	7523	7549	7570	7597	7624	7648	7672	7697	7721	7746	7765	7785	7806
7527	7556	7582	7608	7634	7660	7686	7712	7738	7764	7790	7816	7842	7868	7894
7827	7854	7881	7907	7934	7961	7988	8015	8042	8069	8096	8123	8150	8177	8204
8240	8267	8294	8321	8348	8375	8402	8429	8456	8483	8510	8537	8564	8591	8618
8644	8671	8698	8725	8752	8779	8806	8833	8860	8887	8914	8941	8968	8995	9022
9049	9076	9103	9130	9157	9184	9211	9238	9265	9292	9319	9346	9373	9400	9427
9454	9481	9508	9535	9562	9589	9616	9643	9670	9697	9724	9751	9778	9805	9832
9859	9886	9913	9940	9967	9994	10021	10048	10075	10102	10129	10156	10183	10210	10237
10264	10291	10318	10345	10372	10399	10426	10453	10480	10507	10534	10561	10588	10615	10642
10669	10696	10723	10750	10777	10804	10831	10858	10885	10912	10939	10966	10993	11020	11047
11074	11101	11128	11155	11182	11209	11236	11263	11290	11317	11344	11371	11398	11425	11452
11479	11506	11533	11560	11587	11614	11641	11668	11695	11722	11749	11776	11803	11830	11857
11904	11931	11958	11985	12012	12039	12066	12093	12120	12147	12174	12201	12228	12255	12282
12309	12336	12363	12390	12417	12444	12471	12498	12525	12552	12579	12606	12633	12660	12687
12714	12741	12768	12795	12822	12849	12876	12903	12930	12957	12984	13011	13038	13065	13092
13119	13146	13173	13200	13227	13254	13281	13308	13335	13362	13389	13416	13443	13470	13497
13524	13551	13578	13605	13632	13659	13686	13713	13740	13767	13794	13821	13848	13875	13902
13929	13956	13983	14010	14037	14064	14091	14118	14145	14172	14199	14226	14253	14280	14307
14334	14361	14388	14415	14442	14469	14496	14523	14550	14577	14604	14631	14658	14685	14712
14739	14766	14793	14820	14847	14874	14901	14928	14955	14982	15009	15036	15063	15090	15117
15144	15171	15198	15225	15252	15279	15306	15333	15360	15387	15414	15441	15468	15495	15522
15549	15576	15603	15630	15657	15684	15711	15738	15765	15792	15819	15846	15873	15900	15927
15954	15981	16008	16035	16062	16089	16116	16143	16170	16197	16224	16251	16278	16305	16332
16359	16386	16413	16440	16467	16494	16521	16548	16575	16602	16629	16656	16683	16710	16737
16764	16791	16818	16845	16872	16899	16926	16953	16980	17007	17034	17061	17088	17115	17142
17169	17196	17223	17250	17277	17304	17331	17358	17385	17412	17439	17466	17493	17520	17547
17574	17601	17628	17655	17682	17709	17736	17763	17790	17817	17844	17871	17898	17925	17952
17979	18006	18033	18060	18087	18114	18141	18168	18195	18222	18249	18276	18303	18330	18357
18384	18411	18438	18465	18492	18519	18546	18573	18600	18627	18654	18681	18708	18735	18762
18789	18816	18843	18870	18897	18924	18951	18978	19005	19032	19059	19086	19113	19140	19167
19194	19221	19248	19275	19302	19329	19356	19383	19410	19437	19464	19491	19518	19545	19572
19599	19626	19653	19680	19707	19734	19761	19788	19815	19842	19869	19896	19923	19950	19977
19994	20021	20048	20075	20102	20129	20156	20183	20210	20237	20264	20291	20318	20345	20372
20399	20426	20453	20480	20507	20534	20561	20588	20615	20642	20669	20696	20723	20750	20777
20804	20831	20858	20885	20912	20939	20966	20993	21020	21047	21074	21101	21128	21155	21182
21209	21236	21263	21290	21317	21344	21371	21398	21425	21452	21479	21506	21533	21560	21587
21614	21641	21668	21695	21722	21749	21776	21803	21830	21857	21884	21911	21938	21965	21992
22019	22046	22073	22100	22127	22154	22181	22208	22235	22262	22289	22316	22343	22370	22397
22424	22451	22478	22505	22532	22559	22586	22613	22640	22667	22694	22721	22748	22775	22802
22829	22856	22883	22910	22937	22964	22991	23018	23045	23072	23099	23126	23153	23180	23207
23234	23261	23288	23315	23342	23369	23396	23423	23450	23477	23504	23531	23558	23585	23612
23639	23666	23693	23720	23747	23774	23801	23828	23855	23882	23909	23936	23963	23990	24017
24044	24071	24098	24125	24152	24179	24206	24233	24260	24287	24314	24341	24368	24395	24422
24449	24476	24503	24530	24557	24584	24611	24638	24665	24692	24719	24746	24773	24800	24827
24854	24881	24908	24935	24962	24989	25016	25043	25070	25097	25124	25151	25178	25205	25232
25259	25286	25313	25340	25367	25394	25421	25448	25475	25502	25529	25556	25583	25610	25637
25664	25691	25718	25745	25772	25799	25826	25853	25880	25907	25934	25961	25988	26015	26042
26069	26096	26123	26150	26177	26204	26231	26258	26285	26312	26339	26366	26393	26420	26447
26474	26501	26528	26555	26582	26609	26636	26663	26690	26717	26744	26771	26798	26825	26852
26879	26906	26933	26960	26987	27014	27041	27068	27095	27122	27149	27176	27203	27230	27257
27284	27311	27338	27365	27392	27419	27446	27473	27500	27527	27554	27581	27608	27635	27662
27689	27716	27743	27770	27797	27824	27851	27878	27905	27932	27959	27986	28013	28040	28067
28094	28121	28148	28175	28202	28229	28256	28283	28310	28337	28364	28391	28418	28445	28472
28499	28526	28553	28580	28607	28634	28661	28688	28715	28742	28769	28796	28823	28850	28877
28904	28931	28958	28985	29012	29039	29066	29093	29120	29147	29174	29201	29228	29255	29282
29309	29336	29363	29390	29417	29444	29471	29498	29525	29552	29579	29606	29633	29660	29687
29714	29741	29768	29795	29822	29849	29876	29903	29930	29957	29984	30011	30038	30065	30092
30119	30146	30173	30200	30227	30254	30281	30308	30335	30362	30389	30416	30443	30470	30497
30524	30551	30578	30605	30632	30659	30686	30713	30740	30767	30794	30821	30848	30875	30902
30929	30956	30983	31010	31037	31064	31091	31118	31145	31172	31199	31226	31253	31280	31307
31334	31361	31388	31415	31442	31469	31496	31523	31550	31577	31604	31631	31658	31685	31712
31739	31766	31793	31820	31847	31874	31901	31928	31955	31982	32009	32036	32063	32090	32117
32144	32171	32198	32225	32252	32279	32306	32333	32360	32387	32414	32441	32468	32495	32522
32549	32576	32603	32630	32657	32684	32711	32738	32765	32792	32819	32846	32873	32900	32927
32954	32981	33008	33035	33062	33089	33116	33143	33170	33197	33224	33251	33278	33305	33332
33359	33386	33413	33440	33467	33494	33521	33548	33575	33602	33629	33656	33683	33710	33737
33764	33791	33818	33845	33872	33899	33926	33953	33980	34007	34034	34061	34088	34115	34142
34169	34196	34223	34250	34277	34304	34331	34358	34385	34412	34439	34466	34493	34520	34547
34574	34601	34628	34655	34682	34709	34736	34763	34790	34817	34844	34871	34898	34925	34952
34979	35006	35033	35060	35087	35114	35141	35168	35195	35222	35249	35276	35303	35330	35357
35384	35411	35438	35465	35492	35519	35546	35573	35600	35627	35654	35681	35708	35735	35762
35789	35816	35843	35870	35897	35924	35951	35978	36005	36032	36059	36086	36113	36140	36167
36194	36221	36248	36275	36302	36329	36356	36383	36410	36437	36464	36491	36518	36545	36572
36599	36626	36653	36680	36707	36734	36761	36788	36815	36842	36869	36896	36923	36950	36977
36994	37021	37048	37075	37102	37129	37156	37183	37210	37237	37264	37291	37318	37345	37372
37399	37426	37453	37480	37507	37534	37561	37588	37615	37642	37669	37696	37723	37750	37777
37804	37831	37858	37885	37912	37939	37966	37993	38020	38047	38074	38101	38128	38155	38182
38209	38236	38263	38290	38317	38344	38371	38398	38425	38452	38479	38506	38533	38560	38587
38614	38641	38668	38695	38722	38749	38776	38803	38830	38857	38884	38911	38938	38965	38992
39019	39046	390												