

KD11-K

KD11-K BLT
CQKDAD0

AH-8092D MC
COPYRIGHT 77-79
FICHE 1 OF 2

NOV 1979
digital
MADE IN USA

The main body of the document consists of a large grid of 16 columns and 20 rows of small, illegible data tables or charts. Each cell in the grid appears to contain a small table or a set of data points, but the text is too small to be read. The grid is organized in a regular pattern, with each cell containing a similar structure of data, possibly representing a series of small plots or a data matrix. The overall appearance is that of a dense data table or a series of small charts arranged in a grid.

KD11-K

KD11 K BLT
CQKDAD0

AH 8092D MC
COPYRIGHT 77 79
FICHE 2 OF 2

NOV 1979
digital
MADE IN USA

This image shows a microfiche card with a grid of frames. The frames contain data, likely from a database or a list, organized in columns and rows. The data is printed in a small font and is difficult to read due to the low resolution of the scan. The card is labeled 'KD11-K' and 'CQKDAD0' at the top, and 'AH 8092D MC', 'COPYRIGHT 77 79', and 'FICHE 2 OF 2' on the right side. The 'digital' logo and 'MADE IN USA' are also visible in the top right corner.

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

.SBTTL DOCUMENT LISTING
.TITLE CQKDADO KD11-K BLT
.REM %

PRODUCT CODE: AC-8090D-MC
PRODUCT NAME: CQKDADO KD11-K BLT
PRODUCT DATE: JULY 1979
MAINTAINER: DIAGNOSTIC ENGINEERING

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, 1979 BY DIGITAL EQUIPMENT CORPORATION.

THE FOLLOWING ARE TRADEMARKS OF DIGITAL EQUIPMENT CORPORATION:

DIGITAL	PDP	UNIBUS	MASSBUS
DEC	DECUS	DECTAPE	

42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83

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
- 6.0 CHANGE HISTORY

84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139

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 RETRY 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"

140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195

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

196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251

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.

252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307

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 INDICATED 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:

308 S/B DST FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS THIS
309 COLUMN CONTAINS WHAT THE RESULT (DEST. OPERAND)
310 SHOULD HAVE BEEN (S/B).
311
312 WAS DST FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS THIS
313 COLUMN CONTAINS WHAT THE RESULT (DEST. OPERAND)
314 ACTUALLY WAS AFTER THE TEST.
315
316 DEST FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS THIS
317 COLUMN CONTAINS THE DESTINATION ADDRESS.
318
319 (IR) THIS IS A COPY OF THE TEST INSTRUCTION.
320 THIS WILL BE THE FIRST WORD IN THE CASE OF TWO
321 OR THREE WORD INSTRUCTIONS.
322
323 TEST INDICATES THE TEST NO. (IN OCTAL) THAT FAILED
324
325 (PC) INDICATES THE CONTENTS OF THE PROGRAM COUNTER AT THE
326 TIME OF THE ERROR CALL. THIS IS AN ADDRESS NORMALLY
327 USED TO LOCATE THE ERROR CALL STATEMENT IN
328 THE FAILING TEST.
329
330 (SP) INDICATES THE CONTENTS OF THE STACK POINTER (R6) AT
331 THE TIME OF THE ERROR. NOTE THAT THE ERROR CALL
332 WILL PUSH THE STACK TWICE. IN SP TESTS WHERE THE
333 SP MUST BE RESTORED PRIOR TO CALLING THE ERROR ROUTINE,
334 THEN THE ORIGINAL (UNRESTORED) SP IS TYPED, WITHOUT
335 ADDITIONAL PUSHES FROM THE ERROR CALL.
336
337 (PSW) INDICATES THE CONTENTS OF THE PROCESSOR STATUS WORD
338 AT THE TIME OF THE ERROR CALL
339
340 XXXXXX IS AN OCTAL NUMBER.
341
342 2.) ERROR 2 AND ERROR 4 ARE THE SAME AS FOR ERROR 1 ABOVE
343 EXCEPT THAT IN THIS CASE THE DESTINATION IS A GENERAL
344 REGISTER (WHICH DOES NOT HAVE A UNIBUS ADDRESS). THE OCTAL
345 NUMBER TYPED OUT IN THE "DEST" COLUMN SHOULD BE IGNORED.
346 THE TYPOUT WOULD LOOK AS FOLLOWS:
347
348 S/B DST WAS DST DEST (IR) TEST (PC) (SP) (PSW)
349 IS R3
350 XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX XXXXXX
351
352 3.) ERROR 5, ERROR 6, AND ERROR 7 ARE IDENTICAL TO ERROR 1
353 EXCEPT THAT ONLY THE LAST 5,6, OR 7 COLUMNS (RESPECTIVELY)
354 ARE PRINTED.
355
356 4.) ERROR 3 IS USED IN CASES WHERE THE STACK POINTER IS
357 SPECIFICALLY IN ERROR. THE COLUMNS HAVE THE SAME MEANING AS
358 DESCRIBED FOR ERROR 1 EXCEPT:
359
360 S/B SP IS WHAT THE STACK POINTER SHOULD HAVE BEEN (S/B)
361
362 WAS SP IS WHAT THE STACK POINTER ACTUALLY WAS
363

364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419

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 R0) 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 R0

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

420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475

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

476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531

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 INFORMATION:

532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587

[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:

CQKDA_ 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 N'TH 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 CHECK-POINT 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:

644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699

*
*
*
***>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.

6.0 ***** ECO HISTORY *****
HISTORY STARTS WITH REV. D0
CQKDADO - HARDWARE ECO #M7877-002 REQUIRES THAT BIT 9 IN LOG JAM REG. BE TESTED FOR SET CONDITION. TEST 763 MODIFIED.

%

.TITLE CQKDA-D KD11-K BASIC LOGIC TESTS
:*COPYRIGHT (C) 1977,1979
:*DIGITAL EQUIPMENT CORP.
:*MAYNARD, MASS. 01754

700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755

001000

000011
000012
000015
000200
177776

177774
177772
177570
177570

000000
000001
000002
000003
000004
000005
000006
000007
000006
000007

000000
000040
000100
000140
000200
000240
000300

```
.*
.*
.*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
.*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
.*
.SBTTL OPERATIONAL SWITCH SETTINGS
.*
.*      SWITCH      USE
.*      -----
.*      15          HALT ON ERROR
.*      14          LOOP ON TEST
.*      13          INHIBIT ERROR TYPEOUTS
.*      12          INHIBIT ID MESSAGE & UNEXPECTED TRAP MESSAGES
.*      11          INHIBIT ITERATIONS
.*      10          LOOP ON TEST IN SWR<8:0>
.*      9           LOOP ON ERROR
.*
.ENABLE ABS
.SBTTL BASIC DEFINITIONS

.*INITIAL ADDRESS OF THE STACK POINTER *** 1000 ***
STACK= 1000
.EQUIV EMT,ERROR      ;;BASIC DEFINITION OF ERROR CALL
.EQUIV IOT,SCOPE      ;;BASIC DEFINITION OF SCOPE CALL

.*MISCELLANEOUS DEFINITIONS
HT= 11                ;;CODE FOR HORIZONTAL TAB
LF= 12                ;;CODE FOR LINE FEED
CR= 15                ;;CODE FOR CARRIAGE RETURN
CRLF= 200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
PS= 177776           ;;PROCESSOR STATUS WORD
.EQUIV PS,PSW
STKLMT= 177774        ;;STACK LIMIT REGISTER
PIRQ= 177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
DSWR= 177570          ;;HARDWARE SWITCH REGISTER
DDISP= 177570         ;;HARDWARE DISPLAY REGISTER

.*GENERAL PURPOSE REGISTER DEFINITIONS
R0= %0                ;;GENERAL REGISTER
R1= %1                ;;GENERAL REGISTER
R2= %2                ;;GENERAL REGISTER
R3= %3                ;;GENERAL REGISTER
R4= %4                ;;GENERAL REGISTER
R5= %5                ;;GENERAL REGISTER
R6= %6                ;;GENERAL REGISTER
R7= %7                ;;GENERAL REGISTER
SP= %6                ;;STACK POINTER
PC= %7                ;;PROGRAM COUNTER

.*PRIORITY LEVEL DEFINITIONS
PR0= 0                ;;PRIORITY LEVEL 0
PR1= 40               ;;PRIORITY LEVEL 1
PR2= 100              ;;PRIORITY LEVEL 2
PR3= 140              ;;PRIORITY LEVEL 3
PR4= 200              ;;PRIORITY LEVEL 4
PR5= 240              ;;PRIORITY LEVEL 5
PR6= 300              ;;PRIORITY LEVEL 6
```


756 000340

PR7= 340 ;:PRIORITY LEVEL 7

757

;"SWITCH REGISTER" SWITCH DEFINITIONS

758

SW15= 100000

759 100000

SW14= 40000

760 040000

SW13= 20000

761 020000

SW12= 10000

762 010000

SW11= 4000

763 004000

SW10= 2000

764 002000

SW09= 1000

765 001000

SW08= 400

766 000400

SW07= 200

767 000200

SW06= 100

768 000100

SW05= 40

769 000040

SW04= 20

770 000020

SW03= 10

771 000010

SW02= 4

772 000004

SW01= 2

773 000002

SW00= 1

774 000001

.EQUIV SW09,SW9

775

.EQUIV SW08,SW8

776

.EQUIV SW07,SW7

777

.EQUIV SW06,SW6

778

.EQUIV SW05,SW5

779

.EQUIV SW04,SW4

780

.EQUIV SW03,SW3

781

.EQUIV SW02,SW2

782

.EQUIV SW01,SW1

783

.EQUIV SW00,SW0

784

785

;"DATA BIT DEFINITIONS (BIT00 TO BIT15)

786

BIT15= 100000

787 100000

BIT14= 40000

788 040000

BIT13= 20000

789 020000

BIT12= 10000

790 010000

BIT11= 4000

791 004000

BIT10= 2000

792 002000

BIT09= 1000

793 001000

BIT08= 400

794 000400

BIT07= 200

795 000200

BIT06= 100

796 000100

BIT05= 40

797 000040

BIT04= 20

798 000020

BIT03= 10

799 000010

BIT02= 4

800 000004

BIT01= 2

801 000002

BIT00= 1

802 000001

.EQUIV BIT09,BIT9

803

.EQUIV BIT08,BIT8

804

.EQUIV BIT07,BIT7

805

.EQUIV BIT06,BIT6

806

.EQUIV BIT05,BIT5

807

.EQUIV BIT04,BIT4

808

.EQUIV BIT03,BIT3

809

.EQUIV BIT02,BIT2

810

.EQUIV BIT01,BIT1

811

```
812 .EQUIV BIT00,BIT0
813
814 ;*BASIC "CPU" TRAP VECTOR ADDRESSES
815 000004 ERRVEC= 4 ;;TIME OUT AND OTHER ERRORS
816 000010 RESVEC= 10 ;;RESERVED AND ILLEGAL INSTRUCTIONS
817 000014 TBITVEC=14 ;;"T" BIT
818 000014 TRTVEC= 14 ;;TRACE TRAP
819 000014 BPTVEC= 14 ;;BREAKPOINT TRAP (BPT)
820 000020 IOTVEC= 20 ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
821 000024 PWRVEC= 24 ;;POWER FAIL
822 000030 EMTVEC= 30 ;;EMULATOR TRAP (EMT) **ERROR**
823 000034 TRAPVEC=34 ;;"TRAP" TRAP
824 000060 TKVEC= 60 ;;TTY KEYBOARD VECTOR
825 000064 TPVEC= 64 ;;TTY PRINTER VECTOR
826 000240 PIRQVEC=240 ;;PROGRAM INTERRUPT REQUEST VECTOR
827 .SBTTL TRAP CATCHER
828
829 .=0
830 ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
831 ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
832 ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
833 .=174
834 000174 000000 DISPREG: .WORD 0 ;;SOFTWARE DISPLAY REGISTER
835 000176 000000 SWREG: .WORD 0 ;;SOFTWARE SWITCH REGISTER
836 .SBTTL STARTING ADDRESS(ES)
837 000200 000137 001630 JMP @#START ;;JUMP TO STARTING ADDRESS OF PROGRAM
838 000700 .=700 ;PUT APT HEADER IN STACK AREA
839 .SBTTL APT PARAMETER BLOCK
840
841 ;*****
842 ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
843 ;*****
844 000700 .SX=. ;;SAVE CURRENT LOCATION
845 000024 .=24 ;;SET POWER FAIL TO POINT TO START OF PROGRAM
846 000024 200 ;;FOR APT START UP
847 000044 .=44 ;;POINT TO APT INDIRECT ADDRESS PNTR.
848 000044 $APTHDR ;;POINT TO APT HEADER BLOCK
849 000700 .=$X ;;RESET LOCATION COUNTER
850 ;*****
851 ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
852 ;INTERFACE SPEC.
853
854 000700 $APTHD:
855 000700 000000 $HIBTS: .WORD 0 ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
856 000702 001120 $MADR: .WORD $MAIL ;;ADDRESS OF APT MAILBOX (BITS 0-15)
857 000704 000000 $TSTM: .WORD ;;RUN TIM OF LONGEST TEST
858 000706 000000 $PASTM: .WORD ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
859 000710 000000 $UNITM: .WORD ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
860 000712 000014 .WORD $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
861 .SBTTL ACT11 HOOKS
862
863 ;*****
864 ;HOOKS REQUIRED BY ACT11
865 000714 $SVPC=. ;SAVE PC
866 000046 .=46
867 000046 $ENDAD ;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .SEOP
```

CQKDA-D KD11-K BASIC LOGIC TESTS
CQKDA.D.P11 11-SEP-79 13:31

MACY11 30A(1052) 11-SEP-79^{E 2} 13:53 PAGE 18
ACT11 HOOKS

SEQ 0017

868 000052
869 000052 000000
870 000714

. = 52
. WORD 0
. = \$SVPC

:::2)SET LOC.52 TO ZERO
::: RESTORE PC

871
872
873
874
875
876
877 001000 001000
878 001000 001000
879 001000 000000
880 001002 000
881 001003 000
882 001004 000000
883 001006 000000
884 001010 000000
885 001012 000000
886 001014 000
887 001015 001
888 001016 000000
889 001020 000000
890 001022 000000
891 001024 000000
892 001026 000000
893 001030 000000
894 001032 000000
895 001034 000
896 001035 000
897 001036 000000
898 001040 177570
899 001042 177570
900 001044 177560
901 001046 177562
902 001050 177564
903 001052 177566
904 001054 000
905 001055 002
906 001056 012
907 001057 000
908 001060 000000
909
910 001062 000000
911 001064 000000
912 001066 000000
913 001070 000000
914 001072 000000
915 001074 000000
916 001076 000000
917 001100 000000
918 001102 000000
919 001104 000000
920 001106 000000
921 001110 000000
922 001112 000000
923 001114 077
924 001115 015
925 001116 000012
926

```
.SBTTL COMMON TAGS

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

SCMTAG:      .=1000                ;; START OF COMMON TAGS

                .WORD 0                ;; CONTAINS THE TEST NUMBER
$TSTNM:      .BYTE 0                ;; CONTAINS ERROR FLAG
$ERFLG:      .BYTE 0                ;; CONTAINS SUBTEST ITERATION COUNT
$ICNT:       .WORD 0                ;; CONTAINS SCOPE LOOP ADDRESS
$LPADR:      .WORD 0                ;; CONTAINS SCOPE RETURN FOR ERRORS
$LPERR:      .WORD 0                ;; CONTAINS TOTAL ERRORS DETECTED
$ERTTL:      .WORD 0                ;; CONTAINS ITEM CONTROL BYTE
$ITEMB:      .BYTE 0                ;; CONTAINS MAX. ERRORS PER TEST
$ERMAX:      .BYTE 1                ;; CONTAINS PC OF LAST ERROR INSTRUCTION
$ERRPC:      .WORD 0                ;; CONTAINS ADDRESS OF 'GOOD' DATA
$GDADR:      .WORD 0                ;; CONTAINS ADDRESS OF 'BAD' DATA
$BDADR:      .WORD 0                ;; CONTAINS 'GOOD' DATA
$GDDAT:      .WORD 0                ;; CONTAINS 'BAD' DATA
$BDDAT:      .WORD 0                ;; RESERVED--NOT TO BE USED
                .WORD 0
                .WORD 0
$AUTOB:      .BYTE 0                ;; AUTOMATIC MODE INDICATOR
$INTAG:      .BYTE 0                ;; INTERRUPT MODE INDICATOR
                .WORD 0
SWR:         .WORD DSWR                ;; ADDRESS OF SWITCH REGISTER
DISPLAY:     .WORD DDISP                ;; ADDRESS OF DISPLAY REGISTER
$TKS:        177560                ;; TTY KBD STATUS
$TKB:        177562                ;; TTY KBD BUFFER
$TPS:        177564                ;; TTY PRINTER STATUS REG. ADDRESS
$TPB:        177566                ;; TTY PRINTER BUFFER 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'
$TPFLG:      .BYTE 0                ;; 'TERMINAL AVAILABLE' FLAG (BIT<07>=0=YES)
$REGAD:      .WORD 0                ;; CONTAINS THE ADDRESS FROM
                ;; WHICH ($REG0) WAS OBTAINED
$REG0:       .WORD 0                ;; CONTAINS (($REGAD)+0)
$REG1:       .WORD 0                ;; CONTAINS (($REGAD)+2)
$REG2:       .WORD 0                ;; CONTAINS (($REGAD)+4)
$REG3:       .WORD 0                ;; CONTAINS (($REGAD)+6)
$REG4:       .WORD 0                ;; CONTAINS (($REGAD)+10)
$REG5:       .WORD 0                ;; CONTAINS (($REGAD)+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
$ESCAPE:     0                ;; ESCAPE ON ERROR ADDRESS
$QUES:       .ASCII '??'                ;; QUESTION MARK
$CRLF:       .ASCII <15>                ;; CARRIAGE RETURN
$LF:         .ASCII <10>                ;; LINE FEED
*****
```

927
928
929
930
931 001120
932 001120 000000
933 001122 000000
934 001124 000000
935 001126 000000
936 001130 000000
937 001132 000000
938 001134 000000
939 001136 000000
940 001140
941 001140 000
942 001141 000
943 001142 000000
944 001144 000000
945 001146 000000
946
947
948
949
950
951
952 001150
953

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

954
 955
 956
 957
 958
 959
 960
 961
 962
 963
 964
 965
 966
 967
 968 001150
 969
 970
 971 001150 064640
 972 001152 000000
 973 001154 067764
 974 001156 000000
 975
 976 001160 064640
 977 001162 065061
 978 001164 067764
 979 001166 000000
 980
 981 001170 065013
 982 001172 000000
 983 001174 070006
 984 001176 000000
 985
 986 001200 064640
 987 001202 065072
 988 001204 067764
 989 001206 000000
 990
 991 001210 064666
 992 001212 000000
 993 001214 067772
 994 001216 000000
 995
 996 001220 064660
 997 001222 000000
 998 001224 067770
 999 001226 000000
 1000
 1001 001230 064650
 1002 001232 000000
 1003 001234 067766
 1004 001236 000000
 1005
 1006 001240 064724
 1007 001242 000000
 1008 001244 067764
 1009 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 $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
;*NOTE1: IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($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)
      0
      DT1      ;$REG4, $REG3, $REG2, $REG1,$REG0,$ERRPC,$REG5,$REG6
      0
;ITEM 2
      EM2      ;S/B DST WAS DST DEST (IR) TEST (PC) (SP) (PSW)
      DH2      ;
      DT2      ;$REG4, $REG3, $REG2, $REG1, $REG0,$ERRPC,$REG5,$REG6
      0
;ITEM 3
      EM3      ;S/B SP WAS SP (IR) TEST (PC) (PSW)
      0
      DT3      ;$REG4, $REG3, $REG1,$REG0,$ERRPC,$REG6
      0
;ITEM 4
      EM4      ;S/B DST WAS DST DEST (IR) TEST (PC) (SP) (PSW)
      DH4      ;
      DT4      ;$REG4, $REG3, $REG2, $REG1, $REG0, $ERRPC, $REG5, $REG6
      0
;ITEM 5
      EM5      ;(IR) TEST (PC) (SP) (PSW)
      0
      DT5      ;$REG1, $REG0, $ERRPC, $REG5, $REG6
      0
;ITEM 6
      EM6      ; DEST (IR) TEST (PC) (SP) (PSW)
      0
      DT6      ;$REG2, $REG1, $REG0, $ERRPC, $REG5, $REG6
      0
;ITEM 7
      EM7      ;WAS DST DEST (IR) TEST (PC) (SP) (PSW)
      0
      DT7      ;$REG3, $REG2, $REG1, $REG0, $ERRPC, $REG5, $REG6
      0
;ITEM 10
      EM10     ;S/B RES WAS RES DST OP STC OP TEST (PC) (SP) (PSW)
      0
      DT10     ;$REG4, $REG3, $REG2, $REG1, $REG0, $ERRPC, $REG5, $REG6
      0
```

1010			:ITEM 11	
1011	001250	065261	EM11	:TESTS SKIPPED
1012	001252	065277	DH11	: PC EXPCTD ACTUAL (TEST #'S)
1013	001254	070024	DT11	;\$ERRPC, \$TESTN,\$REGO
1014	001256	000000	0	
1015				
1016			:ITEM 12	
1017	001260	065335	EM12	:MED DID NOT ABORT IN USER MODE
1018	001262	067373	DH23	:PC
1019	001264	067664	DT23	;\$ERRPC
1020	001266	000000	0	
1021				
1022			:ITEM 13	
1023	001270	065374	EM13	:MED EXECUTED IN USER MODE
1024	001272	067373	DH23	:PC
1025	001274	067664	DT23	;\$ERRPC
1026	001276	000000	0	
1027				
1028			:ITEM 14	
1029	001300	065426	EM14	:MED CHANGED PSW
1030	001302	067373	DH23	:PC
1031	001304	067664	DT23	;\$ERRPC
1032	001306	000000	0	
1033				
1034			:ITEM 15	
1035	001310	065446	EM15	:MICROBREAK TRAP-TO-4 DID NOT OCCUR
1036	001312	067304	DH15	:ERRPC MEDCODE MICROBK REG.
1037	001314	067630	DT15	;\$ERRPC,\$TMP0,\$TMP1,0
1038	001316	067756	DF15	:0,0
1039				
1040			:ITEM 16	
1041				
1042	001320	067035	EM16	:CACHE DATA LOGGED INCORRECTLY
1043	001322	067607	DH44	:PC EXPCT RECVD
1044	001324	067670	DT24	;\$ERRPC,\$REG1,\$REG0,0
1045	001326	000000	0	
1046				
1047			:ITEM 17	
1048				
1049	001330	067006	EM45	:CACHE TAG LOGGED WRONG
1050	001332	067607	DH44	:PC EXPCT RECVD
1051	001334	067670	DT24	;\$ERRPC,\$REG0,\$REG1,0
1052	001336	000000	0	
1053				
1054			:ITEM 20	
1055				
1056	001340	065705	EM26	:PHYS. BA LOGGED WRONG
1057	001342	067607	DH44	:PC EXPCT RECVD
1058	001344	067670	DT24	;\$ERRPC,\$REG1,\$REG0,0
1059	001346	000000	0	
1060				
1061			:ITEM 21	
1062	001350	065535	EM21	:CSP CONSTANT WRONG
1063	001352	067336	DH17	:PC MEDCODE EXPECTD RECEIVD
1064	001354	067640	DT21	;\$ERRPC,\$TMP1,\$TMP2,\$REG0,0
1065	001356	067760	DF17	:0,0,0

1066			
1067			:ITEM 22
1068	001360	065560	EM22 ;BAD DATA READ BY A MED
1069	001362	067336	DH17 ;PC MEDCODE EXPECTD RECEIVD
1070	001364	067652	DT22 ;\$ERRPC,\$TMP1,\$TMP2,\$TMP3,0
1071	001366	067760	DF17 ;0,0,0
1072			
1073			:ITEM 23
1074	001370	065607	EM23 ;NO ODD PC TRAP
1075	001372	067373	DH23 ;PC
1076	001374	067664	DT23 ;\$ERRPC
1077	001376	000000	0
1078			
1079			:ITEM 24
1080			
1081	001400	065626	EM24 ;ODD ADR. BIT NOT SET IN CPU ERROR REGISTER OR LOG JAM
1082	001402	067400	DH24 ;PC CPUERR LOGJAM
1083	001404	067670	DT24 ;\$ERRPC,\$REG1,\$REG0
1084	001406	000000	0
1085			
1086			:ITEM 25
1087			
1088	001410	065511	EM17 ;LOG CUA LOGGED INCORRECT U-ADDR
1089	001412	067607	DH44 ;PC EXPCTD RECVD
1090	001414	067670	DT24 ;\$ERRPC \$REG1 \$REG0
1091	001416	000000	0
1092			
1093			:ITEM 26
1094			
1095	001420	065705	EM26 ;PHYS. BA LOGGED WRONG
1096	001422	067437	DH26 ;PC PA<17:16>-EXPCT-PA<15:0> PA<17:16>-RECVD-PA<15:0>
1097	001424	067706	DT26 ;\$ERRPC,\$REG1,\$REG2,\$REG0,\$REG3,0
1098	001426	000000	0
1099			
1100			:ITEM 27
1101			
1102	001430	065732	EM27 ;CACHE PARITY ERROR LOGGED IN BACK UP MODE
1103	001432	067521	DH27 ;PC LOGPBA LOGDATA LOGTAG
1104	001434	067722	DT27 ;\$ERRPC,\$REG3,\$REG1,\$REG2
1105	001436	000000	0
1106			
1107			:ITEM 30
1108			
1109	001440	066002	EM30 ;CACHE PARITY TRAPPED WHEN DISABLED
1110	001442	067373	DH23 ;PC
1111	001444	067664	DT23 ;\$ERRPC
1112	001446	000000	0
1113			
1114			:ITEM 31
1115			
1116	001450	066605	EM31 ;NO CACHE PARITY TRAP
1117	001452	067373	DH23 ;PC
1118	001454	067664	DT23 ;\$ERRPC
1119	001456	000000	0
1120			
1121			:ITEM 32

1122				
1123	001460	066114	EM32	:MEMORY ERROR REGISTERS INCORRECT
1124	001462	067554	DH32	:PC MEMERR
1125	001464	067700	DT25	:\$ERRPC,\$REG0
1126	001466	000000	0	
1127				
1128				:ITEM 33
1129				
1130	001470	066145	EM33	:TIMEOUT BIT NOT SET IN CPU ERROR REGISTER OR LOG JAM
1131	001472	067400	DH24	:PC CPUERR LOGJAM
1132	001474	067670	DT24	:\$ERRPC,\$REG1,\$REG0
1133	001476	000000	0	
1134				
1135				:ITEM 34
1136				
1137	001500	066223	EM34	:NO ILLEGAL INTERNAL ADDRESS TRAP
1138	001502	067373	DH23	:PC
1139	001504	067664	DT23	:\$ERRPC
1140	001506	000000	0	
1141				
1142				:ITEM 35
1143				
1144	001510	066260	EM35	:INTERNAL ADDRESS ERROR BIT NOT SET IN CPU ERROR REGISTER OR LOG JAM
1145	001512	067400	DH24	:PC CPUERR LOGJAM
1146	001514	067670	DT24	:\$ERRPC,\$REG1,\$REG0
1147	001516	000000	0	
1148				
1149				:ITEM 36
1150				
1151	001520	066346	EM36	:LAST INTERRUPT/TRAP VECTOR NOT LOGGED IN FLAG REGISTER
1152	001522	067423	DH25	:PC FLGREG
1153	001524	067700	DT25	:\$ERRPC,\$REG0
1154	001526	000000	0	
1155				
1156				:ITEM 37
1157				
1158	001530	066423	EM37	:LOG FIRST MODE DID NOT INHIBIT ERROR LOG AFTER FIRST ERROR
1159	001532	067400	DH24	:PC CPUERR LOGJAM
1160	001534	067670	DT24	:\$ERRPC,\$REG1,\$REG0
1161	001536	000000	0	
1162				
1163				:ITEM 40
1164				
1165	001540	066516	EM40	:ERROR LOG WAS NOT RE-ENABLED, ODD ADR BIT CLR IN CPUERR
1166	001542	067400	DH24	:PC CPUERR LOGJAM
1167	001544	067670	DT24	:\$ERRPC,\$REG1,\$REG0
1168	001546	000000	0	
1169				
1170				:ITEM 41
1171				
1172	001550	066045	EM41	:INSTRUCTION NOT ABORTED IN CACHE ABORT MODE
1173	001552	067373	DH23	:PC
1174	001554	067664	DT23	:\$ERRPC
1175	001556	000000	0	
1176				
1177				:ITEM 42

```
1178
1179 001560 066632 EM42 ;LO BYTE & TAG PARITY BITS NOT SET IN LOG SERVICE
1180 001562 067570 DH42 ;PC LOGSERVICE
1181 001564 067700 DT25 ;$ERRPC,$REG0,0
1182 001566 000000 0
1183
1184 ;ITEM 43
1185
1186 001570 066720 EM43 ;LO BYTE & TAG PARITY BITS NOT SET IN MEM ERR REGISTER
1187 001572 067554 DH32 ;PC MEMERR
1188 001574 067700 DT25 ;$ERRPC,$REG0
1189 001576 000000 0
1190
1191 ;ITEM 44
1192
1193 001600 067065 EMEIS1 ;EIS SET COND CODES WRONG
1194 001602 067222 DHEIS1 ; PSW REG-WAS-REG+1 REG-S/B-REG+1 PC TEST (IR)
1195 001604 067734 DTEIS1 ;$REGAD $REG2 $REG3 $REG1 $REG4 $ERRPC $REG0 $TMPO
1196 001606 000000 0
1197
1198 ;ITEM 45
1199
1200 001610 067116 EMEIS2 ;EIS GAVE WRONG RESULT
1201 001612 067222 DHEIS1 ; PSW REG-WAS-REG+1 REG-S/B-REG+1 PC TEST (IR)
1202 001614 067734 DTEIS1 ;$REGAD $REG2 $REG3 $REG1 $REG4 $ERRPC $REG0 $TMPO
1203 001616 000000 0
1204
1205 ;ITEM 46
1206
1207 001620 067144 EM46 ;AUTO-INCREMENT (DECREMENT) DID NOT OCCUR
1208 001622 067263 DH46 ; PC (IR) TEST
1209 001624 067746 DT46 ;$ERRPC $TMPO $REG0
1210 001626 000000 0
1211
1212 076600 MED = 076600
1213 140000 UM= 140000
1214 177770 UBREAK= 177770
1215 177744 MEMERR=177744
1216 177766 CPUERR=177766
1217 177746 CCR=177746
1218 000100 WWP=BIT6
1219 000001 DPTRP=BIT0
1220 000200 PABORT=BIT7
1221 000100 LO=BIT6
1222 000200 HI=BIT7
1223 000040 TAG=BIT5
1224
1225 .EQUIV SP,KSP
1226
1227
1228 ;* MED OPERATION CODE DEFINITIONS
1229
1230 000226 WCNSSW=226
1231 000022 RDWHAM1=022
1232 000222 WRWHAM1=222
1233 000144 RDFLAG=144
```

1234 000344
1235 000100
1236 000300
1237 000101
1238 000301
1239 000102
1240 000302
1241 000103
1242 000303
1243 000104
1244 000304
1245 000105
1246 000305
1247 000106
1248 000306
1249 000107
1250 000307
1251 000071
1252
1253
1254
1255
1256
1257 177560
1258 177562
1259 177564
1260 177566
1261 177546
1262

WRFLAG=344
RDLJAM=100
WRLJAM=300
RDLSERVICE=101
WRLSERVICE=301
RDLPBA=102
WRLPBA=302
RDLCUA=103
WRLCUA=303
RDLFGINT=104
WRLFGINT=304
RDLWHAMI=105
WRLWHAMI=305
RDLDATA=106
WRLDATA=306
RDLTAG=107
WRLTAG=307
SWB01=71

;MICRO ADDR. IN SWAB INST.

;ADDRESS ASSIGNMENTS FOR DL11 CONSOLE TERMINAL INTERFACE

RCSR=177560
RDBR = 177562
XCSR = 177564
XDBR = 177566
LKCSR= 177546

;RCVR. CONTROL / STATUS REG. ADDRESS
;RECEIVER DATA BUFFER REG. ADDR.
;TRANSMITTER CONTROL / STATUS REG. ADDR
;TRANSMIT DATA BUFFER REG. ADDR.
;LINE CLOCK ADDRESS

```
1263
1264
1265
1266
1267
1268
1269
1270
1271 001630
1272 001630 000401
1273
1274 001632 000000
1275
1276
1277
1278
1279
1280 001634 000402
1281
1282 001636 000403
1283
1284 001640 000000
1285
1286 001642 000775
1287
1288 001644 000000
1289
1290
1291
1292
1293
1294 001646 100403
1295 001650 001402
1296 001652 102401
1297 001654 103002
1298
1299 001656 000000
1300
1301 001660 000772
1302
1303
1304
1305
1306
1307 001662 000277
1308
1309 001664 100003
1310 001666 001002
1311 001670 102001
1312 001672 103402
1313
1314 001674 000000
1315
1316 001676 000771
```

```

; //////////////////////////////////////
; "BCPT" TESTS /
; //////////////////////////////////////
; *****
; .SBTTL BT001 "BR" TEST - POSITIVE OFFSET
; *****
START:
BT001: BR BT002 ;TEST THE BR FORWARD
E001: HALT ;BR FAILED TO LOAD PC PROPERLY
; *****
; .SBTTL BT002 "BR" TEST - NEGATIVE OFFSET
; *****
BT002: BR 1002 ;GO TO TEST INSTRUCTION
A002: BR BT003 ;GO TO NEXT TEST
EX002: HALT ;JUST IN CASE
1002: BR A002 ;TEST THE BR - NEG. OFFSET
E2002: HALT ;BR FAILED WITH NEG. OFFSET
; *****
; .SBTTL BT003 "BASIC COND. BR" TEST - FLAGS CLEARED
; *****
BT003: BMI E003 ;BR IF "N" SET
; BEQ E003 ;BR IF "Z" SET
; BVS E003 ;BR IF "V" SET
; BCC BT004 ;BR IF "C" CLEAR
E003: HALT ;ERROR - ONE OF THE ABOVE BR'S FAILED
; OR THE FLAGS FAILED TO CLEAR ON "START"
; BR BT003 ;LOCK ON HARD ERROR
; *****
; .SBTTL BT004 "SCC AND COND. BR'S" TEST - FLAGS SET
; *****
BT004: SCC ;MAKE N:C=1111
1004: BPL 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
```

CQKDA-D KD11-K BASIC LOGIC TESTS
CQKDAD.P11 11-SEP-79 13:31

MACY11 30A(1052) 11-SEP-79^{B 3} 13:53 PAGE 28
BT004 "SCC AND COND. BR'S" TEST - FLAGS SET

SEQ 0027

1317
1318
1319
1320
1321

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

1322 001700 000257
1323
1324 001702 100403
1325 001704 001402
1326 001706 102401
1327 001710 103002
1328
1329 001712 000000
1330
1331 001714 000771
1332
1333
1334
1335
1336
1337 001716 000257
1338
1339 001720 005000
1340
1341 001722 001402
1342
1343 001724 000000
1344 001726 000773
1345
1346
1347
1348
1349
1350 001730 005000
1351 001732 000257
1352
1353 001734 005700
1354
1355 001736 001402
1356
1357 001740 000000
1358
1359 001742 000772
1360
1361
1362
1363
1364
1365 001744 005000
1366 001746 000257
1367
1368 001750 005100
1369
1370 001752 100001
1371 001754 103402
1372
1373 001756 000000
1374 001760 000771
1375
1376
1377

```
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
```

1378
1379
1380 001762 005000
1381 001764 000257
1382
1383 001766 005100
1384 001770 005500
1385
1386 001772 001001
1387 001774 103402
1388
1389 001776 000000
1390 002000 000770
1391
1392
1393
1394
1395
1396 002002 005000
1397 002004 000257
1398
1399 002006 012700 177777
1400
1401 002012 005100
1402 002014 001402
1403
1404 002016 000000
1405 002020 000770
1406
1407
1408
1409
1410
1411 002022 005000
1412 002024 005100
1413 002026 000257
1414
1415 002030 012700 000000
1416
1417 002034 005100
1418 002036 005500
1419 002040 001402
1420
1421 002042 000000
1422 002044 000766
1423
1424
1425
1426
1427
1428 002046 012706 001000
1429 002052 012700 177776
1430 002056 000277
1431
1432 002060 005010
1433

```
; *****  
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/B = 000000  
  
        BNE 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,[R]=000000  
; *****  
BT012: CLR 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,[R]=177777  
; *****  
BT013: CLR R0 ;MAKE [R0] = 000000  
        COM R0 ;MAKE [R0] = 177777  
        CCC ;SCOPE SYNC  
  
I013: MOV #0,R0 ;TEST THE MOV - [R0] S/B = 000000  
  
        COM R0 ;MAKE [R0] = 177777, SET "C"  
        ADC R0 ;MAKE [R0] = 000000  
        BEQ BT014 ;BR IF "Z" GOT 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: CLR (R0) ;TEST THE CLR - IT SHOULD CLEAR PSW
```

```
1434 002062 001002          BNE      BT015          ;BR IF CLR MADE "Z" = 0 - IT SHOULD
1435
1436 002064 000000    E014:  HALT          ;ERROR- CLR FAILED TO CLEAR PSW
1437 002066 000767          BR      BT014          ;LOCK ON HARD ERROR
1438
1439 ; *****
1440 ; .SBTTL BT015 "CLR (R)+" TEST - [R] = 177776
1441 ; *****
1442
1443 002070 012700 177776    BT015:  MOV      #PSW,R0      ;R0 POINTS TO PSW
1444 002074 000277          SCC          ;MAKE [PSW] = 017
1445
1446 002076 005020    I015:  CLR      (R0)+        ;TEST THE CLR - IT SHOULD CLEAR PSW
1447
1448 002100 001002          BNE      A015          ;BR IF CLR MADE "Z" = 0 - IT SHOULD
1449
1450 002102 000000    E1015A: HALT          ;ERROR- CLR FAILED TO CLEAR PSW
1451 002104 000771          BR      BT015          ;LOCK ON HARD ERROR
1452
1453 002106 005700    A015:  TST      R0          ;AUTO INC SHOULD ZERO R0
1454
1455 002110 001402          BEQ      BT016          ;BR IF IT DID
1456
1457 002112 000000    E2015:  HALT          ;ERROR - AUTOINC. FAILED
1458 002114 000765          BR      BT015          ;LOCK ON HARD ERROR
1459
1460 ; *****
1461 ; .SBTTL BT016 "COM (R)" TEST - [R] = 177776
1462 ; *****
1463
1464 002116 012700 177776    BT016:  MOV      #PSW,R0      ;R0 POINTS TO PSW
1465 002122 000257          CCC          ;MAKE [PSW] = 000
1466
1467 002124 005110    I016:  COM      (R0)        ;TEST THE COM - [PSW] S/B = 357
1468
1469          BPL      E016          ;N:C=1111 ?
1470          BNE      E016
1471          BVC      E016
1472          BCS      BT017
1473
1474 002136 005010    E016:  CLR      (R0)        ;GO TO KERNEL MODE
1475 002140 000000          HALT          ;ERROR - COM FAILED TO MAKE [PSW] = 357
1476 002142 000765          BR      BT016          ;LOCK ON HARD ERROR
1477
1478 ; *****
1479 ; .SBTTL BT017 "COM (R0)+" TEST - [R0] = 177776
1480 ; *****
1481
1482 002144 012700 177776    BT017:  MOV      #PSW,R0      ;R0 POINTS TO PSW
1483 002150 005010          CLR      (R0)        ;MAKE [PSW] = 000
1484 002152 000257          CCC          ;SCOPE SYNC
1485
1486 002154 005120    I017:  COM      (R0)+        ;TEST THE COM - [PSW] S/B = 357
1487
1488          BPL      EA017          ;N:C = 1111 ?
1489          BNE      EA017
```



```
1546
1547 002276 005100 A021: COM R0 ;SHOULD MAKE [R0]=177777 AND SET 'C'
1548 002300 005500 ADC R0 ;SHOULD MAKE [R0]=000000
1549 002302 001402 BEQ BT022 ;BR IF 'Z' SET
1550
1551 002304 000000 E2021: HALT ;MOV FAILED TO ZERO R0
1552 002306 000763 BR BT021 ;LOCK ON HARD ERROR
1553
1554 ; *****
1555 ; .SBTTL BT022 'MOV #N,@#A' TEST WITH N=17,A=177776
1556 ; *****
1557
1558 002310 000257 BT022: CCC ;MAKE [PSW]=000
1559
1560 002312 012737 000017 177776 I022: MOV #17,@#PSW ;TEST THE MOV
1561
1562 002320 100003 BPL E022 ;N:C=1111
1563 002322 001002 BNE E022
1564 002324 102001 BVC E022
1565 002326 103402 BCS BT023
1566
1567 002330 000000 E022: HALT ;MOV FAILED TO LOAD PSW
1568 002332 000766 BR BT022 ;LOCK ON HARD ERROR
1569
1570 ; *****
1571 ; .SBTTL BT023 'MOV RA,(RB)+' TEST WITH [RA]=17,[RB]=177776
1572 ; *****
1573
1574 002334 012700 177776 BT023: MOV #PSW,R0 ;R0 POINTS TO PSW
1575 002340 012701 000017 MOV #17,R1 ;[SOURCE]=017
1576 002344 000257 CCC ;SCOPE SYNC - MAKE <N:C> = 0000
1577
1578 002346 010120 I023: MOV R1,(R0)+ ;TEST THE MOV
1579
1580 002350 100003 BPL EA023 ;N:C = 1111 ?
1581 002352 001002 BNE EA023
1582 002354 102001 BVC EA023
1583 002356 103402 BCS A023
1584
1585 002360 000000 EA023: HALT ;MOV FAILED TO LOAD PSW
1586 002362 000764 BR BT023 ;LOCK ON HARD ERROR
1587
1588 002364 005700 A023: TST R0 ;DID AUTO INC MAKE R0 GO TO 0?
1589 002366 001402 BEQ BT024 ;BR IF IT DID
1590
1591 002370 000000 E2023: HALT ;MOV FAILED TO AUTO INC. R0
1592 002372 000760 BR BT023 ;LOCK ON HARD ERROR
1593
1594 ; *****
1595 ; .SBTTL BT024 'CMP #N,@#A' TEST WITH N=(A)
1596 ; *****
1597
1598 002374 012700 177776 BT024: MOV #PSW,R0 ;R0 POINTS TO PSW
1599 002400 005010 CLR (R0) ;MAKE [PSW]=000
1600 002402 000273 273 ;MAKE N:C=1011
1601
```

```
1602 002404 022737 000013 177776 1024:  CMP      #13,@#PSW      ;TEST THE CMP
1603                                     BEQ      BT025      ;BR IF "Z" GOT SET
1604 002412 001402                                     E024:  HALT      ;CMP FAILED TO SET "Z"
1605                                     BR      BT024      ;LOCK ON HARD ERROR
1606 002414 000000
1607 002416 000766
1608
1609 ; *****
1610 ; .SBTTL BT025 "CMP #N,@#A" WITH N > (A)
1611 ; *****
1612
1613 002420 000257 BT025:  CCC          ;MAKE [PSW]=000
1614
1615 002422 022737 000017 177776 1025:  CMP      #17,@#PSW      ;TEST THE CMP
1616                                     BEQ      E025      ;BR IF "Z" GOT SET
1617 002430 001401                                     BR      BT026      ;GO TO NEXT TEST
1618 002432 000402
1619
1620 002434 000000 E025:  HALT      ;CMP FAILED TO CLEAR "Z"
1621 002436 000770 BR      BT025      ;LOCK ON HARD ERROR
1622
1623 ; *****
1624 ; .SBTTL BT026 "CMP #N,@#A" WITH N < (A)
1625 ; *****
1626 002440 000277 BT026:  SCC          ;MAKE [PSW]=017
1627
1628 002442 022737 000000 177776 1026:  CMP      #0,@#PSW      ;TEST THE CMP
1629                                     BEQ      E026      ;BR IF "Z" GOT SET
1630 002450 001401                                     BR      BT027      ;GO TO NEXT TEST
1631 002452 000402
1632
1633 002454 000000 E026:  HALT      ;CMP FAILED TO CLEAR "Z"
1634 002456 000770 BR      BT026      ;LOCK ON HARD ERROR
1635
1636 ; *****
1637 ; .SBTTL BT027 "CMP R,#N" TEST WITH [R]=N
1638 ; *****
1639
1640 002460 012700 177777 BT027:  MOV      #-1,R0      ;MAKE [R0]=177777
1641 002464 000257 CCC          ;N:C=0000
1642
1643 002466 020027 177777 1027:  CMP      R0,#-1      ;TEST THE CMP
1644                                     BEQ      BT030      ;BR IF CMP SET "Z"
1645 002472 001402
1646
1647 002474 000000 E027:  HALT      ;CMP FAILED
1648 002476 000770 BR      BT027      ;LOCK ON HARD ERROR
1649
1650 ; *****
1651 ; .SBTTL BT030 "CMP R,#N" TEST WITH [R] > N
1652 ; *****
1653
1654 002500 012700 000001 BT030:  MOV      #1,R0      ;MAKE [R0]=000001
1655 002504 000264 SEZ          ;SET THE "Z" BIT
1656
1657 002506 020027 177777 1030:  CMP      R0,#-1      ;TEST THE CMP
```

```

1658
1659 002512 001002          BNE      BT031          ;BR IF CMP CLEARED "Z"
1660
1661 002514 000000          E030:  HALT          ;CMP FAILED
1662 002516 000770          BR      BT030          ;LOCK ON HARD ERROR
1663 ; *****
1664 .SBTTL BT031 "CMP R,#N" TEST WITH [R] < N
1665 ; *****
1666
1667 002520 012700 000001          BT031: MOV      #1,R0          ;MAKE [R0] = 000001
1668 002524 000264          SEZ          ;SET THE "Z" BIT
1669
1670 002526 020027 000017          I031:  CMP      R0,#17          ;TEST THE CMP
1671
1672 002532 001002          BNE      BT032          ;BR IF CMP CLEARED "Z"
1673
1674 002534 000000          E031:  HALT          ;CMP FAILED TO SET "Z"
1675 002536 000770          BR      BT031          ;LOCK ON HARD ERROR
1676
1677 ; *****
1678 .SBTTL BT032 "CMP (RA)+,RB" TEST WITH [SOURCE]=[RB]
1679 ; *****
1680
1681 002540 012700 177776          BT032: MOV      #PSW,R0          ;R0 POINTS TO PSW
1682 002544 012737 000340 177776          MOV      #340,@#PSW          ;MAKE [PSW]=340
1683 002552 012701 000340          MOV      #340,R1          ;MAKE [DEST]=340
1684 002556 000257          CCC          ;N:C=0000
1685
1686 002560 022001          I032:  CMP      (R0)+,R1          ;TEST THE CMP
1687
1688 002562 001402          BEQ      A032          ;BR IF "Z" GOT SET
1689
1690 002564 000000          EA032: HALT          ;CMP FAILED TO ACCESS PSW
1691 002566 000764          BR      BT032          ;LOCK ON HARD ERROR
1692
1693 002570 005700          A032:  TST      R0          ;"Z" SHOULD SET
1694 002572 001402          BEQ      BT033          ;BR IF "Z" SET
1695
1696 002574 000000          E2032: HALT          ;CMP FAILED TO AUTO INC. R0
1697 002576 000760          BR      BT032          ;LOCK ON HARD ERROR
1698
1699 ; *****
1700 .SBTTL BT033 "CMP (RA)+,RB" TEST WITH [SOURCE]>[RB]
1701 ; *****
1702
1703 002600 012700 177776          BT033: MOV      #PSW,R0          ;R0 POINTS TO PSW
1704 002604 012737 000340 177776          MOV      #340,@#PSW          ;MAKE [PSW]=340
1705 002612 012701 000330          MOV      #330,R1          ;MAKE [DEST]=330
1706 002616 000264          SEZ          ;SET THE "Z" BIT
1707
1708 002620 022001          I033:  CMP      (R0)+,R1          ;TEST THE CMP
1709
1710 002622 001002          BNE      A033          ;BR IF "Z" GOT CLEARED
1711
1712 002624 000000          EA033: HALT          ;CMP FAILED TO ACCESS PSW
1713 002626 000764          BR      BT033          ;LOCK ON HARD ERROR

```

```
1714
1715 002630 005700 A033: TST R0 ;"Z" SHOULD SET
1716 002632 001402 BEQ BT034 ;BR IF "Z" SET
1717
1718 002634 000000 E2033: HALT ;CMP FAILED TO AUTO INC. R0
1719 002636 000760 BR BT033 ;LOCK ON HARD ERROR
1720 ; *****
1721 ; .SBTTL BT034 "CMP (RA)+,RB" TEST WITH [SOURCE]<[RB]
1722 ; *****
1723
1724 002640 012700 177776 BT034: MOV #PSW,R0 ;RO POINTS TO PSW
1725 002644 012737 000330 177776 MOV #330,@#PSW ;MAKE [PSW]=330
1726 002652 012701 000340 MOV #340,R1 ;MAKE [DEST]=340
1727 002656 000264 SEZ ;SET THE "Z" BIT
1728
1729 002660 022001 I034: CMP (R0)+,R1 ;TEST THE CMP
1730
1731 002662 001002 BNE A034 ;BR IF "Z" GOT CLEARED
1732
1733 002664 000000 EA034: HALT ;CMP FAILED TO ACCESS PSW
1734 002666 000764 BR BT034 ;LOCK ON HARD ERROR
1735
1736 002670 005700 A034: TST R0 ;"Z" SHOULD SET
1737 002672 001402 BEQ BT035 ;BR IF "Z" SET
1738
1739 002674 000000 E2034: HALT ;CMP FAILED TO AUTO INC. R0
1740 002676 000760 BR BT034 ;LOCK ON HARD ERROR
1741 ; *****
1742 ; .SBTTL BT035 "CMP RA,RB" TEST WITH [RA] = [RB]
1743 ; *****
1744
1745 002700 012700 125252 BT035: MOV #125252,R0 ;MAKE [R0] = 125252
1746 002704 010001 MOV R0,R1 ;MAKE [R1] = 125252
1747 002706 000257 CCC ;SCOPE SYNC
1748
1749 002710 020100 I035: CMP R1,R0 ;TEST THE CMP
1750
1751 002712 001402 BEQ BT036 ;BR IF "Z" GOT SET
1752
1753 002714 000000 E035: HALT ;ERROR - CMP FAILED TO SET "Z"
1754 002716 000770 BR BT035 ;LOCK ON HARD ERROR
1755 ; *****
1756 ; .SBTTL BT036 "CMP RA,RB" TEST WITH [RA] < [RB]
1757 ; *****
1758
1759 002720 012700 025252 BT036: MOV #25252,R0 ;MAKE [R0] = 25252
1760 002724 005001 CLR R1 ;MAKE [R1] = 000000
1761 002726 000264 SEZ ;SCOPE SYNC - SET "Z"
1762
1763 002730 020100 I036: CMP R1,R0 ;TEST THE CMP
1764
1765 002732 001002 BNE BT037 ;BR IF "Z" GOT CLEARED
1766
1767 002734 000000 E036: HALT ;ERROR - CMP FAILED TO SET "Z"
1768 002736 000770 BR BT036 ;LOCK ON HARD ERROR
1769 ; *****
```

1770
1771
1772
1773 002740 005000
1774 002742 012701 000017
1775 002746 000264
1776
1777 002750 020100
1778
1779 002752 001002
1780
1781 002754 000000
1782 002756 000770
1783
1784
1785
1786
1787
1788 002760 012700 177776
1789 002764 005010
1790 002766 005001
1791 002770 000277
1792
1793 002772 011001
1794
1795 002774 020127 000017
1796 003000 001402
1797
1798 003002 000000
1799 003004 000765
1800
1801
1802
1803
1804 003006 012700 177776
1805 003012 005010
1806 003014 005001
1807 003016 000277
1808
1809 003020 012001
1810
1811 003022 020127 000017
1812 003026 001402
1813
1814 003030 000000
1815 003032 000765
1816
1817 003034 005700
1818 003036 001402
1819
1820 003040 000000
1821 003042 000761
1822
1823
1824
1825

```
.SBTTL BT037 "CMP RA,RB" TEST WITH [RA] > [RB]
; *****
BT037: CLR    R0          ;MAKE [R0] = 000000
        MOV    #17,R1    ;MAKE [R1] = 000017
        SEZ          ;SCOPE SYNC - SET "Z"
1037:  CMP    R1,R0      ;TEST THE CMP
        BNE    BT040     ;BR IF "Z" GOT CLEARED
E037:  HALT          ;ERROR - CMP FAILED TO SET "Z"
        BR     BT037     ;LOCK ON HARD ERROR
; *****
.SBTTL BT040 "MOV (RA),RB" TEST WITH [SOURCE]=[RB]=17
; *****
BT040: MOV    #PSW,R0    ;R0 POINTS TO PSW
        CLR    (R0)      ;MAKE [PSW]=000
        CLR    R1        ;MAKE [R1]=000000
        SCC          ;MAKE N:C=1111
1040:  MOV    (R0),R1    ;TEST THE MOV
        CMP    R1,#17    ;DID R1 GET LOADED WITH 000017 ?
        BEQ    BT041     ;BR IF YES
E040:  HALT          ;MOV FAILED TO LOAD R1
        BR     BT040     ;LOCK ON HARD ERROR
; *****
.SBTTL BT041 "MOV (RA)+,RB" TEST WITH [SOURCE]=[RB]=17
; *****
BT041: MOV    #PSW,R0    ;R0 POINTS TO PSW
        CLR    (R0)      ;MAKE [PSW]=000
        CLR    R1        ;MAKE [R1]=000000
        SCC          ;MAKE N:C=1111
1041:  MOV    (R0)+,R1   ;TEST THE MOV
        CMP    R1,#17    ;DID R1 GET LOADED WITH 000017 ?
        BEQ    A041      ;BR IF YES
EA041: HALT          ;MOV FAILED TO LOAD R1
        BR     BT041     ;LOCK ON HARD ERROR
A041:  TST    R0          ;"Z" SHOULD SET
        BEQ    BT042     ;BR IF "Z" GOT SET
E2041: HALT          ;MOV FAILED TO AUTO INC. R0
        BR     BT041     ;LOCK ON HARD ERROR
; *****
.SBTTL BT042 "XOR RA,RB" TEST WITH [RA] = [RB] = 000000
; *****
```

1826
1827 003044 005000
1828 003046 005001
1829 003050 000257
1830
1831 003052 074100
1832
1833 003054 005700
1834 003056 001402
1835
1836 003060 000000
1837 003062 000770
1838
1839
1840
1841
1842
1843 003064 005000
1844 003066 005100
1845 003070 010001
1846 003072 000257
1847
1848 003074 074100
1849
1850 003076 005700
1851 003100 001402
1852
1853 003102 000000
1854 003104 000767
1855
1856
1857
1858
1859
1860 003106 012701 125252
1861 003112 012700 052525
1862 003116 000257
1863
1864 003120 074100
1865
1866 003122 020027 177777
1867 003126 001402
1868
1869 003130 000000
1870 003132 000400
1871
1872
1873
1874
1875 003134 012700 125252
1876 003140 012701 052525
1877 003144 000257
1878
1879 003146 074100
1880
1881 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 ?
BEQ BT043 ;BR IF YES
E042: HALT ;XOR FAILED
BR BT042

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

BT043: CLR R0 ;MAKE [R0] = 177777
COM R0
MOV R0,R1 ;MAKE [R1] = 177777
CCC ;SCOPE SYNC
I043: XOR R1,R0 ;TEST THE XOR
TST R0 ;RESULT = 000000 ?
BEQ BT044 ;BR IF YES
E043: HALT ;XOR FAILED
BR BT043 ;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 ;XOR FAILED
BR BT045 ;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 ?

```
1882 003154 001402          BEQ      BT046          ;BR IF YES
1883
1884 003156 000000      E045:  HALT          ;XOR FAILED
1885 003160 000765          BR      BT045          ;LOCK ON HARD ERROR
1886
1887 ; *****
1888 ; .SBTTL BT046 GPR ADDRESS INTERRACTION TEST
1889 ; *****
1890
1891 003162 012700 125252      BT046:  MOV      #125252,R0      ;[R0] = 125252
1892 003166 010001          MOV      R0,R1
1893 003170 005101          COM      R1              ;[R1] = 052525
1894 003172 010102          MOV      R1,R2
1895 003174 005102          COM      R2              ;[R2] = 125252
1896 003176 010203          MOV      R2,R3
1897 003200 005103          COM      R3              ;[R3] = 052525
1898 003202 010304          MOV      R3,R4
1899 003204 005104          COM      R4              ;[R4] = 125252
1900 003206 010405          MOV      R4,R5
1901 003210 005105          COM      R5              ;[R5] = 052525
1902
1903 003212 074100      1046:  XOR      R1,R0          ;[R0] S/B = 177777
1904 003214 074200          XOR      R2,R0          ;[R0] S/B = 125252
1905 003216 074300          XOR      R3,R0          ;[R0] S/B = 177777
1906 003220 074400          XOR      R4,R0          ;[R0] S/B = 125252
1907 003222 074500          XOR      R5,R0          ;[R0] S/B = 177777
1908 003224 005100          COM      R0              ;[R0] S/B = 000000
1909
1910 003226 001402          BEQ      A046          ;BR IF [R0] WAS 000000
1911
1912 003230 000000      EA046:  HALT          ;GPR ADDRESSING PROBLEM
1913 003232 000753          BR      BT046          ;LOCK ON HARD ERROR
1914
1915 003234 020627 001000      A046:  CMP      SP,#STACK      ;DID R6 GET DISTURBED
1916 003240 001402          BEQ      BASIC          ;BR IF NOT
1917
1918 003242 000000      E2046:  HALT          ;R6 ADDRESS PROBLEM
1919 003244 000746          BR      BT046          ;LOCK ON HARD ERROR
```


1920
1921
1922
1923
1924 003246 005037 063254
1925 003252 005037 001012
1926 003256 005037 001126
1927 003262 012701 063236
1928 003266 005021
1929 003270 020127 063254
1930 003274 001374
1931 003276 012706 001000
1932 003302 012737 004030 177770
1933 003310 012737 177777 001074
1934
1935
1936
1937
1938
1939 003316
1940 003316 012700 000000
1941 003322 000257
1942
1943 003324
1944 003324 001002
1945
1946 003326 000000
1947 003330 000774
1948
1949
1950
1951
1952 003332
1953 003332 012700 000001
1954 003336 000264
1955
1956 003340 001001
1957
1958 003342 000402
1959
1960 003344 000000
1961 003346 000773
1962
1963
1964
1965
1966 003350
1967 003350 012700 000002
1968 003354 000264
1969
1970 003356
1971 003356 001402
1972
1973 003360 000000
1974 003362 000774
1975

```
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
: / / / / / / BASIC INSTRUCTION TESTS / / / / / /  
XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX  
BASIC: CLR @#ONCE ; SIGNAL PROGRAM HEADER TO BE PRINTED  
CLR @#SERTTL ; CLEAR ERROR COUNT FIRST TIME THROUGH  
CLR @#SPASS ; 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$: MOV #0,R0 ;:LOAD R0 WITH TEST NUMBER  
CCC ;MAKE Z=0  
2$: BNE TST1 ;:TEST THE BNE - IT SHOULD BR  
3$: HALT ;BNE FAILED TO LOAD PC  
BR 1$ ;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$: HALT ;BNE BRANCHED WITH Z=1  
BR 1$ ;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$: HALT ;BEQ FAILED TO LOAD THE PC  
BR 1$ ;LOCK ON HARD ERROR
```

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

1976
 1977
 1978 003364
 1979 003364 012700 000003
 1980 003370 000257
 1981
 1982 003372 001401
 1983
 1984 003374 000402
 1985
 1986 003376 000000
 1987 003400 000773
 1988
 1989
 1990
 1991
 1992 003402
 1993 003402 012700 000004
 1994 003406 005037 177776
 1995 003412 000270
 1996
 1997 003414 100001
 1998
 1999 003416 000402
 2000
 2001 003420 000000
 2002 003422 000771
 2003
 2004
 2005
 2006
 2007 003424
 2008 003424 012700 000005
 2009 003430 005037 177776
 2010 003434 000257
 2011
 2012 003436
 2013 003436 100002
 2014
 2015 003440 000000
 2016 003442 000772
 2017
 2018
 2019
 2020
 2021 003444
 2022 003444 012700 000006
 2023 003450 012705 177776
 2024 003454 005015
 2025 003456 005003
 2026 003460 000277
 2027
 2028 003462 011503
 2029
 2030 003464 020327 000017
 2031 003470 001402

```

;*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$:  HALT                    ;BEQ BRANCHED WITH Z=0
    BR     1$           ;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 SHOULDN'T BR
    BR     TST5         ;;GO TO NEXT TEST
3$:  HALT                    ;BPL BRANCHED WITH N=1
    BR     1$           ;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$:  HALT                    ;BPL FAILED TO LOAD THE PC
    BR     1$           ;LOCK ON HARD ERROR
:*****
;*TEST 6      BASIC 'MOV (RA),RB' TEST - (RA)=177776
:*****
TST6:
    MOV    #6,R0      ;;LOAD R0 WITH TEST NUMBER
    MOV    #PSW,R5    ;SOURCE ADDR = 177776
1$:  CLR    (R5)       ;MAKE [PSW]=000
    CLR    R3         ;[DEST] = 000000
    SCC                      ;MAKE [PSW]=017
2$:  MOV    (R5),R3    ;TEST THE MOV
    CMP    R3,#17     ;CORRECT RESULT ?
    BEQ    TST7       ;;BR IF YES
  
```

```
2032
2033 003472 000000
2034 003474 000767
2035
2036
2037
2038 003476
2039 003476 012700 000007
2040 003502 012702 063312
2041 003506 012704 125252
2042 003512 012737 125252 063312 1$:
2043 003520 000257
2044
2045 003522 020412 2$:
2046
2047 003524 001402
2048
2049 003526 000000 3$:
2050 003530 000770
2051
2052
2053
2054 003532
2055 003532 012700 000010
2056 003536 012702 063312
2057 003542 012704 000001
2058 003546 005037 063312 1$:
2059 003552 000264
2060
2061 003554 020412 2$:
2062
2063 003556 001002
2064
2065 003560 000000 3$:
2066 003562 000771
2067
2068
2069
2070
2071 003564
2072 003564 012700 000011
2073 003570 012704 125252
2074 003574 010403
2075 003576 000257
2076
2077 003600 022703 125252 2$:
2078
2079 003604 001402
2080
2081 003606 000000 3$:
2082 003610 000771
2083
2084 003612 020403 4$:
2085 003614 001402
2086
2087 003616 000000 5$:

3$: HALT ;ERROR-MOV FAILED
BR 1$ ;LOCK ON HARD ERROR
*****
*TEST 7 BASIC "CMP RA,(RB)" TEST - [RA] = [DEST]
*****
TST7:
MOV #7,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,R4 ;RESULT S / B = 125252
1$: MOV #125252,@MBUF0 ;MAKE [DEST] = 125252
CCC ;MAKE N:C=0000

2$: CMP R4,(R2) ;TEST THE CMP
BEQ TST10 ;;BR IF "Z" GOT SET

3$: HALT ;ERROR - CMP FAILED TO SET "Z"
BR 1$ ;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 #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #1,R4 ;RESULT S / B = 000001
1$: CLR @MBUF0 ;MAKE [DEST] = 000000
SEZ ;MAKE N:C=0100

2$: CMP R4,(R2) ;TEST THE CMP
BNE TST11 ;;BR IF "Z" GOT CLEARED

3$: HALT ;ERROR - CMP FAILED TO CLR "Z"
BR 1$ ;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
1$: MOV R4,R3 ;[DEST] = 125252
CCC ;SCOPE SYNC

2$: CMP #125252,R3 ;TEST THE CMP
BEQ 4$ ;BR IF N = [R]

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

4$: CMP R4,R3 ;DID CMP ALTER [DEST]?
BEQ TST12 ;;BR IF NO

5$: HALT ;CMP DELIVERED A RESULT
```

```
2088 003620 000765          BR      1$          ;LOCK ON HARD ERROR
2089
2090
2091          ;*****
2091          ;*TEST 12      BASIC "CMP #N,R" TEST - N NOT EQUAL TO [R]
2092          ;*****
2093 003622
2094 003622 012700 000012      TST12:
2095 003626 005004          MOV     #12,R0          ;;LOAD R0 WITH TEST NUMBER
2096 003630 010403      1$:  CLR     R4          ;RESULT S / B = 000000
2097 003632 000264          MOV     R4,R3          ;[DEST] = 125252
2098          SEZ          ;SCOPE SYNC
2099 003634 022703 000001      2$:  CMP     #1,R3          ;TEST THE CMP
2100
2101 003640 001002          BNE     4$          ;BR IF N NOT EQUAL TO [R]
2102
2103 003642 000000      3$:  HALT          ;CMP FAILED
2104 003644 000771          BR      1$          ;LOCK ON HARD ERROR
2105
2106 003646 020403      4$:  CMP     R4,R3          ;DID CMP ALTER [DEST]?
2107 003650 001402          BEQ     TST13        ;;BR IF NO
2108
2109 003652 000000      5$:  HALT          ;CMP DELIVERED A RESULT
2110 003654 000765          BR      1$          ;LOCK ON HARD ERROR
2111
2112          ;*****
2113          ;*TEST 13      BASIC "MOV RA,(RB)" TEST
2114          ;*****
2115 003656
2116 003656 012700 000013      TST13:
2117 003662 012702 063312      MOV     #13,R0          ;;LOAD R0 WITH TEST NUMBER
2118 003666 012704 177777      MOV     #MBUFO,R2       ;DEST ADDR=MBUFO
2119 003672 005012      1$:  MOV     #-1,R4          ;RESULT S / B = 177777
2120 003674 000257          CLR     (R2)          ;MAKE [DEST] = 000000
2121          CCC          ;SCOPE SYNC - N:C=0000
2122 003676 010412      2$:  MOV     R4,(R2)        ;TEST THE MOV
2123
2124 003700 020412          CMP     R4,(R2)        ;RESULT CORRECT ?
2125 003702 001402          BEQ     TST14        ;;BR IF YES
2126
2127 003704 000000      3$:  HALT          ;ERROR - MOV FAILED
2128 003706 000771          BR      1$          ;LOCK ON HARD ERROR
2129
2130          ;*****
2131          ;*TEST 14      BASIC "MOV #N,(R)" TEST
2132          ;*****
2133 003710
2134 003710 012700 000014      TST14:
2135 003714 012702 063312      MOV     #14,R0          ;;LOAD R0 WITH TEST NUMBER
2136 003720 012704 177777      MOV     #MBUFO,R2       ;DEST ADDR = MBUFO
2137 003724 005012      1$:  MOV     #-1,R4          ;RESULT S / B = 177777
2138 003726 000257          CLR     (R2)          ;MAKE [DEST] = 000000
2139          CCC          ;SCOPE SYNC
2140 003730 012712 177777      2$:  MOV     #-1,(R2)      ;TEST THE MOV
2141
2142 003734 020412          CMP     R4,(R2)        ;RESULT OK ?
2143 003736 001402          BEQ     TST15        ;;BR IF YES
```

```
2144
2145 003740 000000
2146 003742 000770
2147
2148
2149
2150
2151 003744
2152 003744 012700 000015
2153 003750 012704 177401
2154 003754 012702 063316
2155 003760 012705 063312
2156 003764 012712 177777
2157 003770 000257
2158
2159 003772 112765 000001 000004 2$: MOVB #1,4(R5) ;TEST THE MOVB
2160
2161 004000 020412
2162 004002 001402
2163
2164 004004 000000 3$: HALT ;MOVB DELIVERED WRONG RESULT
2165 004006 000766 BR 1$ ;LOCK ON HARD ERROR
2166
2167
2168
2169
2170
2171 004010
2172 004014 012704 000777
2173 004020 012702 063316
2174 004024 012705 063312
2175 004030 012712 177777
2176 004034 000257
2177
2178 004036 112765 000001 000005 2$: MOVB #1,5(R5) ;TEST THE MOVB
2179
2180 004044 020412
2181 004046 001402
2182
2183 004050 000000 3$: HALT ;MOVB DELIVERED WRONG RESULT
2184 004052 000766 BR 1$ ;LOCK ON HARD ERROR
2185
2186
2187
2188
2189 004054
2190 004054 012700 000017
2191 004060 012702 063312
2192 004064 012704 000377
2193 004070 010412
2194 004072 000257
2195
2196 004074 005737 063312 2$: TST @#MBUFO ;TEST THE TST
2197
2198 004100 001401
2199 004102 100002 BPL TST20 ;BR IF "Z" SET - IT SHOULDN'T BE
;BR IF "N" CLEAR - IT SHOULD BE
```

2200
2201 004104 000000
2202 004106 000770
2203
2204
2205
2206 004110
2207 004110 012700 000020
2208 004114 012702 063312
2209 004120 012704 100000
2210 004124 010412
2211 004126 000257
2212
2213 004130 005737 063312
2214
2215 004134 001401
2216 004136 100402
2217
2218 004140 000000
2219 004142 000770
2220 004144 020412
2221 004146 001402
2222
2223 004150 000000
2224 004152 000764
2225
2226
2227
2228
2229 004154
2230 004154 012700 000021
2231 004160 012702 063312
2232 004164 005004
2233 004166 005012
2234 004170 000257
2235
2236 004172 005737 063312
2237
2238 004176 001402
2239
2240 004200 000000
2241 004202 000771
2242
2243 004204 020412
2244 004206 001402
2245
2246 004210 000000
2247 004212 000765
2248
2249
2250
2251
2252 004214
2253 004214 012700 000022
2254 004220 012702 063312
2255 004224 012704 040000

```
3$: HALT ;TST FAILED TO ALTER CODES PROPERLY
BR 1$ ;LOCK ON HARD ERROR
*****
*TEST 20 BASIC 'TST @#A' TEST WITH [A] LT 0
*****
TST20:
MOV #20,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #100000,R4 ;:MAKE S / B = 100000
1$: MOV R4,(R2) ;:MAKE [DEST] = 100000
CCC ;:SCOPE SYNC

2$: TST @#MBUFO ;:TEST THE TST

BEQ 3$ ;:BR IF 'Z' SET - IT SHOULDN'T BE
BMI 4$ ;:BR IF 'N' SET - IT SHOULD BE

3$: HALT ;TST FAILED TO ALTER CODES PROPERLY
BR 1$ ;:LOCK ON HARD ERROR
4$: CMP R4,(R2) ;:DID TST DISTURB [DEST] ?
BEQ TST21 ;:BR IF NOT

5$: HALT ;TST DELIVERED A RESULT
BR 1$ ;:LOCK ON HARD ERROR
*****
*TEST 21 BASIC 'TST @#A' WITH [A] = 0
*****
TST21:
MOV #21,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
CLR R4 ;:RESULT S / B = 0 (IT SHOULDN'T CHANGE
1$: CLR (R2) ;:[DEST] = 0
CCC ;:SCOPE SYNC - Z=0

2$: TST @#MBUFO ;:TEST THE TST

BEQ 4$ ;:BR IF TST SET 'Z'

3$: HALT ;TST FAILED TO SET 'Z'
BR 1$ ;:LOCK ON HARD ERROR

4$: CMP R4,(R2) ;:[DEST] STILL = 000000
BEQ TST22 ;:BR IF YES

5$: HALT ;TST ALTERED THE [DEST]
BR 1$ ;:LOCK ON HARD ERROR
*****
*TEST 22 BASIC 'BIT #N,@#A' WITH BIT SET IN 'A'
*****
TST22:
MOV #22,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #40000,R4 ;:RESULT S / B = 40000
```

```

2256 004230 010412 1$: MOV R4,(R2) ;MAKE [DEST] = 40000
2257 004232 000277 SCC ;SCOPE SYNC - Z=1
2258
2259 004234 032737 040000 063312 2$: BIT #40000,@#MBUF0 ;TEST THE BIT
2260
2261 004242 001002 BNE TST23 ;;BR IF Z=0 - IT SHOULD BE
2262
2263 004244 000000 3$: HALT ;BIT FAILED TO CLEAR "Z"
2264 004246 000770 BR 1$ ;LOCK ON HARD ERROR
2265

```

```

*****
;*TEST 23 BASIC "BIT #N,@#A" WITH BIT CLEAR IN "A"
*****

```

```

2269 004250 TST23:
2270 004250 012700 000023 MOV #23,R0 ;;LOAD R0 WITH TEST NUMBER
2271 004254 012702 063312 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
2272 004260 005012 1$: CLR (R2) ;MAKE [DEST] = 000000
2273 004262 000257 CCC ;SCOPE SYNC - Z=0
2274
2275 004264 032737 040000 063312 2$: BIT #40000,@#MBUF0 ;TEST THE BIT
2276
2277 004272 001402 BEQ 4$ ;BR IF Z=1 - IT SHOULD BE
2278
2279 004274 000000 3$: HALT ;BIT FAILED TO SET "Z"
2280 004276 000770 BR 1$ ;LOCK ON HARD ERROR
2281
2282 004300 005712 4$: TST (R2) ;DID BIT DELIVER A RESULT
2283 004302 001402 BEQ TST24 ;;BR IF NOT
2284
2285 004304 000000 5$: HALT ;BIT DISTURBED THE [DEST]
2286 004306 000764 BR 1$ ;LOCK ON HARD ERROR
2287

```

```

*****
;*TEST 24 BASIC "TST (R)+ " TEST
*****

```

```

2288
2289
2290 TST24:
2291 004310
2292 004310 012700 000024 MOV #24,R0 ;;LOAD R0 WITH TEST NUMBER
2293 .SBTTL USER CONTROLLED BREAKPOINT -- BIT0
2294 004314 032737 000001 063234 BIT #BIT0,@#BPTLOC ;BREAKPOINT HALT SET ??
2295 004322 001401 BEQ .+4 ;BR IF NOT
2296 004324 000000 HALT ;BREAK - DEPRESS CONTINUE TO RESTART
2297 004326 012702 063312 MOV #MBUF0,R2 ;INITIAL DEST ADDR = MBUF0
2298 004332 005012 1$: CLR (R2) ;MAKE [DEST] = 000000
2299 004334 000257 CCC ;SCOPE SYNC
2300
2301 004336 005722 2$: TST (R2)+ ;TEST THE TST
2302
2303 004340 001402 BEQ 4$ ;BR IF "Z" SET - IT SHOULD BE
2304
2305 004342 000000 3$: HALT ;TST FAILED TO SET "Z"
2306 004344 000772 BR 1$ ;LOCK ON HARD ERROR
2307
2308 004346 022702 063314 4$: CMP #MBUF0+2,R2 ;DID REG. GET AUTO-INCREMENTED ?
2309 004352 001402 BEQ TST25 ;;BR IF YES
2310
2311 004354 000000 5$: HALT ;TST FAILED TO UPDATE REGISTER

```

2312 004356 000765
 2313
 2314
 2315
 2316
 2317 004360
 2318 004360 012700 000025
 2319 004364 012702 063330
 2320 004370 012704 000377
 2321 004374 012705 063332
 2322 004400 000270
 2323
 2324 004402 005745
 2325
 2326 004404 100002
 2327
 2328 004406 000000
 2329 004410 000771
 2330
 2331 004412 020502
 2332 004414 001402
 2333
 2334 004416 000000
 2335 004420 000765
 2336
 2337 004422 020412
 2338 004424 001403
 2339
 2340 004426 000000
 2341 004430 010412
 2342 004432 000760
 2343
 2344
 2345
 2346
 2347 004434
 2348 004434 012700 000026
 2349 004440 012702 063312
 2350 004444 005004
 2351 004446 005104
 2352 004450 005012
 2353 004452 000257
 2354
 2355 004454 005137 063312
 2356
 2357 004460 020412
 2358 004462 001402
 2359
 2360 004464 000000
 2361 004466 000770
 2362
 2363
 2364
 2365
 2366 004470
 2367 004470 012700 000027

```

      BR      1$      ;LOCK ON HARD ERROR
:*****
:*TEST 25      BASIC "TST -(R)" TEST
:*****
TST25:
      MOV     #25,R0      ;;LOAD R0 WITH TEST NUMBER
      MOV     #DWTA+6,R2  ;DEST ADDR = DWTA+6
      MOV     #377,R4     ;RESULT S / B = 377
1$:    MOV     #DWTA+10,R5 ;BASE DEST ADDR = DWTA+10
      SEN
      ;SCOPE SYNC

2$:    TST     -(R5)     ;TEST THE TST
      BPL     4$
      ;BR IF 'N' CLEAR

3$:    HALT
      BR      1$      ;TST FAILED TO CLEAR 'N'
      ;LOCK ON HARD ERROR

4$:    CMP     R5,R2     ;DID DEST REG GET DECREMENTED?
      BEQ     6$
      ;BR IF YES

5$:    HALT
      BR      1$      ;ERROR - TST FAILED TO UPDATE DEST REG
      ;LOCK ON HARD ERROR

6$:    CMP     R4,(R2)   ;DID TST ALTER [DEST]?
      BEQ     TST26    ;;BR IF NOT

7$:    HALT
      MOV     R4,(R2)   ;RESTORE [DEST]
      BR      1$      ;LOCK ON HARD ERROR

:*****
:*TEST 26      BASIC "COM @#A" TEST
:*****
TST26:
      MOV     #26,R0      ;;LOAD R0 WITH TEST NUMBER
      MOV     #MBUFO,R2  ;DEST ADDR = MBUFO
      CLR     R4         ;RESULT S / B = 177777
      COM     R4
1$:    CLR     (R2)     ;MAKE [DEST] = 000000
      CCC
      ;SCOPE SYNC

2$:    COM     @#MBUFO ;TEST THE COM
      CMP     R4,(R2)   ;RESULT = 177777 ??
      BEQ     TST27    ;;BR IF YES

3$:    HALT
      BR      1$      ;COM DELIVERED THE WRONG RESULT

:*****
:*TEST 27      BASIC "INC @#A" TEST
:*****
TST27:
      MOV     #27,R0      ;;LOAD R0 WITH TEST NUMBER
  
```


2368 004474 012702 063312
 2369 004500 012704 000100
 2370 004504 012712 000077
 2371 004510 000257
 2372
 2373 004512 005237 063312
 2374
 2375 004516 020412
 2376 004520 001402
 2377
 2378 004522 000000
 2379 004524 000767
 2380

```

MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #100,R4 ;RESULT S / B = 100
1$: MOV #77,(R2) ;[DEST] = 77
    CCC ;SCOPE SYNC

2$: INC @#MBUFO ;TEST THE INC

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

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

 : *TEST 30 BASIC "DEC RN" TEST

2381
 2382
 2383
 2384 004526
 2385 004526 012700 000030
 2386 004532 012703 000001
 2387 004536 000257
 2388
 2389 004540 005303
 2390
 2391 004542 005703
 2392 004544 001402
 2393
 2394 004546 000000
 2395 004550 000770
 2396

```

TST30:
1$: MOV #30,R0 ;:LOAD R0 WITH TEST NUMBER
    MOV #1,R3 ;[DEST] = +1
    CCC ;SCOPE SYNC

2$: DEC R3 ;TEST THE DEC

    TST R3 ;RESULT = 000000 ??
    BEQ TST31 ;:BR IF YES

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

 : *TEST 31 BASIC "DEC @#A" TEST

2397
 2398
 2399
 2400 004552
 2401 004552 012700 000031
 2402 004556 012704 177777
 2403 004562 012702 063312
 2404 004566 005012
 2405 004570 000257
 2406
 2407 004572 005337 063312
 2408
 2409 004576 020412
 2410 004600 001402
 2411
 2412 004602 000000
 2413 004604 000770
 2414

```

TST31:
1$: MOV #31,R0 ;:LOAD R0 WITH TEST NUMBER
    MOV #-1,R4 ;RESULT S / B = 177777
    MOV #MBUFO,R2 ;DEST ADDR = MBUFO
    CLR (R2) ;MAKE [DEST] = 000000
    CCC ;SCOPE SYNC

2$: DEC @#MBUFO ;TEST THE DEC

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

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

 : *TEST 32 BASIC "CLR X(R)" TESTS

2415
 2416
 2417
 2418 004606
 2419 004606 012700 000032
 2420 004612 012702 063314
 2421 004616 005004
 2422 004620 012705 063312
 2423 004624 012712 177777

```

TST32:
1$: 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) ;[DEST] = 177777
  
```

CQKDA-D KD11-K BASIC LOGIC TESTS
CQKAD.P11 11-SEP-79 13:31

MACY11 30A(1052) 11-SEP-79^{J 4} 13:53 PAGE 49
T32 BASIC "CLR X(R)" TESTS

SEQ 0048

2424	004630	000257		CCC		;SCOPE SYNC
2425						
2426	004632	005065	000002	28: CLR	2(R5)	;TEST THE CLR
2427						
2428	004636	020412		CMP	R4,(R2)	;RESULT = 0?
2429	004640	001402		BEQ	TST33	::BR IF YES
2430						

2431 004642 000000
 2432 004644 000765
 2433
 2434
 2435
 2436
 2437 004646
 2438 004646 012700 000033
 2439 004652 012703 125252
 2440 004656 000257
 2441
 2442 004660 006303
 2443
 2444 004662 103402
 2445
 2446 004664 000000
 2447 004666 000771
 2448
 2449 004670 022703 052524
 2450 004674 001402
 2451
 2452 004676 000000
 2453 004700 000764
 2454
 2455
 2456
 2457
 2458 004702
 2459 004702 012700 000034
 2460 004706 012703 052525
 2461 004712 000261
 2462
 2463 004714 006303
 2464
 2465 004716 103002
 2466
 2467 004720 000000
 2468 004722 000771
 2469
 2470 004724 022703 125252
 2471 004730 001402
 2472
 2473 004732 000000
 2474 004734 000764
 2475
 2476
 2477
 2478 004736
 2479 004736 012700 000035
 2480 004742 012703 125252
 2481 004746 000257
 2482
 2483 004750 006103
 2484
 2485 004752 103402
 2486

```

3$: HALT ;CLR FAILED TO ZERO [DEST]
   BR 1$ ;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$: HALT ;ASL FAILED TO SET "C" BIT
   BR 1$ ;LOCK ON HRD ERROR

4$: CMP #52524,R3 ;WAS RESULT = 52524 ??
   BEQ TST34 ;;BR IF YES

5$: HALT ;ASL DELIVERED THE WRONG RESULT
   BR 1$ ;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$: HALT ;ASL FAILED TO CLEAR "C"
   BR 1$ ;LOCK ON HARD ERROR

4$: CMP #125252,R3 ;WRESULT = 125252 ??
   BEQ TST35 ;;BR IF YES

5$: HALT ;ASL DELIVERED WRONG REULT
   BR 1$ ;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
  
```

2487 004754 000000
2488 004756 000771
2489
2490 004760 022703 052524
2491 004764 001402
2492
2493 004766 000000
2494 004770 000764
2495
2496
2497
2498
2499 004772

3\$: HALT ;ROL FAILED TO SET "C"
BR 1\$;LOCK ON HARD ERROR
4\$: CMP #052524,R3 ;RESULT = 052524 ??
BEQ TST36 ;:BR IF YES
5\$: HALT ;ROL DELIVERED WRONG RESULT
BR 1\$;LOCK ON HARD ERROR

2500 004772 012700 000036
2501 004776 012703 052524
2502 005002 000261
2503
2504 005004 006103
2505
2506 005006 103002
2507
2508 005010 000000
2509 005012 000771
2510
2511 005014 022703 125251
2512 005020 001402
2513
2514 005022 000000
2515 005024 000764
2516
2517
2518
2519

: *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 ;ROL FAILED TO CLEAR "C"
BR 1\$;LOCK ON HARD ERROR
4\$: CMP #125251,R3 ;RESULT = 125251 ??
BEQ TST37 ;:BR IF YES
5\$: HALT ;ROL DELIVERED WRONG RESULT
BR 1\$;LOCK ON HARD ERROR

2520 005026
2521 005026 012700 000037
2522 005032 012702 063330
2523 005036 012704 000377
2524 005042 000257
2525
2526 005044 105712
2527
2528 005046 100402
2529
2530 005050 000000
2531 005052 000773
2532
2533 005054 020412
2534 005056 001403
2535
2536 005060 000000
2537 005062 010412
2538 005064 000766
2539
2540
2541
2542 005066

: *TEST 37 BASIS "TSTB (R)" TEST - EVEN ADDRESS

TST37:

1\$: MOV #37,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #DWTA+6,R2 ;DEST ADDR = DWTA+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 ;TSTB FAILED TO SET "N"
BR 1\$;LOCK ON HARD ERROR
4\$: CMP R4,(R2) ;DID TSTB DISTURB [DEST]
BEQ TST40 ;:BR IF NOT
5\$: HALT ;TSTB ALTERED [DEST]
MOV R4,(R2) ;RESTORE [DEST]
BR 1\$;LOCK ON HARD ERROR

: *TEST 40 BASIS "TSTB (R)" TEST - ODD ADDRESS

TST40:

```
2543 005066 012700 000040      MOV      #40,R0          ;;LOAD R0 WITH TEST NUMBER
2544 005072 012702 064040      MOV      #DWTB+6,R2     ;DEST ADDR = DWTB+6
2545 005076 012704 177401      MOV      #177401,R4     ;RESULT S / B = 177401
2546 005102 012703 064041      MOV      #DWTB+7,R3     ;DEST ADDR USED = DWTB+7
2547 005106 000257      1$:      CCC              ;SCOPE SYNC
2548
2549 005110 105713      2$:      TSTB      (R3)      ;TEST THE TSTB
2550
2551 005112 100402      BMI      4$              ;BR IF 'N' SET - IT SHOULD BE
2552
2553 005114 000000      3$:      HALT              ;TSTB FAILED TO SET 'N'
2554 005116 000773      BR      1$              ;LOCK ON HARD ERROR
2555
2556 005120 020412      4$:      CMP      R4,(R2)     ;DID TSTB DISTURB [DEST]
2557 005122 001403      BEQ     TST41           ;;BR IF NOT
2558
2559 005124 000000      5$:      HALT              ;TSTB ALTERED [DEST]
2560 005126 010412      MOV     R4,(R2)        ;RESTORE [DEST]
2561 005130 000766      BR      1$              ;LOCK ON HARD EROR
```

```
:::*****
;*TEST 41      BASIC "TSTB @#A" TEST - EVEN ADDRESS
:::*****
```

```
TST41:
2566 005132      MOV      #41,R0          ;;LOAD R0 WITH TEST NUMBER
2567 005132 012700 000041      MOV      #DWTB+4,R2     ;DEST ADDR = DWTB+4
2568 005136 012702 063326      MOV      #177400,R4     ;RESULT S / B = 177400
2569 005142 012704 177400      1$:      CCC              ;SCOPE SYNC
2570 005146 000257
2571
2572 005150 105737 063326      2$:      TSTB      @#DWTB+4    ;TEST THE TSTB
2573
2574 005154 001402      BEQ     4$              ;BR IF "Z" SET - IT SHOULD BE
2575
2576 005156 000000      3$:      HALT              ;TSTB FAILED TO SET "Z"
2577 005160 000772      BR      1$              ;LOCK ON HARD ERROR
2578
2579 005162 020412      4$:      CMP      R4,(R2)     ;DID TSTB DISTURB [DEST]?
2580 005164 001403      BEQ     TST42           ;;BR IF NOT
2581
2582 005166 000000      5$:      HALT              ;TSTB ALTERED [DEST]
2583 005170 010412      MOV     R4,(R2)        ;RESTORE [DEST]
2584 005172 000765      BR      1$              ;LOCK ON HARD ERROR
```

```
:::*****
;*TEST 42      BASIC "TSTB @#A" TEST - UDD ADDRESS
:::*****
```

```
TST42:
2589 005174      MOV      #42,R0          ;;LOAD R0 WITH TEST NUMBER
2590 005174 012700 000042      MOV      #DWTB+6,R2     ;DEST ADDR = DWTB+6
2591 005200 012702 063330      MOV      #377,R4        ;RESULT S / B = 377
2592 005204 012704 000377      1$:      CCC              ;SCOPE SYNC
2593 005210 000257
2594
2595 005212 105737 063331      2$:      TSTB      @#DWTB+7    ;TEST THE TSTB
2596
2597 005216 001402      BEQ     4$              ;BR IF "Z" SET - IT SHOULD BE
2598
```

2599 005220 000000
2600 005222 000772
2601
2602 005224 020412
2603 005226 001403
2604
2605 005230 000000
2606 005232 010412
2607 005234 000765
2608
2609
2610
2611
2612 005236
2613 005236 012700 000043
2614 005242 010605
2615 005244 012704 177400
2616 005250 010506
2617 005252 005046
2618 005254 000257
2619
2620 005256 105366 000001
2621
2622 005262 020416
2623 005264 001402
2624
2625 005266 000000
2626 005270 000767
2627
2628 005272 010506
2629
2630
2631
2632
2633 005274
2634 005274 012700 000044
2635 005300 005003
2636 005302 000257
2637
2638 005304 013703 063276
2639
2640 005310 022703 063322
2641 005314 001402
2642
2643 005316 000000
2644 005320 000767
2645
2646
2647
2648
2649 005322
2650 005322 012700 000045
2651 005326 012702 063314
2652 005332 012704 125252
2653 005336 012703 063312
2654 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$: MOV R5,SP
    CLR -(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

    CMP #DWTA,R3 ;RESULT = DWTA?
    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
1$: MOV #MBUF0,R3 ;[R3] = BASE DEST ADDR
    CLR (R2) ;[DEST] = 000000
```

```
2655 005344 000257          CCC          ;SCOPE SYNC
2656
2657 005346 012763 125252 000002 2$:  MOV      #125252,2(R3)  ;TEST THE MOV
2658
2659 005354 020412          CMP      R4,(R2)    ;RESULT OK?
2660 005356 001402          BEQ      TST46      ;;BR IF YES
2661
2662 005360 000000          3$:  HALT          ;MOV DELIVERED WRONG RESULT
2663 005362 000765          BR      1$         ;LOCK ON HARD ERROR
2664
2665
2666
2667
2668 005364          ::*****
2669 005364 012700 000046          ;*TEST 46  BASIC "MOV #N,(R)" TEST
2670 005370 012703 063312          ;*****
2671 005374 012704 125252          TST46:
2672 005400 005013          MOV      #46,R0    ;;LOAD R0 WITH TEST NUMBER
2673 005402 000257          MOV      #MBUFO,R3 ;DEST ADDR = MBUFO
2674
2675 005404 012713 125252          1$:  MOV      #125252,R4 ;RESULT S / B = 125252
2676
2677 005410 020413          CLR      (R3)     ;[DEST] = 000000
2678 005412 001402          CCC          ;SCOPE SYNC
2679
2680 005414 000000          2$:  MOV      #125252,(R3) ;TEST THE MOV
2681 005416 000770          CMP      R4,(R3)  ;RESULT OK?
2682
2683
2684
2685
2686 005420          BEQ      TST47      ;;BR IF YES
2687 005420 012700 000047          3$:  HALT          ;MOV DELIVERED WRONG RESULT
2688 005424 012705 063276          BR      1$         ;LOCK ON HARD ERROR
2689 005430 005003
2690 005432 000257
2691
2692 005434 012503          ::*****
2693
2694 005436 022703 063322          ;*TEST 47  BASIC "MOV (RA)+,RB" TEST
2695 005442 000402          ;*****
2696
2697 005444 000000          TST47:
2698 005446 000766          MOV      #47,R0    ;;LOAD R0 WITH TEST NUMBER
2699
2700 005450 022705 063300          1$:  MOV      #ATA,R5  ;SRC ADDR = ATA
2701 005454 001402          CLR      R3       ;[DEST] = 000000
2702
2703 005456 000000          CCC          ;SCOPE SYNC
2704 005460 000761          2$:  MOV      (R5)+,R3 ;TEST THE MOV
2705
2706
2707
2708
2709
2710
2711
2712
2713
2714
2715
2716
2717
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734
2735
2736
2737
2738
2739
2740
2741
2742
2743
2744
2745
2746
2747
2748
2749
2750
2751
2752
2753
2754
2755
2756
2757
2758
2759
2760
2761
2762
2763
2764
2765
2766
2767
2768
2769
2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816
2817
2818
2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874
2875
2876
2877
2878
2879
2880
2881
2882
2883
2884
2885
2886
2887
2888
2889
2890
2891
2892
2893
2894
2895
2896
2897
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928
2929
2930
2931
2932
2933
2934
2935
2936
2937
2938
2939
2940
2941
2942
2943
2944
2945
2946
2947
2948
2949
2950
2951
2952
2953
2954
2955
2956
2957
2958
2959
2960
2961
2962
2963
2964
2965
2966
2967
2968
2969
2970
2971
2972
2973
2974
2975
2976
2977
2978
2979
2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998
2999
```

```
2711 005466 012702 063316      MOV      #MBUF1,R2      ;DEST ADDR = MBUF1
2712 005472 012704 063322      MOV      #DWTA,R4      ;RESULT S / B = #DWTA
2713 005476 005012      1$: CLR      (R2)        ;MAKE [DEST] = 000000
2714 005500 000257      CCC                                ;SCOPE SYNC
2715
2716 005502 013737 063276 063316 2$: MOV      @#ATA,@#MBUF1 ;TEST THE MOV
2717 005510 020412      CMP      R4,(R2)       ;DID RESULT = #DWTA ?
2718 005512 001402      BEQ      TST51         ;:BR IF YES
2719
2720 005514 000000      3$: HALT                                ;MOV DELIVERED THE WRONG RESULT
2721 005516 000767      BR      1$            ;LOCK ON HARD ERROR
2722
2723      ;:*****
2724      ;*TEST 51      BASIC "MOV X(R),PC" TEST
2725      ;:*****
2726 005520      TST51:
2727 005520 012700 000051      1$: MOV      #51,R0      ;;LOAD R0 WITH TEST NUMBER
2728 005524 012705 005532      MOV      #2$,R5        ;[R5] = 2$ (BASE ADDRESS)
2729 005530 000257      CCC                                ;SCOPE SYNC
2730
2731 005532 016507 000010      2$: MOV      4$-2$(R5),PC ;TEST THE MOV - GO TO NEXT TEST VIA 4$
2732
2733 005536 000000      3$: HALT                                ;MOV FAILED TO LOAD THE PC
2734 005540 000771      BR      1$            ;LOCK ON HARD ERROR
2735
2736 005542 005544      4$: .+2                ;POINTER TO NEXT TEST
2737
2738      ;:*****
2739      ;*TEST 52      BASIC "MOV @#A,(R)" TEST
2740      ;:*****
2741 005544      TST52:
2742 005544 012700 000052      MOV      #52,R0      ;;LOAD R0 WITH TEST NUMBER
2743 005550 012704 063322      MOV      #DWTA,R4      ;RESULT S / B = #DWTA
2744 005554 012702 063312      MOV      #MBUF0,R2     ;DEST ADDR = MBUF0
2745 005560 005012      1$: CLR      (R2)        ;MAKE [DEST]=000000
2746 005562 000257      CCC                                ;SCOPE SYNC - Z=0
2747
2748 005564 013712 063276      2$: MOV      @#ATA,(R2) ;TEST THE MOV
2749
2750 005570 020412      CMP      R4,(R2)       ;DID RESULT = #DWTA ??
2751 005572 001402      BEQ      TST53         ;:BR IF YES
2752
2753 005574 000000      3$: HALT                                ;MOV DELIVERED WRONG RESULT
2754 005576 000770      BR      1$            ;LOCK ON HARD ERROR
2755
2756      ;:*****
2757      ;*TEST 53      BASIC "MOV X(RA),RB" TEST
2758      ;:*****
2759 005600      TST53:
2760 005600 012700 000053      MOV      #53,R0      ;;LOAD R0 WITH TEST NUMBER
2761 005604 012705 063276      MOV      #ATA,R5       ;[R5] = BASE ADDR FOR SOURCE (ATA)
2762 005610 005003      1$: CLR      R3        ;MAKE [DEST] = 000000
2763 005612 000257      CCC                                ;SCOPE SYNC
2764
2765 005614 016503 000004      2$: MOV      4(R5),R3   ;TEST THE MOV
2766
```


2767 005620 022703 064630
 2768 005624 001402
 2769
 2770 005626 000000
 2771 005630 000767
 2772
 2773
 2774
 2775
 2776 005632
 2777 005632 012700 000054
 2778 005636 012702 063312
 2779 005642 012704 125252
 2780 005646 012705 063314
 2781 005652 005012
 2782 005654 000257
 2783
 2784 005656 010445
 2785
 2786 005660 020412
 2787 005662 001402
 2788
 2789 005664 000000
 2790 005666 000767
 2791
 2792 005670 020205
 2793 005672 001402
 2794
 2795 005674 000000
 2796 005676 000763
 2797
 2798
 2799
 2800
 2801 005700
 2802 005700 012700 000055
 2803 005704 012704 063322
 2804 005710 012702 063312
 2805 005714 012705 063314
 2806 005720 005012
 2807 005722 000257
 2808
 2809 005724 013745 063276
 2810
 2811 005730 020412
 2812 005732 001402
 2813
 2814 005734 000000
 2815 005736 000766
 2816
 2817 005740 020502
 2818 005742 001402
 2819
 2820 005744 000000
 2821 005746 000762
 2822

```

    CMP    #DBTA,P3      ;RESULT = #DBTA ??
    BEQ    TST54         ;;BR IF YES

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

;*****
;*TEST 54    BASIC "MOV RA,-(RB)" TEST
;*****
TST54:
    MOV    #54,R0        ;;LOAD R0 WITH TEST NUMBER
    MOV    #MBUFO,R2     ;FINAL DEST ADDR = MBUFO
    MOV    #125252,R4    ;RESULT S / B = 125252
1$:    MOV    #MBUFO+2,R5 ;INITIAL DEST ADDR = TEMP2 + 2
    CLR    (R2)          ;MAKE [DEST] = 000000
    CCC                          ;SCOPE SYNC

2$:    MOV    R4,-(R5)    ;TEST THE MOV

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

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

4$:    CMP    R2,R5       ;DID REGISTER GET DECREMENTED ?
    BEQ    TST55        ;;BR IF YES

5$:    HALT                ;MOV FAILED TO UPDATE REGISTER
    BR     1$           ;LOCK ON HARD ERROR

;*****
;*TEST 55    BASIC "MOV @#A,-(R)" TEST
;*****
TST55:
    MOV    #55,R0        ;;LOAD R0 WITH TEST NUMBER
    MOV    #DWTA,R4      ;RESULT S / B = #DWTA
    MOV    #MBUFO,R2     ;DEST ADDR = MBUFO
1$:    MOV    #MBUFO+2,R5 ;INITIAL DEST ADDR = MBUFO+2
    CLR    (R2)          ;MAKE [DEST] = 000000
    CCC                          ;SCOPE SYNC

2$:    MOV    @#ATA,-(R5) ;TEST THE MOV

    CMP    R4,(R2)       ;RESULT = 000000
    BEQ    4$           ;BR IF YES

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

4$:    CMP    R5,R2       ;DID DEST REG GET DECREMENTED ??
    BEQ    TST56        ;;BR IF YES

5$:    HALT                ;MOV FAILED TO UPDATE REGISTER
    BR     1$           ;LOCK ON HARD ERROR
  
```

2823
2824
2825
2826 005750
2827 005750 012700 000056
2828 005754 012702 063312
2829 005760 012704 063322
2830 005764 012705 063276
2831 005770 005012
2832 005772 000257
2833
2834 005774 011537 063312
2835
2836 006000 020412
2837 006002 001402
2838
2839 006004 000000
2840 006006 000770
2841
2842
2843
2844
2845 006010
2846 006010 012700 000057
2847 006014 012702 063312
2848 006020 012704 063322
2849 006024 012705 063300
2850 006030 005012
2851 006032 000257
2852
2853 006034 014537 063312
2854
2855 006040 020412
2856 006042 001402
2857
2858 006044 000000
2859 006046 000766
2860
2861 006050 022705 063276
2862 006054 001402
2863
2864 006056 000000
2865 006060 000761
2866
2867
2868
2869 006062
2870 006062 012700 000060
2871 006066 012705 063276
2872 006072 005003
2873 006074 000257
2874
2875 006076 012503
2876
2877 006100 022703 063322
2878 006104 001402

```
*****
;*TEST 56 BASIC "MOV (R),@#A" TEST
*****
TST56:
MOV #56,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #DWTA,R4 ;RESULT S / B = #DWTA
MOV #ATA,R5 ;SOURCE ADDR = ATA
1$: CLR (R2) ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV (R5),@#MBUFO ;TEST THE MOV

CMP R4,(R2) ;RESULT = #DWTA ??
BEQ TST57 ;;BR IF YES

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

*****
;*TEST 57 BASIC "MOV -(R),@#A" TEST
*****
TST57:
MOV #57,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #DWTA,R4 ;RESULT S / B = #DWTA
1$: MOV #ATA+2,R5 ;INITIAL SOURCE ADDR = ATA+2
CLR (R2) ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV -(R5),@#MBUFO ;TEST THE MOV

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

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

4$: CMP #ATA,R5 ;DID THE SRC REG GET DECREMENTED ?
BEQ TST60 ;;BR IF YES

5$: HALT ;MOV FAILED TO UPDATE SOURCE REG
BR 1$ ;LOCK ON HARD ERROR

*****
;*TEST 60 BASIC "MOV (RA),RB" TEST
*****
TST60:
MOV #60,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #ATA,R5 ;INITIAL SOURCE ADDR = ATA
CLR R3 ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC

2$: MOV (R5)+,R3 ;TEST THE MOV

CMP #DWTA,R3 ;RESULT = #DWTA ?
BEQ 4$ ;BR IF YES
```

2879
2880 006106 000000
2881 006110 000766
2882
2883 006112 022705 063300
2884 006116 001402
2885
2886 006120 000000
2887 006122 000761
2888
2889
2890
2891
2892 006124
2893 006124 012700 000061
2894 006130 012705 063276
2895 006134 005003
2896 006136 000257
2897
2898 006140 016503 000002
2899
2900 006144 022703 064032
2901 006150 001402
2902
2903 006152 000000
2904 006154 000767
2905
2906
2907
2908 006156
2909 006156 012700 000062
2910 006162 012737 063324 063314
2911 006170 012705 063312
2912 006174 005003
2913 006176 000257
2914
2915 006200 017503 000002
2916
2917 006204 022703 177777
2918 006210 001402
2919
2920 006212 000000
2921 006214 000767
2922
2923
2924
2925
2926 006216
2927 006216 012700 000063
2928 006222 012704 125252
2929 006226 012702 063320
2930 006232 010437 063312
2931 006236 012705 063312
2932 006242 005012
2933 006244 000257
2934

3\$: HALT ;MOV DELIVERED WRONG RESULT
BR 1\$;LOCK ON HARD ERROR
4\$: CMP #ATA+2,R5 ;DID SOURCE REG GET INCREMENTED
BEQ TST61 ;;BR IF YES
5\$: HALT ;MOV FAILED TO UPDATE SOURCE REGISTER
BR 1\$;LOCK ON HARD ERROR

;*TEST 61 BASIC 'MOV X(RA),RB' TEST

TST61:
MOV #61,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #ATA,R5 ;;BASE SOURCE ADDR = ATA
1\$: CLR R3 ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC
2\$: MOV 2(R5),R3 ;TEST THE MOV
CMP #DWTB,R3 ;RESULT = #DWTB ?
BEQ TST62 ;;BR IF YES
3\$: HALT ;MOV FAILED TO DELIVER CORRECT RESULT
BR 1\$;LOCK ON HARD ERROR

;*TEST 62 BASIC 'MOV @X(RA),RB' TEST

TST62:
MOV #62,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #DWTB+2,@#MBUF0+2 ;;SET UP ADDRESS TABLE MBUF0
MOV #MBUF0,R5 ;;BASE ADDRESS IN R5
1\$: CLR R3 ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC
2\$: MOV @2(R5),R3 ;TEST THE MOV
CMP #-1,R3 ;RESULT = 177777
BEQ TST63 ;;BR IF YES
3\$: HALT ;MOV DELIVERED THE WRONG RESULT
BR 1\$;LOCK ON HARD ERROR

;*TEST 63 BASIC 'MOV (R)+,X(R)' TEST

TST63:
MOV #63,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #125252,R4 ;;RESULT S / B = 125252
MOV #MBUF1+2,R2 ;;FINAL DEST ADDR = MBUF1+2
MOV R4,@#MBUF0 ;;SOURCE OPERAND = 125252
1\$: MOV #MBUF0,R5 ;[R5] = INITIAL SRC ADDR = MBUF0
CLR (R2) ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC

```

2935 006246 012565 000004 2$: MOV (R5)+,4(R5) ;TEST THE MOV
2936
2937 006252 020412 CMP R4,(R2) ;RESULT = 125252 ?
2938 006254 001402 BEQ 4$ ;BR IF YES
2939
2940 006256 000000 3$: HALT ;MOV DELIVERED WRONG RESULT
2941 006260 000766 BR 1$ ;LOCK ON HARD ERROR
2942
2943 006262 022705 063314 4$: CMP #MBUFO+2,R5 ;DID REGISTER GET INCREMENTED ?
2944 006266 001402 BEQ TST64 ;:BR IF YES
2945
2946 006270 000000 5$: HALT ;MOV FAILED TO UPDATE REGISTER
2947 006272 000761 BR 1$ ;LOCK ON HARD ERROR
2948
2949
2950 ;:*****
2950 ;*TEST 64 BASIC "CMP R,@#A" TEST WITH [R] = [A]
2951 ;:*****
2952 TST64:
2952 006274 .SBTTL USER CONTROLLED BREAKPOINT -- BIT1
2953 006274 012700 000064 MOV #64,R0 ;:LOAD R0 WITH TEST NUMBER
2954 BIT #BIT1,@#BPTLOC ;BREAKPOINT HALT SET ??
2955 006300 032737 000002 063234 BEQ .+4 ;BR IF NOT
2956 006306 001401 HALT ;BREAK - DEPRESS CONTINUE TO RESTART
2957 006310 000000 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
2958 006312 012702 063312 MOV #125252,R4 ;RESULT S / B = 125252
2959 006316 012704 125252 1$: MOV R4,R5 ;[R5] = SOURCE OP = 125252
2960 006322 010405 MOV R4,(R2) ;MAKE [DEST] = 125252
2961 006324 010412 CCC ;SCOPE SYNC
2962 006326 000257
2963
2964 006330 020537 063312 2$: CMP R5,@#MBUFO ;TEST THE CMP
2965 BEQ 4$ ;BR IF "Z" WAS SET - IT SHOULD BE
2966 006334 001402
2967
2968 006336 000000 3$: HALT ;CMP FAILED TO SET "Z"
2969 006340 000770 BR 1$ ;LOCK ON HARD ERROR
2970
2971 006342 020412 4$: CMP R4,(R2) ;IS RESULT STILL = 125252 ?
2972 006344 001402 BEQ TST65 ;:BR IF YES
2973
2974 006346 000000 5$: HALT ;CMP ALTERED [DEST]
2975 006350 000764 BR 1$ ;LOCK ON HARD ERROR
2976
2977 ;:*****
2978 ;*TEST 65 BASIC "CMP R,@#A" WITH [R] NOT EQUAL TO [A]
2979 ;:*****
2980 TST65:
2981 006352 012700 000065 MOV #65,R0 ;:LOAD R0 WITH TEST NUMBER
2982 006356 012702 063312 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
2983 006362 012704 125252 MOV #125252,R4 ;MAKE RESULT S / B = 125252
2984 006366 005005 1$: CLR R5 ;[R5] = SOURCE OP = 000000
2985 006370 010412 MOV R4,(R2) ;MAKE [DEST] = 125252
2986 006372 000277 SCC ;SCOPE SYNC - MAKE Z=1
2987
2988 006374 020537 063312 2$: CMP R5,@#MBUFO ;TEST THE CMP
2989
2990 006400 001002 BNE TST66 ;:BR IF Z=0 - IT SHOULD BE
  
```

2991
2992 006402 000000
2993 006404 000770
2994
2995
2996
2997
2998 006406
2999 006406 012700 000066
3000 006412 012702 063312
3001 006416 012704 177777
3002 006422 005012
3003 006424 000257
3004
3005 006426 052737 177777 063312
3006
3007 006434 020412
3008 006436 001402
3009
3010 006440 000000
3011 006442 000767
3012
3013
3014
3015
3016 006444
3017 006444 012700 000067
3018 006450 012702 063312
3019 006454 012704 000077
3020 006460 012712 177777
3021 006464 000257
3022
3023 006466 042737 177700 063312
3024
3025 006474 020412
3026 006476 001402
3027
3028 006500 000000
3029 006502 000766
3030
3031
3032
3033
3034 006504
3035 006504 012700 000070
3036 006510 005003
3037 006512 005103
3038 006514 000257
3039
3040 006516 042703 177400
3041
3042 006522 022703 000377
3043 006526 001402
3044
3045 006530 000000
3046 006532 000766

```
3$: HALT ;CMP FAILED TO CLEAR "Z"  
BR 1$ ;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  
1$: CLR R3 ;:[DEST] = 177777  
COM R3  
CCC ;:SCOPE SYNC  
2$: 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
```

3047
3048
3049
3050
3051 006534
3052 006534 012700 000071
3053 006540 012704 000357
3054 006544 010605
3055 006546 010506
3056 006550 012746 000377
3057 006554 005746
3058 006556 000257
3059
3060 006560 042766 000020 000002
3061
3062 006566 010602
3063 006570 005722
3064 006572 020412
3065 006574 001402
3066
3067 006576 000000
3068 006600 000762
3069
3070 006602 010506
3071
3072
3073
3074
3075 006604
3076 006604 012700 000072
3077 006610 012703 000002
3078 006614 000257
3079
3080 006616 062703 000002
3081
3082 006622 022703 000004
3083 006626 001402
3084
3085 006630 000000
3086 006632 000766
3087
3088
3089
3090
3091 006634
3092 006634 012700 000073
3093 006640 012702 063312
3094 006644 012704 000004
3095 006650 012712 000002
3096 006654 000257
3097
3098 006656 062712 000002
3099
3100 006662 020412
3101 006664 001402
3102

*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 R5,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
1\$: 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 #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #4,R4 ;:RESULT S / B = 4
1\$: MOV #2,(R2) ;:MAKE [DEST] = 2
CCC ;:SCOPE SYNC
2\$: ADD #2,(R2) ;:TEST THE ADD
CMP R4,(R2) ;:RESULT = 4 ?
BEQ TST74 ;:BR IF YES

3103 006666 000000
3104 006670 000767
3105
3106
3107
3108
3109 006672
3110 006672 012700 000074
3111 006676 012704 000002
3112 006702 012702 063314
3113 006706 012705 063312
3114 006712 005012
3115 006714 000257
3116
3117 006716 062765 000002 000002
3118
3119 006724 020412
3120 006726 001402
3121
3122 006730 000000
3123 006732 000765
3124
3125
3126
3127
3128 006734
3129 006734 012700 000075
3130 006740 012704 177400
3131 006744 010605
3132 006746 010602
3133 006750 005742
3134 006752 010506
3135 006754 010446
3136 006756 000257
3137
3138 006760 122726 000000
3139
3140 006764 001402
3141
3142 006766 000000
3143 006770 000770
3144
3145 006772 020506
3146 006774 001402
3147
3148 006776 000000
3149 007000 000764
3150
3151 007002 020412
3152 007004 001402
3153
3154 007006 000000
3155 007010 000760
3156
3157
3158

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

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

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

CMP R4,(R2) ;:RESULT = 2 ?
BEQ TST75 ;:BR IF YES

3$: HALT ;ADD DELIVERED THE WRONG RESULT
BR 1$ ;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,R5 ;:SAVE SP
MOV SP,R2 ;:SET UP DEST ADDR
TST -(R2) ;:R2 CONTAINS DEST ADDR
1$: MOV R5,SP ;:RESET SP FOR ERROR LOOP
MOV R4,-(SP) ;:MAKE [DEST] = 177400
CCC ;:SCOPE SYNC - "Z" = 0

2$: CMPB #0,(SP)+ ;:TEST THE CMPB

BEQ 4$ ;:BR IF "Z" SET - IT SHOULD BE

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

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

5$: HALT ;CMPB FAILED TO UPDATE SP PROPERLY
BR 1$ ;LOCK ON HARD ERROR

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

7$: HALT ;CMPB MODIFIED [DEST]
BR 1$ ;LOCK ON HARD ERROR.

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

```

3159
3160 007012
3161 007012 012700 000076
3162 007016 012704 177777
3163 007022 012702 063324
3164 007026 012705 063330
3165 007032 010203
3166 007034 000257
3167
3168 007036 122523
3169
3170 007040 001402
3171
3172 007042 000000
3173 007044 000770
3174
3175 007046 022703 063325
3176 007052 001402
3177
3178 007054 000000
3179 007056 000763
3180
3181 007060 022705 063331
3182 007064 001402
3183
3184 007066 000000
3185 007070 000756
3186
3187 007072 020412
3188 007074 001403
3189
3190 007076 000000
3191 007100 010412
3192 007102 000751
3193
3194
3195
3196
3197 007104
3198 007104 012700 000077
3199 007110 012704 177777
3200 007114 012702 063324
3201 007120 012705 063327
3202 007124 012703 063325
3203 007130 000257
3204
3205 007132 122523
3206
3207 007134 001402
3208
3209 007136 000000
3210 007140 000767
3211
3212 007142 022703 063326
3213 007146 001402
3214
  
```

```

*****
TST76:
      MOV      #76,R0          ;;LOAD R0 WITH TEST NUMBER
      MOV      #-1,R4         ;;RESULT S / B = 177777
      MOV      #DWTA+2,R2     ;;DEST ADDR = DWTA+2
1$:   MOV      #DWTA+6,R5     ;;SRC ADDR = DWTA+6
      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$:   CMP      #DWTA+7,R5     ;;DID SRC REG GET UPDATED?
      BEQ      8$             ;;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      TST77         ;;BR IF NOT

9$:   HALT     ;              ;;CMPB DELIVERED A RESULT
      MOV      R4,(R2)        ;;RESTORE [DEST]
      BR      1$             ;;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      #DWTA+2,R2     ;;DEST ADDR = DWTA+2
1$:   MOV      #DWTA+5,R5     ;;SRC ADDR = DWTA+5
      MOV      #DWTA+3,R3     ;;R3 GETS DEST ADDR+1
      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$:   CMP      #DWTA+4,R3     ;;DID DEST REG GET UPDATED?
      BEQ      6$             ;;BR IF YES
  
```


3215 007150 000000
3216 007152 000762
3217
3218 007154 022705 063330
3219 007160 001402
3220
3221 007162 000000
3222 007164 000755
3223
3224 007166 020412
3225 007170 001403
3226
3227 007172 000000
3228 007174 010412
3229 007176 000750
3230
3231
3232
3233
3234 007200
3235 007200 012700 000100
3236 007204 012704 177400
3237 007210 012702 063326
3238 007214 012705 063330
3239 007220 012703 063327
3240 007224 000257
3241
3242 007226 122523
3243
3244 007230 001402
3245
3246 007232 000000
3247 007234 000767
3248
3249 007236 022703 063330
3250 007242 001402
3251
3252 007244 000000
3253 007246 000762
3254
3255 007250 022705 063331
3256 007254 001402
3257
3258 007256 000000
3259 007260 000755
3260
3261 007262 020412
3262 007264 001403
3263
3264 007266 000000
3265 007270 010412
3266 007272 000750
3267
3268
3269
3270

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 TST100 ;:BR IF NOT

9\$: HALT ;CMPB DELIVERED A RESULT
MOV R4,(R2) ;RESTORE [DEST]
BR 1\$;LOCK ON HARD ERROR

::*****
;*TEST 100 BASIC "CMPB (RA)+,(RB)+" - SRC / EVEN,DEST / ODD
::*****
TST100:
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 TST101 ;:BR IF NOT

9\$: HALT ;CMPB DELIVERED A RESULT
MOV R4,(R2) ;RESTORE [DEST]
BR 1\$;LOCK ON HARD ERROR

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

3271 007274
 3272 007274 012700 000101
 3273 007300 012704 177777
 3274 007304 012702 063324
 3275 007310 012705 063327
 3276 007314 010203
 3277 007316 000257
 3278
 3279 007320 122523
 3280
 3281 007322 001402
 3282
 3283 007324 000000
 3284 007326 000770
 3285
 3286 007330 022703 063325
 3287 007334 001402
 3288
 3289 007336 000000
 3290 007340 000763
 3291
 3292 007342 022705 063330
 3293 007346 001402
 3294
 3295 007350 000000
 3296 007352 000756
 3297
 3298 007354 020412
 3299 007356 001403
 3300
 3301 007360 000000
 3302 007362 010412
 3303 007364 000751
 3304
 3305
 3306
 3307
 3308 007366
 3309 007366 012700 000102
 3310 007372 012702 063316
 3311 007376 012703 063312
 3312 007402 012704 177400
 3313 007406 012705 064630
 3314 007412 012712 177777
 3315 007416 000257
 3316
 3317 007420 112563 000004
 3318
 3319 007424 020412
 3320 007426 001402
 3321
 3322 007430 000000
 3323 007432 000765
 3324
 3325 007434 022705 064631
 3326 007440 001402

TST101:
 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 TST102 ;:BR IF NOT
 9\$: HALT ;:CMPB DELIVERED A RESULT
 MOV R4,(R2) ;:RESTORE [DEST]
 BR 1\$;:LOCK ON HARD ERROR
 ;:*****
 ;:*TEST 102 BASIC "MOVB (RA)+,X(RB) - SRC EVEN / DEST EVEN
 ;:*****
 TST102:
 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 #DBTA,R5 ;:SRC ADDR = DBTA
 MOV #-1,(R2) ;:[DEST] = 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 #DBTA+1,R5 ;:DID SRC REG GET INCREMENTED BY +1
 BEQ TST103 ;:BR IF YES

```

3327
3328 007442 000000
3329 007444 000760
3330
3331
3332
3333 007446
3334 007446 012700 000103
3335 007452 012702 063316
3336 007456 012703 063312
3337 007462 012704 000777
3338 007466 012705 064635
3339 007472 012712 177777
3340 007476 000257
3341
3342 007500 112563 000005
3343
3344 007504 020412
3345 007506 001402
3346
3347 007510 000000
3348 007512 000765
3349
3350 007514 022705 064636
3351 007520 001402
3352
3353 007522 000000
3354 007524 000760
3355
3356
3357
3358 007526
3359 007526 012700 000104
3360 007532 012702 063316
3361 007536 012703 063312
3362 007542 012704 000377
3363 007546 012705 064630
3364 007552 012712 177777
3365 007556 000257
3366
3367 007560 112563 000005
3368
3369 007564 020412
3370 007566 001402
3371
3372 007570 000000
3373 007572 000765
3374
3375 007574 022705 064631
3376 007600 001402
3377
3378 007602 000000
3379 007604 000760
3380
3381
3382
  
```

```

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) ;:[DEST] = 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 TST104 ;: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) ;:[DEST] = 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 TST105 ;: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
*****
  
```

3383 007606
 3384 007606 012700 000105
 3385 007612 012702 063316
 3386 007616 012703 063312
 3387 007622 012704 177401
 3388 007626 012705 064635
 3389 007632 012712 177777
 3390 007636 000257
 3391
 3392 007640 112563 000004
 3393
 3394 007644 020412
 3395 007646 001402
 3396
 3397 007650 000000
 3398 007652 000765
 3399
 3400 007654 022705 064636
 3401 007660 001402
 3402
 3403 007662 000000
 3404 007664 000760
 3405
 3406
 3407
 3408
 3409 007666
 3410 007666 012700 000106
 3411 007672 012702 063312
 3412 007676 012704 177401
 3413 007702 012705 064032
 3414 007706 010203
 3415 007710 012713 177777
 3416 007714 000257
 3417
 3418 007716 116523 000002
 3419
 3420 007722 020412
 3421 007724 001402
 3422
 3423 007726 000000
 3424 007730 000766
 3425
 3426 007732 022703 063313
 3427 007736 001402
 3428
 3429 007740 000000
 3430 007742 000761
 3431
 3432
 3433
 3434
 3435 007744
 3436 007744 012700 000107
 3437 007750 012702 063312
 3438 007754 012704 177401

TST105:
 MOV #105,R0 ;:LOAD R0 WITH TEST NUMBER
 MOV #MBUF1,R2 ;:DEST ADDR = MBUF1
 MOV #MBUFO,R3 ;:BASE DEST ADDR = MBUFO
 MOV #177401,R4 ;:RESULT S / B = 177401
 1\$: MOV #DBTB+1,R5 ;:SRC ADDR = DBTB+1
 MOV #-1,(R2) ;:[DEST] = 177777
 CCC ;:SCOPE SYNC
 2\$: MOVB (R5)+,4(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 #DBTB+2,R5 ;:DID SRC REG GET INCREMENTED BY +1
 BEQ TST106 ;: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 #MBUFO,R2 ;:DEST ADDR = MBUFO
 MOV #177401,R4 ;:RESULT S / B = 177401
 MOV #DWTB,R5 ;:SRC ADDR = DWTB
 1\$: MOV R2,R3 ;:R3 GETS DEST ADDR
 MOV #-1,(R3) ;:[DEST] = 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 #MBUFO+1,R3 ;:DID DEST REG GET INCREMENTED?
 BEQ TST107 ;: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 #MBUFO,R2 ;:DEST ADDR = MBUFO
 MOV #177401,R4 ;:RESULT S / B = 177401

3439 007760 012705 064634
3440 007764 010203
3441 007766 012713 177777
3442 007772 000257
3443
3444 007774 116523 000001
3445
3446 010000 020412
3447 010002 001402
3448
3449 010004 000000
3450 010006 000766
3451
3452 010010 022703 063313
3453 010014 001402
3454
3455 010016 000000
3456 010020 000761
3457
3458
3459
3460
3461 010022
3462 010022 012700 000110
3463 010026 012702 063312
3464 010032 012704 000777
3465 010036 012705 064032
3466 010042 012703 063313
3467 010046 012712 177777
3468 010052 000257
3469
3470 010054 116523 000002
3471
3472 010060 020412
3473 010062 001402
3474
3475 010064 000000
3476 010066 000765
3477
3478 010070 022703 063314
3479 010074 001402
3480
3481 010076 000000
3482 010100 000760
3483
3484
3485
3486
3487 010102
3488 010102 012700 000111
3489 010106 012702 063312
3490 010112 012704 000777
3491 010116 012705 064634
3492 010122 012703 063313
3493 010126 012712 177777
3494 010132 000257

1\$: MOV #DBTB,R5 ;SRC ADDR = DBTB
MOV R2,R3 ;R3 GETS DEST ADDR
MOV #-1,(R3) ;[DEST] = 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) ;[DEST] = 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 #DBTB,R5 ;SRC ADDR = DBTB
1\$: MOV #MBUF0+1,R3 ;R3 GETS DEST ADDR = MBUF0+1
MOV #-1,(R2) ;[DEST] = 177777
CCC ;SCOPE SYNC

```
3495
3496 010134 116523 000001 2$: MOVB 1(R5),(R3)+ ;TEST THE MOVB
3497
3498 010140 020412 CMP R4,(R2) ;RESULT OK?
3499 010142 001402 BEQ 4$ ;BR IF YES
3500
3501 010144 000000 3$: HALT ;MOVB DELIVERED WRONG RESULT
3502 010146 000765 BR 1$ ;LOCK ON HARD ERROR
3503
3504 010150 022703 063314 4$: CMP #MBUFO+2,R3 ;DID DEST REG GET INCREMENTED?
3505 010154 001402 BEQ TST112 ;;BR IF YES
3506
3507 010156 000000 5$: HALT ;MOVB FAILED TO AUTO INCREMENT DEST REG
3508 010160 000760 BR 1$ ;LOCK ON HARD ERROR
3509
3510 ;:*****
3511 ;*TEST 112 BASIC 'MOVB -(RA),RB' TEST - SRC EVEN ADDR
3512 ;:*****
3513 010162 TST112:
3514 010162 012700 000112 1$: MOV #112,R0 ;;LOAD R0 WITH TEST NUMBER
3515 010166 012705 063331 MOV #DWTA+7,R5 ;SRC ADDR = DWTA+7
3516 010172 005003 CLR R3 ;[DEST] = 000000
3517 010174 000257 CCC ;SCOPE SYNC
3518
3519 010176 114503 2$: MOVB -(R5),R3 ;TEST THE MOVB
3520
3521 010200 022703 177777 CMP #-1,R3 ;RESULT OK?
3522 010204 001402 BEQ 4$ ;BR IF YES
3523
3524 010206 000000 3$: HALT ;MOVB FAILED - WRONG RESULT
3525 010210 000766 BR 1$ ;LOCK ON HARD ERROR
3526
3527 010212 022705 063330 4$: CMP #DWTA+6,R5 ;SRC REG GET DECREMENTED?
3528 010216 001402 BEQ TST113 ;;BR IF YES
3529
3530 010220 000000 5$: HALT ;MOVB FAILED TO UPDATE SRC REG
3531 010222 000761 BR 1$ ;LOCK ON HARD ERROR
3532
3533 ;:*****
3534 ;*TEST 113 BASIC 'MOVB -(RA),RB' TEST - SRC ODD ADDR
3535 ;:*****
3536 010224 TST113:
3537 010224 012700 000113 1$: MOV #113,R0 ;;LOAD R0 WITH TEST NUMBER
3538 010230 012705 063330 MOV #DWTA+6,R5 ;SRC ADDR = DWTA+6
3539 010234 005003 CLR R3 ;[DEST] = 000000
3540 010236 000257 CCC ;SCOPE SYNC
3541
3542 010240 114503 2$: MOVB -(R5),R3 ;TEST THE MOVB
3543
3544 010242 022703 177777 CMP #-1,R3 ;RESULT OK?
3545 010246 001402 BEQ 4$ ;BR IF YES
3546
3547 010250 000000 3$: HALT ;MOVB FAILED - WRONG RESULT
3548 010252 000766 BR 1$ ;LOCK ON HARD ERROR
3549
3549 010254 022705 063327 4$: CMP #DWTA+5,R5 ;SRC REG GET DECREMENTED?
3550 010260 001402 BEQ TST114 ;;BR IF YES
```

```

3551
3552 010262 000000
3553 010264 000761
3554
3555
3556
3557
3558 010266
3559 010266 012700 000114
3560 010272 010605
3561 010274 012704 177400
3562 010300 010506
3563 010302 012703 064630
3564 010306 012746 177777
3565 010312 010602
3566 010314 005726
3567 010316 000257
3568
3569 010320 112346
3570
3571 010322 022703 064631
3572 010326 001402
3573
3574 010330 000000
3575 010332 000762
3576
3577 010334 020412
3578 010336 001402
3579
3580 010340 000000
3581 010342 000756
3582
3583 010344 020206
3584 010346 001402
3585
3586 010350 000000
3587 010352 000752
3588
3589 010354 010506
3590
3591
3592
3593
3594 010356
3595 010356 012700 000115
3596 010362 010605
3597 010364 012704 177400
3598 010370 010506
3599 010372 012703 064035
3600 010376 012746 177777
3601 010402 010602
3602 010404 005726
3603 010406 000257
3604
3605 010410 112346
3606

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 #177400,R4 ;:RESULT S / B = 177400
1$: MOV R5,SP ;:RESET SP FOR ERROR LOOP
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

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

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 #177400,R4 ;:RESULT S / B = 177400
1$: MOV R5,SP ;:RESET SP FOR ERROR LOOP
MOV #DWTB+3,R3 ;:SRC ADDR = DWTB+3
MOV #-1,-(SP) ;:[DEST] = 177777
MOV SP,R2 ;:R2 GETS DEST ADDR
TST (SF)+ ;:RESET SP
CCC ;:SCOPE SYNC

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

```

3607 010412 022703 064036
 3608 010416 001402
 3609
 3610 010420 000000
 3611 010422 000762
 3612
 3613 010424 020412
 3614 010426 001402
 3615
 3616 010430 000000
 3617 010432 000756
 3618
 3619 010434 020206
 3620 010436 001402
 3621
 3622 010440 000000
 3623 010442 000752
 3624
 3625 010444 010506
 3626
 3627
 3628
 3629
 3630 010446
 3631 010446 012700 000116
 3632 010452 012702 063312
 3633 010456 012704 000001
 3634 010462 012705 064032
 3635 010466 005012
 3636 010470 000257
 3637

```

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

        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

        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      #MBUFO,R2     ;:DEST ADDR = MBUFO
MOV      #1,R4          ;:RESULT S / B = 1
MOV      #DWTB,R5     ;:BASE SRC ADDR = DWTB
1$:      CLR      (R2)   ;:[DEST] = 000000
        CCC          ;:SCOPE SYNC
  
```



```
3638 010472 116537 000006 063312 2$:   MOVB    6(R5),@#MBOF0 ;TEST THE MOVB
3639
3640 010500 020412           CMP     R4,(R2)       ;RESULT OK?
3641 010502 001402           BEQ     TST117        ;;BR IF YES
3642
3643 010504 000000 3$:   HALT                    ;MOVB DELIVERED WRONG RESULT
3644 010506 000767           BR      1$            ;LOCK ON HARD ERROR
3645
3646 :*****
3647 :*TEST 117 BASIC 'MOVB X(R),@#A' TEST - SRC ODD / DEST EVEN
3648 :*****
3648 010510 TST117:
3649 010510 012700 000117       MOV     #117,R0        ;;LOAD R0 WITH TEST NUMBER
3650 010514 012702 063312       MOV     #MBOF0,R2     ;DEST ADDR = MBOF0
3651 010520 012704 000001       MOV     #1,R4         ;RESULT S / B = 1
3652 010524 012705 064634       MOV     #DBTB,R5     ;BASE SRC ADDR = DBTB
3653 010530 005012 1$:   CLR     (R2)         ;[DEST] = 000000
3654 010532 000257           CCC                    ;SCOPE SYNC
3655
3656 010534 116537 000001 063312 2$:   MOVB    1(R5),@#MBOF0 ;TEST THE MOVB
3657
3658 010542 020412           CMP     R4,(R2)       ;RESULT OK?
3659 010544 001402           BEQ     TST120        ;;BR IF YES
3660
3661 010546 000000 3$:   HALT                    ;MOVB DELIVERED WRONG RESULT
3662 010550 000767           BR      1$            ;LOCK ON HARD ERROR
3663
3664 :*****
3665 :*TEST 120 BASIC 'MOVB X(R),@#A' TEST - SRC EVEN / DEST ODD
3666 :*****
3666 010552 TST120:
3667 010552 012700 000120       MOV     #120,R0       ;;LOAD R0 WITH TEST NUMBER
3668 010556 012702 063312       MOV     #MBOF0,R2     ;DEST ADDR = MBOF0
3669 010562 012704 000400       MOV     #400,R4      ;RESULT S / B = 400
3670 010566 012705 064032       MOV     #DWTB,R5     ;BASE SRC ADDR = DWTB
3671 010572 005012 1$:   CLR     (R2)         ;[DEST] = 000000
3672 010574 000257           CCC                    ;SCOPE SYNC
3673
3674 010576 116537 000006 063313 2$:   MOVB    6(R5),@#MBOF0+1 ;TEST THE MOVB
3675
3676 010604 020412           CMP     R4,(R2)       ;RESULT OK?
3677 010606 001402           BEQ     TST121        ;;BR IF YES
3678
3679 010610 000000 3$:   HALT                    ;MOVB DELIVERED WRONG RESULT
3680 010612 000767           BR      1$            ;LOCK ON HARD ERROR
3681
3682 :*****
3683 :*TEST 121 BASIC 'MOVB X(R),@#A' TEST - SRC ODD / DEST ODD
3684 :*****
3684 010614 TST121:
3685 010614 012700 000121       MOV     #121,R0       ;;LOAD R0 WITH TEST NUMBER
3686 :SBTTL USER CONTROLLED BREAKPOINT -- BIT2
3687 010620 032737 000004 063234       BIT     #BIT2,@#BPTLOC ;BREAKPOINT HALT SET ??
3688 010626 001401           BEQ     .+4           ;BR IF NOT
3689 010630 000000           HALT                    ;BREAK - DEPRESS CONTINUE TO RESTART
3690 010632 012702 063312       MOV     #MBOF0,R2     ;DEST ADDR = MBOF0
3691 010636 012704 000400       MOV     #400,R4      ;RESULT S / B = 400
3692 010642 012705 064634       MOV     #DBTB,R5     ;BASE SRC ADDR = DBTB
3693 010646 005012 1$:   CLR     (R2)         ;[DEST] = 000000
```

```
3694 010650 000257          CCC          ;SCOPE SYNC
3695
3696 010652 116537 000001 063313 2$:  MOVB    1(R5),@#MBUF0+1 ;TEST THE MOVB
3697
3698 010660 020412          CMP     R4,(R2)  ;RESULT OK?
3699 010662 001402          BEQ     TST122   ;;BR IF YES
3700
3701 010664 000000          3$:  HALT          ;MOVB DELIVERED WRONG RESULT
3702 010666 000767          BR     1$       ;LOCK ON HARD ERROR
3703
3704
3705          ;:*****
3706          ;*TEST 122      BASIC QUICK VERIFY TEST FOR BMI,BEQ,BVS,BCS-FLAG=0
3707          ;:*****
3708 010670 012700 000122  TST122:
3709 010674 000257          1$:  MOV     #122,R0      ;;LOAD R0 WITH TEST NUMBER
3710
3711 010676 001404          2$:  CCC          ;CLEAR ALL FLAGS
3712 010700 100403          BEQ     3$       ;NO BR SHOULD OCCUR-FLAG=0
3713 010702 102402          BMI     3$       ;NO BR SHOULD OCCUR-FLAG=0
3714 010704 103401          BVS     3$       ;NO BR SHOULD OCCUR-FLAG=0
3715 010706 000402          BCS     3$       ;NO BR SHOULD OCCUR-FLAG=0
3716
3717 010710 000000          BR     TST123   ;;GO TO NEXT TEST
3718 010712 000770          3$:  HALT          ;ONE OF ABOVE BR'S FAILED
3719          BR     1$       ;ERROR LOOP RETURN
3720
3721          ;:*****
3722          ;*TEST 123      BASIC QUICK VERIFY TEST FOR BMI,BEQ,BVS,BCS-FLAG=1
3723          ;:*****
3724 010714 012700 000123  TST123:
3725 010720 000277          1$:  MOV     #123,R0      ;;LOAD R0 WITH TEST NUMBER
3726
3727 010722 001402          SCC          ;MAKE N:C = 1111
3728
3729 010724 000000          21$: BEQ     22$      ;TEST THE BEQ-IT SHOULD BR
3730 010726 000774          3$:  HALT          ;BEQ FAILED
3731          BR     1$       ;ERROR LOOP RETURN
3732 010730 100402          22$: BMI     23$      ;TEST THE BMI-IT SHOULD BR
3733
3734 010732 000000          5$:  HALT          ;BMI FAILED
3735 010734 000771          BR     1$       ;ERROR LOOP RETURN
3736
3737 010736 102402          23$: BVS     24$      ;TEST THE BVS-IT SHOULD BR
3738
3739 010740 000000          7$:  HALT          ;BVS FAILED
3740 010742 000766          BR     1$       ;ERROR LOOP RETURN
3741
3742 010744
3743 010744 103402          24$: BCS     TST124   ;;TEST THE BCS-IT SHOULD BR
3744
3745 010746 000000          9$:  HALT          ;BCS FAILED
3746 010750 000763          BR     1$       ;ERROR LOOP RETURN
3747
3748          ;:*****
3749          ;*TEST 124      BASIC BVC TEST WITH V=1
```



```
3806 ;*TEST 130 BASIC BGE TEST WITH N,V = 10
3807 ;*****
3808 011040 TST130:
3809 011040 012700 000130 MOV #130,R0 ;:LOAD R0 WITH TEST NUMBER
3810
3811 011044 000257 1$: CCC ;CLEAR FLAGS
3812 011046 000270 SEN ;MAKE N,V = 10
3813
3814 011050 002001 2$: BGE 3$ ;TEST THE BGE-IT SHOULDN'T BR
3815 011052 000402 BR TST131 ;:GO TO NEXT TEST
3816
3817 011054 000000 3$: HALT ;BGE FAILED
3818 011056 000772 BR 1$ ;ERROR LOOP RETURN
3819
3820 ;*****
3821 ;*TEST 131 BASIC BGE TEST WITH N,V = 11
3822 ;*****
3823 011060 TST131:
3824 011060 012700 000131 MOV #131,R0 ;:LOAD R0 WITH TEST NUMBER
3825
3826 011064 000257 1$: CCC ;CLEAR FLAGS
3827 011066 000272 272 ;MAKE N,V = 11
3828
3829 011070 2$: BGE TST132 ;:TEST THE BGE-IT SHOULD BR
3830 011070 002002
3831
3832 011072 000000 3$: HALT ;BGE FAILED
3833 011074 000773 BR 1$ ;ERROR LOOP RETURN
3834
3835 ;*****
3836 ;*TEST 132 BASIC BLT TEST WITH N,V = 00
3837 ;*****
3838 011076 TST132:
3839 011076 012700 000132 MOV #132,R0 ;:LOAD R0 WITH TEST NUMBER
3840
3841 011102 000257 1$: CCC ;CLEAR FLAGS
3842
3843 011104 002401 2$: BLT 3$ ;TEST THE BLT-IT SHOULDN'T BR
3844 011106 000402 BR TST133 ;:GO TO NEXT TEST
3845
3846 011110 000000 3$: HALT ;BLT FAILED
3847 011112 000773 BR 1$ ;ERROR LOOP RETURN
3848
3849 ;*****
3850 ;*TEST 133 BASIC BLT TEST WITH N,V = 01
3851 ;*****
3852 011114 TST133:
3853 011114 012700 000133 MOV #133,R0 ;:LOAD R0 WITH TEST NUMBER
3854
3855 011120 000257 1$: CCC ;CLEAR FLAGS
3856 011122 000262 SEV ;MAKE N,V = 01
3857
3858 011124 2$: BLT TST134 ;:TEST THE BLT-IT SHOULD BR
3859 011124 002402
3860
3861 011126 000000 3$: HALT ;BLT FAILED
```

3862 011130 000773
3863
3864
3865
3866
3867 011132
3868 011132 012700 000134
3869
3870 011136 000257
3871 011140 000270
3872
3873 011142
3874 011142 002402
3875
3876 011144 000000
3877 011146 000773
3878
3879
3880
3881
3882 011150
3883 011150 012700 000135
3884
3885 011154 000257
3886 011156 000272
3887
3888 011160 002401
3889 011162 000402
3890
3891 011164 000000
3892 011166 000772
3893
3894
3895
3896
3897 011170
3898 011170 012700 000136
3899
3900 011174 000257
3901 011176 000266
3902
3903 011200 003001
3904 011202 000402
3905
3906 011204 000000
3907 011206 000772
3908
3909
3910
3911
3912 011210
3913 011210 012700 000137
3914
3915 011214 000257
3916 011216 000262
3917

```
BR 1$ ;ERROR LOOP RETURN

*****
*TEST 134 BASIC BLT TEST WITH N,V = 10
*****
TST134:
MOV #134,R0 ;:LOAD R0 WITH TEST NUMBER
1$: CCC ;CLEAR FLAGS
SEN ;SET N - N,V = 10
2$: BLT TST135 ;:TEST THE BLT-IT SHOULD BR
3$: HALT ;BLT FAILED
BR 1$ ;ERROR LOOP RETURN

*****
*TEST 135 BASIC BLT TEST WITH N,V = 11
*****
TST135:
MOV #135,R0 ;:LOAD R0 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 R0 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 R0 WITH TEST NUMBER
1$: CCC ;CLEAR FLAGS
SEV ;SET V
```

3918 011220 003001
3919 011222 000402
3920
3921 011224 000000
3922 011226 000772
3923
3924
3925
3926
3927 011230
3928 011230 012700 000140
3929
3930 011234 000257
3931 011236 000264
3932
3933 011240 003001
3934 011242 000402
3935
3936 011244 000000
3937 011246 000772
3938
3939
3940
3941
3942 011250
3943 011250 012700 000141
3944
3945 011254 000257
3946
3947 011256
3948 011256 003002
3949
3950 011260 000000
3951 011262 000774
3952
3953
3954
3955
3956 011264
3957 011264 012700 000142
3958
3959 011270 000257
3960 011272 000266
3961
3962 011274 003001
3963 011276 000402
3964
3965 011300 000000
3966 011302 000772
3967
3968
3969
3970
3971 011304
3972 011304 012700 000143
3973

```
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 R0 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 R0 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 R0 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 R0 WITH TEST NUMBER
```

3974 011310 000257
3975 011312 000274
3976
3977 011314 003001
3978 011316 000402
3979
3980 011320 000000
3981 011322 000772
3982
3983
3984
3985
3986 011324
3987 011324 012700 000144
3988
3989 011330 000257
3990 011332 000276
3991
3992 011334 003001
3993 011336 000402
3994
3995 011340 000000
3996 011342 000772
3997
3998
3999
4000
4001 011344
4002 011344 012700 000145
4003
4004 011350 000257
4005 011352 000272
4006
4007 011354
4008 011354 003002
4009
4010 011356 000000
4011 011360 000773
4012
4013
4014
4015
4016 011362
4017 011362 012700 000146
4018
4019 011366 000257
4020
4021 011370
4022 011370 101002
4023
4024 011372 000000
4025 011374 000774
4026
4027
4028
4029

```
1$:   CCC           ;CLEAR FLAGS
      274           ;MAKE Z = 1 AND N,V = 10

2$:   BGT          3$ ;TEST THE BLT-IT SHOULDN'T BR
      BR           TST144 ;:GO TO NEXT TEST

3$:   HALT         ;BLT FAILED
      BR          1$  ;ERROR LOOP RETURN

;*****
;*TEST 144 BASIC BGT TEST WITH Z = 1 AND N,V = 11
;*****
TST144:
      MOV          #144,R0 ;:LOAD R0 WITH TEST NUMBER

1$:   CCC           ;CLEAR FLAGS
      276           ;MAKE Z = 1 AND N,V = 11

2$:   BGT          3$ ;TEST THE BGT-IT SHOULD NOT BR
      BR           TST145 ;:GO TO NEXT TEST

3$:   HALT         ;BLT FAILED
      BR          1$  ;ERROR LOOP RETURN

;*****
;*TEST 145 BASIC BGT TEST WITH Z=0 AND N,V=11
;*****
TST145:
      MOV          #145,R0 ;:LOAD R0 WITH TEST NUMBER

1$:   CCC           ;CLEAR FLAGS
      272           ;MAKE N:C=1010

2$:   BGT          TST146 ;:TEST THE BGT - IT SHOULD BR

3$:   HALT         ;BGT FAILED
      BR          1$  ;ERROR LOOP RETURN

;*****
;*TEST 146 BASIC BHI TEST WITH Z,C = 00
;*****
TST146:
      MOV          #146,R0 ;:LOAD R0 WITH TEST NUMBER

1$:   CCC           ;MAKE Z,C = 00

2$:   BHI          TST147 ;:TEST THE BHI-IT SHOULD BR

3$:   HALT         ;BHI FAILED
      BR          1$  ;ERROR LOOP RETURN

;*****
;*TEST 147 BASIC BHI TEST WITH Z,C = 01
;*****
```

4030 011376
4031 011376 012700 000147
4032
4033 011402 000257
4034 011404 000261
4035
4036 011406 101001
4037 011410 000402
4038
4039 011412 000000
4040 011414 000772
4041
4042
4043
4044
4045 011416
4046 011416 012700 000150
4047
4048 011422 000257
4049 011424 000264
4050
4051 011426 101001
4052 011430 000402
4053
4054 011432 000000
4055 011434 000772
4056
4057
4058
4059
4060 011436
4061 011436 012700 000151
4062
4063 011442 000257
4064 011444 000265
4065
4066 011446 101001
4067 011450 000402
4068
4069 011452 000000
4070 011454 000772
4071
4072
4073
4074
4075 011456
4076 011456 012700 000152
4077 011462 012704 177776
4078 011466 012703 000002
4079 011472 000257
4080 011474 000266
4081
4082 011476 005403
4083
4084 011500 100003
4085 011502 001402

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


```
4086 011504 102401          BVS      3$
4087 011506 103402          BCS      4$
4088
4089 011510 000000          3$:     HALT                    ;NEGATE FAILED TO ALTER CODES PROPERLY
4090 011512 000765          BR       1$                    ;ERROR LOOP RETURN
4091
4092 011514 020304          4$:     CMP      R3,R4          ;CORRECT RESULT?
4093 011516 001402          BEQ      TST153                ;;BR IF YES
4094
4095 011520 000000          5$:     HALT                    ;NEG DELIVERED WRONG RESULT
4096 011522 000761          BR       1$                    ;ERROR LOOP RETURN
4097
4098
4099
4100
4101 011524
4102 011524 012700 000153
4103 011530 012704 000002
4104 011534 012702 063312
4105 011540 012712 000004          1$:     MOV      #153,R0        ;;LOAD R0 WITH TEST NUMBER
4106 011544 000257          MOV      #2,R4                ;RESULT S / B = 2
4107
4108 011546 162737 000002 063312 2$:     MOV      #MBUF0,R2        ;R2 POINTS TO DEST
4109
4110
4111
4112
4113
4114
4115
4116
4117
4118
4119 011566
4120 011566 012700 000154          1$:     MOV      #4,(R2)        ;INITIAL [DEST] = 4
4121 011572 012737 000002 063312          CCC
4122 011600 012703 000004          ;CLEAR FLAGS
4123 011604 000257
4124
4125 011606 163703 063312          2$:     SUB      #2,@#MBUF0    ;TEST THE SUB
4126
4127
4128
4129
4130
4131
4132
4133
4134
4135
4136 011622
4137 011622 012700 000155
4138 011626 010605
4139 011630 010506
4140 011632 012703 011652
4141 011636 012746 177777          3$:     CMP      R4,(R2)        ;RESULT=2?
4142
4143
4144
4145
4146
4147
4148
4149
4150
4151
4152
4153
4154
4155
4156
4157
4158
4159
4160
4161
4162
4163
4164
4165
4166
4167
4168
4169
4170
4171
4172
4173
4174
4175
4176
4177
4178
4179
4180
4181
4182
4183
4184
4185
4186
4187
4188
4189
4190
4191
4192
4193
4194
4195
4196
4197
4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240
4241
4242
4243
4244
4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263
4264
4265
4266
4267
4268
4269
4270
4271
4272
4273
4274
4275
4276
4277
4278
4279
4280
4281
4282
4283
4284
4285
4286
4287
4288
4289
4290
4291
4292
4293
4294
4295
4296
4297
4298
4299
4300
4301
4302
4303
4304
4305
4306
4307
4308
4309
4310
4311
4312
4313
4314
4315
4316
4317
4318
4319
4320
4321
4322
4323
4324
4325
4326
4327
4328
4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347
4348
4349
4350
4351
4352
4353
4354
4355
4356
4357
4358
4359
4360
4361
4362
4363
4364
4365
4366
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381
4382
4383
4384
4385
4386
4387
4388
4389
4390
4391
4392
4393
4394
4395
4396
4397
4398
4399
4400
4401
4402
4403
4404
4405
4406
4407
4408
4409
4410
4411
4412
4413
4414
4415
4416
4417
4418
4419
4420
4421
4422
4423
4424
4425
4426
4427
4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446
4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
4497
4498
4499
4500
4501
4502
4503
4504
4505
4506
4507
4508
4509
4510
4511
4512
4513
4514
4515
4516
4517
4518
4519
4520
4521
4522
4523
4524
4525
4526
4527
4528
4529
4530
4531
4532
4533
4534
4535
4536
4537
4538
4539
4540
4541
4542
4543
4544
4545
4546
4547
4548
4549
4550
4551
4552
4553
4554
4555
4556
4557
4558
4559
4560
4561
4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573
4574
4575
4576
4577
4578
4579
4580
4581
4582
4583
4584
4585
4586
4587
4588
4589
4590
4591
4592
4593
4594
4595
4596
4597
4598
4599
4600
4601
4602
4603
4604
4605
4606
4607
4608
4609
4610
4611
4612
4613
4614
4615
4616
4617
4618
4619
4620
4621
4622
4623
4624
4625
4626
4627
4628
4629
4630
4631
4632
4633
4634
4635
4636
4637
4638
4639
4640
4641
4642
4643
4644
4645
4646
4647
4648
4649
4650
4651
4652
4653
4654
4655
4656
4657
4658
4659
4660
4661
4662
4663
4664
4665
4666
4667
4668
4669
4670
4671
4672
4673
4674
4675
4676
4677
4678
4679
4680
4681
4682
4683
4684
4685
4686
4687
4688
4689
4690
4691
4692
4693
4694
4695
4696
4697
4698
4699
4700
4701
4702
4703
4704
4705
4706
4707
4708
4709
4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720
4721
4722
4723
4724
4725
4726
4727
4728
4729
4730
4731
4732
4733
4734
4735
4736
4737
4738
4739
4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754
4755
4756
4757
4758
4759
4760
4761
4762
4763
4764
4765
4766
4767
4768
4769
4770
4771
4772
4773
4774
4775
4776
4777
4778
4779
4780
4781
4782
4783
4784
4785
4786
4787
4788
4789
4790
4791
4792
4793
4794
4795
4796
4797
4798
4799
4800
4801
4802
4803
4804
4805
4806
4807
4808
4809
4810
4811
4812
4813
4814
4815
4816
4817
4818
4819
4820
4821
4822
4823
4824
4825
4826
4827
4828
4829
4830
4831
4832
4833
4834
4835
4836
4837
4838
4839
4840
4841
4842
4843
4844
4845
4846
4847
4848
4849
4850
4851
4852
4853
4854
4855
4856
4857
4858
4859
4860
4861
4862
4863
4864
4865
4866
4867
4868
4869
4870
4871
4872
4873
4874
4875
4876
4877
4878
4879
4880
4881
4882
4883
4884
4885
4886
4887
4888
4889
4890
4891
4892
4893
4894
4895
4896
4897
4898
4899
4900
4901
4902
4903
4904
4905
4906
4907
4908
4909
4910
4911
4912
4913
4914
4915
4916
4917
4918
4919
4920
4921
4922
4923
4924
4925
4926
4927
4928
4929
4930
4931
4932
4933
4934
4935
4936
4937
4938
4939
4940
4941
4942
4943
4944
4945
4946
4947
4948
4949
4950
4951
4952
4953
4954
4955
4956
4957
4958
4959
4960
4961
4962
4963
4964
4965
4966
4967
4968
4969
4970
4971
4972
4973
4974
4975
4976
4977
4978
4979
4980
4981
4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
```

```
4142 011642 000277          SCC          ;N:C = 1111
4143
4144 011644 000203      2$:  RTS      R3          ;TEST THE RTS - GO TO 4$
4145
4146 011646 000000      3$:  HALT          ;RTS FAILED TO LOAD THE PC
4147 011650 000767          BR      1$          ;LOCK ON ERROR
4148
4149 011652 100003      4$:  BPL      5$          ;N:C = 1111 ?
4150 011654 001002          BNE     5$
4151 011656 102001          BVC     5$
4152 011660 103402          BCS     6$
4153
4154 011662 000000      5$:  HALT          ;RTS ALTERED CODES - CLEARED ONE
4155 011664 000761          BR      1$          ;LOCK ON ERROR
4156
4157 011666 020327 177777 6$:  CMP      R3,#-1      ;DID R3 GET LOADED FROM STACK ?
4158 011672 001402          BEQ     8$          ;BR IF YES
4159
4160 011674 000000      7$:  HALT          ;RTS FAILED TO LOAD REG
4161 011676 000754          BR      1$          ;LOCK ON ERROR
4162
4163 011700 020506      8$:  CMP      R5,SP      ;DID RTS POP THE STACK POINTER ?
4164 011702 001402          BEQ     TST156      ;:BR IF YES
4165
4166 011704 000000      9$:  HALT          ;RTS FAILED TO POP SP
4167 011706 000750          BR      1$          ;LOCK ON ERROR
4168
```

```
4169
4170          ;:*****
4171          ;:*TEST 156 BASIC "RTS PC" TEST
4172          ;:*****
```

```
TST156:
4173 011710          MOV      #156,R0      ;;LOAD R0 WITH TEST NUMBER
4174 011714 012700 000156  MOV      SP,R5      ;SAVE THE ORIGINAL SP
4175 011716 010506          MOV      R5,SP      ;RESET SP FOR ERROR LOOP
4176 011720 012746 011734  MOV      #4$,-(SP)   ;PUSH NEW PC ON STACK
4177 011724 000257          CCC          ;SCOPE SYNC
4178
4179 011726 000207      2$:  RTS      PC          ;TEST THE RTS - GO TO 4$
4180
4181 011730 000000      3$:  HALT          ;RTS FAILED TO LOAD PC
4182 011732 000771          BR      1$          ;LOCK ON HARD ERROR
4183
4184 011734 020605      4$:  CMP      SP,R5      ;DID SP GET POPPED ?
4185 011736 001402          BEQ     TST157      ;:BR IF YES
4186
4187 011740 000000      5$:  HALT          ;RTS FAILED TO UPDATE SP
4188 011742 000765          BR      1$          ;LOCK ON HARD ERROR
4189
```

```
4190          ;:*****
4191          ;:*TEST 157 BASIC "JSR PC,@#A" TEST
4192          ;:*****
```

```
TST157:
4193 011744          MOV      #157,R0      ;;LOAD R0 WITH TEST NUMBER
4194 011744 012700 000157  .SBTTL USER CONTROLLED BREAKPOINT -- BIT3
4195          BIT      #BIT3,@#BPTLOC ;BREAKPOINT HALT SET ??
4196 011750 032737 000010 063234 BEQ     .+4          ;BR IF NOT
4197 011756 001401
```

4198	011760	000000			HALT		;BREAK - DEPRESS CONTINUE TO RESTART
4199	011762	010605			MOV	SP,R5	;SAVE ORIGINAL SP
4200	011764	010506		1\$:	MOV	R5,SP	;RESET SP FOR ERROR LOOP
4201	011766	000257			CCC		;SCOPE SYNC
4202							
4203	011770	004737	012000	2\$:	JSR	PC,@#4\$;TEST THE JSR - GO TO 4\$
4204							
4205	011774	000000		3\$:	HALT		;JSR FAILED TO LOAD PC
4206	011776	000772			BR	1\$;LOCK ON HARD ERROR
4207							
4208	012000	022726	011774	4\$:	CMP	#3\$,(SP)+	;DID JSR SAVE OLD PC ON STACK ?
4209	012004	001402			BEQ	TST160	;:BR IF YES
4210							
4211	012006	000000		5\$:	HALT		;JSR FAILED TO SAVE OLD PC
4212	012010	000765			BR	1\$;LOCK ON HARD ERROR
4213							

;*TEST 160 BASIC "RTI" TEST - N:C=0000

4214							
4215							
4216							
4217	012012				TST160:		
4218	012012	012700	000160		MOV	#160,R0	;:LOAD R0 WITH TEST NUMBER
4219	012016	010605			MOV	SP,R5	;SAVE THE SP
4220	012020	010506		1\$:	MOV	R5,SP	;RESET THE SP FOR ERROR LOOP
4221	012022	012746	000357		MOV	#357,-(SP)	;NEW PSW = 357
4222	012026	012746	012046		MOV	#4\$,-(SP)	;NEW PC = 4\$
4223	012032	005037	177776		CLR	@#PSW	;MAKE [PSW] = 000
4224	012036	000257			CCC		;MAKE N:C=0000
4225							
4226	012040	000002		2\$:	RTI		;TEST THE RTI - GO TO 4\$
4227							
4228	012042	000000		3\$:	HALT		;RTI FAILED TO LOAD PC
4229	012044	000765			BR	1\$;LOOP ON HARD ERROR
4230							
4231	012046	013702	177776	4\$:	MOV	@#PSW,R2	;SAVE THE [PSW] IN R2
4232	012052	022702	000357		CMP	#357,R2	;WAS [PSW] = 357 ?
4233	012056	001404			BEQ	6\$;BR IF YES
4234							
4235	012060	010237	177776	5\$:	MOV	R2,@#PSW	;RESTORE THE ERROR PSW
4236	012064	000000			HALT		;RTI FAILED TO LOAD PSW
4237	012066	000754			BR	1\$;LOCK ON HARD ERROR
4238							
4239	012070	020605		6\$:	CMP	SP,R5	;DID SP GET UPDATED OK ?
4240	012072	001402			BEQ	TST161	;:BR IF YES
4241							
4242	012074	000000		7\$:	HALT		;RTI FAILED TO UPDATE THE SP
4243	012076	000750			BR	1\$;LOCK ON HARD ERROR
4244							

;*TEST 161 BASIC "RTI" TEST WITH N:C=1111

4245							
4246							
4247							
4248	012100				TST161:		
4249	012100	012700	000161		MOV	#161,R0	;:LOAD R0 WITH TEST NUMBER
4250	012104	010605			MOV	SP,R5	;SAVE THE SP IN R5
4251	012106	010506		1\$:	MOV	R5,SP	;RESET SP FOR ERROR LOOP
4252	012110	005046			CLR	-(SP)	;NEW PSW = 000000
4253	012112	012746	012130		MOV	#4\$,-(SP)	;NEW PC = 4\$

```
4254 012116 012737 000357 177776      MOV      #357,@#PSW      ;MAKE OLD PSW = 357
4255 012124 000240                      NOP                      ;SCOPE SYNC
4256
4257 012126 000002      2$:      RTI                      ;TEST THE RTI - GO TO 4$
4258
4259 012130 013702 177776      4$:      MOV      @#PSW,R2      ;GET THE PSW
4260 012134 022702 000000      CMP      #0,R2          ;WAS [PSW]=000
4261 012140 001404      BEQ      TST162         ;:BR IF YES
4262
4263 012142 010237 177776      3$:      MOV      R2,@#PSW      ;RESTORE ERROR PSW
4264 012146 000000      HALT                    ;RTI FAILED TO CLEAR PSW
4265 012150 000756      BR       1$             ;LOCK ON HARD ERROR
4266
4267
4268
4269
4270 012152
4271 012152 012700 000162      MOV      #162,R0        ;;LOAD R0 WITH TEST NUMBER
4272 012156 010605      MOV      SP,R5          ;SAVE THE SP
4273 012160 010506      1$:      MOV      R5,SP          ;RESET SP FOR ERROR LOOP
4274 012162 012737 012220 000020      MOV      #4$,@#20      ;SET UP IOT VECTOR
4275 012170 012737 000357 000022      MOV      #357,@#22
4276 012176 012766 177777 177776      MOV      #-1,-2(SP)    ;IOT SHOULD CHANGE -1 TO 0
4277 012204 005037 177776      CLR      @#PSW         ;MAKE [PSW] = 000
4278 012210 000257      CCC                    ;SCOPE SYNC
4279
4280 012212 000004      2$:      IOT                      ;TEST THE IOT
4281
4282 012214 000000      3$:      HALT                    ;IOT FAILED TO LOAD PC
4283 012216 000760      BR       1$             ;LOCK ON HARD ERROR
4284
4285 012220 013702 177776      4$:      MOV      @#PSW,R2      ;GET THE PSW
4286 012224 022702 000357      CMP      #357,R2       ;DID IOT LOAD A 357 ?
4287 012230 001404      BEQ      6$             ;BR IF YES
4288
4289 012232 010237 177776      5$:      MOV      R2,@#PSW      ;RESTORE ERROR PSW
4290 012236 000000      HALT                    ;IOT FAILED TO LOAD PSW
4291 012240 000747      BR       1$             ;LOCK ON HARD ERROR
4292
4293 012242 022726 012214      6$:      CMP      #3$, (SP)+    ;DID IOT SAVE OLD PC ?
4294 012246 001404      BEQ      8$             ;BR IF YES
4295
4296 012250 010237 177776      7$:      MOV      R2,@#PSW      ;RESTORE ERROR PSW
4297 012254 000000      HALT                    ;IOT FAILED TO SAVE OLD PC
4298 012256 000740      BR       1$             ;LOCK ON HARD ERROR
4299
4300 012260 005726      8$:      TST      (SP)+        ;DID IOT SAVE OLD PSW ?
4301 012262 001404      BEQ      TST163        ;:BR IF YES
4302
4303 012264 010237 177776      9$:      MOV      R2,@#PSW      ;RESTORE ERROR PSW
4304 012270 000000      HALT                    ;IOT FAILED TO SAVE OLD PSW
4305 012272 000732      BR       1$             ;LOCK ON HARD ERROR
```

```
:::*****  
:*TEST 162      BASIC "IOT" TEST -VERIFY LOADING PSW WITH 357  
:::*****
```

TST162:

```
:::*****  
:*TEST 163      BASIC "IOT" TEST - VERIFY LINKAGE TO SCOPE SERVICE  
:::*****
```

4306
4307
4308
4309

```

4310 012274
4311 012274 012700 000163
4312 012300 010605
4313 012302 010506
4314 012304 005037 063244
4315 012310 012737 061612 000020
4316 012316 005037 000022
4317 012322 000257
4318
4319 012324 000004
4320
4321 012326 005137 063244
4322 012332 001402
4323
4324 012334 000000
4325 012336 000761
4326
4327 012340 010506
4328
4329
4330
4331 012342
4332 012342 012700 000164
4333 012346 010605
4334 012350 010506
4335 012352 012737 012410 000020
4336 012360 012737 000357 000022
4337 012366 012766 177777 177776
4338 012374 005037 177776
4339 012400 000257
4340
4341 012402 000004
4342
4343 012404 000000
4344 012406 000760
4345
4346 012410 013702 177776
4347 012414 022702 000357
4348 012420 001404
4349
4350 012422 010237 177776
4351 012426 000000
4352 012430 000747
4353
4354 012432 022726 012404
4355 012436 001404
4356
4357 012440 010237 177776
4358 012444 000000
4359 012446 000740
4360
4361 012450 005726
4362 012452 001404
4363
4364 012454 010237 177776
4365 012460 000000
  
```

```

TST163:
MOV #163,R0 ;;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;;SAVE SP
1$: MOV R5,SP ;;RESET SP FOR ERROR LOOP
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$: MOV R5,SP ;;RESET SP IN CASE OF ERROR
;*****
;*TEST 164 BASIC "IOT" TEST -VERIFY LOADING PSW WITH 357
;*****
TST164:
MOV #164,R0 ;;LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;;SAVE THE SP
1$: MOV R5,SP ;;RESET SP FOR ERROR LOOP
MOV #4$,@#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$: CMP #3$,(SP)+ ;;DID IOT SAVE OLD PC ?
BEQ 8$ ;;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
  
```

```

4366 012462 000732          BR      1$          ;LOCK ON HARD ERROR
4367
4368
4369          ;:*****
4370          ;*TEST 165      BASIC IOT TEST - VERIFY LOADING PSW WITH 000
4371          ;:*****
4372          TST165:
4372 012464 012700 000165      MOV      #165,R0          ;;LOAD R0 WITH TEST NUMBER
4373 012470 010605          MOV      SP,R5          ;SAVE THE SP
4374 012472 010506      1$:  MOV      R5,SP          ;RESET SP FOR ERROR LOOP
4375 012474 012737 012520 000020  MOV      #4$,@#20        ;SET UP IOT VECTOR
4376 012502 005037 000022      CLR      @#22          ;
4377 012506 012737 000340 177776  MOV      #340,@#PSW      ;MAKE [PSW] = 340
4378 012514 000277          SCC          ;MAKE N:C=1111
4379
4380 012516 000004      2$:  IOT          ;TEST THE IOT
4381
4382 012520 013702 177776      4$:  MOV      @#PSW,R2      ;GET THE [PSW]
4383 012524 001404          BEQ      6$          ;BR IF [PSW] = 000
4384
4385 012526 010237 177776      3$:  MOV      R2,@#PSW      ;RESTORE THE ERROR PSW
4386 012532 000000          HALT          ;IOT FAILED TO CLEAR THE PSW
4387 012534 000756          BR      1$          ;LOCK ON HARD ERROR
4388
4389 012536 010506      6$:  MOV      R5,SP          ;RESET THE SP BEFORE CONTINUING
4390
4391          ;:*****
4392          ;*TEST 166      BASIC "TRAP" TEST - LINKAGE TO PRINT ROUTINE
4393          ;:*****
4394          TST166:
4395 012540 012700 000166      MOV      #166,R0          ;;LOAD R0 WITH TEST NUMBER
4396 012544 010605          MOV      SP,R5          ;SAVE THE SP
4397 012546 010506      1$:  MOV      R5,SP          ;RESET SP FOR ERROR LOOP
4398 012550 005037 063236      CLR      @#PRIFLG        ;INITIALIZE TEST FLAG
4399 012554 005037 000036      CLR      @#36          ;SET UP THE "TRAP" VECTOR
4400 012560 012737 062202 000034  MOV      #PRINA,@#34
4401 012566 000257          CCC          ;SCOPE SYNC
4402
4403 012570 104401      2$:  TYPE          ;TEST THE TRAP
4404
4405 012572 012737 063166 000034  MOV      #$TRAP,@#34      ;SETUP TRAP VECTOR
4406 012600 012737 000340 000036  MOV      #340,@#36
4407 012606 005137 063236      COM      @#PRIFLG        ;SHOULD MAKE [PRIFLG] = 000000
4408 012612 001402          BEQ      TST167        ;;BR IF IT DID
4409
4410 012614 000000      3$:  HALT          ;TRAP FAILED TO LINK TO PRINT SERV.
4411 012616 000753          BR      1$          ;LOCK ON HARD ERROR
4412
4413          ;:*****
4414          ;*TEST 167      BASIC "EMT" TEST - LINKAGE TO ERROR SERVICE
4415          ;:*****
4416          TST167:
4417 012620 012700 000167      MOV      #167,R0          ;;LOAD R0 WITH TEST NUMBER
4418 012624 010605          MOV      SP,R5          ;SAVE THE SP
4419 012626 010506      1$:  MOV      R5,SP          ;RESET SP FOR ERROR LOOP
4420 012630 012737 062040 000030  MOV      #ERRA,@#30      ;SET UP THE EMT VECTOR
4421 012636 005037 000032      CLR      @#32

```

```
4422 012642 005037 063240 CLR @#ERRFLG ;EMT SERVICE WILL COM [ERRFLG]
4423 012646 000257 CCC ;SCOPE SYNC
4424
4425 012650 104000 2$: ERROR ;TEST THE EMT
4426
4427 012652 005137 063240 COM @#ERRFLG ;DID EMT SERV. COM ERRFLG?
4428 012656 001402 BEQ TST170 ;:BR IF YES
4429
4430 012660 000000 3$: HALT ;EMT DID NOT LINK PROPERLY
4431 012662 000761 BR 1$ ;LOCK ON HARD ERROR
4432
4433 ;:*****
4434 ;*TEST 170 BASIC TEST OF RSVD INSTR. TRAP LINKAGE
4435 ;:*****
4436 012664 TST170:
4437 012670 012700 000170 MOV #170,R0 ;:LOAD R0 WITH TEST NUMBER
4438 012672 012737 061114 000010 MOV SP,R5 ;SAVE THE SP
4439 012700 012737 000340 000012 MOV #RSVTST,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
4440 012706 010506 1$: MOV #340,@#12
4441 012710 005037 063246 MOV R5,SP ;RESET SP FOR ERROR LOOP
4442 CLR @#RSVFLG ;INITIALIZE TEST FLAG THAT WILL GET
4443 012714 000257 CCC ;COMPLEMENTED BY TRAP SERVICE
4444 ;SCOPE SYNC
4445 012716 000007 2$: 000007 ;FORCE RSVD INSTR. TRAP
4446
4447 012720 005137 063246 COM @#RSVFLG ;TEST FLAG SHOULD GO TO 000000
4448 012724 001402 BEQ 4$ ;BR IF TRAP SPRUNG
4449
4450 012726 000000 3$: HALT ;RSVD INSTR. TRAP FAILED
4451 012730 000766 BR 1$ ;LOCK ON HARD ERROR
4452
4453 012732 012737 061122 000010 4$: MOV #RSERR,@#10 ;SET UP RSVD INSTR TRAP VECTOR TO POINT
4454 012740 012737 000340 000012 MOV #340,@#12 ;TO ERROR SERVICE ROUTINE
4455
4456 ;:*****
4457 ;*TEST 171 BASIC TEST OF BUS TIMEOUT TRAP LINKAGE
4458 ;:*****
4459 012746 TST171:
4460 012746 012700 000171 MOV #171,R0 ;:LOAD R0 WITH TEST NUMBER
4461 012752 010605 MOV SP,R5 ;SAVE THE SP
4462 012754 012737 061212 000004 MOV #BETST,@#4 ;SET UP THE BUS ERROR VECTOR
4463 012762 012737 000340 000006 MOV #340,@#6
4464 012770 010506 1$: MOV R5,SP ;RESET SP FOR ERROR LOOP
4465 012772 005037 063250 CLR @#BERFLG ;INITIALIZE TEST FLAG THAT WILL GET
4466 ;COMPLEMENTED BY TRAP SERVICE
4467 012776 000257 CCC ;SCOPE SYNC
4468
4469 013000 005737 177700 2$: TST @#177700 ;FORCE BUS TIMEOUT USING R0 ADDR.
4470
4471 013004 005137 063250 COM @#BERFLG ;TEST FLAG SHOULD GO TO 000000
4472 013010 001402 BEQ TST172 ;:BR IF TRAP SPRUNG
4473
4474 013012 000000 3$: HALT ;BUS ERROR FAILED TO SPRING TRAP
4475 013014 000765 BR 1$ ;LOCK ON HARD ERROR
4476
4477 ;:*****
```

4478
4479
4480 013016
4481 013016 012700 000172
4482 013022 005067 050264
4483 013026 005367 050260
4484 013032 001375
4485 013034 012737 013074 000004
4486 013042 012737 000340 000006
4487 013050 010605
4488 013052 010506
4489 013054 012702 177560
4490 013060 000257
4491
4492 013062 005722
4493 013064 005722
4494 013066 005722
4495 013070 005712
4496
4497 013072 000403
4498
4499 013074 005742
4500 013076 000000
4501 013100 000764
4502
4503 013102 012737 061220 000004
4504 013110 012737 000340 000006
4505
4506
4507
4508 013116
4509 013116 012700 000173
4510 013122 012702 177564
4511 013126 012704 000200
4512 013132 005012
4513 013134 005001
4514 013136 000257
4515
4516 013140 020412
4517
4518 013142 001405
4519 013144 005301
4520 013146 001374
4521
4522 013150 011203
4523 013152 000000
4524 013154 000766
4525
4526
4527
4528
4529 013156
4530 013156 012700 000174
4531 013162 012702 177564
4532 013166 012704 000200
4533 013172 005012

```
;*TEST 172 BASIC TEST FOR ACCESSING DL11 REGISTERS
:*****
TST172:
MOV #172,R0 ;;LOAD R0 WITH TEST NUMBER
CLR MBUF0 ;;INIT STALL COUNTER
11$: DEC MBUF0 ;;COUNT THE TIMER
BNE 11$ ;;BR IF NO TIMEOUT
MOV #3$,@#4 ;SET UP BUS TIMEOUT VECTOR
MOV #340,@#6
MOV SP,R5 ;;SAVE TH SP
1$: MOV R5,SP ;;RESET SP FOR ERROR LOOP
MOV #RCSR,R2 ;;[R2] = STARTING DL11 ADDR.
CCC ;;SCOPE SYNC

2$: TST (R2)+ ;;REFERENCE DL11 - RCSR
TST (R2)+ ;;REFERENCE DL11 - RDBR
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$: MOV #BERR,@#4 ;;SET UP BUS ERROR VECTOR TO POINT
MOV #340,@#6 ;;TO ERROR SERVICE ROUTINE
:*****
;*TEST 173 BASIC TEST OF DL11 - XCSR - READY(1)
:*****
TST173:
MOV #173,R0 ;;LOAD R0 WITH TEST NUMBER
MOV #XCSR,R2 ;;DEST ADDR = XCSR
MOV #200,R4 ;;RESULT S / B = 200
1$: CLR (R2) ;;CLEAR [DEST]
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
1$: CLR (R2) ;;CLEAR MAINT. BIT
```


4534 013174 000257
4535
4536 013176 020412
4537
4538 013200 001403
4539
4540 013202 011203
4541 013204 000000
4542 013206 000771
4543

CCC ;SCOPE SYNC
2\$: CMP R4,(R2) ;TEST MAINT(0)
BEQ TST175 ;:BR IF MAINT BIT CLEAR
3\$: MOV (R2),R3 ;GET THE WAS DATA
HALT ;CAN'T CLEAR MAINT BIT
BR 1\$;LOCK ON HARD ERROR

4544
4545
4546
4547 013210
4548 013210 012700 000175
4549 013214 012702 177564
4550 013220 012704 000204
4551 013224 012712 000004
4552 013230 000257
4553
4554 013232 020412
4555
4556 013234 001403
4557
4558 013236 011203
4559 013240 000000
4560 013242 000770
4561

;*TEST 175 BASIC TEST OF DL11 XCSR - MAINT BIT = 1

TST175:
MOV #175,R0 ;:LOAD R0 WITH TEST NUMBER
MOV #XCSR,R2 ;DEST ADDR = XCSR
MOV #204,R4 ;RESULT S / B = 204
1\$: MOV #4,(R2) ;SET THE MAINT. BIT
CCC ;SCOPE SYNC
2\$: CMP R4,(R2) ;TEST MAINT.(1)
BEQ TST176 ;:BR IF IT WAS
3\$: MOV (R2),R3 ;GET THE WAS DATA
HALT ;CAN'T SET MAINT BIT IN XCSR
BR 1\$;LOCK O HARD ERROR

4562
4563
4564
4565
4566
4567
4568
4569
4570
4571
4572
4573 013244
4574 013244 012700 000176
4575 013250 012702 177560
4576 013254 105762 000002
4577 013260 105762 000002
4578 013264 012703 063266
4579 013270 012704 063256
4580 013274 012705 000010
4581 013300 012762 000004 000004
4582
4583 013306 005001
4584 013310 112462 000006
4585 013314 105712
4586 013316 100404
4587 013320 005301
4588 013322 001374
4589

;*TEST 176 BASIC DL11 OUT / IN ECHO TEST (MAINT MODE)
;THIS ROUTINE USES THE MAINTENANCE MODE FEATURE OF THE DL11 TO
;TURN AROUND A STRING OF 8 CHARACTERS TO THE DL11. THIS STRING CONSISTS
;OF ALTERNATING NULL / DELETE CHARS WHICH ARE NON PRINTING. THE 8 CHARS
;ARE OUTPUT THEN READ BACK INTO A CORE BUFFER AND THEN THE INPUT AND
;OUTPUT CORE BUFFERS ARE CHECKED FOR EQUIVALENCE. IF AN ERROR IS DET-
;ECTED DURING THE COMPARISON THE ROUTINE HALTS WITH THE WAS AND S / B
;DATA IN R3 AND R4 RESPECTFULLY. A TIMER IS EMPLOYED TO PREVENT THE
;TEST FROM HANGING IF RECEIVER DONE DOES NOT RESPOND.

TST176:
MOV #176,R0 ;:LOAD R0 WITH TEST NUMBER
6\$: MOV #RCSR,R2 ;R2 POINTS TO DL11 - START ADDR
TSTB 2(R2) ;REFERENCE DL11 INPUT DATA BUFFER TWICE
TSTB 2(R2) ;TO FLUSH RCVR "DONE" BIT
MOV #IBUF,R3 ;R3 POINTS TO CORE INPUT BUFFER
MOV #OBUF,R4 ;R4 POINTS TO CORE OUTPUT BUFFER
MOV #10,R5 ;R5 WILL COUNT 8 CHARS OUTPUT
MOV #4,4(R2) ;TURN ON MAINT MODE
1\$: CLR R1 ;R1 USED AS TIMEOUT COUNTER
MOVB (R4)+,6(R2) ;LOAD OUTPUT BUFFER IN DL11
2\$: TSTB (R2) ;RECEIVER DONE SET ?
BMI 3\$;BR IF YES
DEC R1 ;COUNT THE TIMER
BNE 2\$;BR IF NO TIMEOUT

4590	013324	000000			HALT				;DL11 FAILED TO RESPOND IN TIME
4591	013326	000750			BR	6\$;LOCK ON HARD ERROR
4592									
4593	013330	116223	000002	3\$:	MOVB	2(R2),(R3)+			;READ THE DL11 INPUT BUFFER INTO CORE
4594	013334	005305			DEC	R5			;COUNT ONE CHAR
4595	013336	001363			BNE	1\$;BR IF NOT DONE 8 CHARS
4596									
4597	013340	005062	000004		CLR	4(R2)			;TURN OFF MAINT. MODE
4598	013344	012705	000010		MOV	#10,R5			;RESET CHAR COUNTER
4599	013350	012703	063266		MOV	#1BUF,R3			;RESET INBUF POINTER
4600	013354	012704	063256		MOV	#OBUF,R4			;RESET OUTBUF POINTER
4601									
4602	013360	122324		4\$:	CMPB	(R3)+,(R4)+			;INPUT = OUTPUT ??
4603	013362	001003			BNE	5\$;BR IF NOT
4604	013364	005305			DEC	R5			;COUNT ONE CHECKED
4605	013366	001374			BNE	4\$;BR UNTIL 8 DONE
4606	013370	000410			BR	CITST			;GO TO NEXT TEST
4607									
4608	013372	114303		5\$:	MOVB	-(R3),R3			;WAS DATA IN R3 [BITS 7:0]
4609	013374	114404			MOVB	-(R4),R4			;S / B DATA IN R4 [BITS 7:0]
4610	013376	042703	177400		BIC	#177400,R3			;STRIP OFF BITS <15:08>
4611	013402	042704	177400		BIC	#177400,R4			;..
4612	013406	000000			HALT				;RECEIVED DATA NOT EQUAL TO OUTPUT DATA
4613	013410	000717			BR	6\$;LOCK ON HARD ERROR

```
4614 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4615 ;////////////////COMPREHENSIVE INSTRUCTION TESTS////////////////////
4616 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4617
4618 013412 012737 061260 000020 CITST: MOV    $$SCOPE,@#20    ;SET UP IOT VECTOR
4619 013420 005037 000022          CLR    @#22
4620 013424 012737 061620 000030          MOV    $ERROR,@#30    ;SET UP EMT VECTOR
4621 013432 012737 000340 000032          MOV    #340,@#32
4622 013440 012737 063166 000034          MOV    $TRAP,@#34    ;SET UP TRAP VECTOR
4623 013446 012737 000340 000036          MOV    #340,@#36
4624 013454 012737 060664 000024          MOV    $PWRDN,@#24   ;SET UP POWER FAIL VECTOR
4625 013462 012737 000340 000026          MOV    #340,@#26
4626 013470 105737 001141          TSTB   @#$ENVM       ;DO NOT SIZE BIT SET?
4627 013474 100003          BPL    3$           ;BR IF NOT - USE HARDWARE SWITCH REG
4628 013476 012737 001142 001040          MOV    $SWREG,@#SWR  ;USE APT SWITCH REG.
4629 013504 032777 010000 165326 3$: BIT    #SW12,@SWR   ;INHIBIT PRINTING INTRO. I.D. MESSAGE?
4630 013512 001007          BNE    1$           ;BR IF YES
4631 013514 005737 063254          TST    @#ONCE       ;FIRST TIME INTO "CIT" TESTS ?
4632 013520 001004          BNE    1$           ;BR IF NOT - PRINT ID ONLY ONCE
4633 013522 005137 063254          COM    @#ONCE       ;SET FLAG TO INHIBIT PRINTING AGAIN
4634 013526 104401          TYPE
4635 013530 065141          IDENT1            ;IDENTIFY THIS PROGRAM
4636 013532 005037 177776          CLR    @#PSW        ;ADDR OF THE ID MESSAGE
4637 013536 012737 003316 001006 1$: MOV    #TST0,@#$LPADR ;SET CPU PRIORITY TO LEVEL 000
4638 013544 012737 000040 001110          MOV    #40,@#$TIMES ;INITIALIZE SCOPE LOOP RETURN
4639 013552 010037 001124          MOV    R0,@#$TESTN  ;ITERATE ON BIT SECTION 32 TIMES
4640                                     ;PREVENT MISSED TEST ERROR ON
4641                                     ;FIRST SCOPE CALL
```

```
4642 ;*****
4643 ;*TEST 177      BCC TEST WITH C=1
4644 ;*****
4645 TST177:
4646 013556 000004          SCOPE              ;CALL THE SCOPE LOOP UTILITY
4647 013560 012700 000177          MOV    #177,R0     ;:LOAD R0 WITH TEST NUMBER
4648 013564 013701 013572          MOV    @#2$,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
4649 013570 000261          SEC
4650                                     ;MAKE C=1
4651 013572 103001          2$: BCC    3$           ;TEST THE BCC, IT SHOULDN'T BR
4652 013574 000401          BR     TST200      ;:GO TO SCOPE EXIT
4653
4654 013576 104005          3$: ERROR 5        ;BCC FAILED
4655
```

```
4656 ;*****
4657 ;*TEST 200      BCC TEST WITH C=0
4658 ;*****
4659 TST200:
4660 013600 000004          SCOPE              ;CALL THE SCOPE LOOP UTILITY
4661 013602 012700 000200          MOV    #200,R0     ;:LOAD R0 WITH TEST NUMBER
4662 013606 013701 013614          MOV    @#2$,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
4663 013612 000241          CLC
4664                                     ;MAKE C=0
4665
4666 013614          2$: BCC    TST201      ;:TEST THE BCC-IT SHOULD BR
4667
4668 013616 104005          3$: ERROR 5        ;BCC FAILED
4669
```

4670
4671
4672
4673 013620
4674 013620 000004
4675 013622 012700 000201
4676 013626 013701 013644
4677 013632 012704 000017
4678 013636 012702 177776
4679
4680 013642 000277
4681
4682 013644 103004
4683
4684 013646 013703 177776
4685 013652 020304
4686 013654 001401
4687
4688 013656 104001
4689
4690
4691
4692
4693 013660
4694 013660 000004
4695 013662 012700 000202
4696 013666 013701 013704
4697 013672 012704 000017
4698 013676 012702 177776
4699
4700 013702 000277
4701
4702 013704 000401
4703
4704 013706 104005
4705
4706 013710 013703 177776
4707 013714 020304
4708 013716 001401
4709
4710 013720 104001
4711
4712
4713
4714
4715 013722
4716 013722 000004
4717 013724 012700 000203
4718 013730 013701 013744
4719 013734 005004
4720 013736 012702 177776
4721
4722 013742 000257
4723
4724 013744 103404
4725

```
*****  
*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 @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #17,R4 ;:S/B PSW  
MOV #PSW,R2 ;:DEST = PSW FOR ERROR CALL  
  
SCC ;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 @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #17,R4 ;:S/B PSW  
MOV #PSW,R2 ;:DEST = PSW FOR ERROR CALL  
  
SCC ;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 @#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$: BCS 3$ ;TEST THE BCS-IT SHOULDN'T BR
```

4726 013746 013703 177776
4727 013752 005703
4728 013754 001401
4729
4730 013756 104001
4731
4732
4733
4734
4735 013760
4736 013760 000004
4737 013762 012700 000204
4738 013766 013701 014002
4739 013772 005004
4740 013774 012702 177776
4741
4742 014000 000257
4743
4744 014002 000401
4745
4746 014004 104005
4747
4748 014006 013703 177776
4749 014012 005703
4750 014014 001401
4751
4752 014016 104001
4753
4754
4755
4756
4757 014020
4758 014020 000004
4759 014022 012700 000205
4760 014026 013701 014034
4761 014032 000257
4762
4763 014034 003401
4764 014036 000401
4765
4766 014040 104005
4767
4768
4769
4770
4771 014042
4772 014042 000004
4773 014044 012700 000206
4774 014050 013701 014060
4775 014054 000257
4776 014056 000264
4777
4778 014060
4779 014060 003401
4780
4781 014062 104005

```
MOV @#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
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
```

4782
4783
4784
4785
4786 014064
4787 014064 000004
4788 014066 012700 000207
4789 014072 013701 014102
4790 014076 000257
4791 014100 000262
4792
4793 014102
4794 014102 003401
4795
4796 014104 104005
4797
4798
4799
4800
4801 014106
4802 014106 000004
4803 014110 012700 000210
4804 014114 013701 014124
4805 014120 000257
4806 014122 000270
4807
4808 014124
4809 014124 003401
4810
4811 014126 104005
4812
4813
4814
4815
4816 014130
4817 014130 000004
4818 014132 012700 000211
4819 014136 013701 014146
4820 014142 000257
4821 014144 000272
4822
4823 014146 003401
4824 014150 000401
4825
4826 014152 104005
4827
4828
4829
4830
4831 014154
4832 014154 000004
4833 014156 012700 000212
4834 014162 013701 014170
4835 014166 000257
4836
4837 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  
CCC ;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  
CCC ;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  
CCC ;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  
CCC ;MAKE Z,C = 00  
  
2$: BLOS 3$ ;:TEST THE BLOS-IT SHOULDN'T BR
```

4838 014172 000401
4839
4840 014174 104005
4841
4842
4843
4844
4845 014176
4846 014176 000004
4847 014200 012700 000213
4848 014204 013701 014214
4849 014210 000257
4850 014212 000261
4851
4852 014214
4853 014214 101401
4854
4855 014216 104005
4856
4857
4858
4859
4860 014220
4861 014220 000004
4862 014222 012700 000214
4863 014226 013701 014236
4864 014232 000257
4865 014234 000264
4866
4867 014236
4868 014236 101401
4869
4870 014240 104005
4871
4872
4873
4874
4875 014242
4876 014242 000004
4877 014244 012700 000215
4878 014250 013701 014260
4879 014254 000257
4880 014256 000265
4881
4882 014260
4883 014260 101401
4884
4885 014262 104005
4886
4887
4888
4889
4890 014264
4891 014264 000004
4892 014266 012700 000216
4893 014272 013701 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 R0 WITH TEST NUMBER
MOV @#2\$,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 R0 WITH TEST NUMBER
MOV @#2\$,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 R0 WITH TEST NUMBER
MOV @#2\$,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 R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

```
4894 014276 005004
4895 014300 012703 177777
4896 014304 000257
4897 014306 000263
4898
4899 014310 006703 2$: SXT R3 ;TEST THE SXT
4900
4901 014312 100403
4902 014314 001002
4903 014316 102401
4904 014320 103401
4905
4906 014322 104002 3$: ERROR 2 ;SXT FAILED TO ALTER CODES PROPERLY
4907
4908 014324 005703
4909 014326 001401 4$: TST R3 ;DID RESULT = 0?
      BEQ TST217 ;:BR IF IT DID
4910
4911 014330 104002 5$: ERROR 2 ;SXT DELIVERED WRONG RESULT TO R3
4912
4913
4914
4915
4916 014332
4917 014332 000004
4918 014334 012700 000217
4919 014340 013701 014366
4920
4921 014344 032737 000020 063234 .SBTTL USER CONTROLLED BREAKPOINT -- BIT4
4922 014352 001401
4923 014354 000000
4924
4925 014356 005004
4926 014360 012703 177777
4927 014364 000257
4928
4929 014366 006703 2$: SXT R3 ;TEST THE SXT
4930 014370 103001
      BCC TST220 ;:BR IF "C" STILL CLEAR
4931
4932 014372 104002 3$: ERROR 2 ;SXT AFFECTED "C" BIT
4933
4934
4935
4936
4937 014374
4938 014374 000004
4939 014376 012700 000220
4940 014402 013701 014416
4941 014406 012704 177777
4942 014412 005003
4943 014414 000277
4944
4945 014416 006703 2$: SXT R3 ;TEST THE SXT
4946
4947 014420 100003
4948 014422 001402
      BPL 3$
      BEQ 3$ ;N:C = 1001?
      BVS 3$
4949 014424 102401
```


4950 014426 103401
 4951
 4952 014430 104002
 4953
 4954 014432 010305
 4955 014434 005105
 4956 014436 001401
 4957
 4958 014440 104002
 4959
 4960
 4961
 4962
 4963 014442
 4964 014442 000004
 4965 014444 012700 000221
 4966 014450 013701 014466
 4967 014454 012704 177777
 4968 014460 005003
 4969 014462 000257
 4970 014464 000276
 4971
 4972 014466 006703
 4973 014470 103001
 4974
 4975 014472 104002
 4976
 4977
 4978
 4979
 4980 014474
 4981 014474 000004
 4982 014476 012700 000222
 4983 014502 013701 014524
 4984 014506 012702 063312
 4985 014512 005004
 4986 014514 012712 177777
 4987 014520 000257
 4988 014522 000263
 4989
 4990 014524 006712
 4991
 4992 014526 100403
 4993 014530 001002
 4994 014532 102401
 4995 014534 103401
 4996
 4997 014536 104001
 4998
 4999 014540 005712
 5000 014542 001401
 5001
 5002 014544 104001
 5003
 5004 014546 012702 063312
 5005 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 R0 WITH TEST NUMBER
 MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #-1,R4 ;RESULT S / B = 177777
 CLR R3 ;INITIAL DEST OP = 0
 CCC ;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 R0 WITH TEST NUMBER
 MOV @#2\$,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 [DEST] = 177777
 CCC ;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 [DEST]
 11\$: MOV #MBUF0,R2 ;DEST ADDR = MBUF0
 MOV @#12\$,R1 ;LOAD R1 WITH TEST INSTR WORD

```
5006 014556 012712 177777      MOV      #-1,(R2)      ;INITIAL [DEST] = 177777
5007 014562 000257              CCC                  ;CLEAR CODES
5008 014564 000263              263                  ;MAKE N:C = 0011
5009
5010 014566 006722      12$:  SXT      (R2)+      ;TEST SXT - DM2
5011
5012 014570 100403              BMI      7$           ;N:C = 0101 ?
5013 014572 001002              BNE      7$
5014 014574 102401              BVS      7$
5015 014576 103401              BCS      6$
5016
5017 014600 104001      7$:  ERROR  1           ;SXT FAILED TO ALTER CODES PROPERLY
5018
5019 014602 005737 063312      6$:  TST      @#MBUFO    ;DID RESULT GET ZEROED ?
5020 014606 001401              BEQ      8$           ;BR IF YES
5021
5022 014610 104001      9$:  ERROR  1           ;SXT FAILED TO ZERO [DEST]
5023
5024 014612 020227 063314      8$:  CMP      R2,#MBUFO+2 ;WAS IT REALLY MODE 2 ?
5025 014616 001401              BEQ      TST223      ;:BR IF YES
5026
5027 014620 104001              ERROR  1           ;SXT FAILED TO AUTO INCREMENT
5028
5029
5030
5031
5032
5033
5034
5035
5036
5037
5038
5039
5040
5041
5042
5043
5044
5045
5046
5047
5048
5049
5050
5051
5052
5053
5054
5055
5056
5057
5058
5059
5060
5061
```

```
*****
;*TEST 223      SXT MODE 1 TEST WITH N = 0 AND C = 0
*****
TST223:
      SCOPE                  ;CALL THE SCOPE LOOP UTILITY
      MOV      #223,R0       ;:LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
      CLR      R4            ;RESULT S / B = 0
      MOV      #MBUFO,R2     ;R2 POINTS TO DEST OP
      MOV      #-1,(R2)     ;INITIAL [DEST] = 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 R0 WITH TEST NUMBER
      MOV      @#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #-1,R4       ;RESULT S / B = 177777
      MOV      #MBUFO,R2     ;R2 POINTS TO DEST OP
      CLR      (R2)         ;INITIAL [DEST] = 0
      SCC                  ;MAKE N:C = 1111

2$:  SXT      (R2)          ;TEST THE SXT

      BPL      3$
      BEQ      3$           ;N:C = 1001?
```

5062 014712 102401
5063 014714 103401
5064
5065 014716 104001
5066
5067 014720 021204
5068 014722 001401
5069
5070 014724 104001
5071
5072
5073
5074
5075 014726
5076 014726 000004
5077 014730 012700 000225
5078 014734 013701 014756
5079 014740 012704 177777
5080 014744 012702 063312
5081 014750 005012
5082 014752 000257
5083 014754 000276
5084
5085 014756 006712
5086 014760 103001
5087
5088 014762 104001
5089
5090
5091
5092
5093 014764
5094 014764 000004
5095 014766 012700 000226
5096 014772 013701 015012
5097 014776 012704 177400
5098 015002 012703 000377
5099 015006 000257
5100 015010 000273
5101
5102 015012 000303
5103
5104 015014 100403
5105 015016 001002
5106 015020 102401
5107 015022 103001
5108
5109 015024 104002
5110
5111 015026 020403
5112 015030 001401
5113
5114 015032 104002
5115
5116
5117

BVS 3\$
BCS 4\$
3\$: ERROR 1 ;SXT FAILED TO ALTER CODES PROPERLY
4\$: CMP (R2),R4 ;RESULT = 177777?
BEQ TST225 ;:BR IF YES
5\$: ERROR 1 ;SXT DELIVERED WRONG RESULT
:*****
:*TEST 225 SXT MODE 1 TEST WITH N = 1 AND C = 0
:*****
TST225:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #225,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S / B = 177777
MOV #MBUFO,R2 ;R2 POINTS TO DEST OP
CLR (R2) ;INITIAL [DEST] = 0
CCC ;CLEAR FLAGS
276 ;MAKE N:C = 1110
2\$: SXT (R2) ;TEST THE SXT
BCC TST226 ;:BR IF "C" UNAFFECTED
3\$: ERROR 1 ;SXT SET THE "C" BIT
:*****
:*TEST 226 SWAB MODE 0 TEST WITH POS. RESULT
:*****
TST226:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #226,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,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 TST227 ;:BR IF YES
5\$: ERROR 2 ;SWAB DELIVERED WRONG RESULT
:*****
:*TEST 227 SWAB MODE 0 TEST WITH NEG. RESULT

5118
5119 015034
5120 015034 000004
5121 015036 012700 000227
5122 015042 013701 015062
5123 015046 012704 000377
5124 015052 012703 177400
5125 015056 000257
5126 015060 000267
5127
5128 015062 000303
5129
5130 015064 100003
5131 015066 001402
5132 015070 102401
5133 015072 103001
5134
5135 015074 104002
5136
5137 015076 020403
5138 015100 001401
5139
5140 015102 104002
5141
5142
5143
5144
5145 015104
5146 015104 000004
5147 015106 012700 000230
5148 015112 013701 015136
5149 015116 012704 177400
5150 015122 012702 063312
5151 015126 012712 000377
5152 015132 000257
5153 015134 000273
5154
5155 015136 000312
5156
5157 015140 100403
5158 015142 001002
5159 015144 102401
5160 015146 103001
5161
5162 015150 104001
5163
5164 015152 020412
5165 015154 001401
5166
5167 015156 104001
5168
5169 015160 013701 015200
5170 015164 012702 063312
5171 015170 012712 000377
5172 015174 000257
5173 015176 000273

```
.....  
TST227:  
      SCOPE                ;CALL THE SCOPE LOOP UTILITY  
      MOV #227,R0          ;:LOAD R0 WITH TEST NUMBER  
      MOV @#2$,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 TST230         ;:BR IF OK  
  
5$:   ERROR 2              ;:SWAB DELIVERED WRONG RESULT  
  
.....  
*TEST 230 SWAB MODE 1 AND 2 TEST WITH POS. RESULT  
.....  
TST230:  
      SCOPE                ;CALL THE SCOPE LOOP UTILITY  
      MOV #230,R0          ;:LOAD R0 WITH TEST NUMBER  
      MOV @#2$,R1         ;:LOAD R1 WITH TEST INSTRUCTION WORD  
      MOV #177400,R4      ;:RESULT S / B = 177400  
      MOV #MBUFO,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  
  
      ERROR 1              ;:SWAB DELIVERED WRONG RESULT  
  
5$:   MOV @#20$,R1        ;:LOAD R1 WITH TEST INSTR. WORD  
      MOV #MBUFO,R2       ;:R2 POINTS TO DEST OP  
      MOV #377,(R2)       ;:[DEST] = 000377  
      CCC                 ;:CLEAR FLAGS  
      273                 ;:MAKE N:C = 1011
```

```
5174
5175 015200 000322      20$: SWAB (R2)+ ;TEST THE SWAB - DM2
5176
5177 015202 100403      BMI 7$ ;N:C = 0100
5178 015204 001002      BNE 7$
5179 015206 102401      BVS 7$
5180 015210 103001      BCC 6$
5181
5182 015212 104001      7$: ERROR 1 ;SWAB FAILED TO SET CODES PROPERLY
5183
5184 015214 020437 063312 6$: CMP R4,@#M0000 ;CORRECT RESULT ?
5185 015220 001401      BEQ 8$ ;BR IF YES
5186
5187 015222 104001      9$: ERROR 1 ;SWAB DELIVERED THE WRONG RESULT
5188
5189 015224 020227 063314 8$: CMP R2,#M0000+2 ;DID AUTO INCREMENT OCCUR ?
5190 015230 001401      BEQ TST231 ;:BR IF YES
5191
5192 015232 104001      ERROR 1 ;SWAB FAILED TO AUTO INC REG.
5193
5194
5195
5196
5197 015234
5198 015234 000004
5199 015236 012700 000231
5200 015242 013701 015266
5201 015246 012704 000377
5202 015252 012702 063312
5203 015256 012712 177400
5204 015262 000257
5205 015264 000267
5206
5207 015266 000312      2$: SWAB (R2) ;TEST THE SWAB
5208
5209 015270 100003      BPL 3$
5210 015272 001402      BEQ 3$ ;N:C = 1000?
5211 015274 102401      BVS 3$
5212 015276 103001      BCC 4$
5213
5214 015300 104001      3$: ERROR 1 ;SWAB FAILED TO ALTER CODES PROPERLY
5215
5216 015302 020412      4$: CMP R4,(R2) ;CORRECT RESULT?
5217 015304 001401      BEQ TST232 ;:BR IF YES
5218
5219 015306 104001      5$: ERROR 1 ;SWAB DELIVERED WRONG RESULT
5220
5221
5222
5223
5224 015310
5225 015310 000004
5226 015312 012700 000232
5227 015316 013701 015332
5228 015322 005004
5229 015324 005003
```

```
5230 015326 000257          CCC          ;CLEAR FLAGS
5231 015330 000273          273          ;MAKE N:C = 1011
5232
5233 015332 005403          2$: NEG      R3          ;TEST THE NEG
5234
5235 015334 100403          BMI      3$          ;
5236 015336 001002          BNE      3$          ;N:C = 0100 ONLY "Z" SET?
5237 015340 102401          BVS      3$
5238 015342 103001          BCC      4$
5239
5240 015344 104002          3$: ERROR  2          ;NEG FAILED TO ALTER CODES PROPERLY
5241
5242 015346 020304          4$: CMP      R3,R4          ;WAS RESULT = 0
5243 015350 001401          BEQ      TST233        ;;BR IF YES
5244
5245 015352 104002          5$: ERROR  2          ;NEG DELIVERED WRONG RESULT
5246
```

```
::*****
:*TEST 233      NEG MODE 0 TEST : [DEST] LT 0
*****
TST233:
```

```
5250 015354
5251 015354 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5252 015356 012700 000233  MOV      #233,R0          ;;LOAD R0 WITH TEST NUMBER
5253 015362 013701 015402  MOV      @#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
5254 015366 012704 000002  MOV      #2,R4           ;RESULT S / B = 2
5255 015372 012703 177776  MOV      #-2,R3          ;INITIAL [DEST] = 177776
5256 015376 000257          CCC          ;CLEAR FLAGS
5257 015400 000276          276          ;MAKE N:C = 1110
5258
5259 015402 005403          2$: NEG      R3          ;TEST THE NEG
5260
5261 015404 100403          BMI      3$          ;
5262 015406 001402          BEQ      3$          ;N:C = 0001?
5263 015410 102401          BVS      3$
5264 015412 103401          BCS      4$
5265
5266 015414 104002          3$: ERROR  2          ;NEG FAILED TO ALTER CODES PROPERLY
5267
5268 015416 020304          4$: CMP      R3,R4          ;RESULT = 2?
5269 015420 001401          BEQ      TST234        ;;BR IF YES
5270
5271 015422 104002          5$: ERROR  2          ;NEG DELIVERED WRONG RESULT
5272
```

```
::*****
:*TEST 234      NEG MODE 0 TEST : [DEST] = 100000 (8)
*****
TST234:
```

```
5273
5274
5275
5276 015424
5277 015424 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5278 015426 012700 000234  MOV      #234,R0          ;;LOAD R0 WITH TEST NUMBER
5279 015432 013701 015450  MOV      @#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD
5280 015436 012704 100000  MOV      #100000,R4       ;RESULT S / B = 100000
5281 015442 010403          MOV      R4,R3          ;INITIAL [DEST] = 100000
5282 015444 000257          CCC          ;CLEAR FLAGS
5283 015446 000264          SEZ          ;MAKE N:C = 01000
5284
5285 015450 005403          2$: NEG      R3          ;TEST THE NEG
```

```
5286  
5287 015452 100003          BPL      3$  
5288 015454 001402          BEQ      3$          ;N:C = 1011?  
5289 015456 102001          BVC      3$  
5290 015460 103401          BCS      4$  
5291  
5292 015462 104002          3$:      ERROR    2          ;NEG FAILED TO ALTER CODES PROPERLY  
5293  
5294 015464 020304          4$:      CMP      R3,R4          ;RESULT STILL 100000?  
5295 015466 001401          BEQ      TST235        ;:BR IF YES  
5296  
5297 015470 104002          5$:      ERROR    2          ;NEG DELIVERED WRONG RESULT  
5298  
5299
```

```
::*****  
:*TEST 235      NEG MODE 1 TEST : [DEST] = 0  
:*****  
TST235:
```

```
5302 015472  
5303 015472 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY  
5304 015474 012700 000235  MOV      #235,R0          ;:LOAD R0 WITH TEST NUMBER  
5305 015500 013701 015520  MOV      @#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD  
5306 015504 012702 063312  MOV      #MBUFO,R2        ;R2 POINTS TO DEST OP  
5307 015510 005004          CLR      R4              ;RESULT S / B = 0  
5308 015512 005012          CLR      (R2)            ;INITIAL [DEST] = 0  
5309 015514 000257          CCC          ;CLEAR FLAGS  
5310 015516 000273          273          ;MAKE N:C = 1011
```

```
5311  
5312 015520 005412          2$:      NEG      (R2)          ;TEST THE NEG  
5313  
5314 015522 100403          BMI      3$  
5315 015524 001002          BNE      3$          ;N:C = 0100?  
5316 015526 102401          BVS      3$  
5317 015530 103001          BCC      4$  
5318  
5319 015532 104001          3$:      ERROR    1          ;NEG FAILED TO ALTER CODES PROPERLY  
5320  
5321 015534 021204          4$:      CMP      (R2),R4        ;RESULT = 0?  
5322 015536 001401          BEQ      TST236        ;:BR IF YES  
5323  
5324 015540 104001          5$:      ERROR    1          ;NEG DELIVERED WRONG RESULT  
5325
```

```
::*****  
:*TEST 236      NEG MODE 1 TEST : [DEST] GT 0  
:*****  
TST236:
```

```
5329 015542  
5330 015542 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY  
5331 015544 012700 000236  MOV      #236,R0          ;:LOAD R0 WITH TEST NUMBER  
5332 015550 013701 015574  MOV      @#2$,R1          ;LOAD R1 WITH TEST INSTRUCTION WORD  
5333 015554 012702 063312  MOV      #MBUFO,R2        ;R2 POINTS TO DEST OP  
5334 015560 012704 177776  MOV      #-2,R4           ;RESULT S / B = 177776  
5335 015564 012712 000002  MOV      #2,(R2)          ;INITIAL [DEST] = 2  
5336 015570 000257          CCC          ;CLEAR FLAGS  
5337 015572 000266          266          ;MAKE N:C = 0110  
5338  
5339 015574 005412          2$:      NEG      (2)          ;TEST THE NEG  
5340  
5341 015576 100003          BPL      3$
```

5342 015600 001402
5343 015602 10240i
5344 015604 103401
5345
5346 015606 104001
5347
5348 015610 021204
5349 015612 001401
5350
5351 015614 104001
5352
5353
5354
5355
5356 015616
5357 015616 000004
5358 015620 012700 000237
5359 015624 013701 015650
5360 015630 012702 063312
5361 015634 012704 000002
5362 015640 012712 177776
5363 015644 000257
5364 015646 000276
5365
5366 015650 005412
5367
5368 015652 100403
5369 015654 001402
5370 015656 102401
5371 015660 103401
5372
5373 015662 104001
5374
5375 015664 021204
5376 015666 001401
5377
5378 015670 104001
5379
5380
5381
5382
5383 015672
5384 015672 000004
5385 015674 012700 000240
5386 015700 013701 015722
5387 015704 012702 063312
5388 015710 012704 100000
5389 015714 010412
5390 015716 000257
5391 015720 000264
5392
5393 015722 005412
5394
5395 015724 100003
5396 015726 001402
5397 015730 102001

BEQ 3\$;N:C = 1001?
BVS 3\$
BCS 4\$
3\$: ERROR 1 ;NEG FAILED TO ALTER CODES PROPERLY
4\$: CMP (R2),R4 ;CORRECT RESULT?
BEQ 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 R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,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 R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,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\$

5398 015732 103401
5399
5400 015734 104001
5401
5402 015736 021204
5403 015740 001401
5404
5405 015742 104001
5406
5407
5408
5409
5410 015744
5411 015744 000004
5412 015746 012700 000241
5413 015752 013701 015772
5414 015756 012704 052525
5415 015762 012703 125252
5416 015766 000257
5417 015770 000276
5418
5419 015772 006003
5420
5421 015774 100403
5422 015776 001402
5423 016000 102401
5424 016002 103001
5425
5426 016004 104002
5427
5428 016006 020403
5429 016010 001401
5430
5431 016012 104002
5432
5433
5434
5435
5436 016014
5437 016014 000004
5438 016016 012700 000242
5439 016022 013701 016040
5440 016026 005004
5441 016030 012703 000001
5442 016034 000257
5443 016036 000270
5444
5445 016040 006003
5446
5447 016042 100403
5448 016044 001002
5449 016046 102001
5450 016050 103401
5451
5452 016052 104002
5453

BCS 4\$
3\$: ERROR 1 ;NEG FAILED TO ALTER CODES PROPERLY
4\$: CMP (R2),R4 ;CORRECT RESULT = 100000?
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 @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #52525,R4 ;RESULT S / B = 52525
MOV #125252,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 @#2\$,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\$
BVC 3\$
BCS 4\$
3\$: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY

5454 016054 020403
 5455 016056 001401
 5456
 5457 016060 104002
 5458
 5459
 5460
 5461
 5462 016062
 5463 016062 000004
 5464 016064 012700 000243
 5465 016070 013701 016110
 5466 016074 012704 125252
 5467 016100 012703 052525
 5468 016104 000257
 5469 016106 000267
 5470
 5471 016110 006003
 5472
 5473 016112 100003
 5474 016114 001402
 5475 016116 102401
 5476 016120 103401
 5477
 5478 016122 104002
 5479
 5480 016124 020403
 5481 016126 001401
 5482
 5483 016130 104002
 5484
 5485
 5486
 5487
 5488 016132
 5489 016132 000004
 5490 016134 012700 000244
 5491 016140 013701 016156
 5492 016144 005004
 5493 016146 012703 000001
 5494 016152 000257
 5495 016154 000270
 5496
 5497 016156 006003
 5498
 5499 016160 100403
 5500 016162 001002
 5501 016164 102001
 5502 016166 103401
 5503
 5504 016170 104002
 5505
 5506 016172 020403
 5507 016174 001401
 5508
 5509 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    @#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
    MOV    #125252,R4    ;RESULT S / B = 125252
    MOV    #52525,R3     ;[DEST] = 052525
    CCC
    267                 ;CLEAR FLAGS
                          ;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    @#2$,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$
    BVC    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
  
```

5510
5511
5512
5513
5514 016200
5515 016200 000004
5516 016202 012700 000245
5517 016206 013701 016226
5518 016212 012704 152525
5519 016216 012703 125252
5520 016222 000257
5521 016224 000265
5522
5523 016226 006003
5524
5525 016230 100003
5526 016232 001402
5527 016234 102001
5528 016236 103001
5529
5530 016240 104002
5531
5532 016242 020403
5533 016244 001401
5534
5535 016246 104002
5536
5537
5538
5539
5540 016250
5541 016250 000004
5542 016252 012700 000246
5543 016256 013701 016276
5544 016262 012704 025252
5545 016266 012703 052525
5546 016272 000257
5547 016274 000274
5548
5549 016276 006003
5550
5551 016300 100403
5552 016302 001402
5553 016304 102001
5554 016306 103401
5555
5556 016310 104002
5557
5558 016312 020403
5559 016314 001401
5560
5561 016316 104002
5562
5563
5564
5565

```
*****  
: *TEST 245 ASR TEST - DMO - N:C = 0101  
*****  
TST245:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #245,R0 ;LOAD R0 WITH TEST NUMBER  
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #152525,R4 ;RESULT S / B = 152525  
MOV #125252,R3 ;[DEST] = 125252  
CCC ;CLEAR FLAGS  
265 ;N:C = 0101  
  
2$: ROR R3 ;TEST THE ROR  
  
BPL 3$ ;N:C = 1010 ?  
BEQ 3$  
BVC 3$  
BCC 4$  
  
3$: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY  
  
4$: CMP R4,R3 ;CORRECT RESULT ?  
BEQ TST246 ;BR IF YES  
  
5$: ERROR 2 ;ROR DELIVERED THE WRONG RESULT  
  
*****  
: *TEST 246 ASR TEST - DMO - N:C = 1100  
*****  
TST246:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #246,R0 ;LOAD R0 WITH TEST NUMBER  
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #25252,R4 ;RESULT S / B = 25252  
MOV #52525,R3 ;[DEST] = 52525  
CCC ;CLEAR FLAGS  
274 ;N:C = 1100  
  
2$: ROR R3 ;TEST THE ROR  
  
BMI 3$ ;N:C = 0011 ?  
BEQ 3$  
BVC 3$  
BCS 4$  
  
3$: ERROR 2 ;ROR FAILED TO ALTER CODES PROPERLY  
  
4$: CMP R4,R3 ;CORRECT RESULT ?  
BEQ TST247 ;BR IF YES  
  
5$: ERROR 2 ;ROR DELIVERED THE WRONG RESULT  
  
*****  
: *TEST 247 ROR TEST - DM1 - N:C = 1110  
*****
```

5566 016320
5567 016320 000004
5568 016322 012700 000247
5569 016326 013701 016352
5570 016332 012702 063312
5571 016336 012704 052525
5572 016342 012712 125252
5573 016346 000257
5574 016350 000276
5575
5576 016352 006012
5577
5578 016354 100403
5579 016356 001402
5580 016360 102401
5581 016362 103001
5582
5583 016364 104001
5584
5585 016366 020412
5586 016370 001402
5587 016372 011203
5588 016374 104001
5589
5590
5591
5592
5593 016376
5594 016376 000004
5595 016400 012700 000250
5596 016404 013701 016426
5597 016410 012702 063312
5598 016414 005004
5599 016416 012712 000001
5600 016422 000257
5601 016424 000270
5602
5603 016426 006012
5604
5605 016430 100403
5606 016432 001002
5607 016434 102001
5608 016436 103401
5609
5610 016440 104001
5611
5612 016442 020412
5613 016444 001402
5614
5615 016446 011203
5616 016450 104001
5617
5618
5619
5620
5621 016452

TST247:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #247,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #52525,R4 ;:RESULT S / B = 52525
MOV #125252,(R2) ;:[DEST] = 125252
CCC ;:CLEAR FLAGS
276 ;:N:C = 1110

2\$: ROR (R2) ;:TEST THE ROR

BMI 3\$;:N:C = 0000 ?
BEQ 3\$
BVS 3\$
BCC 4\$

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

4\$: CMP R4,(R2) ;:CORRECT RESULT ?
BEQ TST250 ;:BR IF YES
MOV (R2),R3 ;:GET THE WAS DATA
5\$: ERROR 1 ;:ROR DELIVERED WRONG RESULT

:::*****
:*TEST 250 ROR TEST - DM1 - N:C = 1000
:::*****

TST250:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #250,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
CLR R4 ;:RESULT S / B = 000000
MOV #1,(R2) ;:[DEST] = 1
CCC ;:CLEAR FLAGS
SEN ;:N:C = 1000

2\$: ROR (R2) ;:TEST THE ROR

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

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

4\$: CMP R4,(R2) ;:CORRECT RESULT ?
BEQ TST251 ;:BR IF YES
MOV (R2),R3 ;:GET THE WAS DATA
5\$: ERROR 1 ;:ROR DELIVERED WRONG RESULT

:::*****
:*TEST 251 ROR TEST - DM1 - N:C = 0111
:::*****

TST251:

```
5622 016452 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5623 016454 012700 000251  MOV #251,R0    ;:LOAD R0 WITH TEST NUMBER
5624 016460 013701 016504  MOV @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
5625 016464 012702 063312  MOV #MUF0,R2   ;DEST ADDR = MUF0
5626 016470 012704 125252  MOV #125252,R4 ;RESULT S / B = 125252
5627 016474 012712 052525  MOV #52525,(R2);[DEST] = 52525
5628 016500 000257          CCC          ;CLEAR FLAGS
5629 016502 000267          267          ;N:C = 0111
5630
5631 016504 006012 2$: ROR (R2)      ;TEST THE ROR
5632
5633 016506 100003          BPL 3$        ;N:C = 1001 ?
5634 016510 001402          BEQ 3$
5635 016512 102401          BVS 3$
5636 016514 103401          BCS 4$
5637
5638 016516 104001 3$: ERROR 1      ;ROR FAILED TO ALTER CODES PROPERLY
5639
5640 016520 020412 4$: CMP R4,(R2)    ;CORRECT RESULT ?
5641 016522 001402          BEQ TST252    ;:BR IF YES
5642
5643 016524 011203          MOV (R2),R3   ;GET THE WAS DATA
5644 016526 104001 5$: ERROR 1      ;ROR DELIVERED WRONG RESULT
5645
```

```
::*****
:*TEST 252 ASR TEST - DM1 - N:C = 1000
*****
TST252:
```

```
5650 016530 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5651 016532 012700 000252  MOV #252,R0    ;:LOAD R0 WITH TEST NUMBER
5652 016536 013701 016560  MOV @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
5653 016542 012702 063312  MOV #MUF0,R2   ;DEST ADDR = MUF0
5654 016546 005004          CLR R4        ;RESULT S / B = 0000C0
5655 016550 012712 000001  MOV #1,(R2)    ;[DEST] = 1
5656 016554 000257          CCC          ;CLEAR FLAGS
5657 016556 000270          SEN          ;N:C = 1000
5658
5659 016560 006012 2$: ROR (R2)      ;TEST THE ROR
5660
5661 016562 100403          BMI 3$        ;N:C = 0111 ?
5662 016564 001002          BNE 3$
5663 016566 102001          BVC 3$
5664 016570 103401          BCS 4$
5665
5666 016572 104001 3$: ERROR 1      ;ROR FAILED TO ALTER CODES PROPERLY
5667
5668 016574 020412 4$: CMP R4,(R2)    ;CORRECT RESULT ?
5669 016576 001402          BEQ TST253    ;:BR IF YES
5670
5671 016600 011203          MOV (R2),R3   ;GET THE WAS DATA
5672 016602 104001 5$: ERROR 1      ;ROR DELIVERED WRONG RESULT
5673
```

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

5677 016604


```

5734 016740 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5735 016742 012700 000255  MOV #255,R0    ;;LOAD R0 WITH TEST NUMBER
5736 016746 013701 016772  MOV @#2$,R1   ;LOAD R1 WITH TEST INSTRUCTION WORD
5737 016752 012702 063312  MOV #MBUF0,R2 ;DEST ADDR = MBUF0
5738 016756 012704 000177  MOV #177,R4   ;RESULT S / B = 177
5739 016762 010203        MOV R2,R3     ;R3 CONTAINS DEST ADDR
5740 016764 012712 000377  MOV #377,(R2) ;[DEST] = 377
5741 016770 000257        CCC              ;SCOPE SYNC "C" = 0
5742
5743 016772 106023        2$: RORB (R3)+ ;TEST THE RORB
5744
5745 016774 103401        BCS 4$        ;BR IF ROR SET "C"
5746
5747 016776 104001        3$: ERROR 1    ;ROR FAILED TO SET "C"
5748
5749 017000 022703 063313  4$: CMP #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED ?
5750 017004 001401        BEQ 6$        ;BR IF YES
5751
5752 017006 104005        5$: ERROR 5    ;RORB FAILED TO UPDATE DEST REG
5753
5754 017010 020412        6$: CMP R4,(R2)  ;CORRECT RESULT ?
5755 017012 001402        BEQ TST256   ;;BR IF YES
5756
5757 017014 011203        MOV (R2),R3   ;GET THE WAS DATA
5758 017016 104001        7$: ERROR 1    ;RORB DELIVERED WRONG RESULT
5759

```

```

;*****
;*TEST 256 RORB TEST - DM1 - EVEN ADDRESS
;*****

```

```

5760
5761
5762
5763 017020
5764 017020 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
5765 017022 012700 000256  MOV #256,R0    ;;LOAD R0 WITH TEST NUMBER
5766 017026 013701 017054  MOV @#2$,R1   ;LOAD R1 WITH TEST INSTRUCTION WORD
5767 017032 012702 063312  MOV #MBUF0,R2 ;DEST ADDR = MBUF0
5768 017036 012704 000377  MOV #377,R4   ;RESULT S / B = 377
5769 017042 010203        MOV R2,R3     ;R3 CONTAINS DEST ADDR
5770 017044 012712 000376  MOV #376,(R2) ;[DEST] = 376
5771 017050 000257        CCC              ;CLEAR FLAGS
5772 017052 000261        SEC              ;SCOPE SYNC - SET "C"
5773
5774 017054 106013        2$: RORB (R3)   ;TEST THE RORB
5775
5776 017056 103001        BCC 4$        ;BR IF "C" CLR - IT SHOULD BE
5777
5778 017060 104001        3$: ERROR 1    ;RORB FAILED TO CLR "C"
5779
5780 017062 020412        4$: CMP R4,(R2)  ;CORRECT RESULT ?
5781 017064 001402        BEQ TST257   ;;BR IF YES
5782
5783 017066 011203        MOV (R2),R3   ;GET THE WAS DATA
5784 017070 104001        5$: ERROR 1    ;RORB DELIVERED WRONG RESULT
5785

```

```

;*****
;*TEST 257 RORB TEST - DM2 - ODD ADDRESS
;*****

```

```

5786
5787
5788
5789 017072
TST257:

```

5790	017072	000004			SCOPE		;CALL THE SCOPE LOOP UTILITY	
5791	017074	012700	000257		MOV	#257,R0	::LOAD R0 WITH TEST NUMBER	
5792	017100	013701	017142		MOV	@#2\$,R1	;LOAD R1 WITH TEST INSTRUCTION WORD	
5793					.SBTTL	USER CONTROLLED BREAKPOINT	-- BITS	
5794	017104	032737	000040	063234	BIT	#BIT5,@#BPTLOC	;BREAKPOINT HALT SET ??	
5795	017112	001401			BEQ	+.4	;BR IF NOT	
5796	017114	000000			HALT		;BREAK - DEPRESS CONTINUE TO RESTART	
5797	017116	012702	063313		MOV	#MBUFO+1,R2	;DEST ADDR = MBUFO+1	
5798	017122	012704	077777		MOV	#77777,R4	;RESULT S / B = 77777	
5799	017126	012705	063312		MOV	#MBUFO,R5	;POINT R5 TO CHECK RESULT	
5800	017132	010203			MOV	R2,R3	;R3 CONTAINS DEST ADDR	
5801	017134	012715	177777		MOV	#-1,(R5)	;[DEST] = 177777	
5802	017140	000257			CCC		;SCOPE SYNC - "C" = 0	
5803								
5804	017142	106023			2\$: RORB	(R3)+	;TEST THE RORB	
5805								
5806	017144	103401				BCC	4\$;BR IF "C" IS SET - IT SHOULD BE
5807								
5808	017146	104001			3\$: ERROR	1	;RORB FAILED TO SET "C"	
5809								
5810	017150	022703	063314		4\$: CMP	#MBUFO+2,R3	;DID DEST REG GET INCREMENTED ?	
5811	017154	001401				BEQ	6\$;BR IF YES
5812								
5813	017156	104005			5\$: ERROR	5	;RORB FAILED TO UPDATE DEST REG	
5814								
5815	017160	020415			6\$: CMP	R4,(R5)	;CORRECT RESULT ?	
5816	017162	001402				BEQ	TST260	;BR IF YES
5817								
5818	017164	011503				MOV	(R5),R3	;GET THE WAS DATA
5819	017166	104001			7\$: ERROR	1	;RORB DELIVERED WRONG RESULT	

 ;*TEST 260 RORB TEST - DM1 - ODD ADDRESS

 TST260:

5824	017170							
5825	017170	000004			SCOPE		;CALL THE SCOPE LOOP UTILITY	
5826	017172	012700	000260		MOV	#260,R0	::LOAD R0 WITH TEST NUMBER	
5827	017176	013701	017226		MOV	@#2\$,R1	;LOAD R1 WITH TEST INSTRUCTION WORD	
5828	017202	012702	063313		MOV	#MBUFO+1,R2	;DEST ADDR = MBUFO+1	
5829	017206	012704	177777		MOV	#-1,R4	;RESULT S / B = 177777	
5830	017212	012705	063312		MOV	#MBUFO,R5	;POINT R5 TO CHECK RESULT	
5831	017216	010203			MOV	R2,R3	;R3 CONTAINS DEST ADDR	
5832	017220	012715	177377		MOV	#177377,(R5)	;[DEST] = 177377	
5833	017224	000261			SEC		;SCOPE SYNC - SET "C"	
5834								
5835	017226	106023			2\$: RORB	(R3)+	;TEST THE RORB	
5836								
5837	017230	103001				BCC	4\$;BR IF "C" CLEAR - IT SHOULD BE
5838								
5839	017232	104001			3\$: ERROR	1	;RORB FAILED TO CLEAR "C"	
5840								
5841	017234	020415			4\$: CMP	R4,(R5)	;CORRECT RESULT ?	
5842	017236	001402				BEQ	TST261	;BR IF YES
5843								
5844	017240	011503				MOV	(R5),R3	;GET THE WAS DATA
5845	017242	104001			5\$: ERROR	1	;RORB DELIVERED WRONG RESULT	

5846
 5847
 5848
 5849
 5850 017244
 5851 017244 000004
 5852 017246 012700 000261
 5853 017252 013701 017302
 5854 017256 012702 063313
 5855 017262 012704 000377
 5856 017266 012705 063312
 5857 017272 010203
 5858 017274 012715 000777
 5859 017300 000257
 5860
 5861 017302 106223
 5862
 5863 017304 103401
 5864
 5865 017306 104001
 5866
 5867 017310 022703 063314
 5868 017314 001401
 5869
 5870 017316 104005
 5871
 5872 017320 020415
 5873 017322 001402
 5874
 5875 017324 011503
 5876 017326 104001
 5877
 5878
 5879
 5880
 5881 017330
 5882 017330 000004
 5883 017332 012700 000262
 5884 017336 013701 017366
 5885 017342 012702 063313
 5886 017346 012704 140377
 5887 017352 012705 063312
 5888 017356 010203
 5889 017360 012715 100377
 5890 017364 000261
 5891
 5892 017366 106213
 5893
 5894 017370 103001
 5895
 5896 017372 104001
 5897
 5898 017374 020415
 5899 017376 001402
 5900
 5901 017400 011503

```

:*****
:*TEST 261      ASRB TEST - DM2 - ODD ADDRESS
:*****
TST261:
      SCOPE                                ;CALL THE SCOPE LOOP UTILITY
      MOV      #261,R0                      ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,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)                    ;[DEST] = 777
      CCC                                       ;SCOPE SYNC "C" = 0

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

      BCS      4$                            ;BR IF CARRY SET - IT SHOULD BE

3$:   ERROR    1                            ;ASRB FAILED TO SET THE CARRY

4$:   CMP      #MBUF0+2,R3                  ;DID DEST REG GET INCREMENTED ?
      BEQ      6$                            ;BR IF YES

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

6$:   CMP      R4,(R5)                      ;CORRECT RESULT ?
      BEQ      TST262                       ;;BR IF YES

7$:   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 R0 WITH TEST NUMBER
      MOV      @#2$,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)                 ;[DEST] = 100377
      SEC                                       ;SCOPE SYNC - "C" = 1

2$:   ASRB      (R3)                        ;TEST THE ASRB

      BCC      4$                            ;BR IF CARRY CLEAR - IT SHOULD BE

3$:   ERROR    1                            ;ASRB FAILED TO CLEAR THE CARRY

4$:   CMP      R4,(R5)                      ;CORRECT RESULT ?
      BEQ      TST263                       ;;BR IF YES

      MOV      (R5),R3                      ;GET THE WAS DATA
  
```

5902 017402 104001
5903
5904
5905
5906
5907 017404
5908 017404 000004
5909 017406 012700 000263
5910 017412 013701 017436
5911 017416 012702 063312
5912 017422 012704 000077
5913 017426 010203
5914 017430 012712 000177
5915 017434 000257
5916
5917 017436 106223
5918
5919 017440 103401
5920
5921 017442 104001
5922
5923 017444 022703 063313
5924 017450 001401
5925
5926 017452 104005
5927
5928 017454 020412
5929 017456 001402
5930
5931 017460 011203
5932 017462 104001
5933
5934
5935
5936
5937 017464
5938 017464 000004
5939 017466 012700 000264
5940 017472 013701 017516
5941 017476 012702 063312
5942 017502 012704 000303
5943 017506 010203
5944 017510 012712 000206
5945 017514 000261
5946
5947 017516 106213
5948
5949 017520 103001
5950
5951 017522 104001
5952
5953 017524 020412
5954 017526 001402
5955
5956 017530 011203
5957 017532 104001

5\$: ERROR 1 ;ASRB DELIVERED WRONG RESULT
:*****
:*TEST 263 ASRB TEST - DM2 - EVEN ADDRESS
:*****
TST263:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #263,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #77,R4 ;:RESULT S / B = 77
MOV R2,R3 ;:R3 CONTAINS DEST ADDR
MOV #177,(R2) ;:[DEST] = 177
CCC ;SCOPE SYNC - "C" = 0
2\$: ASRB (R3)+ ;TEST THE ASRB
BCS 4\$;BR IF "C" = 1 - IT SHOULD BE
3\$: ERROR 1 ;ASRB FAILED TO SET "C"
4\$: CMP #MBUFO+1,R3 ;DID DEST REG GET INCREMENTED ?
BEQ 6\$;BR IF YES
5\$: ERROR 5 ;ASRB FAILED TO UPDATE DEST REG
6\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST264 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
7\$: ERROR 1 ;ASRB DELIVERED WRONG RESULT
:*****
:*TEST 264 ASRB TEST - DM1 - EVEN ADDRESS
:*****
TST264:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #264,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #303,R4 ;:RESULT S / B = 303
MOV R2,R3 ;:R3 CONTAINS DEST ADDR
MOV #206,(R2) ;:[DEST] = 206
SEC ;SCOPE SYNC - "C" = 1
2\$: ASRB (R3) ;TEST THE CLRASRB
BCC 4\$;BR IF CARRY CLEAR - IT SHOULD BE
3\$: ERROR 1 ;ASRB FAILED TO CLEAR THE CARRY
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST265 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
5\$: ERROR 1 ;ASRB DELIVERED WRONG RESULT

5958
5959
5960
5961
5962 017534
5963 017534 000004
5964 017536 012700 000265
5965 017542 013701 017556
5966 017546 005004
5967 017550 005003
5968 017552 000257
5969 017554 000273
5970
5971 017556 005703
5972
5973 017560 100403
5974 017562 001002
5975 017564 102401
5976 017566 103001
5977
5978 017570 104002
5979
5980 017572 020403
5981 017574 001401
5982
5983 017576 104002
5984
5985
5986
5987
5988 017600
5989 017600 000004
5990 017602 012700 000266
5991 017606 013701 017624
5992 017612 005004
5993 017614 005104
5994 017616 010403
5995 017620 000257
5996 017622 000264
5997
5998 017624 005703
5999
6000 017626 100003
6001 017630 001402
6002 017632 102401
6003 017634 103001
6004
6005 017636 104002
6006
6007 017640 020403
6008 017642 001401
6009
6010 017644 104002
6011
6012
6013

```
*****
;*TEST 265      TST DMO TEST - N:C = 1011
*****
TST265:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
MOV    #265,R0            ;;LOAD R0 WITH TEST NUMBER
MOV    @#2$,R1           ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR    R4                ;RESULT S / B = 000000
CLR    R3                ;[DEST] = 000000
CCC                    ;CLEAR CODES
273                   ;N:C=1011

2$:   TST    R3          ;TEST THE TST

      BMI    3$          ;N:C = 0100 ?
      BNE    3$
      BVS    3$
      BCC    4$

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

4$:   CMP    R4,R3      ;RESULT OK ?
      BEQ    TST266     ;;BR IF YES

5$:   ERROR  2          ;TST ALTERED THE [DEST]

*****
;*TEST 266      TST DMO TEST - N:C = 0100
*****
TST266:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
MOV    #266,R0            ;;LOAD R0 WITH TEST NUMBER
MOV    @#2$,R1           ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR    R4                ;RESULT S / B = 177777
COM    R4                ;[DEST] = 177777
MOV    R4,R3            ;CLEAR CODES
264                   ;N:C=0100

2$:   TST    R3          ;TEST THE TST

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

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

4$:   CMP    R4,R3      ;RESULT OK ?
      BEQ    TST267     ;;BR IF YES

5$:   ERROR  2          ;TST ALTERED THE [DEST]

*****
;*TEST 267      CLR DMO TEST - N:C = 1011
*****
```

6014
6015 017646
6016 017646 000004
6017 017650 012700 000267
6018 017654 013701 017672
6019 017660 005004
6020 017662 012703 177777
6021 017666 000257
6022 017670 000273
6023
6024 017672 005003
6025
6026 017674 100403
6027 017676 001002
6028 017700 102401
6029 017702 103001
6030
6031 017704 104002
6032
6033 017706 020403
6034 017710 001401
6035
6036 017712 104002
6037
6038
6039
6040
6041 017714
6042 017714 000004
6043 017716 012700 000270
6044 017722 013701 017736
6045 017726 005004
6046 017730 012703 177777
6047 017734 000257
6048
6049 017736 005003
6050
6051 017740 100403
6052 017742 001002
6053 017744 102401
6054 017746 103001
6055
6056 017750 104002
6057
6058 017752 020403
6059 017754 001401
6060
6061 017756 104002
6062
6063
6064
6065
6066 017760
6067 017760 000004
6068 017762 012700 000271
6069 017766 013701 020006

TST267:

```
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #267,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #-1,R3 ;[DEST] = 177777
CCC ;CLEAR CODES
273 ;N:C = 1011

2$: CLR R3 ;TEST THE CLR

BMI 3$ ;N:C = 0100 ?
BNE 3$
BVS 3$
BCC 4$

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

4$: CMP R4,R3 ;RESULT OK ?
BEQ TST270 ;:BR IF YES

5$: ERROR 2 ;CLR DELIVERED THE WRONG RESULT
```

*TEST 270 CLR DMO TEST - N:C = 0000

TST270:

```
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #270,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #-1,R3 ;[DEST] = 177777
CCC ;CLEAR CODES

2$: CLR R3 ;TEST THE CLR

BMI 3$ ;N:C = 0100 ?
BNE 3$
BVS 3$
BCC 4$

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

4$: CMP R4,R3 ;RESULT OK ?
BEQ TST271 ;:BR IF YES

5$: ERROR 2 ;CLR DELIVERED THE WRONG RESULT
```

*TEST 271 COM DMO TEST - N:C = 0110

TST271:

```
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #271,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
```

```
6070 017772 012704 125252      MOV      #125252,R4      ;RESULT S / B = 125252
6071 017776 012703 052525      MOV      #52525,R3      ;[DEST] = 52525
6072 020002 000257      CCC      ;CLEAR CODES
6073 020004 000266      266      ;N:C = 0110
6074
6075 020006 005103      2$:      COM      R3      ;TEST THE COM
6076
6077 020010 100003      BPL      3$      ;N:C = 1001 ?
6078 020012 001402      BEQ      3$
6079 020014 102401      BVS      3$
6080 020016 103401      BCS      4$
6081
6082 020020 104002      3$:      ERROR      2      ;COM FAILED TO ALTER THE CODES PROPERLY
6083
6084 020022 020403      4$:      CMP      R4,R3      ;RESULT OK ?
6085 020024 001401      BEQ      TST272      ;:BR IF YES
6086
6087 020026 104002      5$:      ERROR      2      ;COM DELIVERED THE WRONG RESULT
6088
6089
6090
6091
6092 020030
6093 020030 000004      ;:*****
6094 020032 012700 000272      ;*TEST 272      COM DMO TEST - N:C = 1001
6095 020036 013701 020054      ;:*****
6096 020042 005004      TST272:
6097 020044 012703 177777      SCOPE      ;CALL THE SCOPE LOOP UTILITY
6098 020050 000257      MOV      #272,R0      ;:LOAD R0 WITH TEST NUMBER
6099 020052 000271      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
6100
6101 020054 005103      2$:      COM      R3      ;TEST THE COM
6102
6103 020056 100403      BMI      3$      ;N:C = 0101 ?
6104 020060 001002      BNE      3$
6105 020062 102401      BVS      3$
6106 020064 103401      BCS      4$
6107
6108 020066 104002      3$:      ERROR      2      ;COM FAILED TO ALTER THE CODES PROPERLY
6109
6110 020070 020403      4$:      CMP      R4,R3      ;RESULT OK ?
6111 020072 001401      BEQ      TST273      ;:BR IF YES
6112
6113 020074 104002      5$:      ERROR      2      ;COM DELIVERED THE WRONG RESULT
6114
6115
6116
6117
6118 020076
6119 020076 000004      ;:*****
6120 020100 012700 000273      ;*TEST 273      INC DMO TEST - N:C = 1011
6121 020104 013701 020122      ;:*****
6122 020110 005004      TST273:
6123 020112 012703 177777      SCOPE      ;CALL THE SCOPE LOOP UTILITY
6124 020116 000257      MOV      #273,R0      ;:LOAD R0 WITH TEST NUMBER
6125 020120 000273      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        CLR      R4      ;RESULT S / B = 000000
        MOV      #-1,R3      ;[DEST] = 177777
        CCC      ;CLEAR CODES
        273      ;N:C = 1011
```

```
6126
6127 020122 005203      2$:   INC      R3           ;TEST THE INC
6128
6129 020124 100403      BMI     3$           ;N:C = 0101 ?
6130 020126 001002      BNE     3$
6131 020130 102401      BVS     3$
6132 020132 103401      BCS     4$
6133
6134 020134 104002      3$:   ERROR    2           ;INC FAILED TO ALTER THE CODES PROPERLY
6135
6136 020136 020403      4$:   CMP      R4,R3       ;RESULT OK ?
6137 020140 001401      BEQ     TST274        ;;BR IF YES
6138
6139 020142 104002      5$:   ERROR    2           ;INC DELIVERED THE WRONG RESULT
6140
6141
6142
6143
6144 020144
6145 020144 000004
6146 020146 012700 000274
6147 020152 013701 020172
6148 020156 012704 100000
6149 020162 012703 077777
6150 020166 000257
6151 020170 000264
6152
6153 020172 005203      2$:   INC      R3           ;TEST THE INC
6154
6155 020174 100003      BPL     3$           ;N:C = 1010 ?
6156 020176 001402      BEQ     3$
6157 020200 102001      BVC     3$
6158 020202 103001      BCC     4$
6159
6160 020204 104002      3$:   ERROR    2           ;INC FAILED TO ALTER THE CODES PROPERLY
6161
6162 020206 020403      4$:   CMP      R4,R3       ;RESULT OK ?
6163 020210 001401      BEQ     TST275        ;;BR IF YES
6164
6165 020212 104002      5$:   ERROR    2           ;INC DELIVERED THE WRONG RESULT
6166
6167
6168
6169
6170 020214
6171 020214 000004
6172 020216 012700 000275
6173 020222 013701 020240
6174 020226 005004
6175 020230 012703 000001
6176 020234 000257
6177 020236 000273
6178
6179 020240 005303      2$:   DEC      R3           ;TEST THE DEC
6180
6181 020242 100403      BMI     3$           ;N:C = 0101 ?

;*****
;*TEST 274      INC DMO TEST - N:C = 0100
;*****
TST274:
SCOPE           ;CALL THE SCOPE LOOP UTILITY
MOV      #274,R0 ;:LOAD R0 WITH TEST NUMBER
MOV      @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #100000,R4 ;:RESULT S / B = 100000
MOV      #77777,R3 ;:[DEST] = 77777
CCC
264           ;CLEAR CODES
           ;N:C = 0100

;*****
;*TEST 275      DEC DMO TEST - N:C = 1011
;*****
TST275:
SCOPE           ;CALL THE SCOPE LOOP UTILITY
MOV      #275,R0 ;:LOAD R0 WITH TEST NUMBER
MOV      @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
CLR      R4      ;:RESULT S / B = 000000
MOV      #1,R3   ;:[DEST] = 1
CCC
273           ;CLEAR CODES
           ;N:C = 1011
```

6182 020244 001002
6183 020246 102401
6184 020250 103401
6185
6186 020252 104002
6187
6188 020254 020403
6189 020256 001401
6190
6191 020260 104002
6192
6193
6194
6195
6196 020262
6197 020262 000004
6198 020264 012700 000276
6199 020270 013701 020310
6200 020274 012704 077777
6201 020300 012703 100000
6202 020304 000257
6203 020306 000274
6204
6205 020310 005303
6206
6207 020312 100403
6208 020314 001402
6209 020316 102001
6210 020320 103001
6211
6212 020322 104002
6213
6214 020324 020403
6215 020326 001401
6216
6217 020330 104002
6218
6219
6220
6221
6222 020332
6223 020332 000004
6224 020334 012700 000277
6225 020340 013701 020354
6226 020344 012704 177777
6227 020350 005003
6228 020352 000257
6229
6230 020354 005303

```
BNE 3$  
BVS 3$  
BCS 4$  
3$: ERROR 2 ;DEC FAILED TO ALTER THE CODES PROPERLY  
4$: CMP R4,R3 ;RESULT OK ?  
BEQ TST276 ;;BR IF YES  
5$: ERROR 2 ;DEC DELIVERED THE WRONG RESULT  
:*****  
:*TEST 276 DEC DMO TEST - N:C = 1100  
:*****  
TST276:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #276,R0 ;;LOAD R0 WITH TEST NUMBER  
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #77777,R4 ;RESULT S / B = 77777  
MOV #100000,R3 ;[DEST] = 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 TST277 ;;BR IF YES  
5$: ERROR 2 ;DEC DELIVERED THE WRONG RESULT  
:*****  
:*TEST 277 DEC DMO TEST - N:C = 0000  
:*****  
TST277:  
SCOPE ;CALL THE SCOPE LOOP UTILITY  
MOV #277,R0 ;;LOAD R0 WITH TEST NUMBER  
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #-1,R4 ;RESULT S / B = 177777  
CLR R3 ;[DEST] = 000000  
CCC ;CLEAR CODES  
2$: DEC R3 ;TEST THE DEC
```

```

6231
6232 020356 100003          BPL      3$           ;N:C = 1000 ?
6233 020360 001402          BEQ      3$
6234 020362 102401          BVS      3$
6235 020364 103001          BCC      4$
6236
6237 020366 104002          3$:      ERROR    2           ;DEC FAILED TO ALTER THE CODES PROPERLY
6238
6239 020370 020403          4$:      CMP      R4,R3         ;RESULT OK ?
6240 020372 001401          BEQ      TST300        ;:BR IF YES
6241
6242 020374 104002          5$:      ERROR    2           ;DEC DELIVERED THE WRONG RESULT
6243

```

```

*****
;*TEST 300      ASL DMO TEST - N:C = 1000
*****
TST300:

```

```

6247 020376
6248 020376 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6249 020400 012700 000300  MOV      #300,R0    ;:LOAD R0 WITH TEST NUMBER
6250 020404 013701 020422  MOV      @#2$,R1    ;:LOAD R1 WITH TEST INSTRUCTION WORD
6251 020410 005004          CLR      R4         ;:RESULT S / B = 000000
6252 020412 012703 100000  MOV      #100000,R3 ;:[DEST] = 100000
6253 020416 000257          CCC
6254 020420 000270          SEN           ;CLEAR CODES
6255
6256 020422 006303          2$:      ASL      R3           ;TEST THE ASL
6257
6258 020424 100403          BMI      3$         ;N:C = 0111 ?
6259 020426 001002          BNE      3$
6260 020430 102001          BVC      3$
6261 020432 103401          BCS      4$
6262

```

```

6263 020434 104002          3$:      ERROR    2           ;ASL FAILED TO ALTER THE CODES PROPERLY
6264
6265 020436 020403          4$:      CMP      R4,R3         ;RESULT OK ?
6266 020440 001401          BEQ      TST301        ;:BR IF YES
6267
6268 020442 104002          5$:      ERROR    2           ;ASL DELIVERED THE WRONG RESULT
6269

```

```

*****
;*TEST 301      ASL DMO TEST - N:C = 0101
*****
TST301:

```

```

6273 020444
6274 020444 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6275 020446 012700 000301  MOV      #301,R0    ;:LOAD R0 WITH TEST NUMBER
6276 020452 013701 020472  MOV      @#2$,R1    ;:LOAD R1 WITH TEST INSTRUCTION WORD
6277 020456 012704 100000  MOV      #100000,R4 ;:RESULT S / B = 100000
6278 020462 012703 040000  MOV      #40000,R3  ;:[DEST] = 40000
6279 020466 000257          CCC           ;CLEAR CODES
6280 020470 000265          265          ;N:C = 0101
6281

```

```

6282 020472 006303          2$:      ASL      R3           ;TEST THE ASL
6283
6284 020474 100003          BPL      3$         ;N:C = 1010 ?
6285 020476 001402          BEQ      3$
6286 020500 102001          BVC      3$

```


6287 020502 103001
6288
6289 020504 104002
6290
6291 020506 020403
6292 020510 001401
6293
6294 020512 104002
6295
6296
6297
6298
6299 020514
6300 020514 000004
6301 020516 012700 000302
6302 020522 013701 020536
6303 020526 005004
6304 020530 005003
6305 020532 000257
6306 020534 000262
6307
6308 020536 006303
6309
6310 020540 100403
6311 020542 001002
6312 020544 102401
6313 020546 103001
6314
6315 020550 104002
6316
6317 020552 020403
6318 020554 001401
6319
6320 020556 104002
6321
6322
6323
6324
6325 020560
6326 020560 000004
6327 020562 012700 000303
6328 020566 013701 020606
6329 020572 012704 052525
6330 020576 012703 125252
6331 020602 000257
6332 020604 000275
6333
6334 020606 006103
6335
6336 020610 100403
6337 020612 001402
6338 020614 102001
6339 020616 103401
6340
6341 020620 104002
6342

BCC 4\$
3\$: ERROR 2 ;ASL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,R3 ;RESULT OK ?
BEQ TST302 ;;BR IF YES
5\$: ERROR 2 ;ASL DELIVERED THE WRONG RESULT
:*****
:*TEST 302 ASL DMO TEST - N:C = 0010
:*****
TST302:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #302,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
CLR R3 ;[DEST] = 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 TST303 ;;BR IF YES
5\$: ERROR 2 ;ASL DELIVERED THE WRONG RESULT
:*****
:*TEST 303 ROL DMO TEST - N:C = 1101
:*****
TST303:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #303,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #52525,R4 ;RESULT S / B = 52525
MOV #125252,R3 ;[DEST] = 125252
CCC ;CLEAR CODES
275 ;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

6343 020622 020403
6344 020624 001401
6345
6346 020626 104002
6347
6348
6349
6350
6351 020630
6352 020630 000004
6353 020632 012700 000304
6354 020636 013701 020656
6355 020642 012704 125253
6356 020646 012703 052525
6357 020652 000257
6358 020654 000265
6359
6360 020656 006103
6361
6362 020660 100003
6363 020662 001402
6364 020664 102001
6365 020666 103001
6366
6367 020670 104002
6368 020672 020403
6369 020674 001401
6370
6371 020676 104002
6372
6373
6374
6375
6376 020700
6377 020700 000004
6378 020702 012700 000305
6379 020706 013701 020722
6380 020712 005004
6381 020714 005003
6382 020716 000257
6383 020720 000262
6384
6385 020722 006103
6386
6387 020724 100403
6388 020726 001002
6389 020730 102401
6390 020732 103001
6391
6392 020734 104002
6393
6394 020736 020403
6395 020740 001401
6396
6397 020742 104002
6398

4\$: CMP R4,R3 ;RESULT OK ?
BEQ TST304 ;:BR IF YES

5\$: ERROR 2 ;ROL DELIVERED THE WRONG RESULT

:::*****
:*TEST 304 ROL DMO TEST - N:C = 0101
:::*****

TST304:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #304,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #125253,R4 ;RESULT S / B = 125253
MOV #52525,R3 ;[DEST] = 52525
CCC ;CLEAR CODES
265 ;N:C = 0101

2\$: ROL R3 ;TEST THE ROL

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

3\$: ERROR 2 ;ROL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,R3 ;RESULT OK ?
BEQ TST305 ;:BR IF YES

5\$: ERROR 2 ;ROL DELIVERED THE WRONG RESULT

:::*****
:*TEST 305 ROL DMO TEST - N:C = 0010
:::*****

TST305:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #305,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
CLR R3 ;[DEST] = 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 TST306 ;:BR IF YES

5\$: ERROR 2 ;ROL DELIVERED THE WRONG RESULT

6399
6400
6401
6402 020744
6403 020744 000004
6404 020746 012700 000306
6405 020752 013701 020772
6406 020756 012704 100000
6407 020762 012703 077777
6408 020766 000257
6409 020770 000265
6410
6411 020772 005503
6412
6413 020774 100003
6414 020776 001402
6415 021000 102001
6416 021002 103001
6417
6418 021004 104002
6419
6420 021006 020403
6421 021010 001401
6422
6423 021012 104002
6424
6425
6426
6427
6428 021014
6429 021014 000004
6430 021016 012700 000307
6431 021022 013701 021040
6432 021026 005004
6433 021030 012703 177777
6434 021034 000257
6435 021036 000273
6436
6437 021040 005503
6438
6439 021042 100403
6440 021044 001002
6441 021046 102401
6442 021050 103401
6443
6444 021052 104002
6445
6446 021054 020403
6447 021056 001401
6448
6449 021060 104002
6450
6451
6452
6453
6454 021062

```
.....  
*TEST 306      ADC DMO TEST - N:C = 0101  
.....  
TST306:  
      SCOPE      ;CALL THE SCOPE LOOP UTILITY  
      MOV      #306,R0      ;:LOAD R0 WITH TEST NUMBER  
      MOV      @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD  
      MOV      #100000,R4    ;:RESULT S / B = 100000  
      MOV      #77777,R3    ;:[DEST] = 77777  
      CCC      ;CLEAR CODES  
      265      ;N:C = 0101  
  
2$:   ADC      R3      ;TEST THE ADC  
  
      BPL      3$      ;N:C = 1010 ?  
      BEQ      3$  
      BVC      3$  
      BCC      4$  
  
3$:   ERROR    2      ;ADC FAILED TO ALTER THE CODES PROPERLY  
  
4$:   CMP      R4,R3    ;RESULT OK ?  
      BEQ      TST307   ;:BR IF YES  
  
5$:   ERROR    2      ;ADC DELIVERED THE WRONG RESULT
```

```
.....  
*TEST 307      ADC DMO TEST - N:C = 1011  
.....  
TST307:  
      SCOPE      ;CALL THE SCOPE LOOP UTILITY  
      MOV      #307,R0      ;:LOAD R0 WITH TEST NUMBER  
      MOV      @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD  
      CLR      R4          ;:RESULT S / B = 000000  
      MOV      #-1,R3     ;:[DEST] = 177777  
      CCC      ;CLEAR CODES  
      273      ;N:C = 1011  
  
2$:   ADC      R3      ;TEST THE ADC  
  
      BMI      3$      ;N:C = 0101 ?  
      BNE      3$  
      BVS      3$  
      BCS      4$  
  
3$:   ERROR    2      ;ADC FAILED TO ALTER THE CODES PROPERLY  
  
4$:   CMP      R4,R3    ;RESULT OK ?  
      BEQ      TST310   ;:BR IF YES  
  
5$:   ERROR    2      ;ADC DELIVERED THE WRONG RESULT
```

```
.....  
*TEST 310      ADC DMO TEST - N:C = 1010  
.....  
TST310:
```

```
6455 021062 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6456 021064 012700 000310  MOV      #310,R0  ;:LOAD R0 WITH TEST NUMBER
6457 021070 013701 021110  MOV      @#2$,R1  ;:LOAD R1 WITH TEST INSTRUCTION WORD
6458 021074 012704 177777  MOV      #-1,R4   ;:RESULT S / B = 177777
6459 021100 012703 177777  MOV      #-1,R3   ;:[DEST] = 177777
6460 021104 000257          CCC          ;:CLEAR CODES
6461 021106 000272          272          ;:N:C = 1010
6462
6463 021110 005503          2$:  ADC      R3          ;:TEST THE ADC
6464
6465 021112 100003          BPL      3$          ;:N:C = 1000 ?
6466 021114 001402          BEQ      3$
6467 021116 102401          BVS      3$
6468 021120 103001          BCC      4$
6469
6470 021122 104002          3$:  ERROR    2          ;:ADC FAILED TO ALTER THE CODES PROPERLY
6471
6472 021124 020403          4$:  CMP      R4,R3     ;:RESULT OK ?
6473 021126 001401          BEQ      TST311     ;:BR IF YES
6474
6475 021130 104002          5$:  ERROR    2          ;:ADC DELIVERED THE WRONG RESULT
6476
6477
6478
6479
6480 021132          ;:*****
6481 021132 000004          ;:*TEST 311      SBC DMO TEST - N:C = 1011
6482 021134 012700 000311  ;:*****
6483 021140 013701 021156  TST311:
6484 021144 005004          SCOPE          ;:CALL THE SCOPE LOOP UTILITY
6485 021146 012703 000001  MOV      #311,R0  ;:LOAD R0 WITH TEST NUMBER
6486 021152 000257          MOV      @#2$,R1  ;:LOAD R1 WITH TEST INSTRUCTION WORD
6487 021154 000273          CLR      R4       ;:RESULT S / B = 000000
6488
6489 021156 005603          2$:  SBC      R3          ;:[DEST] = +1
6490
6491 021160 100403          BMI      3$          ;:CLEAR CODES
6492 021162 001002          BNE      3$          ;:N:C = 1011
6493 021164 102401          BVS      3$
6494 021166 103001          BCC      4$
6495
6496 021170 104002          3$:  ERROR    2          ;:TEST THE SBC
6497
6498 021172 020403          BMI      3$          ;:N:C = 0100 ?
6499 021174 001401          BNE      3$
6500
6501 021176 104002          3$:  ERROR    2          ;:SBC FAILED TO ALTER THE CODES PROPERLY
6502
6503
6504
6505
6506 021200          4$:  CMP      R4,R3     ;:RESULT OK ?
6507 021200 000004          BEQ      TST312     ;:BR IF YES
6508 021202 012700 000312  5$:  ERROR    2          ;:SBC DELIVERED THE WRONG RESULT
6509 021206 013701 021226  ;:*****
6510 021212 012704 077777  ;:*TEST 312      SBC DMO TEST - N:C = 0101
;:*****
TST312:
SCOPE          ;:CALL THE SCOPE LOOP UTILITY
MOV      #312,R0  ;:LOAD R0 WITH TEST NUMBER
MOV      @#2$,R1  ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #077777,R4 ;:RESULT S / B = 077777
```

6511 021216 012703 100000
6512 021222 000257
6513 021224 000265
6514
6515 021226 005603
6516
6517 021230 100403
6518 021232 001402
6519 021234 102001
6520 021236 103001
6521
6522 021240 104002
6523
6524 021242 020403
6525 021244 001401
6526
6527 021246 104002
6528
6529
6530
6531
6532 021250
6533 021250 000004
6534 021252 012700 000313
6535 021256 013701 021276
6536 021262 012704 000001
6537 021266 012703 000001
6538 021272 000257
6539 021274 000276
6540
6541 021276 005603
6542
6543 021300 100403
6544 021302 001402
6545 021304 102401
6546 021306 103001
6547
6548 021310 104002
6549
6550 021312 020403
6551 021314 001401
6552
6553 021316 104002
6554
6555
6556
6557
6558 021320
6559 021320 000004
6560 021322 012700 000314
6561 021326 013701 021344
6562 021332 012704 177777
6563 021336 005003
6564 021340 000257
6565 021342 000267
6566

```
MOV #100000,R3 ;[DEST] = 100000
CCC ;CLEAR CODES
265 ;N:C = 0101

2$: SBC R3 ;TEST THE SBC

BMI 3$ ;N:C = 0010 ?
BEQ 3$
BVC 3$
BCC 4$

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

4$: CMP R4,R3 ;RESULT OK ?
BEQ TST313 ;;BR IF YES

5$: ERROR 2 ;SBC DELIVERED THE WRONG RESULT

;*****
;*TEST 313 SBC DMO TEST - N:C = 1110
;*****
TST313:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #313,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,R4 ;RESULT S / B = 1
MOV #1,R3 ;[DEST] = 1
CCC ;CLEAR CODES
276 ;N:C = 1110

2$: SBC R3 ;TEST THE SBC

BMI 3$ ;N:C = 0000 ?
BEQ 3$
BVS 3$
BCC 4$

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

4$: CMP R4,R3 ;RESULT OK ?
BEQ TST314 ;;BR IF YES

5$: ERROR 2 ;SBC DELIVERED THE WRONG RESULT

;*****
;*TEST 314 SBC DMO TEST - N:C = 0111
;*****
TST314:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #314,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S / B = 177777
CLR R3 ;[DEST] = 000000
CCC ;CLEAR CODES
267 ;N:C = 0111
```

6567 021344 005603
6568
6569 021346 100003
6570 021350 001402
6571 021352 102401
6572 021354 103401
6573
6574 021356 104002
6575
6576 021360 020403
6577 021362 001401
6578
6579 021364 104002
6580
6581
6582
6583
6584 021366
6585 021366 000004
6586 021370 012700 000315
6587 021374 013701 021414
6588 021400 012702 063312
6589 021404 005004
6590 021406 005012
6591 021410 000257
6592 021412 000273
6593
6594 021414 005712
6595
6596 021416 100403
6597 021420 001002
6598 021422 102401
6599 021424 103001
6600
6601 021426 104001
6602
6603 021430 020412
6604 021432 001402
6605
6606 021434 011203
6607 021436 104001
6608
6609
6610
6611
6612 021440
6613 021440 000004
6614 021442 012700 000316
6615 021446 013701 021472
6616 021452 012702 063312
6617 021456 005004
6618 021460 005104
6619 021462 012712 177777
6620 021466 000257
6621 021470 000264
6622

2\$: SBC R3 ;TEST THE SBC
BPL 3\$;N:C = 1001 ?
BEQ 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 2 ;SBC FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,R3 ;RESULT OK ?
BEQ TST315 ;;BR IF YES
5\$: ERROR 2 ;SBC DELIVERED THE WRONG RESULT

;*TEST 315 TST DMI TEST - N:C = 1011

TST315:

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #315,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
CLR (R2) ;[DEST] = 000000
CCC ;CLEAR CODES
273 ;N:C=1011
2\$: TST (R2) ;TEST THE TST
BMI 3\$;N:C = 0100 ?
BNE 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 1 ;TST FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ TST316 ;;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;TST ALTERED THE [DEST]

;*TEST 316 TST DMI TEST - N:C = 0100

TST316:

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #316,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 177777
COM R4 ;[DEST] = 177777
MOV #-1,(R2) ;CLEAR CODES
CCC ;N:C=0100
264

6623 021472 005712
6624
6625 021474 100003
6626 021476 001402
6627 021500 102401
6628 021502 103001
6629
6630 021504 104001
6631
6632 021506 020412
6633 021510 001402
6634
6635 021512 011203
6636 021514 104001
6637
6638
6639
6640
6641 021516
6642 021516 000004
6643 021520 012700 000317
6644 021524 013701 021560
6645
6646 021530 032737 000100 063234
6647 021536 001401
6648 021540 000000
6649 021542 012702 063312
6650 021546 005004
6651 021550 012712 177777
6652 021554 000257
6653 021556 000273
6654
6655 021560 005012
6656
6657 021562 100403
6658 021564 001002
6659 021566 102401
6660 021570 103001
6661
6662 021572 104001
6663
6664 021574 020412
6665 021576 001402
6666
6667 021600 011203
6668 021602 104001
6669
6670
6671
6672
6673 021604
6674 021604 000004
6675 021606 012700 000320
6676 021612 013701 021632
6677 021616 012702 063312
6678 021622 005004

```
2$: TST (R2) ;TEST THE TST
    BPL 3$ ;N:C = 1000 ?
    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 [DEST]

;*****
;*TEST 317 CLR DM1 TEST - N:C = 1011
;*****
TST317:
    SCOPE ;CALL THE SCOPE LOOP UTILITY
    MOV #317,R0 ;;LOAD R0 WITH TEST NUMBER
    MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
.SBTTL USER CONTROLLED BREAKPOINT -- BIT6
    BIT #BIT6,@#BPTLOC ;BREAKPOINT HALT SET ??
    BEQ .+4 ;BR IF NOT
    HALT ;BREAK - DEPRESS CONTINUE TO RESTART
    MOV #MBUFO,R2 ;DEST ADDR = MBUFO
    CLR R4 ;RESULT S / B = 000000
    MOV #-1,(R2) ;[DEST] = 177777
    CCC ;CLEAR CODES
    273 ;N:C = 1011
2$: CLR (R2) ;TEST THE CLR
    BMI 3$ ;N:C = 0100 ?
    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 @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
    MOV #MBUFO,R2 ;DEST ADDR = MBUFO
    CLR R4 ;RESULT S / B = 000000
```

```
6679 021624 013712 063324      MOV @#DWTA+2,(R2) ;[DEST] = 177777
6680 021630 000257              CCC ;CLEAR CODES
6681
6682 021632 005022      2$: CLR (R2)+ ;TEST THE CLR
6683
6684 021634 100403              BMI 3$ ;N:C = 0100 ?
6685 021636 001002              BNE 3$
6686 021640 102401              BVS 3$
6687 021642 103001              BCC 4$
6688
6689 021644 104001      3$: ERROR 1 ;CLR FAILED TO ALTER THE CODES PROPERLY
6690
6691 021646 022702 063314      4$: CMP #MBUFO+2,R2 ;DID CLR INCREMENT DEST REG
6692 021652 001401              BEQ 6$ ;BR IF YES
6693
6694 021654 104005      5$: ERROR 5 ;CLR FAILED TO UPDATE DEST REG
6695
6696 021656 020442      6$: CMP R4,-(R2) ;RESULT OK ?
6697 021660 001402              BEQ TSI321 ;:BR IF YES
6698
6699 021662 011203      7$: MOV (R2),R3 ;GET THE WAS DATA
6700 021664 104001              ERROR 1 ;CLR DELIVERED THE WRONG RESULT
6701
6702
6703
6704
```

```
::*****
:*TEST 321 COM DM1 TEST - N:C = 0110
:*****
TST321:
```

```
6705 021666 000004              SCOPE ;CALL THE SCOPE LOOP UTILITY
6706 021666 000004              MOV #321,R0 ;:LOAD R0 WITH TEST NUMBER
6707 021670 012700 000321      MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
6708 021674 013701 021720      MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
6709 021700 012702 063312      MOV #125252,R4 ;:RESULT S / B = 125252
6710 021704 012704 125252      MOV #52525,(R2) ;[DEST] = 52525
6711 021710 012712 052525      CCC ;CLEAR CODES
6712 021714 000257              266 ;N:C = 0110
6713 021716 000266
6714
6715 021720 005112      2$: COM (R2) ;TEST THE CLR
6716
6717 021722 100003              BPL 3$ ;N:C = 1001 ?
6718 021724 001402              BEQ 3$
6719 021726 102401              BVS 3$
6720 021730 103401              BCS 4$
6721
6722 021732 104001      3$: ERROR 1 ;COM FAILED TO ALTER THE CODES PROPERLY
6723 021734 020412      4$: CMP R4,(R2) ;:RESULT OK ?
6724 021736 001402              BEQ TST322 ;:BR IF YES
6725
6726 021740 011203      5$: MOV (R2),R3 ;GET THE WAS DATA
6727 021742 104001              ERROR 1 ;COM DELIVERED THE WRONG RESULT
6728
6729
```

```
::*****
:*TEST 322 COM DM1 TEST - N:C = 1001
:*****
TST322:
```

```
6730
6731
6732 021744 000004              SCOPE ;CALL THE SCOPE LOOP UTILITY
6733 021744 000004              MOV #322,R0 ;:LOAD R0 WITH TEST NUMBER
6734 021746 012700 000322
```


6735	021752	013701	021774	MOV	@#2\$,R1	:LOAD R1 WITH TEST INSTRUCTION WORD
6736	021756	012702	063312	MOV	#MBUFO,R2	:DEST ADDR = MBUFO
6737	021762	005004		CLR	R4	:RESULT S / B = 000000
6738	021764	012712	177777	MOV	#-1,(R2)	: [DEST] = 177777
6739	021770	000257		CCC		:CLEAR CODES
6740	021772	000271		271		:N:C = 1001
6741						
6742	021774	005112		2\$: COM	(R2)	:TEST THE COM
6743						
6744	021776	100403		BMI	3\$:N:C = 0101 ?
6745	022000	001002		BNE	3\$	
6746	022002	102401		BVS	3\$	
6747	022004	103401		BCS	4\$	
6748						
6749	022006	104001		3\$: ERROR	1	:COM FAILED TO ALTER THE CODES PROPERLY
6750	022010	020412		4\$: CMP	R4,(R2)	:RESULT OK ?
6751	022012	001402		BEQ	TST323	::BR IF YES
6752						
6753	022014	011203		MOV	(R2),R3	:GET THE WAS DATA
6754	022016	104001		5\$: ERROR	1	:COM DELIVERED THE WRONG RESULT

::*****
 : *TEST 323 INC DM1 TEST - N:C = 1011
 :*****
 TST323:

6759	022020			SCOPE		:CALL THE SCOPE LOOP UTILITY
6760	022020	000004		MOV	#323,R0	::LOAD R0 WITH TEST NUMBER
6761	022022	012700	000323	MOV	@#2\$,R1	:LOAD R1 WITH TEST INSTRUCTION WORD
6762	022026	013701	022050	MOV	#MBUFO,R2	:DEST ADDR = MBUFO
6763	022032	012702	063312	CLR	R4	:RESULT S / B = 000000
6764	022036	005004		MOV	#-1,(R2)	: [DEST] = 177777
6765	022040	012712	177777	CCC		:CLEAR CODES
6766	022044	000257		273		:N:C = 1011
6767	022046	000273				
6768						
6769	022050	005212		2\$: INC	(R2)	:TEST THE INC
6770						
6771	022052	100403		BMI	3\$:N:C = 0101 ?
6772	022054	001002		BNE	3\$	
6773	022056	102401		BVS	3\$	
6774	022060	103401		BCS	4\$	
6775						
6776	022062	104001		3\$: ERROR	1	:INC FAILED TO ALTER THE CODES PROPERLY
6777	022064	020412		4\$: CMP	R4,(R2)	:RESULT OK ?
6778	022066	001402		BEQ	TST324	::BR IF YES
6779						
6780	022070	011203		MOV	(R2),R3	:GET THE WAS DATA
6781	022072	104001		5\$: ERROR	1	:INC DELIVERED THE WRONG RESULT

::*****
 : *TEST 324 INC DM1 TEST - N:C = 0100
 :*****
 TST324:

6786	022074			SCOPE		:CALL THE SCOPE LOOP UTILITY
6787	022074	000004		MOV	#324,R0	::LOAD R0 WITH TEST NUMBER
6788	022076	012700	000324	MOV	@#2\$,R1	:LOAD R1 WITH TEST INSTRUCTION WORD
6789	022102	013701	022126	MOV	#MBUFC,R2	:DEST ADDR = MBUFO
6790	022106	012702	063312			

6791 022112 012704 100000
6792 022116 012712 077777
6793 022122 000257
6794 022124 000264
6795
6796 022126 005212
6797
6798 022130 100003
6799 022132 001402
6800 022134 102001
6801 022136 103001
6802
6803 022140 104001
6804 022142 020412
6805 022144 001402
6806
6807 022146 011203
6808 022150 104001
6809

MOV #100000,R4 ;RESULT S / B = 100000
MOV #77777,(R2) ;[DEST] = 77777
CCC ;CLEAR CODES
264 ;N:C = 0100
2\$: INC (R2) ;TEST THE INC
BPL 3\$;N:C = 1010 ?
BEQ 3\$
BVC 3\$
BCC 4\$
3\$: ERROR 1 ;INC FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ TST325 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
5\$: ERROR 1 ;INC DELIVERED THE WRONG RESULT

6810
6811
6812
6813 022152
6814 022152 000004
6815 022154 012700 000325
6816 022160 013701 022202
6817 022164 012702 063312
6818 022170 005004
6819 022172 012712 000001
6820 022176 000257
6821 022200 000273
6822
6823 022202 005312
6824
6825 022204 100403
6826 022206 001002
6827 022210 102401
6828 022212 103401
6829
6830 022214 104001
6831 022216 020412
6832 022220 001402
6833
6834 022222 011203
6835 022224 104001
6836

: *TEST 325 DEC DM1 TEST - N:C = 1011

TST325:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #325,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
MOV #1,(R2) ;[DEST] = 1
CCC ;CLEAR CODES
273 ;N:C = 1011
2\$: DEC (R2) ;TEST THE DEC
BMI 3\$;N:C = 0101 ?
BNE 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 1 ;DEC FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ TST326 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
5\$: ERROR 1 ;DEC DELIVERED THE WRONG RESULT

6837
6838
6839
6840 022226
6841 022226 000004
6842 022230 012700 000326
6843 022234 013701 022260
6844 022240 012702 063312
6845 022244 012704 077777
6846 022250 012712 100000

: *TEST 326 DEC DM1 TEST - N:C = 1100

TST326:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #326,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,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

```
6847 022254 000257          CCC          ;CLEAR CODES
6848 022256 000274          274          ;N:C = 1100
6849
6850 022260 005312          2$: DEC      (R2)          ;TEST THE DEC
6851
6852 022262 100403          BMI      3$          ;N:C = 0010 ?
6853 022264 001402          BEQ      3$
6854 022266 102001          BVC      3$
6855 022270 103001          BCC      4$
6856
6857 022272 104001          3$: ERROR    1          ;DEC FAILED TO ALTER THE CODES PROPERLY
6858 022274 020412          4$: CMP      R4,(R2)      ;RESULT OK ?
6859 022276 001402          BEQ      TST327        ;:BR IF YES
6860
6861 022300 011203          MOV      (R2),R3        ;GET THE WAS DATA
6862 022302 104001          5$: ERROR    1          ;DEC DELIVERED THE WRONG RESULT
6863
```

```
::*****
:*TEST 327      DEC DM1 TEST - N:C = 0000
::*****
TST327:
```

```
6867 022304          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6868 022304 000004          MOV      #327,R0        ;:LOAD R0 WITH TEST NUMBER
6869 022306 012700 000327          MOV      @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
6870 022312 013701 022332          MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
6871 022316 012702 063312          MOV      #-1,R4         ;RESULT S / B = 177777
6872 022322 012704 177777          CLR      (R2)          ;[DEST] = 000000
6873 022326 005012          CCC          ;CLEAR CODES
6874 022330 000257
6875
6876 022332 005312          2$: DEC      (R2)          ;TEST THE DEC
6877
6878 022334 100003          BPL      3$          ;N:C = 1000 ?
6879 022336 001402          BEQ      3$
6880 022340 102401          BVS      3$
6881 022342 103001          BCC      4$
6882
6883 022344 104001          3$: ERROR    1          ;DEC FAILED TO ALTER THE CODES PROPERLY
6884 022346 020412          4$: CMP      R4,(R2)      ;RESULT OK ?
6885 022350 001402          BEQ      TST330        ;:BR IF YES
6886
6887 022352 011203          MOV      (R2),R3        ;GET THE WAS DATA
6888 022354 104001          5$: ERROR    1          ;DEC DELIVERED THE WRONG RESULT
6889
```

```
::*****
:*TEST 330      ASL DM1 TEST - N:C = 1000
::*****
TST330:
```

```
6893 022356          SCOPE          ;CALL THE SCOPE LOOP UTILITY
6894 022356 000004          MOV      #330,R0        ;:LOAD R0 WITH TEST NUMBER
6895 022360 012700 000330          MOV      @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
6896 022364 013701 022406          MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
6897 022370 012702 063312          CLR      R4            ;RESULT S / B = 000000
6898 022374 005004          MOV      #100000,(R2)   ;[DEST] = 100000
6899 022376 012712 100000          CCC          ;CLEAR CODES
6900 022402 000257          SEN          ;N:C = 1000
6901 022404 000270
6902
```

6903 022406 006312

6904

6905 022410 100403

6906 022412 001002

6907 022414 102001

6908 022416 103401

6909

6910 022420 104001

6911 022422 020412

6912 022424 001402

6913

6914 022426 011203

6915 022430 104001

6916

6917

6918

6919

6920 022432

6921 022432 000004

6922 022434 012700

6923 022440 013701

6924 022444 012702

6925 022450 012704

6926 022454 012712

6927 022460 000257

6928 022462 000265

6929

6930 022464 006312

6931

6932 022466 100003

6933 022470 001402

6934 022472 102001

6935 022474 103001

6936

6937 022476 104001

6938 022500 020412

6939 022502 001402

6940

6941 022504 011203

6942 022506 104001

6943

6944

6945

6946

6947 022510

6948 022510 000004

6949 022512 012700

6950 022516 013701

6951 022522 012702

6952 022526 005004

6953 022530 005012

6954 022532 000257

6955 022534 000262

6956

6957 022536 006312

6958

000331
022464
063312
100000
040000

000332
022536
063312

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 TST331 ;;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
:*****
TST331:

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #331,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #100000,R4 ;RESULT S / B = 100000
MOV #40000,(R2) ;[DEST] = 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 TST332 ;;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
:*****
TST332:

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #332,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
CLR (R2) ;[DEST] = 000000
CCC ;CLEAR CODES
SEV ;N:C = 0010

2\$: ASL (R2) ;TEST THE ASL

6959 022540 100403
6960 022542 001002
6961 022544 102401
6962 022546 103001
6963
6964 022550 104001
6965 022552 020412
6966 022554 001402
6967
6968 022556 011203
6969 022560 104001
6970
6971
6972
6973
6974 022562
6975 022562 000004
6976 022564 012700 000333
6977 022570 013701 022614
6978 022574 012702 063312
6979 022600 012704 052525
6980 022604 012712 125252
6981 022610 000257
6982 022612 000275
6983
6984 022614 006112
6985
6986 022616 100403
6987 022620 001402
6988 022622 102001
6989 022624 103401
6990
6991 022626 104001
6992 022630 020412
6993 022632 001402
6994
6995 022634 011203
6996 022636 104001
6997
6998
6999
7000
7001 022640
7002 022640 000004
7003 022642 012700 000334
7004 022646 013701 022672
7005 022652 012702 063312
7006 022656 012704 125253
7007 022662 012712 052525
7008 022666 000257
7009 022670 000265
7010
7011 022672 006112
7012
7013 022674 100003
7014 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 TST333 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
5\$: ERROR 1 ;ASL DELIVERED THE WRONG RESULT

:TEST 333 ROL DM1 TEST - N:C = 1101

TST333:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #333,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #52525,R4 ;RESULT S / B = 52525
MOV #125252,(R2) ;[DEST] = 125252
CCC ;CLEAR CODES
275 ;N:C = 1101

2\$: ROL (R2) ;TEST THE ROL
BMI 3\$;N:C = 0011 ?
BEQ 3\$
BVC 3\$
BCS 4\$
3\$: ERROR 1 ;ROL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ TST334 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
5\$: ERROR 1 ;ROL DELIVERED THE WRONG RESULT

:TEST 334 ROL DM1 TEST - N:C = 0101

TST334:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #334,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125253,R4 ;RESULT S / B = 125253
MOV #52525,(R2) ;[DEST] = 52525
CCC ;CLEAR CODES
265 ;N:C = 0101

2\$: ROL (R2) ;TEST THE ROL
BPL 3\$;N:C = 1010 ?
BEQ 3\$

7015 022700 102001
7016 022702 103001
7017
7018 022704 104001
7019 022706 020412
7020 022710 001402
7021
7022 022712 011203
7023 022714 104001
7024
7025
7026
7027
7028 022716
7029 022716 000004
7030 022720 012700 000335
7031 022724 013701 022744
7032 022730 012702 063312
7033 022734 005004
7034 022736 005012
7035 022740 000257
7036 022742 000262
7037
7038 022744 006112
7039
7040 022746 100403
7041 022750 001002
7042 022752 102401
7043 022754 103001
7044
7045 022756 104001
7046 022760 020412
7047 022762 001402
7048
7049 022764 011203
7050 022766 104001
7051
7052
7053
7054
7055 022770
7056 022770 000004
7057 022772 012700 000336
7058 022776 013701 023022
7059 023002 012702 063312
7060 023006 012704 100000
7061 023012 012712 077777
7062 023016 000257
7063 023020 000265
7064
7065 023022 005512
7066
7067 023024 100003
7068 023026 001402
7069 023030 102001
7070 023032 103001

BVC 3\$
BCC 4\$
3\$: ERROR 1 ;ROL FAILED TO ALTER THE CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT OK ?
BEQ TST335 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
5\$: ERROR 1 ;ROL DELIVERED THE WRONG RESULT

:TEST 335 ROL DM1 TEST - N:C = 0010

TST335:

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #335,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
CLR (R2) ;[DEST] = 000000
CCC ;CLEAR CODES
SEV ;N:C = 0010

2\$: ROL (R2) ;TEST THE ROL
BNI 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 TST336 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
5\$: ERROR 1 ;ROL DELIVERED THE WRONG RESULT

:TEST 336 ADC DM1 TEST - N:C = 0101

TST336:

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #336,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #100000,R4 ;RESULT S / B = 100000
MOV #77777,(R2) ;[DEST] = 77777
CCC ;CLEAR CODES
265 ;N:C = 0101

2\$: ADC (R2) ;TEST THE ADC
BPL 3\$;N:C = 1010 ?
BEQ 3\$
BVC 3\$
BCC 4\$

7071
 7072 023034 10400i
 7073 023036 020412
 7074 023040 001402
 7075
 7076 023042 011203
 7077 023044 104001
 7078
 7079
 7080
 7081
 7082 023046
 7083 023046 000004
 7084 023050 012700 000337
 7085 023054 013701 023076
 7086 023060 012702 063312
 7087 023064 005004
 7088 023066 012712 177777
 7089 023072 000257
 7090 023074 000273
 7091
 7092 023076 005512
 7093
 7094 023100 100403
 7095 023102 001002
 7096 023104 102401
 7097 023106 103401
 7098
 7099 023110 104001
 7100 023112 020412
 7101 023114 001402
 7102
 7103 023116 011203
 7104 023120 104001
 7105
 7106
 7107
 7108
 7109 023122
 7110 023122 000004
 7111 023124 012700 000340
 7112 023130 013701 023154
 7113 023134 012702 063312
 7114 023140 012704 177777
 7115 023144 012712 177777
 7116 023150 000257
 7117 023152 000272
 7118
 7119 023154 005512
 7120
 7121 023156 100003
 7122 023160 001402
 7123 023162 102401
 7124 023164 103001
 7125
 7126 023166 104001

3\$: ERROR 1 ;ADC FAILED TO ALTER THE CODES PROPERLY
 4\$: CMP R4,(R2) ;RESULT OK ?
 BEQ TST337 ;:BR IF YES

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

:::*****
 ;*TEST 337 ADC DM1 TEST - N:C = 1011
 :::*****
 TST337:

SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #337,R0 ;:LOAD R0 WITH TEST NUMBER
 MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
 CLR R4 ;RESULT S / B = 000000
 MOV #-1,(R2) ;[DEST] = 177777
 CCC ;CLEAR CODES
 273 ;N:C = 1011

2\$: ADC (R2) ;TEST THE ADC
 BMI 3\$;N:C = 0101 ?
 BNE 3\$
 BVS 3\$
 BCS 4\$

3\$: ERROR 1 ;ADC FAILED TO ALTER THE CODES PROPERLY
 4\$: CMP R4,(R2) ;RESULT OK ?
 BEQ TST340 ;:BR IF YES

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

:::*****
 ;*TEST 340 ADC DM1 TEST - N:C = 1010
 :::*****
 TST340:

SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #340,R0 ;:LOAD R0 WITH TEST NUMBER
 MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
 MOV #-1,R4 ;RESULT S / B = 177777
 MOV #-1,(R2) ;[DEST] = 177777
 CCC ;CLEAR CODES
 272 ;N:C = 1010

2\$: ADC (R2) ;TEST THE ADC
 BPL 3\$;N:C = 1000 ?
 BEQ 3\$
 BVS 3\$
 BCC 4\$

3\$: ERROR 1 ;ADC FAILED TO ALTER THE CODES PROPERLY

```

7127 023170 020412
7128 023172 001402
7129
7130 023174 011203
7131 023176 104001
7132
7133
7134
7135
7136 023200
7137 023200 000004
7138 023202 012700 000341
7139 023206 013701 023230
7140 023212 012702 063312
7141 023216 005004
7142 023220 012712 000001
7143 023224 000257
7144 023226 000273
7145
7146 023230 005612
7147
7148 023232 100403
7149 023234 001002
7150 023236 102401
7151 023240 103001
7152
7153 023242 104001
7154 023244 020412
7155 023246 001402
7156
7157 023250 011203
7158 023252 104001
7159
7160
7161
7162
7163 023254
7164 023254 000004
7165 023256 012700 000342
7166 023262 013701 023306
7167 023266 012702 063312
7168 023272 012704 077777
7169 023276 012712 100000
7170 023302 000257
7171 023304 000265
7172
7173 023306 005612
7174
7175 023310 100403
7176 023312 001402
7177 023314 102001
7178 023316 103001
7179
7180 023320 104001
7181 023322 020412
7182 023324 001402
  
```

```

4$:  CMP      R4,(R2)      ;RESULT OK ?
      BEQ      TST341      ;;BR IF YES

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

;*****
;*TEST 341      SBC DM1 TEST - N:C = 1011
;*****
TST341:
      SCOPE                    ;CALL THE SCOPE LOOP UTILITY
      MOV      #341,R0        ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
      CLR      R4             ;RESULT S / B = 000000
      MOV      #1,(R2)        ;[DEST] = +1
      CCC                      ;CLEAR CODES
      273                    ;N:C = 1011

2$:  SBC      (R2)          ;TEST THE SBC

      BMI      3$            ;N:C = 0100 ?
      BNE      3$
      BVS      3$
      BCC      4$

3$:  ERROR    1           ;SBC FAILED TO ALTER THE CODES PROPERLY
4$:  CMP      R4,(R2)      ;RESULT OK ?
      BEQ      TST342      ;;BR IF YES

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

;*****
;*TEST 342      SBC DM1 TEST - N:C = 0101
;*****
TST342:
      SCOPE                    ;CALL THE SCOPE LOOP UTILITY
      MOV      #342,R0        ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
      MOV      #077777,R4     ;RESULT S / B = 077777
      MOV      #100000,(R2)   ;[DEST] = 100000
      CCC                      ;CLEAR CODES
      265                    ;N:C = 0101

2$:  SBC      (R2)          ;TEST THE SBC

      BMI      3$            ;N:C = 0010 ?
      BEQ      3$
      BVC      3$
      BCC      4$

3$:  ERROR    1           ;SBC FAILED TO ALTER THE CODES PROPERLY
4$:  CMP      R4,(R2)      ;RESULT OK ?
      BEQ      TST343      ;;BR IF YES
  
```


7183
7184 023326 011203
7185 023330 104001
7186
7187
7188
7189
7190 023332
7191 023332 000004
7192 023334 012700 000343
7193 023340 013701 023364
7194 023344 012702 063312
7195 023350 012704 000001
7196 023354 012712 000001
7197 023360 000257
7198 023362 000276
7199
7200 023364 005612
7201
7202 023366 100403
7203 023370 001402
7204 023372 102401
7205 023374 103001
7206
7207 023376 104001
7208 023400 020412
7209 023402 001402
7210
7211 023404 011203
7212 023406 104001
7213
7214
7215
7216
7217 023410
7218 023410 000004
7219 023412 012700 000344
7220 023416 013701 023440
7221 023422 012702 063312
7222 023426 012704 177777
7223 023432 005012
7224 023434 000257
7225 023436 000267
7226
7227 023440 005612
7228
7229 023442 100003
7230 023444 001402
7231 023446 102401
7232 023450 103401
7233
7234 023452 104001
7235 023454 020412
7236 023456 001402
7237
7238 023460 011203

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

;*****
; *TEST 343 SBC DM1 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) ;[DEST] = 1
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
5$: ERROR 1 ;SBC DELIVERED THE WRONG RESULT

;*****
; *TEST 344 SBC DM1 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 = 177777
CLR (R2) ;[DEST] = 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
```

7239 023462 104001
7240
7241
7242
7243
7244 023464
7245 023464 000004
7246 023466 012700 000345
7247 023472 013701 023512
7248 023476 012704 177776
7249 023502 012703 177402
7250 023506 000257
7251 023510 000266
7252
7253 023512 105403
7254
7255 023514 100003
7256 023516 001402
7257 023520 102401
7258 023522 103401
7259
7260 023524 104002
7261
7262 023526 020403
7263 023530 001401
7264
7265 023532 104002
7266
7267
7268
7269
7270 023534
7271 023534 000004
7272 023536 012700 000346
7273 023542 013701 023562
7274 023546 012704 177400
7275 023552 012703 177400
7276 023556 000257
7277 023560 000263
7278
7279 023562 105403
7280
7281 023564 100403
7282 023566 001002
7283 023570 102401
7284 023572 103001
7285
7286 023574 104002
7287
7288 023576 020403
7289 023600 001401
7290
7291 023602 104002
7292
7293
7294

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 ;:[DEST] = 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
5\$: 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 ;:[DEST] = 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
5\$: ERROR 2 ;NEGB DELIVERED THE WRONG RESULT
:*****
:*TEST 347 NEGB - MODE 0 TEST - N:C = 1101

7295
7296 023604
7297 023604 000004
7298 023606 012700 000347
7299 023612 013701 023632
7300 023616 012704 177600
7301 023622 012703 177600
7302 023626 000257
7303 023630 000275
7304
7305 023632 105403
7306
7307 023634 100003
7308 023636 001402
7309 023640 102001
7310 023642 103401
7311
7312 023644 104002
7313
7314 023646 020403
7315 023650 001401
7316
7317 023652 104002
7318
7319
7320
7321
7322 023654
7323 023654 000004
7324 023656 012700 000350
7325 023662 013701 023702
7326 023666 012704 177400
7327 023672 012703 177777
7328 023676 000257
7329 023700 000273
7330
7331 023702 105003
7332
7333 023704 100403
7334 023706 001002
7335 023710 102401
7336 023712 103001
7337
7338 023714 104002
7339
7340 023716 020403
7341 023720 001401
7342
7343 023722 104002
7344
7345
7346
7347
7348 023724
7349 023724 000004
7350 023726 012700 000351

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

```

7351 023732 013701 023752      MOV    @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
7352 023736 012704 177400      MOV    #177400,R4  ;RESULT S / B = 000 (LO BYTE)
7353 023742 012703 177777      MOV    #-1,R3      ;[DEST] = 177777
7354 023746 000257              CCC                ;CLEAR FLAGS
7355 023750 000264              SEZ                ;N:C = 0100
7356
7357 023752 105003      2$:   CLR B    R3      ;TEST THE CLR B
7358
7359 023754 100403      BMI    3$           ;N:C = 0100 ?
7360 023756 001002      BNE    3$
7361 023760 102401      BVS    3$
7362 023762 103001      BCC    4$
7363
7364 023764 104002      3$:   ERROR    2      ;CLR B FAILED TO SET CODES PROPERLY
7365
7366 023766 020403      4$:   CMP    R4,R3      ;RESULT CORRECT ?
7367 023770 001401      BEQ    TST352       ;:BR IF YES
7368
7369 023772 104002      5$:   ERROR    2      ;CLR B DELIVERED THE WRONG RESULT
7370
7371
7372
7373
7374 023774
7375 023774 000004
7376 023776 012700 000352      SCOPE              ;CALL THE SCOPE LOOP UTILITY
7377 024002 013701 024032      MOV    #352,R0      ;:LOAD R0 WITH TEST NUMBER
7378 024006 012702 063313      MOV    @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
7379 024012 012704 000377      MOV    #MBUFO+1,R2  ;DEST ADDR = MBUFO+1
7380 024016 012705 063312      MOV    #377,R4      ;RESULT S / B = 377
7381 024022 010203      MOV    #MBUFO,R5    ;POINT R5 TO CHECK RESULT
7382 024024 012715 177777      MOV    R2,R3        ;R3 CONTAINS DEST ADDR
7383 024030 000257      MOV    #-1,(R5)     ;[DEST] = 177777
7384
7385 024032 105023      2$:   CLR B    (R3)+   ;TEST THE CLR B
7386
7387 024034 022703 063314      CMP    #MBUFO+2,R3  ;DID DEST REG GET INCREMENTED ?
7388 024040 001401      BEQ    4$           ;BR IF YES
7389
7390 024042 104005      3$:   ERROR    5      ;CLR B FAILED TO UPDATE DEST REG
7391
7392 024044 020415      4$:   CMP    R4,(R5)  ;CORRECT RESULT ?
7393 024046 001402      BEQ    TST353       ;:BR IF YES
7394
7395 024050 011503      MOV    (R5),R3      ;GET THE WAS DATA
7396 024052 104001      5$:   ERROR    1      ;CLR B DELIVERED WRONG RESULT
7397
7398
7399
7400
7401 024054
7402 024054 000004
7403 024056 012700 000353      SCOPE              ;CALL THE SCOPE LOOP UTILITY
7404 024062 013701 024112      MOV    #353,R0      ;:LOAD R0 WITH TEST NUMBER
7405 024066 012702 063313      MOV    @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
7406 024072 012704 000377      MOV    #MBUFO+1,R2  ;DEST ADDR = MBUFO+1
                          MOV    #377,R4      ;RESULT S / B = 377

```

7407 024076 012705 063312
7408 024102 010203
7409 024104 012715 177777
7410 024110 000257
7411
7412 024112 105013
7413
7414 024114 020415
7415 024116 001402
7416
7417 024120 011503
7418 024122 104001
7419
7420
7421
7422
7423 024124
7424 024124 000004
7425 024126 012700 000354
7426 024132 013701 024156
7427 024136 012702 063312
7428 024142 012704 177400
7429 024146 010203
7430 024150 012712 177777
7431 024154 000257
7432
7433 024156 105023
7434
7435 024160 022703 063313
7436 024164 001401
7437
7438 024166 104005
7439
7440 024170 020412
7441 024172 001402
7442
7443 024174 011203
7444 024176 104001
7445
7446
7447
7448
7449 024200
7450 024200 000004
7451 024202 012700 000355
7452 024206 013701 024232
7453 024212 012702 063312
7454 024216 012704 177400
7455 024222 010203
7456 024224 012712 177777
7457 024230 000257
7458
7459 024232 105013
7460
7461 024234 020412
7462 024236 001402

MOV #MBUFO,R5 ;POINT R5 TO CHECK RESULT
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R5) ;[DEST] = 177777
CCC ;SCOPE SYNC
2\$: CLR B (R3) ;TEST THE CLR B
CMP R4,(R5) ;CORRECT RESULT ?
BEQ TST354 ;:BR IF YES
3\$: MOV (R5),R3 ;GET THE WAS DATA
ERROR 1 ;CLR B DELIVERED WRONG RESULT
;*****
;*TEST 354 CLR B TEST - DM2 - EVEN ADDRESS
;*****
TST354:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #354,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #177400,R4 ;RESULT S / B = 177400
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R2) ;[DEST] = 177777
CCC ;SCOPE SYNC
2\$: CLR B (R3)+ ;TEST THE CLR B
CMP #MBUFO+1,R3 ;DID DEST REG GET INCREMENTED ?
BEQ 4\$;BR IF YES
3\$: ERROR 5 ;CLR B FAILED TO UPDATE DEST REG
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST355 ;:BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CLR B DELIVERED WRONG RESULT
;*****
;*TEST 355 CLR B TEST - DM1 - EVEN ADDRESS
;*****
TST355:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #355,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #177400,R4 ;RESULT S / B = 177400
MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R2) ;[DEST] = 177777
CCC ;SCOPE SYNC
2\$: CLR B (R3) ;TEST THE CLR B
CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST356 ;:BR IF YES

```

7463
7464 024240 011203
7465 024242 104001
7466
7467
7468
7469
7470 024244
7471 024244 000004
7472 024246 012700 000356
7473 024252 013701 024302
7474 024256 012702 063313
7475 024262 012704 000777
7476 024266 012705 063312
7477 024272 010203
7478 024274 012715 177777
7479 024300 000257
7480
7481 024302 105423
7482
7483 024304 022703 063314
7484 024310 001401
7485
7486 024312 104005
7487
7488 024314 020415
7489 024316 001402
7490
7491 024320 011503
7492 024322 104001
7493
7494
7495
7496
7497 024324
7498 024324 000004
7499 024326 012700 000357
7500 024332 013701 024374
7501
7502 024336 032737 000200 063234
7503 024344 001401
7504 024346 000000
7505 024350 012702 063313
7506 024354 012704 000777
7507 024360 012705 063312
7508 024364 010203
7509 024366 012715 177777
7510 024372 000257
7511
7512 024374 105413
7513
7514 024376 020415
7515 024400 001402
7516
7517 024402 011503

      MOV      (R2),R3      ;GET THE WAS DATA
3$:   ERROR    1            ;CLR8 DELIVERED WRONG RESULT

;*****
;*TEST 356      NEGB TEST - DM2 - ODD ADDRESS
;*****
TST356:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #356,R0      ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #MBUF0+1,R2  ;DEST ADDR = MBUF0+1
      MOV      #777,R4      ;RESULT S / B = 777
      MOV      #MBUF0,R5    ;POINT R5 TO CHECK RESULT
      MOV      R2,R3        ;R3 CONTAINS DEST ADDR
      MOV      #-1,(R5)     ;[DEST] = 177777
      CCC                  ;SCOPE SYNC

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

      CMP      #MBUF0+2,R3  ;DID DEST REG GET INCREMENTED ?
      BEQ     4$            ;BR IF YES

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

4$:   CMP      R4,(R5)     ;CORRECT RESULT ?
      BEQ     TST357       ;;BR IF YES

5$:   MOV      (R5),R3     ;GET THE WAS DATA
      ERROR    1            ;NEGB DELIVERED WRONG RESULT

;*****
;*TEST 357      NEGB TEST - DM1 - ODD ADDRESS
;*****
TST357:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #357,R0      ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
      .SBTTL  USER CONTROLLED BREAKPOINT -- BIT7
      BIT      #BIT7,@#BPTLOC ;BREAKPOINT HALT SET ??
      BEQ     .+4           ;BR IF NOT
      HALT                    ;BREAK - DEPRESS CONTINUE TO RESTART
      MOV      #MBUF0+1,R2  ;DEST ADDR = MBUF0+1
      MOV      #777,R4      ;RESULT S / B = 777
      MOV      #MBUF0,R5    ;POINT R5 TO CHECK RESULT
      MOV      R2,R3        ;R3 CONTAINS DEST ADDR
      MOV      #-1,(R5)     ;[DEST] = 177777
      CCC                  ;SCOPE SYNC

2$:   NEGB     (R3)         ;TEST THE NEGB

      CMP      R4,(R5)     ;CORRECT RESULT ?
      BEQ     TST360       ;;BR IF YES

      MOV      (R5),R3     ;GET THE WAS DATA
  
```

7518 024404 104001
7519
7520
7521
7522
7523 024406
7524 024406 000004
7525 024410 012700 000360
7526 024414 013701 024440
7527 024420 012702 063312
7528 024424 012704 177401
7529 024430 010203
7530 024432 012712 177777
7531 024436 000257
7532
7533 024440 105423
7534
7535 024442 022703 063313
7536 024446 001401
7537
7538 024450 104005
7539
7540 024452 020412
7541 024454 001402
7542
7543 024456 011203
7544 024460 104001
7545
7546
7547
7548
7549 024462
7550 024462 000004
7551 024464 012700 000361
7552 024470 013701 024514
7553 024474 012702 063312
7554 024500 012704 177401
7555 024504 010203
7556 024506 012712 177777
7557 024512 000257
7558
7559 024514 105413
7560
7561 024516 020412
7562 024520 001402
7563
7564 024522 011203
7565 024524 104001
7566
7567
7568
7569
7570 024526
7571 024526 000004
7572 024530 012700 000362
7573 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 @#2\$,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) ;:[DEST] = 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
MOV (R2),R3 ;GET THE WAS DATA
5\$: 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 @#2\$,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) ;:[DEST] = 177777
CCC ;SCOPE SYNC
2\$: NEGB (R3) ;TEST THE NEGB
CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST362 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
3\$: 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 @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD

7574 024540 005004
7575 024542 012705 177777
7576 024546 012703 000001
7577 024552 000257
7578 024554 000272
7579

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

7580 024556 060503
7581
7582 024560 100403
7583 024562 001002
7584 024564 102401
7585 024566 103401
7586

2\$: ADD R5,R3 ;TEST THE ADD
BMI 3\$;N:C = 0101
BNE 3\$
BVS 3\$
BCS 4\$

7587 024570 104002
7588
7589 024572 020403
7590 024574 001401
7591
7592 024576 104002
7593

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

7594
7595
7596
7597 024600

;*TEST 363 ADD TEST - SMO,DMO - N:C = 0101

TST363:

7598 024600 000004
7599 024602 012700 000363
7600 024606 013701 024632
7601 024612 012704 100006
7602 024616 012705 077777
7603 024622 012703 000007
7604 024626 000257
7605 024630 000265
7606

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #363,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100006,R4 ;RESULT S / B = 100006
MOV #77777,R5 ;SRC OPR = 77777
MOV #7,R3 ;[DEST] = 7
CCC ;CLEAR FLAGS
265 ;N:C = 0101

7607 024632 060503
7608
7609 024634 100003
7610 024636 001402
7611 024640 102001
7612 024642 103001
7613

2\$: ADD R5,R3 ;TEST THE ADD
BPL 3\$;N:C = 1010
BEQ 3\$
BVC 3\$
BCC 4\$

7614 024644 104002
7615
7616 024646 020403
7617 024650 001401
7618
7619 024652 104002
7620

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

7621
7622
7623
7624 024654

;*TEST 364 ADD SM1,DMO TEST

TST364:

7625 024654 000004
7626 024656 012700 000364
7627 024662 013701 024702
7628 024666 012704 063322
7629 024672 012705 063276

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #364,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #DWTA,R4 ;RESULT S / B = #DWTA
MOV #ATA,R5 ;SOURCE ADDR = ATA


```
7630 024676 005003          CLR    R3          ;[DEST] = 0
7631 024700 000257          CCC          ;SCOPE SYNC
7632
7633 024702 061503    2$:  ADD    (R5),R3      ;TEST THE ADD - SM1,DMO
7634
7635 024704 020403          CMP    R4,R3      ;RESULT = #DWTA?
7636 024706 001401          BEQ    4$         ;BR IF YES
7637
7638 024710 104002    3$:  ERROR  2          ;ADD DELIVERED WRONG RESULT
7639
7640 024712 022705 063276    4$:  CMP    #ATA,R5      ;DID ADD CHANGE REG.
7641 024716 001401          BEQ    TST365     ;:BR IF NOT
7642
7643 024720 104005    5$:  ERROR  5          ;REG GOT MODIFIED
7644
7645
7646
7647
7648 024722
7649 024722 000004          :*****
7650 024724 012700 000365    :*TEST 365      ADD SM2,DMO TEST
7651 024730 013701 024750    :*****
7652 024734 012704 063322    TST365:
7653 024740 012705 063276          SCOPE          ;CALL THE SCOPE LOOP UTILITY
7654 024744 005003          MOV    #365,R0    ;:LOAD R0 WITH TEST NUMBER
7655 024746 000257          MOV    @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
7656
7657 024750 062503    2$:  ADD    (R5)+,R3    ;TEST THE ADD - SM2,DMO
7658
7659 024752 020403          CMP    R4,R3      ;RESULT = #DWTA
7660 024754 001401          BEQ    4$         ;BR IF YES
7661
7662 024756 104002    3$:  ERROR  2          ;ADD DELIVERED WRONG RESULT
7663
7664 024760 022705 063300    4$:  CMP    #ATA+2,R5  ;DID ADD AUTO INCREMENT SOURCE REG?
7665 024764 001401          BEQ    TST366     ;:BR IF YES
7666
7667 024766 104005    5$:  ERROR  5          ;ADD FAILED TO UPDATE SOURCE REG.
7668
7669
7670
7671
7672 024770
7673 024770 000004          :*****
7674 024772 012700 000366    :*TEST 366      ADD SM3,DMO TEST
7675 024776 013701 025022    :*****
7676 025002 012704 063322    TST366:
7677 025006 012705 063306          SCOPE          ;CALL THE SCOPE LOOP UTILITY
7678 025012 010437 063312          MOV    #366,R0    ;:LOAD R0 WITH TEST NUMBER
7679 025016 005003          MOV    @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
7680 025020 000257          MOV    #DWTA,R4   ;RESULT S / B = #DWTA
7681
7682 025022 063503    2$:  ADD    @ (R5)+,R3  ;TEST THE ADD - SM3,DMO
7683
7684 025024 020437 063312          MOV    #ATA+10,R5 ;R5 POINTS TO SOURCE ADDR
7685 025030 001401          MOV    R4,@#MBOF0 ;[SOURCE] = #DWTA
                          CLR    R3          ;[DEST] = 0
                          CCC          ;SCOPE SYNC
```

```
7686
7687 025032 104002 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7688
7689 025034 022705 063310 4$: CMP #ATA+12,R5 ;DID ADD AUTO INCREMENT SOURCE REG?
7690 025040 001401 BEQ TST367 ;;BR IF YES
7691
7692 025042 104005 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.
7693
7694 ::*****
7695 :*TEST 367 ADD SM4,DMO TEST
7696 ::*****
7697 TST367:
7698 025044 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
7699 025046 012700 000367 MOV #367,R0 ;;LOAD R0 WITH TEST NUMBER
7700 025052 013701 025072 MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
7701 025056 012704 063322 MOV #DWTA,R4 ;RESULT S / B = #DWTA
7702 025062 012705 063300 MOV #ATA+2,R5 ;SOURCE ADDR = ATA
7703 025066 005003 CLR R3 ;[DEST] = 0
7704 025070 000257 CCC ;SCOPE SYNC
7705
7706 025072 064503 2$: ADD -(R5),R3 ;TEST THE ADD - SM4,DMO
7707
7708 025074 020403 CMP R4,R3 ;RESULT = #DWTA?
7709 025076 001401 BEQ 4$ ;BR IF YES
7710
7711 025100 104002 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7712
7713 025102 022705 063276 4$: CMP #ATA,R5 ;DID SOURCE REG GET DECREMENTED?
7714 025106 001401 BEQ TST370 ;;BR IF YES
7715
7716 025110 104005 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG
7717
7718 ::*****
7719 :*TEST 370 ADD SM5,DMO TEST
7720 ::*****
7721 TST370:
7722 025112 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
7723 025114 012700 000370 MOV #370,R0 ;;LOAD R0 WITH TEST NUMBER
7724 025120 013701 025144 MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
7725 025124 012704 063322 MOV #DWTA,R4 ;RESULT S / B = #DWTA
7726 025130 012705 063310 MOV #ATA+12,R5 ;R5 POINTS TO SOURCE ADDR
7727 025134 010437 063312 MOV R4,@#MBUFO ;[SOURCE] = #DWTA
7728 025140 005003 CLR R3 ;[DEST] = 0
7729 025142 000257 CCC ;SCOPE SYNC
7730
7731 025144 065503 2$: ADD @-(R5),R3 ;TEST THE ADD - SM5,DMO
7732
7733 025146 020437 063312 CMP R4,@#MBUFO ;RESULT = #DWTA?
7734 025152 001401 BEQ 4$ ;BR IF YES
7735
7736 025154 104002 3$: ERROR 2 ;ADD DELIVERED WRONG RESULT
7737
7738 025156 022705 063306 4$: CMP #ATA+10,R5 ;DID ADD DECREMENT SOURCE REG?
7739 025162 001401 BEQ TST371 ;;BR IF YES
7740
7741 025164 104005 5$: ERROR 5 ;ADD FAILED TO UPDATE SOURCE REG.
```

7742
7743
7744
7745
7746 025166
7747 025166 000004
7748 025170 012700 000371
7749 025174 013701 025214
7750 025200 012704 063312
7751 025204 012705 063276
7752 025210 005003
7753 025212 000257
7754
7755 025214 066503 000010
7756
7757 025220 020403
7758 025222 001401
7759
7760 025224 104002
7761
7762
7763
7764
7765 025226
7766 025226 000004
7767 025230 012700 000372
7768 025234 013701 025260
7769 025240 012704 063322
7770 025244 012705 063276
7771 025250 010437 063312
7772 025254 005003
7773 025256 000257
7774
7775 025260 067503 000010
7776
7777 025264 020403
7778 025266 001401
7779
7780 025270 104002
7781
7782
7783
7784
7785 025272
7786 025272 000004
7787 025274 012700 000373
7788 025300 013701 025324
7789 025304 012702 063312
7790 025310 012704 063322
7791 025314 012705 063276
7792 025320 005012
7793 025322 000257
7794
7795 025324 061512
7796
7797 025326 020412

```
*****  
: *TEST 371      ADD SM6,DMO TEST  
*****  
TST371:  
      SCOPE                ;CALL THE SCOPE LOOP UTILITY  
      MOV      #371,R0      ;:LOAD R0 WITH TEST NUMBER  
      MOV      @#2$,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,DMO  
  
      CMP      R4,R3        ;RESULT =MBUF0?  
      BEQ      TST372      ;:BR IF YES  
  
3$:   ERROR   2             ;ADD DELIVERED WRONG RESULT  
  
*****  
: *TEST 372      ADD SM7,DMO TEST  
*****  
TST372:  
      SCOPE                ;CALL THE SCOPE LOOP UTILITY  
      MOV      #372,R0      ;:LOAD R0 WITH TEST NUMBER  
      MOV      @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD  
      MOV      #DWTA,R4     ;:RESULT S / B = #DWTA  
      MOV      #ATA,R5      ;:BASE SOURCE ADDR = ATA  
      MOV      R4,@#MBUF0   ;:[SOURCE] = #DWTA  
      CLR      R3           ;:[DEST] = 0  
      CCC                      ;SCOPE SYNC  
  
2$:   ADD      @10(R5),R3   ;TEST THE ADD - SM7,DMO  
  
      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      @#2$,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?
```

7798 025330 001402
 7799
 7800 025332 011203
 7801 025334 104001
 7802
 7803
 7804
 7805
 7806 025336
 7807 025336 000004
 7808 025340 012700 000374
 7809 025344 013701 025370
 7810 025350 012702 063312
 7811 025354 012704 063322
 7812 025360 012705 063276
 7813 025364 005012
 7814 025366 000257
 7815
 7816 025370 062512
 7817
 7818 025372 020412
 7819 025374 001402
 7820
 7821 025376 011203
 7822 025400 104001
 7823
 7824
 7825
 7826
 7827 025402
 7828 025402 000004
 7829 025404 012700 000375
 7830 025410 013701 025436
 7831 025414 012702 063312
 7832 025420 012704 063322
 7833 025424 012705 063276
 7834 025430 010203
 7835 025432 005012
 7836 025434 000257
 7837
 7838 025436 061523
 7839
 7840 025440 020412
 7841 025442 001406
 7842
 7843 025444 010337 063316
 7844 025450 011203
 7845 025452 104001
 7846
 7847 025454 013703 063316
 7848 025460 022703 063314
 7849 025464 001401
 7850
 7851 025466 104005
 7852
 7853

```

      BEQ      TST374          ;;BR IF YES
      MOV      (R2),R3        ;GET WAS DATA
3$:  ERROR    1              ;ADD DELIVERED WRONG RESULT
      ;*****
      ;*TEST 374  ADD SM2,DM1 TEST
      ;*****
TST374:
      SCOPE                    ;CALL THE SCOPE LOOP UTILITY
      MOV      #374,R0        ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,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

      MOV      (R2),R3        ;GET WAS DATA
3$:  ERROR    1              ;ADD DELIVERED WRONG RESULT
      ;*****
      ;*TEST 375  ADD SM1,DM2 TEST
      ;*****
TST375:
      SCOPE                    ;CALL THE SCOPE LOOP UTILITY
      MOV      #375,R0        ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,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      R2,R3          ;[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

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

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

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

```

7854
7855
7856 025470
7857 025470 000004
7858 025472 012700 000376
7859 025476 013701 025524
7860 025502 012702 063312
7861 025506 012704 063322
7862 025512 012705 063276
7863 025516 010203
7864 025520 005012
7865 025522 000257
7866
7867 025524 062523
7868
7869 025526 020412
7870 025530 001406
7871
7872 025532 010337 063316
7873 025536 011203
7874 025540 104001
7875
7876 025542 013703 063316
7877 025546 022703 063314
7878 025552 001401
7879
7880 025554 104005
7881
7882
7883
7884
7885 025556
7886 025556 000004
7887 025560 012700 000377
7888 025564 013701 025614
7889 025570 012702 063312
7890 025574 012704 063322
7891 025600 012705 063276
7892 025604 012703 063306
7893 025610 005012
7894 025612 000257
7895
7896 025614 061533
7897
7898 025616 020412
7899 025620 001406
7900
7901 025622 010337 063316
7902 025626 011203
7903 025630 104001
7904
7905 025632 013703 063316
7906 025636 022703 063310
7907 025642 001401
7908
7909 025644 104005
  
```

```

;*TEST 376      ADD SM2,DM2 TEST
:*****
TST376:
      SCOPE      ;CALL THE SCOPE LOOP UTILITY
MOV    #376,R0   ;:LOAD R0 WITH TEST NUMBER
MOV    @#2$,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    R2,R3     ;:[R3] = DEST ADDR
CLR    (R2)      ;:[DEST] = 0
CCC                               ;:SCOPE SYNC

2$:   ADD    (R5)+,(R3)+ ;TEST THE ADD - SM2,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    TST377       ;:BR IF YES

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

:*****
;*TEST 377      ADD SM1,DM3 TEST
:*****
TST377:
      SCOPE      ;CALL THE SCOPE LOOP UTILITY
MOV    #377,R0   ;:LOAD R0 WITH TEST NUMBER
MOV    @#2$,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    #ATA+10,R3 ;:[R3] = ADDR OF DEST ADDR
CLR    (R2)      ;:[DEST] = 0
CCC                               ;:SCOPE SYNC

2$:   ADD    (R5),@(R3)+ ;TEST THE ADD - SM1,DM3

      CMP    R4,(R2)   ;:RESULT = #DWTA?
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+12,R3 ;:DID ADD INCREMENT DEST REG
BEQ    TST400       ;:BR IF YES

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

7910
 7911
 7912
 7913
 7914 025646
 7915 025646 000004
 7916 025650 012700 000400
 7917 025654 013701 025704
 7918 025660 012702 063312
 7919 025664 012704 063322
 7920 025670 012705 063276
 7921 025674 012703 063306
 7922 025700 005012
 7923 025702 000257
 7924
 7925 025704 062533
 7926
 7927 025706 020412
 7928 025710 001406
 7929
 7930 025712 010337 063316
 7931 025716 011203
 7932 025720 104001
 7933
 7934 025722 013703 063316
 7935 025726 022703 063310
 7936 025732 001401
 7937
 7938 025734 104005
 7939
 7940
 7941
 7942
 7943 025736
 7944 025736 000004
 7945 025740 012700 000401
 7946 025744 013701 025774
 7947 025750 012702 063312
 7948 025754 012704 063322
 7949 025760 012705 063276
 7950 025764 012703 063314
 7951 025770 005012
 7952 025772 000257
 7953
 7954 025774 061543
 7955
 7956 025776 020412
 7957 026000 001406
 7958
 7959 026002 010337 063316
 7960 026006 011203
 7961 026010 104001
 7962
 7963 026012 013703 063316
 7964 026016 020302
 7965 026020 001401

```

:*****
:TEST 400      ADD SM2,DM3 TEST
:*****
TST400:
SCOPE                ;CALL THE SCOPE LOOP UTILITY
MOV #400,R0          ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,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 #ATA+10,R3       ;:[R3] = ADDR OF DEST ADDR
CLR (R2)             ;:[DEST] = 0
CCC                  ;:SCOPE SYNC

2$:  ADD (R5)+,@(R3)+ ;:TEST THE ADD - SM2,DM3

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

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

MOV @#MBUF1,R3       ;:RESTORE R3
CMP #ATA+12,R3       ;:DID ADD INCREMENT DEST REG
BEQ TST401           ;:BR IF YES

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

:*****
:TEST 401      ADD SM1,DM4 TEST
:*****
TST401:
SCOPE                ;CALL THE SCOPE LOOP UTILITY
MOV #401,R0          ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,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 #MBUF0+2,R3      ;:R3 POINTS TO DEST ADDR +2
CLR (R2)             ;:[DEST] = 0
CCC                  ;:SCOPE SYNC

2$:  ADD (R5),-(R3)   ;:TEST THE ADD - SM1,DM4

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

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

MOV @#MBUF1,R3       ;:RESTORE R3
CMP R3,R2            ;:DID ADD INCREMENT DEST REG?
BEQ TST402           ;:BR IF YES
  
```

```

7966
7967 026022 104005      5$:      ERROR      5      ;ADD FAILED TO UPDATE DEST REG.
7968
7969
7970
7971
7972 026024
7973 026024 000004
7974 026026 012700 000402
7975 026032 013701 026062
7976 026036 012702 063312
7977 026042 012704 063322
7978 026046 012705 063276
7979 026052 012703 063314
7980 026056 005012
7981 026060 000257
7982
7983 026062 061543      2$:      ADD      (R5),-(R3)      ;TEST THE ADD - SM2,DM4
7984
7985 026064 020412
7986 026066 001406
7987
7988 026070 010337 063316
7989 026074 011203
7990 026076 104001      3$:      ERROR      1      ;ADD DELIVERED WRONG RESULT
7991
7992 026100 013703 063316
7993 026104 020302      4$:      CMP      R4,(R2)      ;RESULT = #DWTA?
7994 026106 001401      BEQ      4$      ;BR IF YES
7995
7996 026110 104005      5$:      ERROR      5      ;ADD FAILED TO UPDATE DEST REG.
7997
7998
7999
8000
8001 026112
8002 026112 000004
8003 026114 012700 000403
8004 026120 013701 026150
8005 026124 012702 063312
8006 026130 012704 063322
8007 026134 012705 063276
8008 026140 012703 063310
8009 026144 005012
8010 026146 000257
8011
8012 026150 061553      2$:      ADD      (R5),@(R3)      ;TEST THE ADD - SM1,DM5
8013
8014 026152 020412
8015 026154 001406
8016
8017 026156 010337 063316
8018 026162 011203
8019 026164 104001      3$:      ERROR      1      ;ADD DELIVERED WRONG RESULT
8020
8021 026166 013703 063316
  
```

```

*****
*TEST 402      ADD SM2,DM4 TEST
*****
TST402:
  
```

```

SCOPE      ;CALL THE SCOPE LOOP UTILITY
MOV #402,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #DWTA,R4 ;:RESULT S / B = #DWTA
MOV #ATA,R5 ;:SOURCE ADDR = ATA
MOV #MBUFO+2,R3 ;:R3 POINTS TO DEST ADDR +2
CLR (R2) ;:[DEST] = 0
CCC ;SCOPE SYNC
  
```

```

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

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

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

```

*****
*TEST 403      ADD SM1,DM5 TEST
*****
TST403:
  
```

```

SCOPE      ;CALL THE SCOPE LOOP UTILITY
MOV #403,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #DWTA,R4 ;:RESULT S / B = #DWTA
MOV #ATA,R5 ;:SOURCE ADDR = ATA
MOV #ATA+12,R3 ;:R3 CONTAINS ADDR OF DEST ADDR PLUS 2
CLR (R2) ;:[DEST] = 0
CCC ;SCOPE SYNC
  
```

```

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

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

MOV @#MBUF1,R3 ;:RESTORE R3
  
```

8022 026172 022703 063306
 8023 026176 001401
 8024
 8025 026200 104005
 8026
 8027
 8028
 8029
 8030 026202
 8031 026202 000004
 8032 026204 012700 000404
 8033 026210 013701 026240
 8034 026214 012702 063312
 8035 026220 012704 063322
 8036 026224 012705 063276
 8037 026230 012703 063310
 8038 026234 005012
 8039 026236 000257
 8040
 8041 026240 062553
 8042
 8043 026242 020412
 8044 026244 001406
 8045
 8046 026246 010337 063316
 8047 026252 011203
 8048 026254 104001
 8049
 8050 026256 013703 063316
 8051 026262 022703 063306
 8052 026266 001401
 8053
 8054 026270 104005
 8055
 8056
 8057
 8058
 8059 026272
 8060 026272 000004
 8061 026274 012700 000405
 8062 026300 013701 026330
 8063 026304 012702 063316
 8064 026310 012704 063322
 8065 026314 012705 063276
 8066 026320 012703 063312
 8067 026324 005012
 8068 026326 000257
 8069
 8070 026330 061563 000004
 8071
 8072 026334 020412
 8073 026336 001402
 8074
 8075 026340 011203
 8076 026342 104001
 8077

```

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     @#2$,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     #ATA+12,R3    ;R3 CONTAINS ADDR OF DEST ADDR PLUS 2
      CLR     (R2)          ;[DEST] = 0
      CCC                    ;SCOPE SYNC

2$:   ADD     (R5)+,@-(R3)  ;TEST THE ADD - SM2,DM5

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

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

      MOV     @#MBUF1,R3    ;RESTORE R3
4$:   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     @#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV     #MBUF0+4,R2   ;DEST ADDR = MBUF0+4
      MOV     #DWTA,R4      ;RESULT S / B = #DWTA
      MOV     #ATA,R5       ;SOURCE ADDR = ATA
      MOV     #MBUF0,R3     ;[R3] = BASE DEST ADDR
      CLR     (R2)          ;[DEST] = 0
      CCC                    ;SCOPE SYNC

2$:   ADD     (R5),4(R3)    ;TEST THE ADD - SM1,DM6

      CMP     R4,(R2)       ;RESULT = #DWTA?
      BEQ     TST406        ;;BR IF YES

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


8078
8079
8080
8081 026344
8082 026344 000004
8083 026346 012700 000406
8084 026352 013701 026402
8085 026356 012702 063316
8086 026362 012704 063322
8087 026366 012705 063276
8088 026372 012703 063312
8089 026376 005012
8090 026400 000257
8091
8092 026402 062563 000004
8093
8094 026406 020412
8095 026410 001402
8096
8097 026412 011203
8098 026414 104001
8099
8100
8101
8102
8103 026416
8104 026416 000004
8105 026420 012700 000407
8106 026424 013701 026452
8107 026430 012702 063312
8108 026434 012704 063322
8109 026440 012705 063276
8110 026444 010503
8111 026446 005012
8112 026450 000257
8113
8114 026452 061573 000010
8115
8116 026456 020412
8117 026460 001402
8118
8119 026462 011203
8120 026464 104001
8121
8122
8123
8124
8125 026466
8126 026466 000004
8127 026470 012700 000410
8128 026474 013701 026522
8129 026500 012702 063312
8130 026504 012704 063322
8131 026510 012705 063276
8132 026514 010503
8133 026516 005012

```
*****  
*TEST 406      ADD SM2,DM6 TEST  
*****  
TST406:  
      SCOPE                ;CALL THE SCOPE LOOP UTILITY  
      MOV      #406,R0      ;:LOAD R0 WITH TEST NUMBER  
      MOV      @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD  
      MOV      #MBUF0+4,R2  ;:DEST ADDR = MBUF0+4  
      MOV      #DWTA,R4     ;:RESULT S / B = #DWTA  
      MOV      #ATA,R5      ;:SOURCE ADDR = ATA  
      MOV      #MBUF0,R3    ;:[R3] = BASE DEST ADDR  
      CLR      (R2)         ;:[DEST] = 0  
      CCC                     ;SCOPE SYNC  
  
2$:   ADD      (R5)+,4(R3)   ;TEST THE ADD - SM2,DM6  
  
      CMP      R4,(R2)      ;RESULT = #DWTA?  
      BEQ      TST407       ;:BR IF YES  
  
      MOV      (R2),R3      ;GET WAS DATA  
3$:   ERROR   1             ;ADD DELIVERED WRONG RESULT  
  
*****  
*TEST 407      ADD SM1,DM7 TEST  
*****  
TST407:  
      SCOPE                ;CALL THE SCOPE LOOP UTILITY  
      MOV      #407,R0      ;:LOAD R0 WITH TEST NUMBER  
      MOV      @#2$,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      R5,R3        ;:BASE DEST ADDR = ATA  
      CLR      (R2)         ;:[DEST] = 0  
      CCC                     ;SCOPE SYNC  
  
2$:   ADD      (R5),@10(R3) ;TEST THE ADD - SM1,DM7  
  
      CMP      R4,(R2)      ;RESULT = #DWTA?  
      BEQ      TST410       ;:BR IF YES  
  
      MOV      (R2),R3      ;GET WAS DATA  
3$:   ERROR   1             ;ADD DELIVERED WRONG RESULT  
  
*****  
*TEST 410      ADD SM2,DM7 TEST  
*****  
TST410:  
      SCOPE                ;CALL THE SCOPE LOOP UTILITY  
      MOV      #410,R0      ;:LOAD R0 WITH TEST NUMBER  
      MOV      @#2$,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      R5,R3        ;:BASE DEST ADDR = ATA  
      CLR      (R2)         ;:[DEST] = 0
```

8134 026520 000257
8135
8136 026522 062573 000010
8137
8138 026526 020412
8139 026530 001402
8140
8141 026532 011203
8142 026534 104001
8143
8144
8145
8146
8147 026536
8148 026536 000004
8149 026540 012700 000411
8150 026544 013701 026560
8151 026550 005004
8152 026552 005003
8153 026554 000257
8154 026556 000272
8155
8156 026560 074403
8157
8158 026562 100403
8159 026564 001002
8160 026566 102401
8161 026570 103001
8162
8163 026572 104002
8164
8165 026574 020403
8166 026576 001401
8167
8168 026600 104002
8169
8170
8171
8172
8173 026602
8174 026602 000004
8175 026604 012700 000412
8176 026610 013701 026630
8177 026614 005004
8178 026616 012705 177777
8179 026622 010503
8180 026624 000257
8181 026626 000265
8182
8183 026630 074503
8184
8185 026632 100403
8186 026634 001002
8187 026636 102401
8188 026640 103401
8189

```
CCC ;SCOPE SYNC
2$: ADD (R5)+,@10(R3) ;TEST THE ADD - SM2,DM7
CMP R4,(R2) ;RESULT = #DWTA?
BEQ TST411 ;;BR IF YES
3$: MOV (R2),R3 ;GET WAS DATA
ERROR 1 ;ADD DELIVERED WRONG RESULT
:*****
:*TEST 411 "XOR RA,RB" TEST - A=B=000000 N:C=1010
:*****
TST411:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #411,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT AND MASK = 000000
CLR R3 ;[DEST] = 000000
CCC ;SCOPE SYNC
272 ;MAKE N:C=1010
2$: XOR R4,R3 ;TEST THE XOR
BMI 3$ ;N:C=0100 ??
BNE 3$
BVS 3$
BCC 4$
3$: ERROR 2 ;XOR FAILED TO SET FLAGS PROPERLY
4$: CMP R4,R3 ;RESULT CORRECT?
BEQ TST412 ;;BR IF YES
5$: ERROR 2 ;XOR DELIVERED THE WRONG RESULT
:*****
:*TEST 412 "XOR RA,RB" TEST - A=B=177777 N:C=0101
:*****
TST412:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #412,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT = 000000
MOV #-1,R5 ;MASK = 177777
MOV R5,R3 ;[DEST]=177777
CCC ;SCOPE SYNC
265 ;MAKE N:C=0101
2$: XOR R5,R3 ;TEST THE XOR
BMI 3$ ;N:C=0101 ??
BNE 3$
BVS 3$
BCS 4$
```

8190 026642 104002
8191
8192 026644 020403
8193 026646 001401
8194
8195 026650 104002
8196
8197
8198
8199
8200 026652
8201 026652 000004
8202 026654 012700 000413
8203 026660 013701 026704
8204 026664 012704 177777
8205 026670 012705 125252
8206 026674 012703 052525
8207 026700 000257
8208 026702 000266
8209
8210 026704 074503
8211
8212 026706 100003
8213 026710 001402
8214 026712 102401
8215 026714 103001
8216
8217 026716 104002
8218
8219 026720 020403
8220 026722 001401
8221
8222 026724 104002
8223
8224
8225
8226
8227 026726
8228 026726 000004
8229 026730 012700 000414
8230 026734 013701 026760
8231 026740 012704 177777
8232 026744 012705 052525
8233 026750 012703 125252
8234 026754 000257
8235 026756 000271
8236
8237 026760 074503
8238
8239 026762 100003
8240 026764 001402
8241 026766 102401
8242 026770 103401
8243
8244 026772 104002
8245

```
3$: ERROR 2 ;XOR FAILED TO SET FLAGS PROPERLY
4$: CMP R4,R3 ;RESULT CORRECT?
   BEQ TST413 ;:BR IF YES
5$: ERROR 2 ;XOR DELIVERED THE WRONG RESULT
:*****
:*TEST 413 "XOR RA,RB" TEST - A=125252,B=052525 N:C=0110
:*****
TST413:
   SCOPE ;CALL THE SCOPE LOOP UTILITY
   MOV #413,R0 ;:LOAD R0 WITH TEST NUMBER
   MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
   MOV #-1,R4 ;RESULT S/B = 177777
   MOV #125252,R5 ;MASK=125252
   MOV #052525,R3 ;[DEST] = 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$
   BCC 4$
3$: ERROR 2 ;XOR FAILED TO SET FLAGS PROPERLY
4$: CMP R4,R3 ;RESULT CORRECT?
   BEQ TST414 ;:BR IF YES
5$: ERROR 2 ;XOR DELIVERED THE WRONG RESULT
:*****
:*TEST 414 "XOR RA,RB" TEST - A=052525,B=125252 N:C=1001
:*****
TST414:
   SCOPE ;CALL THE SCOPE LOOP UTILITY
   MOV #414,R0 ;:LOAD R0 WITH TEST NUMBER
   MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
   MOV #-1,R4 ;RESULT S/B = 177777
   MOV #52525,R5 ;MASK=052525
   MOV #125252,R3 ;[DEST] = 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
```

8246 026774 020403
8247 026776 00140i
8248
8249 027000 104002
8250
8251
8252
8253
8254 027002
8255 027002 000004
8256 027004 012700 000415
8257 027010 013701 027032
8258 027014 005004
8259 027016 005005
8260 027020 012702 063312
8261 027024 005012
8262 027026 000257
8263 027030 000272
8264
8265 027032 074512
8266
8267 027034 100403
8268 027036 001002
8269 027040 102401
8270 027042 103001
8271
8272 027044 104001
8273
8274 027046 020412
8275 027050 001402
8276
8277 027052 011203
8278 027054 104001
8279
8280
8281
8282
8283 027056
8284 027056 000004
8285 027060 012700 000416
8286 027064 013701 027112
8287 027070 005004
8288 027072 012705 177777
8289 027076 012702 063312
8290 027102 012712 177777
8291 027106 000257
8292 027110 000265
8293
8294 027112 074512
8295
8296 027114 100403
8297 027116 001002
8298 027120 102401
8299 027122 103401
8300
8301 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 @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
CLR R5 ;MASK = 000000
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR (R2) ;[DEST] = 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\$
BCC 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 @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #-1,R5 ;MASK = 177777
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #-1,(R2) ;[DEST] = 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

8302
 8303 027126 020412
 8304 027130 001402
 8305
 8306 027132 011203
 8307 027134 104001
 8308
 8309
 8310
 8311
 8312 027136
 8313 027136 000004
 8314 027140 012700 000417
 8315 027144 013701 027206
 8316
 8317 027150 032737 000400 063234
 8318 027156 001401
 8319 027160 000000
 8320 027162 012704 177777
 8321 027166 012705 125252
 8322 027172 012702 063312
 8323 027176 012712 052525
 8324 027202 000257
 8325 027204 000266
 8326
 8327 027206 074512
 8328
 8329 027210 100003
 8330 027212 001402
 8331 027214 102401
 8332 027216 103001
 8333
 8334 027220 104001
 8335
 8336 027222 020412
 8337 027224 001402
 8338
 8339 027226 011203
 8340 027230 104001
 8341
 8342
 8343
 8344
 8345 027232
 8346 027232 000004
 8347 027234 012700 000420
 8348 027240 013701 027270
 8349 027244 012704 177777
 8350 027250 012705 052525
 8351 027254 012702 063312
 8352 027260 012712 125252
 8353 027264 000257
 8354 027266 000271
 8355
 8356 027270 074512
 8357

```

4$:    CMP    R4,(R2)      ;RESULT CORRECT?
      BEQ    TST417       ;;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
;*****
TST417:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV    #417,R0       ;;LOAD R0 WITH TEST NUMBER
      MOV    @#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
      .SBTTL USER CONTROLLED BREAKPOINT -- BIT8
      BIT    #BIT8,@#BPTLOC ;BREAKPOINT HALT SET ??
      BEQ    .+4           ;BR IF NOT
      HALT                ;BREAK - DEPRESS CONTINUE TO RESTART
      MOV    #-1,R4        ;RESULT S/B = 177777
      MOV    #125252,R5    ;MASK = 125252
      MOV    #MBOF0,R2     ;DEST ADDR = MBOF0
      MOV    #052525,(R2)  ;[DEST] = 052525
      CCC                ;SCOPE SYNC
      266                 ;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    TST420       ;;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
;*****
TST420:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV    #420,R0       ;;LOAD R0 WITH TEST NUMBER
      MOV    @#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV    #-1,R4        ;RESULT S/B = 177777
      MOV    #52525,R5     ;MASK = 052525
      MOV    #MBOF0,R2     ;DEST ADDR = MBOF0
      MOV    #125252,(R2)  ;[DEST] = 125252
      CCC                ;SCOPE SYNC
      271                 ;MAKE N:C=1001

2$:    XOR    R5,(R2)      ;TEST THE XOR
  
```

8358 027272 100003
8359 027274 001402
8360 027276 102401
8361 027300 103401
8362
8363 027302 104001
8364
8365 027304 020412
8366 027306 001402
8367
8368 027310 011203
8369 027312 104001
8370
8371
8372
8373
8374 027314
8375 027314 000004
8376 027316 012700 000421
8377 027322 013701 027342
8378 027326 005004
8379 027330 012703 052525
8380 027334 010305
8381 027336 000257
8382 027340 000273
8383
8384 027342 160503
8385
8386 027344 100403
8387 027346 001002
8388 027350 102401
8389 027352 103001
8390
8391 027354 104002
8392
8393 027356 020304
8394 027360 001401
8395
8396 027362 104002
8397
8398
8399
8400
8401 027364
8402 027364 000004
8403 027366 012700 000422
8404 027372 013701 027412
8405 027376 005004
8406 027400 012703 125252
8407 027404 010305
8408 027406 000257
8409 027410 000273
8410
8411 027412 160503
8412
8413 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 TST421 ;: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) = +,+
:*****
TST421:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #421,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,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\$
BNE 3\$;DID N:C = 0100
BVS 3\$
BCC 4\$
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R3,R4 ;WAS RESULT = 0?
BEQ TST422 ;:BR IF YES
5\$: ERROR 2 ;SUB DELIVERED WRONG RESULT
:*****
:*TEST 422 SUB TEST SMO,DMO - (SRC) = (DEST) = -,-
:*****
TST422:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #422,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,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\$

8414 027416 001002
8415 027420 102401
8416 027422 103001
8417
8418 027424 104002
8419
8420 027426 020304
8421 027430 001401
8422
8423 027432 104002
8424
8425
8426
8427
8428 027434
8429 027434 000004
8430 027436 012700 000423
8431 027442 013701 027466
8432 027446 012704 000002
8433 027452 012703 000001
8434 027456 012705 177777
8435 027462 000257
8436 027464 000276
8437
8438 027466 160503
8439
8440 027470 100403
8441 027472 001402
8442 027474 102401
8443 027476 103401
8444
8445 027500 104002
8446
8447 027502 020304
8448 027504 001401
8449
8450 027506 104002
8451
8452
8453
8454
8455 027510
8456 027510 000004
8457 027512 012700 000424
8458 027516 013701 027542
8459 027522 012704 177776
8460 027526 012703 177777
8461 027532 012705 000001
8462 027536 000257
8463 027540 000267
8464
8465 027542 160503
8466
8467 027544 100003
8468 027546 001402
8469 027550 102401

BNE 3\$;N:C = 0100?
BVS 3\$
BCC 4\$
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R3,R4 ;RESULT = 0?
BEQ TST423 ;: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 @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #2,R4 ;RESULT S / B = 2
MOV #1,R3 ;[R3] = DEST OP = 1
MOV #-1,R5 ;[R5] = SRC OP = -1
CCC ;CLEAR FLAGS
276 ;MAKE N:C = 1110
2\$: SUB R5,R3 ;TEST THE SUB
BMI 3\$
BEQ 3\$;N:C = 0001
BVS 3\$
BCS 4\$
3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R3,R4 ;RESULT = +2?
BEQ TST424 ;: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 @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-2,R4 ;RESULT S / B = -2
MOV #-1,R3 ;[R3] = [DEST] = -1
MOV #1,R5 ;[R5] = [SOURCE] = +1
CCC ;CLEAR FLAGS
267 ;MAKE N:C = 0111
2\$: SUB R5,R3 ;TEST THE SUB
BPL 3\$
BEQ 3\$;N:C = 1000
BVS 3\$

8470 027552 103001
 8471
 8472 027554 104002
 8473
 8474 027556 020403
 8475 027560 001401
 8476
 8477 027562 104002
 8478
 8479
 8480
 8481
 8482 027564
 8483 027564 000004
 8484 027566 012700 000425
 8485 027572 013701 027616
 8486 027576 012704 077777
 8487 027602 012703 100000
 8488 027606 012705 000001
 8489 027612 000257
 8490 027614 000274
 8491
 8492 027616 160503
 8493
 8494 027620 100403
 8495 027622 001402
 8496 027624 102001
 8497 027626 103001
 8498
 8499 027630 104002
 8500
 8501 027632 020304
 8502 027634 001401
 8503
 8504 027636 104002
 8505
 8506
 8507
 8508
 8509 027640
 8510 027640 000004
 8511 027642 012700 000426
 8512 027646 013701 027674
 8513 027652 012702 063312
 8514 027656 012704 177777
 8515 027662 012705 000001
 8516 027666 005012
 8517 027670 000257
 8518 027672 000266
 8519
 8520 027674 160512
 8521
 8522 027676 100003
 8523 027700 001402
 8524 027702 102401
 8525 027704 103401

BCC 4\$

3\$: ERROR 2 ;SUB DID NOT ALTER CODES PROPERLY

4\$: CMP R4,R3 ;RESULT = -2?
 BEQ TST425 ;;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 @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #77777,R4 ;RESULT = 77777
 MOV #100000,R3 ;[R3] = DEST OP = 100000
 MOV #1,R5 ;[R5] = SRC OP = 1
 CCC ;CLEAR FLAGS
 274 ;MAKE N:C = 1100

2\$: SUB R5,R3 ;TEST THE SUB

BMI 3\$
 BEQ 3\$;N:C = 0011 ('V' BIT SHOULD SET)
 BVC 3\$
 BCC 4\$

3\$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY

4\$: CMP R3,R4 ;RESULT = 77777?
 BEQ TST426 ;;BR IF YES

5\$: ERROR 2 ;SUB DELIVERED WRONG RESULT

 ;*TEST 426 SUB TEST - SMO,DM1 - N:C = 0110

 TST426:
 SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #426,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 = 177777
 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 3\$;N:C = 1001
 BEQ 3\$
 BVS 3\$
 BCS 4\$


```
8526
8527 027706 10400i 3$: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY
8528
8529 027710 020412 4$: CMP R4,(R2) ;CORRECT RESULT ?
8530 027712 001402 BEQ TST427 ;;BR IF YES
8531
8532 027714 011203 5$: MOV (R2),R3 ;GET THE WAS DATA
8533 027716 104001 ERROR 1 ;SUB DELIVERED THE WRONG RESULT
8534
8535
8536
8537
8538 027720
8539 027720 000004
8540 027722 012700 000427
8541 027726 013701 027754
8542 027732 012702 063312
8543 027736 005004
8544 027740 012705 177777
8545 027744 012712 177777
8546 027750 000257
8547 027752 000272
8548
8549 027754 160512 2$: SUB R5,(R2) ;TEST THE SUB
8550
8551 027756 100403 BMI 3$ ;N:C = 0100
8552 027760 001002 BNE 3$
8553 027762 102401 BVS 3$
8554 027764 103001 BCC 4$
8555
8556 027766 104001 3$: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY
8557
8558 027770 020412 4$: CMP R4,(R2) ;CORRECT RESULT ?
8559 027772 001402 BEQ TST430 ;;BR IF YES
8560
8561 027774 011203 5$: MOV (R2),R3 ;GET THE WAS DATA
8562 027776 104001 ERROR 1 ;SUB DELIVERED THE WRONG RESULT
8563
8564
8565
8566
8567 030000
8568 030000 000004
8569 030002 012700 000430
8570 030006 013701 030034
8571 030012 012702 063312
8572 030016 012704 077777
8573 030022 012705 000001
8574 030026 012712 100000
8575 030032 000257
8576
8577 030034 160512 2$: SUB R5,(R2) ;TEST THE SUB
8578
8579 030036 100403 BMI 3$ ;N:C = 0010
8580 030040 001402 BEQ 3$
8581 030042 102001 BVC 3$

;*****
;*TEST 427 SUB TEST - SMO,DM1 - N:C = 1010
;*****
TST427:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #427,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
MOV #-1,R5 ;SRC OPR = 177777
MOV #-1,(R2) ;[DEST] = 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 TST430 ;;BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;SUB DELIVERED THE WRONG RESULT

;*****
;*TEST 430 SUB TEST - SMO,DM1 - N:C = 0000
;*****
TST430:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #430,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #77777,R4 ;RESULT S / B = 77777
MOV #+1,R5 ;SRC OPR = +1
MOV #100000,(R2) ;[DEST] = 100000
CCC ;CLEAR FLAGS

2$: SUB R5,(R2) ;TEST THE SUB

BMI 3$ ;N:C = 0010
BEQ 3$
BVC 3$
```

8582 030044 103001
8583
8584 030046 104001
8585
8586 030050 020412
8587 030052 001402
8588
8589 030054 011203
8590 030056 104001
8591

BCC 4\$
3\$: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST431 ;;BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
5\$: ERROR 1 ;SUB DELIVERED THE WRONG RESULT

8592
8593
8594
8595 030060
8596 030060 000004
8597 030062 012700 000431
8598 030066 013701 030110
8599 030072 012704 177777
8600 030076 012705 064034
8601 030102 005003
8602 030104 000257
8603 030106 000266
8604

:::*****
:*TEST 431 SUB TEST - SM1,DM0 - N:C = 0110
:::*****
TST431:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #431,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S / B = 177777
MOV #DWTB+2,R5 ;SRC ADDR = DWTB+2
CLR R3 ;[DEST] = 000000
CCC ;CLEAR FLAGS
266 ;N:C = 0110

8605 030110 161503
8606
8607 030112 100003
8608 030114 001402
8609 030116 102401
8610 030120 103401
8611
8612 030122 104002
8613
8614 030124 020403
8615 030126 001401
8616
8617 030130 104002
8618
8619

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 TST432 ;;BR IF YES
5\$: ERROR 2 ;SUB DELIVERED THE WRONG RESULT

8620
8621
8622 030132
8623 030132 000004
8624 030134 012700 000432
8625 030140 013701 030160
8626 030144 005004
8627 030146 012705 063324
8628 030152 011503
8629 030154 000257
8630 030156 000272
8631

:::*****
:*TEST 432 SUB TEST - SM1,DM0 - N:C = 1010
:::*****
TST432:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #432,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #DWTA+2,R5 ;SRC ADDR = DWTA+2
MOV (R5),R3 ;[DEST] = 177777
CCC ;CLEAR FLAGS
272 ;N:C = 1010

8632 030160 161503
8633
8634 030162 100403
8635 030164 001002
8636 030166 102401
8637 030170 103001

2\$: SUB (R5),R3 ;TEST THE SUB
BMI 3\$;N:C = 0100
BNE 3\$
BVS 3\$
BCC 4\$

8638
8639 030172 104002
8640
8641 030174 020403
8642 030176 001401
8643
8644 030200 104002
8645
8646
8647
8648
8649 030202
8650 030202 000004
8651 030204 012700 000433
8652 030210 013701 030236
8653 030214 012704 077777
8654 030220 012705 063316
8655 030224 012703 100000
8656 030230 012715 000001
8657 030234 000257
8658
8659 030236 161503
8660
8661 030240 100403
8662 030242 001402
8663 030244 102001
8664 030246 103001
8665
8666 030250 104002
8667
8668 030252 020403
8669 030254 001401
8670
8671 030256 104002
8672
8673
8674
8675
8676 030260
8677 030260 000004
8678 030262 012700 000434
8679 030266 013701 030320
8680 030272 012702 063312
8681 030276 012704 177777
8682 030302 012705 063316
8683 030306 012715 000001
8684 030312 005012
8685 030314 000257
8686 030316 000266
8687
8688 030320 161512
8689
8690 030322 100003
8691 030324 001402
8692 030326 102401
8693 030330 103401

```
3$: ERROR 2 ;SUB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,R3 ;CORRECT RESULT ?
   BEQ TSI433 ;:BR IF YES
5$: ERROR 2 ;SUB DELIVERED THE WRONG RESULT

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

8694
8695 030332 104001
8696
8697 030334 020412
8698 030336 001402
8699
8700 030340 011203
8701 030342 104001
8702
8703
8704
8705
8706 030344
8707 030344 000004
8708 030346 012700 000435
8709 030352 013701 030406
8710 030356 012702 063312
8711 030362 012704 177777
8712 030366 012705 063316
8713 030372 012715 000001
8714 030376 005012
8715 030400 010203
8716 030402 000257
8717 030404 000266
8718
8719 030406 161523
8720
8721 030410 100003
8722 030412 001402
8723 030414 102401
8724 030416 103401
8725
8726 030420 104005
8727
8728 030422 020412
8729 030424 001402
8730
8731 030426 011203
8732 030430 104001
8733
8734
8735
8736
8737 030432
8738 030432 000004
8739 030434 012700 000436
8740 030440 013701 030464
8741 030444 012702 063312
8742 030450 012704 125252
8743 030454 010205
8744 030456 012712 052526
8745 030462 000257
8746
8747 030464 005425
8748
8749 030466 020412

```
3$: ERROR 1 ;SUB FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
   BEQ TST435 ;: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 @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
   MOV #MBUF0,R2 ;DEST ADDR = MBUF0
   MOV #-1,R4 ;RESULT S / B = 177777
   MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
   MOV #+1,(R5) ;[SOURCE] = 000001
   CLR (R2) ;[DEST] = 000000
   MOV R2,R3 ;R3 GETS DEST ADDR
   CCC ;CLEAR FLAGS
   266 ;N:C = 0110
2$: 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 TST436 ;: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 @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
   MOV #MBUF0,R2 ;DEST ADDR = MBUF0
   MOV #125252,R4 ;RESULT S / B = 125252
   MOV R2,R5 ;[R5] = DEST ADDR
   MOV #52526,(R2) ;[DEST] = 52526
   CCC ;SCOPE SYNC
2$: NEG (R5)+ ;TEST THE NEG - MODE 2
   CMP R4,(R2) ;RESULT = 125252?
```

8750 030470 001402
8751
8752 030472 011203
8753 030474 104001
8754
8755 030476 022705 063314
8756 030502 001401
8757
8758 030504 104005
8759
8760
8761
8762
8763 030506
8764 030506 000004
8765 030510 012700 000437
8766 030514 013701 030542
8767 030520 012702 063312
8768 030524 012704 125252
8769 030530 012705 063306
8770 030534 012712 052526
8771 030540 000257
8772
8773 030542 005435
8774
8775 030544 020412
8776 030546 001402
8777
8778 030550 011203
8779 030552 104001
8780
8781 030554 022705 063310
8782 030560 001401
8783
8784 030562 104005
8785
8786
8787
8788
8789 030564
8790 030564 000004
8791 030566 012700 000440
8792 030572 013701 030620
8793 030576 012702 063312
8794 030602 012704 125252
8795 030606 012705 063314
8796 030612 012712 052526
8797 030616 000257
8798
8799 030620 005445
8800
8801 030622 020412
8802 030624 001402
8803
8804 030626 011203
8805 030630 104001

```
      BEQ      4$      ;BR IF YES
      MOV      (R2),R3 ;GET THE WAS DATA
3$:  ERROR    1      ;NEG DELIVERED WRONG RESULT
      CMP      #MBUFO+2,R5 ;DID REG. GET AUTO INCREMENTED?
4$:  BEQ      TST437 ;:BR IF YES
      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      @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
      MOV      #125252,R4     ;RESULT S / B = 125252
      MOV      #ATA+10,R5     ;[ATA+10] = MBUFO
      MOV      #52526,(R2)    ;[DEST] = 52526
      CCC                    ;SCOPE SYNC
      NEG      @(R5)+         ;TEST THE NEG - MODE 3
      CMP      R4,(R2)        ;RESULT = 125252?
      BEQ      4$            ;BR IF YES
      MOV      (R2),R3        ;GET WAS DATA
3$:  ERROR    1      ;NEG DELIVERED WRONG RESULT
      CMP      #ATA+12,R5     ;DID REG GET AUTO INCREMENTED?
4$:  BEQ      TST440 ;:BR IF YES
      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      @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
      MOV      #125252,R4     ;RESULT S / B = 125252
      MOV      #MBUFO+2,R5    ;[R5] = DEST ADDR + 2
      MOV      #52526,(R2)    ;[DEST] = 52526
      CCC                    ;SCOPE SYNC
      NEG      -(R5)         ;TEST THE NEG - MODE 4
      CMP      R4,(R2)        ;RESULT = 125252?
      BEQ      4$            ;BR IF YES
      MOV      (R2),R3        ;GET WAS DATA
3$:  ERROR    1      ;NEG DELIVERED WRONG RESULT
```

8806
8807 030632 020502
8808 030634 001401
8809
8810 030636 104005

4\$: CMP R5,R2 ;DID REG GET AUTO INCREMENTED?
BEQ TSI441 ;;BR IF YES
5\$: ERROR 5 ;NEG FAILED TO UPDATE REG

8811
 8812
 8813
 8814
 8815 030640
 8816 030640 000004
 8817 030642 012700 000441
 8818 030646 013701 030674
 8819 030652 012702 063312
 8820 030656 012704 125252
 8821 030662 012705 063310
 8822 030666 012712 052526
 8823 030672 000257
 8824
 8825 030674 005455
 8826
 8827 030676 020412
 8828 030700 001402
 8829
 8830 030702 011203
 8831 030704 104001
 8832
 8833 030706 022705 063306
 8834 030712 001401
 8835
 8836 030714 104005
 8837
 8838
 8839
 8840
 8841 030716
 8842 030716 000004
 8843 030720 012700 000442
 8844 030724 013701 030752
 8845 030730 012702 063312
 8846 030734 012704 125252
 8847 030740 012705 063310
 8848 030744 012712 052526
 8849 030750 000257
 8850
 8851 030752 005465 000002
 8852
 8853 030756 020412
 8854 030760 001402
 8855
 8856 030762 011203
 8857 030764 104001
 8858
 8859
 8860
 8861
 8862 030766
 8863 030766 000004
 8864 030770 012700 000443
 8865 030774 013701 031022
 8866 031000 012702 063312

```

:*****
:*TEST 441      NEG DM5 TEST
:*****
TST441:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #441,R0      ;:LOAD R0 WITH TEST NUMBER
      MOV      @#2$,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

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

4$:   CMP      #ATA+10,R5 ;:DID NEG UPDATE REG
      BEQ      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      @#2$,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] = 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

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

:*****
:*TEST 443      NEG DM7 TEST
:*****
TST443:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #443,R0      ;:LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1     ;:LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #MBUF0,R2   ;:DEST ADDR = MBUF0
  
```

8867 031004 012704 125252
8868 031010 012705 063276
8869 031014 012712 052526
8870 031020 000257
8871
8872 031022 005475 000010
8873
8874 031026 020412
8875 031030 001402
8876
8877 031032 011203
8878 031034 104001
8879
8880
8881
8882
8883 031036
8884 031036 000004
8885 031040 012700 000444
8886 031044 013701 031074
8887 031050 005004
8888 031052 005104
8889 031054 012702 063312
8890 031060 012705 063324
8891 031064 010203
8892 031066 005012
8893 031070 000257
8894 031072 000264
8895
8896 031074 011513
8897
8898 031076 100003
8899 031100 001402
8900 031102 102401
8901 031104 103001
8902
8903 031106 104001
8904
8905 031110 020412
8906 031112 001403
8907
8908 031114 005003
8909 031116 051203
8910 031120 104001
8911
8912
8913
8914
8915 031122
8916 031122 000004
8917 031124 012700 000445
8918 031130 013701 031160
8919 031134 005004
8920 031136 005104
8921 031140 012702 063312
8922 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
3$: ERROR 1 ;NEG DELIVERED WRONG RESULT

;*****
;*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 @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
MOV R2,R3 ;BASE DEST ADDR = MBUFO
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
5$: ERROR 1 ;MOV DELIVERED THE WRONG RESULT

;*****
;*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 @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
```



```
8923 031150 010203      MOV      R2,R3      ;BASE DEST ADDR = MBUFO
8924 031152 005012      CLR      (R2)      ;MAKE [DEST] = 000000
8925 031154 000257      CCC      ;CLEAR FLAGS
8926 031156 000264      264      ;N:C = 0100
8927
8928 031160 012513      2$:      MOV      (R5)+,(R3) ;TEST THE MOV - SM2,DM1
8929
8930 031162 100003      BPL      3$      ;N:C = 1000 ?
8931 031164 001402      BEQ      3$
8932 031166 102401      BVS      3$
8933 031170 103001      BCC      4$
8934
8935 031172 104001      3$:      ERROR    1      ;MOV FAILED TO ALTER CODES PROPERLY
8936
8937 031174 020412      4$:      CMP      R4,(R2)    ;RESULT CORRECT ??
8938 031176 001403      BEQ      TST446    ;:BR IF YES
8939
8940 031200 005003      CLR      R3      ;GET THE WAS DATA
8941 031202 051203      BIS      (R2),R3
8942 031204 104001      5$:      ERROR    1      ;MOV DELIVERED THE WRONG RESULT
8943
8944
8945
8946
8947 031206
8948 031206 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
8949 031210 012700 000446      MOV      #446,R0  ;:LOAD R0 WITH TEST NUMBER
8950 031214 013701 031244      MOV      @#2$,R1  ;LOAD R1 WITH TEST INSTRUCTION WORD
8951 031220 005004      CLR      R4      ;RESULT S / B = 000000
8952 031222 012702 063312      MOV      #MBUFO,R2 ;DEST ADDR = MBUFO
8953 031226 012705 063322      MOV      #DWTA,R5 ;SOURCE ADDR = DWTA
8954 031232 010203      MOV      R2,R3    ;BASE DEST ADDR = MBUFO
8955 031234 005012      CLR      (R2)    ;MAKE [DEST] = 177777
8956 031236 005112      COM      (R2)
8957 031240 000257      CCC      ;CLEAR FLAGS
8958 031242 000273      273      ;N:C = 1011
8959
8960 031244 011513      2$:      MOV      (R5),(R3) ;TEST THE MOV - SM1,DM1
8961
8962 031246 100403      BMI      3$      ;N:C = 0101 ?
8963 031250 001002      BNE      3$
8964 031252 102401      BVS      3$
8965 031254 103401      BCS      4$
8966
8967 031256 104001      3$:      ERROR    1      ;MOV FAILED TO ALTER CODES PROPERLY
8968
8969 031260 020412      4$:      CMP      R4,(R2)    ;RESULT CORRECT ??
8970 031262 001403      BEQ      TST447    ;:BR IF YES
8971
8972 031264 005003      CLR      R3      ;GET THE WAS DATA
8973 031266 051203      BIS      (R2),R3
8974 031270 104001      5$:      ERROR    1      ;MOV DELIVERED THE WRONG RESULT
8975
8976
8977
8978
```

8979 031272
8980 031272 000004
8981 031274 012700 000447
8982 031300 013701 031330
8983 031304 005004
8984 031306 012702 063312
8985 031312 012705 063322
8986 031316 010203
8987 031320 005012
8988 031322 005112
8989 031324 000257
8990 031326 000273
8991
8992 031330 012513
8993
8994 031332 100403
8995 031334 001002
8996 031336 102401
8997 031340 103401
8998
8999 031342 104001
9000
9001 031344 020412
9002 031346 001403
9003
9004 031350 005003
9005 031352 051203
9006 031354 104001
9007
9008
9009
9010
9011 031356
9012 031356 000004
9013 031360 012700 000450
9014 031364 013701 031414
9015 031370 005004
9016 031372 005104
9017 031374 012702 063312
9018 031400 012705 063324
9019 031404 010203
9020 031406 005012
9021 031410 000257
9022 031412 000264
9023
9024 031414 011523
9025
9026 031416 100003
9027 031420 001402
9028 031422 102401
9029 031424 103001
9030
9031 031426 104001
9032
9033 031430 022703 063314
9034 031434 001401

TST447:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #447,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;:RESULT S / B = 000000
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #DWTA,R5 ;:SOURCE ADDR = DWTA
MOV R2,R3 ;:BASE DEST ADDR = MBUFO
CLR (R2) ;:MAKE [DEST] = 177777
COM (R2)
CCC ;:CLEAR FLAGS
273 ;:N:C = 1011
2\$: MOV (R5)+,(R3) ;:TEST THE MOV - SM2,DM1
BMI 3\$;:N:C = 0101 ?
BNE 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 1 ;:MOV FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;:RESULT CORRECT ??
BEQ TST450 ;:BR IF YES
CLR R3 ;:GET THE WAS DATA
BIS (R2),R3
5\$: ERROR 1 ;:MOV DELIVERED THE WRONG RESULT
:*****
:*TEST 450 MOV SM1,DM2 TEST - N:C = 0100
:*****
TST450:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #450,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;:RESULT S / B = 177777
COM R4
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV #DWTA+2,R5 ;:SOURCE ADDR = D'.IA
MOV R2,R3 ;:BASE DEST ADDR = MBUFO
CLR (R2) ;:MAKE [DEST] = 000000
CCC ;:CLEAR FLAGS
264 ;:N:C = 0100
2\$: MOV (R5),(R3)+ ;:TEST THE MOV - SM1,DM2
BPL 3\$;:N:C = 1000 ?
BEQ 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 1 ;:MOV FAILED TO ALTER CODES PROPERLY
4\$: CMP #MBUFO+2,R3 ;:DID MOV INCREMENT DEST REG ?
BEQ 6\$;:BR IF YES

9035
 9036 031436 104005
 9037
 9038 031440 020412
 9039 031442 001403
 9040
 9041 031444 005003
 9042 031446 051203
 9043 031450 104001
 9044
 9045
 9046
 9047
 9048 031452
 9049 031452 000004
 9050 031454 012700 000451
 9051 031460 013701 031510
 9052 031464 005004
 9053 031466 005104
 9054 031470 012702 063312
 9055 031474 012705 063324
 9056 031500 010203
 9057 031502 005012
 9058 031504 000257
 9059 031506 000264
 9060
 9061 031510 012523
 9062
 9063 031512 100003
 9064 031514 001402
 9065 031516 102401
 9066 031520 103001
 9067
 9068 031522 104001
 9069
 9070 031524 022703 063314
 9071 031530 001401
 9072
 9073 031532 104005
 9074
 9075 031534 020412
 9076 031536 001403
 9077
 9078 031540 005003
 9079 031542 051203
 9080 031544 104001
 9081
 9082
 9083
 9084
 9085 031546
 9086 031546 000004
 9087 031550 012700 000452
 9088 031554 013701 031606
 9089 031560 005004
 9090 031562 005104

```

5$:  ERROR 5 ;MOV FAILED TO UPDATE DEST REG
6$:  CMP R4,(R2) ;RESULT CORRECT ??
    BEQ TST451 ;:BR IF YES
    CLR R3 ;GET THE WAS DATA
7$:  BIS (R2),R3
    ERROR 1 ;MOV DELIVERED THE WRONG RESULT

:*****
:*TEST 451 MOV SM2,DM2 TEST - N:C = 0100
:*****
TST451:
    SCOPE ;CALL THE SCOPE LOOP UTILITY
    MOV #451,R0 ;:LOAD R0 WITH TEST NUMBER
    MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
    CLR R4 ;RESULT S / B = 177777
    COM R4
    MOV #MBUFO,R2 ;DEST ADDR = MBUFO
    MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA
    MOV R2,R3 ;BASE DEST ADDR = MBUFO
    CLR (R2) ;MAKE [DEST] = 000000
    CCC ;CLEAR FLAGS
    264 ;N:C = 0100

2$:  MOV (R5)+,(R3)+ ;TEST THE MOV - SM2,DM2
    BPL 3$ ;N:C = 1000 ?
    BEQ 3$
    BVS 3$
    BCC 4$

3$:  ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY
4$:  CMP #MBUFO+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 ;GET THE WAS DATA
7$:  BIS (R2),R3
    ERROR 1 ;MOV DELIVERED THE WRONG RESULT

:*****
:*TEST 452 MOV SM1,DM3 TEST - N:C = 0100
:*****
TST452:
    SCOPE ;CALL THE SCOPE LOOP UTILITY
    MOV #452,R0 ;:LOAD R0 WITH TEST NUMBER
    MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
    CLR R4 ;RESULT S / B = 177777
    COM R4
  
```

```

9091 031564 012702 063312      MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
9092 031570 012705 063324      MOV      #DWTA+2,R5    ;SOURCE ADDR = DWTA+2
9093 031574 012703 063306      MOV      #ATA+10,R3    ;BASE DEST ADDR = ATA+10
9094 031600 005012              CLR      (R2)          ;MAKE [DEST] = 000000
9095 031602 000257              CCC                ;CLEAR FLAGS
9096 031604 000264              264                ;N:C = 0100
9097
9098 031606 011533      2$:      MOV      (R5),@(R3)+ ;TEST THE MOV - SM1,DM3
9099
9100 031610 100003              BPL      3$          ;N:C = 1000 ?
9101 031612 001402              BEQ      3$
9102 031614 102401              BVS      3$
9103 031616 103001              BCC      4$
9104
9105 031620 104001      3$:      ERROR    1          ;MOV FAILED TO ALTER CODES PROPERLY
9106
9107 031622 022703 063310      4$:      CMP      #ATA+12,R3 ;DID MOV INCREMENT DEST REG ?
9108 031626 001401              BEQ      6$          ;BR IF YES
9109
9110 031630 104005      5$:      ERROR    5          ;MOV FAILED TO UPDATE DEST REG
9111
9112 031632 020412      6$:      CMP      R4,(R2)    ;RESULT CORRECT ??
9113 031634 001403              BEQ      TST453      ;:BR IF YES
9114
9115 031636 005003              CLR      R3          ;GET THE WAS DATA
9116 031640 051203              BIS      (R2),R3
9117 031642 104001      7$:      ERROR    1          ;MOV DELIVERED THE WRONG RESULT
9118
9119
9120
9121
9122
9123
9124
9125
9126
9127
9128
9129
9130
9131
9132
9133
9134
9135
9136
9137
9138
9139
9140
9141
9142
9143
9144
9145
9146

```

```

:*****
:*TEST 453      MOV SM2,DM3 TEST - N:C = 0100
:*****
TST453:

```

```

SCOPE          ;CALL THE SCOPE LOOP UTILITY
MOV      #453,R0 ;:LOAD R0 WITH TEST NUMBER
MOV      @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
CLR      R4      ;RESULT S / B = 177777
COM      R4
MOV      #MBUFO,R2 ;DEST ADDR = MBUFO
MOV      #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
MOV      #ATA+10,R3 ;BASE DEST ADDR = ATA+10
CLR      (R2)    ;MAKE [DEST] = 000000
CCC                ;CLEAR FLAGS
264                ;N:C = 0100
2$:      MOV      (R5)+,@(R3)+ ;TEST THE MOV - SM2,DM3
BPL      3$          ;N:C = 1000 ?
BEQ      3$
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

```

9147 031726 104005
9148
9149 031730 020412
9150 031732 001403
9151
9152 031734 005003
9153 031736 051203
9154 031740 104001
9155
9156
9157
9158
9159 031742
9160 031742 000004
9161 031744 012700 000454
9162 031750 013701 032002
9163 031754 005004
9164 031756 005104
9165 031760 012702 063312
9166 031764 012705 063324
9167 031770 012703 063314
9168 031774 005012
9169 031776 000257
9170 032000 000264
9171
9172 032002 011543
9173
9174 032004 100003
9175 032006 001402
9176 032010 102401
9177 032012 103001
9178
9179 032014 104001
9180
9181 032016 020203
9182 032020 001401
9183
9184 032022 104005
9185
9186 032024 020412
9187 032026 001403
9188
9189 032030 005003
9190 032032 051203
9191 032034 104001
9192
9193
9194
9195
9196 032036
9197 032036 000004
9198 032040 012700 000455
9199 032044 013701 032076
9200 032050 005004
9201 032052 005104
9202 032054 012702 063312

5\$: ERROR 5 ;MOV FAILED TO UPDATE DEST REG
6\$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ TST454 ;:BR IF YES
CLR R3 ;GET THE WAS DATA
BIS (R2),R3
7\$: 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 @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4
MOV #MBOF0,R2 ;DEST ADDR = MBOF0
MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
MOV #MBOF0+2,R3 ;BASE DEST ADDR = MBOF0+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 TST455 ;:BR IF YES
CLR R3 ;GET THE WAS DATA
BIS (R2),R3
7\$: 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 @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4
MOV #MBOF0,R2 ;DEST ADDR = MBOF0

```

9203 032060 012705 063324      MOV      #DWTA+2,R5      ;SOURCE ADDR = DWTA+2
9204 032064 012703 063314      MOV      #MBUFO+2,R3    ;BASE DEST ADDR = MBUFO+2
9205 032070 005012              CLR      (R2)           ;MAKE [DEST] = 000000
9206 032072 000257              CCC                     ;CLEAR FLAGS
9207 032074 000264              264                    ;N:C = 0100
9208
9209 032076 012543      2$:  MOV      (R5)+,-(R3) ;TEST THE MOV - SM2,DM4
9210
9211 032100 100003              BPL      3$             ;N:C = 1000 ?
9212 032102 001402              BEQ      3$
9213 032104 102401              BVS      3$
9214 032106 103001              BCC      4$
9215
9216 032110 104001      3$:  ERROR    1           ;MOV FAILED TO ALTER CODES PROPERLY
9217
9218 032112 020203      4$:  CMP      R2,R3      ;DID MOV INCREMENT DEST REG ?
9219 032114 001401              BEQ      6$            ;BR IF YES
9220
9221 032116 104005      5$:  ERROR    5           ;MOV FAILED TO UPDATE DEST REG
9222
9223 032120 020412      6$:  CMP      R4,(R2)    ;RESULT CORRECT ??
9224 032122 001403              BEQ      TST456        ;;BR IF YES
9225
9226 032124 005003              CLR      R3            ;GET THE WAS DATA
9227 032126 051203              BIS      (R2),R3
9228 032130 104001      7$:  ERROR    1           ;MOV DELIVERED THE WRONG RESULT
9229
9230
9231
9232
9233
9234
9235
9236
9237
9238 032132 000004
9239 032134 012700 000456
9240 032140 013701 032204
9241
9242
9243 032144 032737 001000 063234
9244 032152 001401
9245 032154 000000
9246 032156 005004
9247 032160 005104
9248 032162 012702 063312
9249 032166 012705 063324
9250 032172 012703 063310
9251 032176 005012
9252 032200 000257
9253 032202 000264
9254
9255
9256
9257 032204 011553      2$:  MOV      (R5),@(R3) ;TEST THE MOV - SM1,DM5
9258
9259
9260
9261
9262
9263
9264
9265
9266
9267
9268
9269
9270
9271
9272
9273
9274
9275
9276
9277
9278
9279
9280
9281
9282
9283
9284
9285
9286
9287
9288
9289
9290
9291
9292
9293
9294
9295
9296
9297
9298
9299
9300
9301
9302
9303
9304
9305
9306
9307
9308
9309
9310
9311
9312
9313
9314
9315
9316
9317
9318
9319
9320
9321
9322
9323
9324
9325
9326
9327
9328
9329
9330
9331
9332
9333
9334
9335
9336
9337
9338
9339
9340
9341
9342
9343
9344
9345
9346
9347
9348
9349
9350
9351
9352
9353
9354
9355
9356
9357
9358
9359
9360
9361
9362
9363
9364
9365
9366
9367
9368
9369
9370
9371
9372
9373
9374
9375
9376
9377
9378
9379
9380
9381
9382
9383
9384
9385
9386
9387
9388
9389
9390
9391
9392
9393
9394
9395
9396
9397
9398
9399
9400
9401
9402
9403
9404
9405
9406
9407
9408
9409
9410
9411
9412
9413
9414
9415
9416
9417
9418
9419
9420
9421
9422
9423
9424
9425
9426
9427
9428
9429
9430
9431
9432
9433
9434
9435
9436
9437
9438
9439
9440
9441
9442
9443
9444
9445
9446
9447
9448
9449
9450
9451
9452
9453
9454
9455
9456
9457
9458
9459
9460
9461
9462
9463
9464
9465
9466
9467
9468
9469
9470
9471
9472
9473
9474
9475
9476
9477
9478
9479
9480
9481
9482
9483
9484
9485
9486
9487
9488
9489
9490
9491
9492
9493
9494
9495
9496
9497
9498
9499
9500

```

```

*****
;*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      @#2$,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 = 177777
COM      R4
MOV      #MBUFO,R2 ;DEST ADDR = MBUFO
MOV      #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
MOV      #ATA+12,R3 ;BASE DEST ADDR = ATA+12
CLR      (R2)      ;MAKE [DEST] = 000000
CCC                     ;CLEAR FLAGS
264                    ;N:C = 0100

```

```

2$:  MOV      (R5),@(R3) ;TEST THE MOV - SM1,DM5
;N:C = 0100 ?
3$:  ERROR    1           ;MOV FAILED TO ALTER CODES PROPERLY

```

9259 032220 022703 063306
9260 032224 001401
9261
9262 032226 104005
9263
9264 032230 020412
9265 032232 001403
9266
9267 032234 005003
9268 032236 051203
9269 032240 104001
9270
9271
9272
9273
9274 032242
9275 032242 000004
9276 032244 012700 000457
9277 032250 013701 032302
9278 032254 005004
9279 032256 005104
9280 032260 012702 063312
9281 032264 012705 063324
9282 032270 012703 063310
9283 032274 005012
9284 032276 000257
9285 032300 000264
9286
9287 032302 012553
9288
9289 032304 100003
9290 032306 001402
9291 032310 102401
9292 032312 103001
9293
9294 032314 104001
9295
9296 032316 022703 063306
9297 032322 001401
9298
9299 032324 104005
9300
9301 032326 020412
9302 032330 001403
9303
9304 032332 005003
9305 032334 051203
9306 032336 104001
9307
9308
9309
9310
9311 032340
9312 032340 000004
9313 032342 012700 000460
9314 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 TST457 ;:BR IF YES

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

:*TEST 457 MOV SM2,DM5 TEST - N:C = 0100

TST457:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #457,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 177777
COM R4
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
MOV #ATA+12,R3 ;BASE DEST ADDR = ATA+12
CLR (R2) ;MAKE [DEST] = 00000
CCC ;CLEAR FLAGS
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 TST460 ;:BR IF YES

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

:*TEST 460 MOV SM1,DM6 TEST - N:C = 0100

TST460:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #460,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

```

9315 032352 005004          CLR      R4          ;RESULT S / B = 177777
9316 032354 005104          COM      R4
9317 032356 012702 063320    MOV      #MBUFO+6,R2 ;DEST ADDR = MBUFO+6
9318 032362 012705 063324    MOV      #DWTA+2,R5  ;SOURCE ADDR = DWTA+2
9319 032366 012703 063312    MOV      #MBUFO,R3   ;BASE DEST ADDR = MBUFO
9320 032372 005012          CLR      (R2)        ;MAKE [DEST] = 000000
9321 032374 000257          CCC
9322 032376 000264          264              ;CLEAR FLAGS
9323                                     ;N:C = 0100
9324 032400 011563 000006    2$: MOV      (R5),6(R3) ;TEST THE MOV - SM1,DM6
9325                                     ;N:C = 1000 ?
9326 032404 100003          BPL      3$
9327 032406 001402          BEQ      3$
9328 032410 102401          BVS      3$
9329 032412 103001          BCC      4$
9330
9331 032414 104001          3$: ERROR 1          ;MOV FAILED TO ALTER CODES PROPERLY
9332
9333 032416 020412          4$: CMP      R4,(R2)  ;RESULT CORRECT ??
9334 032420 001403          BEQ      TST461     ;:BR IF YES
9335
9336 032422 005003          CLR      R3          ;GET THE WAS DATA
9337 032424 051203          BIS      (R2),R3
9338 032426 104001          5$: ERROR 1          ;MOV DELIVERED THE WRONG RESULT
9339
9340                                     ;:*****
9341                                     ;*TEST 461      MOV SM2,DM6 TEST - N:C = 0100
9342                                     ;:*****
9343 TST461:
9344 032430 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
9345 032432 012700 000461    MOV      #461,R0    ;:LOAD R0 WITH TEST NUMBER
9346 032436 013701 032470    MOV      @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
9347 032442 005004          CLR      R4          ;RESULT S / B = 177777
9348 032444 005104          COM      R4
9349 032446 012702 063320    MOV      #MBUFO+6,R2 ;DEST ADDR = MBUFO+6
9350 032452 012705 063324    MOV      #DWTA+2,R5  ;SOURCE ADDR = DWTA+2
9351 032456 012703 063312    MOV      #MBUFO,R3   ;BASE DEST ADDR = MBUFO
9352 032462 005012          CLR      (R2)        ;MAKE [DEST] = 000000
9353 032464 000257          CCC
9354 032466 000264          264              ;CLEAR FLAGS
9355                                     ;N:C = 0100
9356 032470 012563 000006    2$: MOV      (R5)+,6(R3) ;TEST THE MOV - SM2,DM6
9357                                     ;N:C = 1000 ?
9358 032474 100003          BPL      3$
9359 032476 001402          BEQ      3$
9360 032500 102401          BVS      3$
9361 032502 103001          BCC      4$
9362
9363 032504 104001          3$: ERROR 1          ;MOV FAILED TO ALTER CODES PROPERLY
9364
9365 032506 020412          4$: CMP      R4,(R2)  ;RESULT CORRECT ??
9366 032510 001403          BEQ      TST462     ;:BR IF YES
9367
9368 032512 005003          CLR      R3          ;GET THE WAS DATA
9369 032514 051203          BIS      (R2),R3
9370 032516 104001          5$: ERROR 1          ;MOV DELIVERED THE WRONG RESULT

```


9371
9372
9373
9374
9375 032520
9376 032520 000004
9377 032522 012700 000462
9378 032526 013701 032560
9379 032532 005004
9380 032534 005104
9381 032536 012702 063312
9382 032542 012705 063324
9383 032546 012703 063276
9384 032552 005012
9385 032554 000257
9386 032556 000264
9387
9388 032560 011573 000010
9389
9390 032564 100003
9391 032566 001402
9392 032570 102401
9393 032572 103001
9394
9395 032574 104001
9396
9397 032576 020412
9398 032600 001403
9399
9400 032602 005003
9401 032604 051203
9402 032606 104001
9403
9404
9405
9406
9407 032610
9408 032610 000004
9409 032612 012700 000463
9410 032616 013701 032650
9411 032622 005004
9412 032624 005104
9413 032626 012702 063312
9414 032632 012705 063324
9415 032636 012703 063276
9416 032642 005012
9417 032644 000257
9418 032646 000264
9419
9420 032650 011573 000010
9421
9422 032654 100003
9423 032656 001402
9424 032660 102401
9425 032662 103001
9426

```

:*****
:*TEST 462      MOV SM1,DM7 TEST - N:C = 0100
:*****
TST462:
SCOPE          ;CALL THE SCOPE LOOP UTILITY
MOV #462,R0    ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4        ;RESULT S / B = 177777
COM R4
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #DWTA+2,R5;SOURCE ADDR = DWTA+2
MOV #ATA,R3   ;BASE DEST ADDR = ATA
CLR (R2)     ;MAKE [DEST] = 000000
CCC         ;CLEAR FLAGS
264        ;N:C = 0100

2$: MOV (R5),@10(R3) ;TEST THE MOV - SM1,DM7

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 TST463 ;;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
:*****
TST463:
SCOPE          ;CALL THE SCOPE LOOP UTILITY
MOV #463,R0    ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4        ;RESULT S / B = 177777
COM R4
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #DWTA+2,R5;SOURCE ADDR = DWTA+2
MOV #ATA,R3   ;BASE DEST ADDR = ATA
CLR (R2)     ;MAKE [DEST] = 000000
CCC         ;CLEAR FLAGS
264        ;N:C = 0100

2$: MOV (R5),@10(R3) ;TEST THE MOV - SM2,DM7

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

```

9427 032664 104001
9428
9429 032666 020412
9430 032670 001403
9431
9432 032672 005003
9433 032674 051203
9434 032676 104001
9435
9436
9437
9438
9439 032700
9440 032700 000004
9441 032702 012700 000464
9442 032706 013701 032726
9443 032712 012702 063312
9444 032716 010004
9445 032720 010205
9446 032722 005012
9447 032724 000257
9448
9449 032726 010015
9450
9451 032730 020412
9452 032732 001402
9453
9454 032734 011203
9455 032736 104001
9456
9457
9458
9459
9460 032740
9461 032740 000004
9462 032742 012700 000465
9463 032746 013701 032766
9464 032752 012702 063312
9465 032756 010004
9466 032760 010205
9467 032762 005012
9468 032764 000257
9469
9470 032766 010025
9471
9472 032770 020412
9473 032772 001402
9474
9475 032774 011203
9476 032776 104001
9477
9478
9479
9480
9481 033000
9482 033000 000004

3\$: ERROR 1 ;MOV FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;RESULT CORRECT ??
BEQ TST464 ;:BR IF YES
CLR R3 ;GET THE WAS DATA
BIS (R2),R3
5\$: ERROR 1 ;MOV DELIVERED THE WRONG RESULT
:*****
:*TEST 464 MOV SMO,DM1 TEST
:*****
TST464:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #464,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
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
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TST465 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
3\$: ERROR 1 ;MOV DELIVERED THE WRONG RESULT
:*****
:*TEST 465 MOV SMO,DM2 TEST
:*****
TST465:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #465,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
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
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TST466 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA
3\$: ERROR 1 ;MOV DELIVERED THE WRONG RESULT
:*****
:*TEST 466 MOV SMO,DM3 TEST
:*****
TST466:
SCOPE ;CALL THE SCOPE LOOP UTILITY

9483 033002 012700 000466
9484 033006 013701 033030
9485 033012 012702 063312
9486 033016 010004
9487 033020 012705 063306
9488 033024 005012
9489 033026 000257
9490
9491 033030 010035
9492
9493 033032 020412
9494 033034 001402
9495
9496 033036 011203
9497 033040 104001
9498
9499
9500
9501
9502 033042
9503 033042 000004
9504 033044 012700 000467
9505 033050 013701 033072
9506 033054 012702 063312
9507 033060 010004
9508 033062 012705 063314
9509 033066 005012
9510 033070 000257
9511
9512 033072 010045
9513
9514 033074 020412
9515 033076 001402
9516
9517 033100 011203
9518 033102 104001
9519
9520
9521
9522
9523 033104
9524 033104 000004
9525 033106 012700 000470
9526 033112 013701 033134
9527 033116 012702 063312
9528 033122 010004
9529 033124 012705 063310
9530 033130 005012
9531 033132 000257
9532
9533 033134 010055
9534
9535 033136 020412
9536 033140 001402
9537
9538 033142 011203

```
MOV #466,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV R0,R4 ;:RESULT S / B = TEST NUMBER
MOV #ATA+10,R5 ;:BASE DEST ADDR = ATA+10
CLR (R2) ;:[DEST] = 000000
CCC ;:SCOPE SYNC

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

CMP R4,(R2) ;:CORRECT RESULT
BEQ TST467 ;:BR IF YES

MOV (R2),R3 ;:GET THE WAS DATA
3$: 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 @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV R0,R4 ;:RESULT S / B = TEST NUMBER
MOV #MBUFO+2,R5 ;:R5 CONTAINS BASE DEST ADDR
CLR (R2) ;:[DEST] = 000000
CCC ;:SCOPE SYNC

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

CMP R4,(R2) ;:CORRECT RESULT ?
BEQ TST470 ;:BR IF YES

MOV (R2),R3 ;:GET THE WAS DATA
3$: 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 @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
MOV R0,R4 ;:RESULT S / B = TEST NUMBER
MOV #ATA+12,R5 ;:R5 CONTAINS BASE DEST ADDR
CLR (R2) ;:[DEST] = 000000
CCC ;:SCOPE SYNC

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

CMP R4,(R2) ;:CORRECT RESULT ?
BEQ TST471 ;:BR IF YES

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

9539 033144 104001
9540
9541
9542
9543
9544
9545 033146
9546 033146 000004
9547 033150 012700 000471
9548 033154 013701 033176
9549 033160 012702 063316
9550 033164 010004
9551 033166 012705 063312
9552 033172 005012
9553 033174 000257
9554
9555 033176 010065 000004
9556
9557 033202 020412
9558 033204 001402
9559
9560 033206 011203
9561 033210 104001
9562
9563
9564
9565
9566 033212
9567 033212 000004
9568 033214 012700 000472
9569 033220 013701 033242
9570 033224 012704 177652
9571 033230 012705 000252
9572 033234 005003
9573 033236 000257
9574 033240 000266
9575
9576 033242 110503
9577
9578 033244 100003
9579 033246 001402
9580 033250 102401
9581 033252 103001
9582
9583 033254 104002
9584
9585 033256 020403
9586 033260 001401
9587
9588 033262 104002
9589
9590
9591
9592
9593 033264
9594 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 @#2\$,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) ;:[DEST] = 000000
CCC ;:SCOPE SYNC
2\$: MOV R0,4(R5) ;:TEST THE MOV
CMP R4,(R2) ;:RESULT CORRECT ?
BEQ TST472 ;:BR IF YES
3\$: MOV (R2),R3 ;:GET THE WAS DATA
ERROR 1 ;:MOV DELIVERED THE WRONG RESULT
:*****
:*TEST 472 MOVB TEST - SMO,DMO - EXTEND 1'S
:*****
TST472:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #472,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177652,R4 ;:RESULT S / B = 177652
MOV #252,R5 ;:SOURCE OP = 252
CLR R3 ;:[DEST] = 000000
CCC ;:CLEAR FLAGS
266 ;:N:C = 0110
2\$: MOVB R5,R3 ;:TEST THE MOVB
BPL 3\$;:N:C = 1000 ?
BEQ 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 2 ;:MOVB FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;:RESULT CORRECT ?
BEQ TST473 ;:BR IF YES
5\$: ERROR 2 ;:MOVB DELIVERED THE WRONG RESULT
:*****
:*TEST 473 MOVB TEST - SMO,DMO - EXTEND 0'S
:*****
TST473:
SCOPE ;CALL THE SCOPE LOOP UTILITY

```
9595 033266 012700 000473      MOV      #473,R0          ;;LOAD R0 WITH TEST NUMBER
9596 033272 013701 033314      MOV      @#2$,R1        ;;LOAD R1 WITH TEST INSTRUCTION WORD
9597 033276 005004                CLR      R4              ;RESULT S / B = 000000
9598 033300 012705 177400      MOV      #177400,R5     ;SOURCE OP = 177400
9599 033304 005003                CLR      R3              ;[DEST] = 177777
9600 033306 005103                COM      R3
9601 033310 000257                CCC
9602 033312 000271                271                      ;CLEAR FLAGS
9603                                ;N:C = 1001
9604 033314 110503      2$:  MOV B      R5,R3          ;TEST THE MOV B
9605
9606 033316 100403                BMI      3$              ;N:C = 0101 ?
9607 033320 001002                BNE      3$
9608 033322 102401                BVS      3$
9609 033324 103401                BCS      4$
9610
9611 033326 104002      3$:  ERROR      2              ;MOV B FAILED TO ALTER CODES PROPERLY
9612
9613 033330 020403      4$:  CMP      R4,R3          ;RESULT CORRECT ?
9614 033332 001401                BEQ      TST474         ;;BR IF YES
9615
9616 033334 104002      5$:  ERROR      2              ;MOV B DELIVERED THE WRONG RESULT
9617
9618                                ;*****
9619                                ;*TEST 474      MOV B TEST - SM1,DMO - SOURCE ADDR EVEN
9620                                ;*****
9621                                TST474:
9622 033336 000004                SCOPE                    ;CALL THE SCOPE LOOP UTILITY
9623 033340 012700 000474      MOV      #474,R0        ;;LOAD R0 WITH TEST NUMBER
9624 033344 013701 033364      MOV      @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
9625 033350 005004                CLR      R4              ;RESULT S / B = 000000
9626 033352 012705 064630      MOV      #DBTA,R5     ;SOURCE ADDR = DBTA
9627 033356 005003                CLR      R3              ;[DEST] = 177777
9628 033360 005103                COM      R3
9629 033362 000257                CCC                      ;SCOPE SYNC
9630
9631 033364 111503      2$:  MOV B      (R5),R3        ;TEST THE MOV B
9632
9633 033366 020403                CMP      R4,R3          ;RESULT CORRECT ?
9634 033370 001401                BEQ      TST475         ;;BR IF YES
9635
9636 033372 104002      3$:  ERROR      2              ;MOV B DELIVERED THE WRONG RESULT
9637
9638                                ;*****
9639                                ;*TEST 475      MOV B TEST - SM1,DMO - SOURCE ADDR ODD
9640                                ;*****
9641                                TST475:
9642 033374 000004                SCOPE                    ;CALL THE SCOPE LOOP UTILITY
9643 033376 012700 000475      MOV      #475,R0        ;;LOAD R0 WITH TEST NUMBER
9644 033402 013701 033424      MOV      @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
9645 033406 012704 000125      MOV      #125,R4        ;RESULT S / B = 125
9646 033412 012705 064633      MOV      #DBTA+3,R5    ;SOURCE ADDR = DBTA+3
9647 033416 012703 177400      MOV      #177400,R3     ;[DEST] = 177400
9648 033422 000257                CCC                      ;SCOPE SYNC
9649
9650 033424 111503      2$:  MOV B      (R5),R3        ;TEST THE MOV B
```

```

9651
9652 033426 020403          CMP    R4,R3          ;RESULT CORRECT ?
9653 033430 001401          BEQ    TST476         ;;BR IF YES
9654
9655 033432 104002          3$:   ERROR    2          ;MOV B DELIVERED THE WRONG RESULT
9656
9657          ;;*****
9658          ;*TEST 476      MOV B TEST - SM2,DMO - SOURCE ADDR ODD
9659          ;;*****
9660 033434          TST476:
9661 033434 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
9662 033436 012700 000476      MOV    #476,R0      ;;LOAD R0 WITH TEST NUMBER
9663 033442 013701 033462      MOV    @#2$,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
9664 033446 012704 177777      MOV    #-1,R4      ;RESULT S / B = 177777
9665 033452 012705 064631      MOV    #DBTA+1,R5  ;SOURCE ADDR = DBTA+1
9666 033456 005003          CLR    R3          ;[DEST] = 000000
9667 033460 000257          CCC          ;SCOPE SYNC
9668
9669 033462 112503          2$:   MOV B    (R5)+,R3      ;TEST THE MOV B
9670
9671 033464 020403          CMP    R4,R3          ;RESULT CORRECT ?
9672 033466 001401          BEQ    4$           ;BR IF YES
9673
9674 033470 104002          3$:   ERROR    2          ;MOV B DELIVERED THE WRONG RESULT
9675
9676 033472 022705 064632      4$:   CMP    #DBTA+2,R5  ;DID MOV B INCREMENT SRC REG ?
9677 033476 001401          BEQ    TST477       ;;BR IF YES
9678
9679 033500 104005          5$:   ERROR    5          ;MOV B FAILED TO UPDATE SRC REG
9680
9681          ;;*****
9682          ;*TEST 477      MOV B TEST - SM2,DMO - SOURCE ADDR EVEN
9683          ;;*****
9684 033502          TST477:
9685 033502 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
9686 033504 012700 000477      MOV    #477,R0      ;;LOAD R0 WITH TEST NUMBER
9687 033510 013701 033530      MOV    @#2$,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
9688 033514 005004          CLR    R4          ;RESULT S / B = 000000
9689 033516 012705 064630      MOV    #DBTA,R5    ;SOURCE ADDR = DBTA
9690 033522 012703 177400      MOV    #177400,R3  ;[DEST] = 177400
9691 033526 000257          CCC          ;SCOPE SYNC
9692
9693 033530 112503          2$:   MOV B    (R5)+,R3      ;TEST THE MOV B
9694
9695 033532 020403          CMP    R4,R3          ;RESULT CORRECT ?
9696 033534 001401          BEQ    4$           ;BR IF YES
9697
9698 033536 104002          3$:   ERROR    2          ;MOV B DELIVERED THE WRONG RESULT
9699
9700 033540 022705 064631      4$:   CMP    #DBTA+1,R5  ;DID MOV B INCREMENT SRC REG ?
9701 033544 001401          BEQ    TST500       ;;BR IF YES
9702
9703 033546 104005          5$:   ERROR    5          ;MOV B FAILED TO UPDATE SOURCE REG
9704
9705          ;;*****
9706          ;*TEST 500      MOV B TEST - SM1,DM1 - SRC ADR ODD / DST ADR EVEN
  
```

```

9707
9708 033550
9709 033550 000004
9710 033552 012700 000500
9711 033556 013701 033602
9712 033562 012702 063312
9713 033566 012704 000377
9714 033572 012705 064631
9715 033576 005012
9716 033600 000257
9717
9718 033602 111512
9719
9720 033604 020412
9721 033606 001402
9722
9723 033610 011203
9724 033612 104001
9725
9726
9727
9728
9729 033614
9730 033614 000004
9731 033616 012700 000501
9732 033622 013701 033650
9733 033626 012702 063312
9734 033632 012704 000377
9735 033636 012705 064631
9736 033642 005012
9737 033644 010203
9738 033646 000257
9739
9740 033650 111523
9741
9742 033652 020412
9743 033654 001402
9744
9745 033656 011203
9746 033660 104001
9747
9748 033662 022703 063313
9749 033666 001401
9750
9751 033670 104005
9752
9753
9754
9755
9756 033672
9757 033672 000004
9758 033674 012700 000502
9759 033700 013701 033730
9760 033704 012702 063312
9761 033710 012704 000377
9762 033714 012705 064631
  
```

```

*****
TST500:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #500,R0                          ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2                        ;DEST ADDR = MBUFO
MOV #377,R4                          ;RESULT S / B = 377
MOV #DBTA+1,R5                       ;SRC ADDR = DBTA +1
CLR (R2)                             ;[DEST] = 000000
CCC                                   ;CLEAR FLAGS - SCOPE SYNC

2$:  MOV B (R5),(R2)                  ;TEST THE MOV B
    CMP R4,(R2)                      ;CORRECT RESULT ?
    BEQ TST501                       ;;BR IF YES

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

*****
;*TEST 501 MOV B TEST - SM1,DM2 - SRC ADR ODD / DST ADR EVEN
*****
TST501:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #501,R0                          ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2                        ;DEST ADDR = MBUFO
MOV #377,R4                          ;RESULT S / B = 377
MOV #DBTA+1,R5                       ;SRC ADDR = DBTA +1
CLR (R2)                             ;[DEST] = 000000
MOV R2,R3                            ;[R3] = DEST ADDR
CCC                                   ;CLEAR FLAGS - SCOPE SYNC

2$:  MOV B (R5),(R3)+                 ;TEST THE MOV B
    CMP R4,(R2)                      ;CORRECT RESULT ?
    BEQ 4$                            ;BR IF YES

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

4$:  CMP #MBUFO+1,R3                 ;DID MOV B INCREMENT THE DEST REG ?
    BEQ TST502                       ;;BR IF YES

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

*****
;*TEST 502 MOV B TEST - SM1,DM3 - SRC ADR ODD / DST ADR EVEN
*****
TST502:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV #502,R0                          ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2                        ;DEST ADDR = MBUFO
MOV #377,R4                          ;RESULT S / B = 377
MOV #DBTA+1,R5                       ;SRC ADDR = DBTA +1
  
```

9763 033720 005012
9764 033722 012703 063306
9765 033726 000257
9766
9767 033730 111533
9768
9769 033732 022703 063310
9770 033736 001401
9771
9772 033740 104005
9773
9774 033742 020412
9775 033744 001402
9776
9777 033746 011203
9778 033750 104001
9779
9780
9781
9782
9783 033752
9784 033752 000004
9785 033754 012700 000503
9786 033760 013701 034010
9787 033764 012702 063312
9788 033770 012704 000377
9789 033774 012705 064631
9790 034000 005012
9791 034002 012703 063313
9792 034006 000257
9793
9794 034010 111543
9795
9796 034012 020302
9797 034014 001401
9798
9799 034016 104005
9800
9801 034020 020412
9802 034022 001402
9803
9804 034024 011203
9805 034026 104001
9806
9807
9808
9809
9810 034030
9811 034030 000004
9812 034032 012700 000504
9813 034036 013701 034066
9814 034042 012702 063312
9815 034046 012704 000377
9816 034052 012705 064631
9817 034056 005012
9818 034060 012703 063310

```

CLR      (R2)          ;[DEST] = 000000
MOV      #ATA+10,R3   ;BASE DEST ADDR = ATA +10
CCC      ;CLEAR FLAGS - SCOPE SYNC

2$:      MOV B      (R5),@(R3)+ ;TEST THE MOV B

        CMP      #ATA+12,R3   ;DID DEST REG GET INCREMENTED ?
        BEQ      4$          ;BR IF YES

3$:      ERROR    5          ;MOV B FAILED TO UPDATE DEST REG

4$:      CMP      R4,(R2)     ;CORRECT RESULT ?
        BEQ      TST503      ;;BR IF YES

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

;*****
;*TEST 503      MOV B TEST - SM1,DM4 - SRC ADR ODD / DST ADR EVEN
;*****
TST503:
        SCOPE
        MOV      #503,R0     ;CALL THE SCOPE LOOP UTILITY
        MOV      @#2$,R1    ;;LOAD R0 WITH TEST NUMBER
        MOV      #MBUF0,R2  ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV      #377,R4    ;DEST ADDR = MBUF0
        MOV      #DBTA+1,R5 ;RESULT S / B = 377
        CLR      (R2)       ;SRC ADDR = DBTA +1
        MOV      #MBUF0+1,R3 ;[DEST] = 000000
        CCC      ;INITIAL DEST ADDR = MBUF0+1
                ;CLEAR FLAGS - SCOPE SYNC

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

        CMP      R3,R2     ;DID MOV B DECREMENT DEST REG ?
        BEQ      4$          ;BR IF YES

3$:      ERROR    5          ;MOV B FAILED TO UPDATE DEST REG

4$:      CMP      R4,(R2)     ;CORRECT RESULT ?
        BEQ      TST504      ;;BR IF YES

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

;*****
;*TEST 504      MOV B TEST - SM1,DM5 - SRC ADR ODD / DST ADR EVEN
;*****
TST504:
        SCOPE
        MOV      #504,R0     ;CALL THE SCOPE LOOP UTILITY
        MOV      @#2$,R1    ;;LOAD R0 WITH TEST NUMBER
        MOV      #MBUF0,R2  ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV      #377,R4    ;DEST ADDR = MBUF0
        MOV      #DBTA+1,R5 ;RESULT S / B = 377
        CLR      (R2)       ;SRC ADDR = DBTA +i
        MOV      #ATA+12,R3 ;[DEST] = 000000
                ;INITIAL DEST ADDR = ATA +12
    
```


9819 034064 000257
 9820
 9821 034066 111553
 9822
 9823 034070 022703 063306
 9824 034074 001401
 9825
 9826 034076 104005
 9827
 9828 034100 020412
 9829 034102 001402
 9830
 9831 034104 011203
 9832 034106 104001
 9833
 9834
 9835
 9836
 9837 034110
 9838 034110 000004
 9839 034112 012700 000505
 9840 034116 013701 034146
 9841 034122 012702 063312
 9842 034126 012704 000377
 9843 034132 012705 064631
 9844 034136 005012
 9845 034140 012703 063320
 9846 034144 000257
 9847
 9848 034146 111563 177772
 9849
 9850 034152 020412
 9851 034154 001402
 9852
 9853 034156 011203
 9854 034160 104001
 9855
 9856
 9857
 9858
 9859 034162
 9860 034162 000004
 9861 034164 012700 000506
 9862 034170 013701 034220
 9863 034174 012702 063312
 9864 034200 012704 000377
 9865 034204 012705 064631
 9866 034210 005012
 9867 034212 012703 063276
 9868 034216 000257
 9869
 9870 034220 111573 000010
 9871
 9872 034224 020412
 9873 034226 001402
 9874

```

      CCC                                ;CLEAR FLAGS - SCOPE SYNC
2$:  MOV B (R5),@-(R3)                  ;TEST THE MOV B
      CMP #ATA+10,R3                    ;DID MOV B DECREMENT DEST REG ?
      BEQ 4$                             ;BR IF YES
3$:  ERROR 5                             ;MOV B FAILED TO UPDATE DEST REG
4$:  CMP R4,(R2)                         ;CORRECT RESULT ?
      BEQ TST505                          ;: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
;*****
TST505:
      SCOPE                               ;CALL THE SCOPE LOOP UTILITY
      MOV #505,R0                          ;:LOAD R0 WITH TEST NUMBER
      MOV @#2$,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV #MBUF0,R2                         ;DEST ADDR = MBUF0
      MOV #377,R4                           ;RESULT S / B = 377
      MOV #DBTA+1,R5                        ;SRC ADDR = DBTA +1
      CLR (R2)                               ;[DEST] = 000000
      MOV #MBUF0+6,R3                       ;BASE DEST ADDR = MBUF0+6
      CCC                                    ;CLEAR FLAGS - SCOPE SYNC
2$:  MOV B (R5),-6(R3)                    ;TEST THE MOV B
      CMP R4,(R2)                         ;CORRECT RESULT ?
      BEQ TST506                          ;: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
;*****
TST506:
      SCOPE                               ;CALL THE SCOPE LOOP UTILITY
      MOV #506,R0                          ;:LOAD R0 WITH TEST NUMBER
      MOV @#2$,R1                          ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV #MBUF0,R2                         ;DEST ADDR = MBUF0
      MOV #377,R4                           ;RESULT S / B = 377
      MOV #DBTA+1,R5                        ;SRC ADDR = DBTA +1
      CLR (R2)                               ;[DEST] = 000000
      MOV #ATA,R3                           ;BASE DEST ADDR = ATA
      CCC                                    ;CLEAR FLAGS - SCOPE SYNC
2$.  MOV B (R5),@10(R3)                   ;TEST THE MOV B
      CMP R4,(R2)                         ;CORRECT RESULT ?
      BEQ TST507                          ;:BR IF YES
  
```

9875 034230 011203
9876 034232 10400i
9877
9878
9879
9880
9881 034234
9882 034234 000004
9883 034236 012700 000507
9884 034242 013701 034270
9885 034246 012702 063312
9886 034252 012704 000377
9887 034256 012703 177777
9888 034262 010205
9889 034264 005012
9890 034266 000257
9891
9892 034270 110315
9893
9894 034272 020412
9895 034274 001402
9896
9897 034276 011203
9898 034300 104001
9899
9900
9901
9902
9903 034302
9904 034302 000004
9905 034304 012700 000510
9906 034310 013701 034336
9907 034314 012702 063312
9908 034320 012704 000377
9909 034324 012703 177777
9910 034330 010205
9911 034332 005012
9912 034334 000257
9913
9914 034336 110325
9915
9916 034340 020412
9917 034342 001402
9918
9919 034344 011203
9920 034346 104001
9921
9922
9923
9924
9925 034350
9926 034350 000004
9927 034352 012700 000511
9928 034356 013701 034406
9929 034362 012702 063312
9930 034366 012704 000377

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

*****
*TEST 507 MOV B SMO,DM1 TEST
*****
TST507:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #507,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,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 TST510 ;BR IF YES

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

*****
*TEST 510 MOV B SMO,DM2 TEST
*****
TST510:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #510,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,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 TST511 ;BR IF YES

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

*****
*TEST 511 MOV B SMO,DM3 TEST
*****
TST511:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #511,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
```

9931 034372 012703 177777
9932 034376 012705 063306
9933 034402 005012
9934 034404 000257
9935
9936 034406 110335
9937
9938 034410 020412
9939 034412 001402
9940
9941 034414 011203
9942 034416 104001
9943
9944
9945
9946
9947 034420
9948 034420 000004
9949 034422 012700 000512
9950 034426 013701 034456
9951 034432 012702 063312
9952 034436 012704 177400
9953 034442 012703 177777
9954 034446 012705 063314
9955 034452 005012
9956 034454 000257
9957
9958 034456 110345
9959
9960 034460 020412
9961 034462 001402
9962
9963 034464 011203
9964 034466 104001
9965
9966
9967
9968
9969 034470
9970 034470 000004
9971 034472 012700 000513
9972 034476 013701 034526
9973 034502 012702 063312
9974 034506 012704 000377
9975 034512 012703 177777
9976 034516 012705 063314
9977 034522 005012
9978 034524 000257
9979
9980 034526 110365 177776
9981
9982 034532 020412
9983 034534 001402
9984
9985 034536 011203
9986 034540 104001

MOV #-1,R3 ;SOURCE OP IN R3
MOV #ATA+10,R5 ;BASE DEST ADDR = ATA+10
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2\$: MOV B R3,@(R5)+ ;TEST THE MOV B
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TST512 ;;BR IF YES
3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED THE WRONG RESULT
:*****
:*TEST 512 MOV B SMO,DM4 TEST
:*****
TST512:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #512,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #177400,R4 ;RESULT S / B = 177400
MOV #-1,R3 ;R3 CONTAINS SOURCE OP
MOV #MBUFO+2,R5 ;BASE DEST ADDR = MBUFO+2
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2\$: MOV B R3,-(R5) ;TEST THE MOV B
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TST513 ;;BR IF YES
3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED THE WRONG RESULT
:*****
:*TEST 513 MOV B SMO,DM6 TEST
:*****
TST513:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #513,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #377,R4 ;RESULT S / B = 377
MOV #-1,R3 ;R3 CONTAINS SOURCE OP
MOV #MBUFO+2,R5 ;BASE DEST ADDR = MBUFO+2
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2\$: MOV B R3,-2(R5) ;TEST THE MOV B
CMP R4,(R2) ;RESULT CORRECT ?
BEQ TST514 ;;BR IF YES
3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;MOV B DELIVERED THE WRONG RESULT

9987
 9988
 9989
 9990
 9991 034542
 9992 034542 000004
 9993 034544 012700 000514
 9994 034550 013701 034574
 9995 034554 012704 177777
 9996 034560 012705 125252
 9997 034564 012703 052525
 9998 034570 000257
 9999 034572 000267
 10000
 10001 034574 050503
 10002
 10003 034576 100003
 10004 034600 001402
 10005 034602 102401
 10006 034604 103401
 10007
 10008 034606 104002
 10009
 10010 034610 020403
 10011 034612 001401
 10012
 10013 034614 104002
 10014
 10015
 10016
 10017
 10018 034616
 10019 034616 000004
 10020 034620 012700 000515
 10021 034624 013701 034654
 10022
 10023 034630 032737 002000 063234
 10024 034636 001401
 10025 034640 000000
 10026 034642 005004
 10027 034644 005005
 10028 034646 005003
 10029 034650 000257
 10030 034652 000270
 10031
 10032 034654 050503
 10033
 10034 034656 100403
 10035 034660 001002
 10036 034662 102401
 10037 034664 103001
 10038
 10039 034666 104002
 10040
 10041 034670 020403
 10042 034672 001401

```

:*****
:*TEST 514      BIS TEST - SMO,DMO - N:C = 0111
:*****
TST514:
      SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #514,R0                            ;:LOAD R0 WITH TEST NUMBER
MOV      @#2$,R1                            ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #-1,R4                             ;:RESULT S / B = 177777
MOV      #125252,R5                         ;:SRC OPR = 125252
MOV      #52525,R3                          ;:[DEST] = 52525
CCC                                           ;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      TST515                       ;;BR IF YES

5$:   ERROR    2                            ;BIS DELIVERED THE WRONG RESULT

:*****
:*TEST 515      BIS TEST - SMO,DMO - N:C = 1000
:*****
TST515:
      SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #515,R0                            ;:LOAD R0 WITH TEST NUMBER
MOV      @#2$,R1                            ;:LOAD R1 WITH TEST INSTRUCTION WORD
.SBTTL  USER CONTROLLED BREAKPOINT -- BIT10
      BIT      #BIT10,@#BPTLOC             ;BREAKPOINT HALT SET ??
      BEQ      .+4                          ;BR IF NOT
      HALT                                         ;BREAK-DEPRESS CONTINUE TO CONTINUE
      CLR      R4                             ;RESULT S / B = 000000
      CLR      R5                             ;SRC OPR = 000000
      CLR      R3                             ;:[DEST] = 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      TST516                       ;;BR IF YES
  
```

10043
 10044 034674 104002
 10045
 10046
 10047
 10048
 10049 034676
 10050 034676 000004
 10051 034700 012700 000516
 10052 034704 013701 034730
 10053 034710 012704 100000
 10054 034714 012705 077777
 10055 034720 012703 177777
 10056 034724 000257
 10057 034726 000267
 10058
 10059 034730 040503
 10060
 10061 034732 100003
 10062 034734 001402
 10063 034736 102401
 10064 034740 103401
 10065
 10066 034742 104002
 10067
 10068 034744 020403
 10069 034746 001401
 10070
 10071 034750 104002
 10072
 10073
 10074
 10075
 10076 034752
 10077 034752 000004
 10078 034754 012700 000517
 10079 034760 013701 034776
 10080 034764 005004
 10081 034766 005005
 10082 034770 005003
 10083 034772 000257
 10084 034774 000270
 10085
 10086 034776 040503
 10087
 10088 035000 100403
 10089 035002 001002
 10090 035004 102401
 10091 035006 103001

```

5$:      ERROR      2              ;BIS DELIVERED THE WRONG RESULT
:*****
:*TEST 516      BIC TEST - SMO,DMO - N:C = 0111
:*****
TST516:
          SCOPE              ;CALL THE SCOPE LOOP UTILITY
MOV      #516,R0             ;:LOAD R0 WITH TEST NUMBER
MOV      @#2$,R1             ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV      #100000,R4          ;:RESULT S / B = 100000
MOV      #77777,R5           ;:SRC OPR = 77777
MOV      #-1,R3              ;:[DEST] = 177777
CCC                      ;: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      TST517        ;:BR IF YES

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

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

```
10092
10093 035010 104002 3$: ERROR 2 ;BIC FAILED TO ALTER CODES PROPERLY
10094
10095 035012 020403 4$: CMP R4,R3 ;CORRECT RESULT ?
10096 035014 001401 BEQ TST520 ;:BR IF YES
10097
10098 035016 104002 5$: ERROR 2 ;BIC DELIVERED THE WRONG RESULT
10099
10100
10101
10102
10103 035020
10104 035020 000004
10105 035022 012700 000520
10106 035026 013701 035052
10107 035032 012704 100000
10108 035036 012705 100000
10109 035042 012703 100000
10110 035046 000257
10111 035050 000267
10112
10113 035052 030503 2$: BIT R5,R3 ;TEST THE BIT
10114
10115 035054 100003 BPL 3$ ;N:C = 1001
10116 035056 001402 BEQ 3$
10117 035060 102401 BVS 3$
10118 035062 103401 BCS 4$
10119
10120 035064 104002 3$: ERROR 2 ;BIT FAILED TO ALTER CODES PROPERLY
10121
10122 035066 020403 4$: CMP R4,R3 ;CORRECT RESULT ?
10123 035070 001402 BEQ TST521 ;:BR IF YES
10124
10125 035072 011203 5$: MOV (R2),R3 ;GET THE WAS DATA
10126 035074 104002 ERROR 2 ;BIT DELIVERED A RESULT
10127
10128
10129
10130
10131 035076
10132 035076 000004
10133 035100 012700 000521
10134 035104 013701 035126
10135 035110 012704 125252
10136 035114 012705 052525
10137 035120 010403
10138 035122 000257
10139 035124 000270
10140
10141 035126 030503 2$: BIT R5,R3 ;TEST THE BIT
10142
10143 035130 100403 BMI 3$ ;N:C = 0100
10144 035132 001002 BNE 3$
10145 035134 102401 BVS 3$
10146 035136 103001 BCC 4$
10147
```

10148 035140 104002
 10149
 10150 035142 020403
 10151 035144 001401
 10152
 10153 035146 104002
 10154
 10155
 10156
 10157
 10158 035150
 10159 035150 000004
 10160 035152 012700 000522
 10161 035156 013701 035200
 10162 035162 012704 000001
 10163 035166 005005
 10164 035170 012703 000001
 10165 035174 000257
 10166 035176 000266
 10167
 10168 035200 020503
 10169
 10170 035202 100003
 10171 035204 001402
 10172 035206 102401
 10173 035210 103401
 10174
 10175 035212 104002
 10176
 10177 035214 020403
 10178 035216 001401
 10179
 10180 035220 104002
 10181
 10182
 10183
 10184
 10185 035222
 10186 035222 000004
 10187 035224 012700 000523
 10188 035230 013701 035252
 10189 035234 012704 177777
 10190 035240 012705 177777
 10191 035244 010403
 10192 035246 000257
 10193 035250 000272
 10194
 10195 035252 020503
 10196
 10197 035254 100403
 10198 035256 001002
 10199 035260 102401
 10200 035262 103001
 10201
 10202 035264 104002
 10203

```

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

4$: CMP R4,R3 ;CORRECT RESULT ?
   BEQ T522 ;: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 @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
   MOV #+1,R4 ;RESULT S / B = +1
   CLR R5 ;SRC OPR = 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 T523 ;: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 OPR = 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
   BNE 3$
   BVS 3$
   BCC 4$

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

```

10204 035266 020403
 10205 035270 001401
 10206
 10207 035272 104002
 10208
 10209
 10210
 10211
 10212 035274
 10213 035274 000004
 10214 035276 012700 000524
 10215 035302 013701 035324
 10216 035306 012704 000001
 10217 035312 012705 100000
 10218 035316 012703 000001
 10219 035322 000257
 10220
 10221 035324 020503
 10222
 10223 035326 100403
 10224 035330 001402
 10225 035332 102001
 10226 035334 103001
 10227
 10228 035336 104002
 10229
 10230 035340 020403
 10231 035342 001401
 10232
 10233 035344 104002
 10234
 10235
 10236
 10237
 10238 035346
 10239 035346 000004
 10240 035350 012700 000525
 10241 035354 013701 035404
 10242 035360 012702 063312
 10243 035364 012704 177777
 10244 035370 012705 125252
 10245 035374 012712 052525
 10246 035400 000257
 10247 035402 000267
 10248
 10249 035404 050512
 10250
 10251 035406 100003
 10252 035410 001402
 10253 035412 102401
 10254 035414 103401
 10255
 10256 035416 104001
 10257
 10258 035420 020412
 10259 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
;*****
T51524:
      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 OPR = 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,DM1 - N:C = 0111
;*****
T51525:
      SCOPE                    ;CALL THE SCOPE LOOP UTILITY
      MOV      #525,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 = 177777
      MOV      #125252,R5    ;SRC OPR = 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
  
```


10260
 10261 035424 011203
 10262 035426 104001
 10263
 10264
 10265
 10266
 10267 035430
 10268 035430 000004
 10269 035432 012700 000526
 10270 035436 013701 035460
 10271 035442 012702 063312
 10272 035446 005004
 10273 035450 005005
 10274 035452 005012
 10275 035454 000257
 10276 035456 000270
 10277
 10278 035460 050512
 10279
 10280 035462 100403
 10281 035464 001002
 10282 035466 102401
 10283 035470 103001
 10284
 10285 035472 104001
 10286
 10287 035474 020412
 10288 035476 001402
 10289
 10290 035500 011203
 10291 035502 104001
 10292
 10293
 10294
 10295
 10296 035504
 10297 035504 000004
 10298 035506 012700 000527
 10299 035512 013701 035542
 10300 035516 012702 063312
 10301 035522 012704 100000
 10302 035526 012705 077777
 10303 035532 012712 177777
 10304 035536 000257
 10305 035540 000267
 10306
 10307 035542 040512
 10308
 10309 035544 100003
 10310 035546 001402
 10311 035550 102401
 10312 035552 103401
 10313
 10314 035554 104001
 10315

```

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

;*****
;*TEST 526      BIS TEST - SMO,DM1 - N:C = 1000
;*****
TST526:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #526,R0     ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #MBUFO,R2   ;DEST ADDR = MBUFO
      CLR      R4          ;RESULT S / B = 000000
      CLR      R5          ;SRC OPR = 000000
      CLR      (R2)        ;[DEST] = 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      TST527     ;;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
;*****
TST527:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #527,R0     ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1     ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #MBUFO,R2   ;DEST ADDR = MBUFO
      MOV      #100000,R4  ;RESULT S / B = 100000
      MOV      #77777,R5   ;SRC OPR = 77777
      MOV      #-1,(R2)    ;[DEST] = 177777
      CCC                      ;CLEAR FLAGS
      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
  
```

10316 035556 020412
10317 035560 001402
10318
10319 035562 011203
10320 035564 104001
10321
10322
10323
10324
10325 035566
10326 035566 000004
10327 035570 012700 000530
10328 035574 013701 035616
10329 035600 012702 063312
10330 035604 005004
10331 035606 005005
10332 035610 005012
10333 035612 000257
10334 035614 000270
10335
10336 035616 040512
10337
10338 035620 100403
10339 035622 001002
10340 035624 102401
10341 035626 103001
10342
10343 035630 104001
10344
10345 035632 020412
10346 035634 001402
10347
10348 035636 011203
10349 035640 104001
10350
10351
10352
10353
10354 035642
10355 035642 000004
10356 035644 012700 000531
10357 035650 013701 035700
10358 035654 012702 063312
10359 035660 012704 100000
10360 035664 012705 100000
10361 035670 012712 100000
10362 035674 000257
10363 035676 000267
10364
10365 035700 030512
10366
10367 035702 100003
10368 035704 001402
10369 035706 102401
10370 035710 103401
10371

```
4$:  CMP      R4,(R2)      ;CORRECT RESULT ?  
     BEQ      TST530      ;;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  
;*****  
TST530:  
     SCOPE                    ;CALL THE SCOPE LOOP UTILITY  
     MOV      #530,R0        ;;LOAD R0 WITH TEST NUMBER  
     MOV      @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD  
     MOV      #MBUFO,R2      ;DEST ADDR = MBUFO  
     CLR      R4             ;RESULT S / B = 000000  
     CLR      R5             ;SRC OPR = 000000  
     CLR      (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      TST531      ;;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  
;*****  
TST531:  
     SCOPE                    ;CALL THE SCOPE LOOP UTILITY  
     MOV      #531,R0        ;;LOAD R0 WITH TEST NUMBER  
     MOV      @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD  
     MOV      #MBUFO,R2      ;DEST ADDR = MBUFO  
     MOV      #100000,R4     ;RESULT S / B = 100000  
     MOV      #100000,R5     ;SRC OPR = 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$
```

10372 035712 104001
10373
10374 035714 020412
10375 035716 001402
10376
10377 035720 011203
10378 035722 104001
10379
10380
10381
10382
10383 035724
10384 035724 000004
10385 035726 012700 000532
10386 035732 013701 035762
10387 035736 012702 063312
10388 035742 012704 052525
10389 035746 012705 125252
10390 035752 012712 052525
10391 035756 000257
10392 035760 000270
10393
10394 035762 030512
10395
10396 035764 100403
10397 035766 001002
10398 035770 102401
10399 035772 103001
10400
10401 035774 104001
10402
10403 035776 020412
10404 036000 001402
10405
10406 036002 011203
10407 036004 104001
10408
10409
10410
10411 036006
10412 036006 000004
10413 036010 012700 000533
10414 036014 013701 036044
10415 036020 012702 063312
10416 036024 012704 177777
10417 036030 012705 177777
10418 036034 012712 177777
10419 036040 000257
10420 036042 000272
10421
10422 036044 020512
10423
10424 036046 100403
10425 036050 001002
10426 036052 102401
10427 036054 103001

```
3$: ERROR 1 ;BIT FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
   BEQ T532 ;:BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
   ERROR 1 ;BIT DELIVERED A RESULT
;*****
;*TEST 532 BIT TEST - SMO,DM1 - N:C = 1000
;*****
T532:
   SCOPE ;CALL THE SCOPE LOOP UTILITY
   MOV #532,R0 ;:LOAD R0 WITH TEST NUMBER
   MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
   MOV #MBUF0,R2 ;DEST ADDR = MBUF0
   MOV #52525,R4 ;RESULT S / B = 52525
   MOV #125252,R5 ;SRC OPR = 125252
   MOV #52525,(R2) ;[DEST] = 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 T533 ;:BR IF YES
5$: MOV (R2),R3 ;GET THE WAS DATA
   ERROR 1 ;BIT DELIVERED A RESULT
;*****
;*TEST 533 CMP TEST - SMO,DM1 - N:C = 1010
;*****
T533:
   SCOPE ;CALL THE SCOPE LOOP UTILITY
   MOV #533,R0 ;:LOAD R0 WITH TEST NUMBER
   MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
   MOV #MBUF0,R2 ;DEST ADDR = MBUF0
   MOV #-1,R4 ;RESULT S / B = -1
   MOV #-1,R5 ;SRC OPR = 177777
   MOV #-1,(R2) ;[DEST] = 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$
```

```

10428
10429 036056 104001
10430
10431 036060 020412
10432 036062 001402
10433
10434 036064 011203
10435 036066 104001
10436
10437
10438
10439
10440 036070
10441 036070 000004
10442 036072 012700 000534
10443 036076 013701 036124
10444 036102 012702 063312
10445 036106 012704 000001
10446 036112 005005
10447 036114 012712 000001
10448 036120 000257
10449 036122 000266
10450
10451 036124 020512
10452
10453 036126 100003
10454 036130 001402
10455 036132 102401
10456 036134 103401
10457
10458 036136 104001
10459
10460 036140 020412
10461 036142 001402
10462
10463 036144 011203
10464 036146 104001
10465
10466
10467
10468
10469 036150
10470 036150 000004
10471 036152 012700 000535
10472 036156 013701 036204
10473 036162 012702 063312
10474 036166 012704 000001
10475 036172 012705 100000
10476 036176 012712 000001
10477 036202 000257
10478
10479 036204 020512
10480
10481 036206 100403
10482 036210 001402
10483 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 @#2$,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] = +1
   CCC ;CLEAR FLAGS
   266 ;N:C = 0110

2$: CMP R5,(R2) ;TEST THE CMP

   BPL 3$ ;N:C = 1001
   BEQ 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 @#2$,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] = +1
   CCC ;CLEAR FLAGS

2$: CMP R5,(R2) ;TEST THE CMP

   BMI 3$ ;N:C = 0010
   BEQ 3$
   BVC 3$
    
```

10484 036214 103001
10485
10486 036216 104001
10487
10488 036220 020412
10489 036222 001402
10490
10491 036224 011203
10492 036226 104001
10493
10494
10495
10496
10497 036230
10498 036230 000004
10499 036232 012700 000536
10500 036236 013701 036262
10501 036242 012704 177777
10502 036246 012705 063332
10503 036252 012703 052525
10504 036256 000257
10505 036260 000267
10506
10507 036262 051503
10508
10509 036264 100003
10510 036266 001402
10511 036270 102401
10512 036272 103401
10513
10514 036274 104002
10515
10516 036276 020403
10517 036300 001401
10518
10519 036302 104002
10520
10521
10522
10523
10524 036304
10525 036304 000004
10526 036306 012700 000537
10527 036312 013701 036332
10528 036316 005004
10529 036320 012705 063322
10530 036324 005003
10531 036326 000257
10532 036330 000270
10533
10534 036332 051503
10535
10536 036334 100403
10537 036336 001002
10538 036340 102401
10539 036342 103001

BCC 4\$
3\$: ERROR 1 ;CMP FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST536 ;: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

TST536:
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 ;[DEST] = 52525
CCC ;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 TST537 ;:BR IF YES
5\$: ERROR 2 ;BIS DELIVERED THE WRONG RESULT

:*TEST 537 BIS TEST - SM1,DM0 - N:C = 1000

TST537:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #537,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 ;[DEST] = 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\$

10540
10541 036344 104002
10542
10543 036346 020403
10544 036350 001401
10545
10546 036352 104002
10547
10548
10549
10550
10551 036354
10552 036354 000004
10553 036356 012700 000540
10554 036362 013701 036412
10555 036366 012704 100000
10556 036372 012705 063316
10557 036376 012703 177777
10558 036402 012715 077777
10559 036406 000257
10560 036410 000267
10561
10562 036412 041503
10563
10564 036414 100003
10565 036416 001402
10566 036420 102401
10567 036422 103401
10568
10569 036424 104002
10570
10571 036426 020403
10572 036430 001401
10573
10574 036432 104002
10575
10576
10577
10578
10579 036434
10580 036434 000004
10581 036436 012700 000541
10582 036442 013701 036462
10583 036446 005004
10584 036450 012705 063322
10585 036454 005003
10586 036456 000257
10587 036460 000270
10588
10589 036462 041503
10590
10591 036464 100403
10592 036466 001002
10593 036470 102401
10594 036472 103001
10595

3\$: ERROR 2 ;BIS FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST540 ;;BR IF YES
5\$: ERROR 2 ;BIS DELIVERED THE WRONG RESULT

;*TEST 540 BIC TEST - SM1,DMO - N:C = 0111

TST540:
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 ;[DEST] = 177777
MOV #77777,(R5) ;SRC OPR = 77777
CCC ;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 TST541 ;;BR IF YES
5\$: ERROR 2 ;BIC DELIVERED THE WRONG RESULT

;*TEST 541 BIC TEST - SM1,DMO - N:C = 1000

TST541:
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 ;[DEST] = 000000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

2\$: BIC (R5),R3 ;TEST THE BIC
BMI 3\$;N:C = 0100
BNE 3\$
BVS 3\$
BCC 4\$

10596 036474 104002
10597
10598 036476 020403
10599 036500 001401
10600
10601 036502 104002
10602
10603
10604
10605
10606 036504
10607 036504 000004
10608 036506 012700 000542
10609 036512 013701 036534
10610 036516 012704 100000
10611 036522 012705 063324
10612 036526 010403
10613 036530 000257
10614 036532 000267
10615
10616 036534 031503
10617
10618 036536 100003
10619 036540 001402
10620 036542 102401
10621 036544 103401
10622
10623 036546 104002
10624
10625 036550 020403
10626 036552 001401
10627
10628 036554 104002
10629
10630
10631
10632
10633 036556
10634 036556 000004
10635 036560 012700 000543
10636 036564 013701 036606
10637 036570 012704 052525
10638 036574 012705 063332
10639 036600 010403
10640 036602 000257
10641 036604 000270
10642
10643 036606 031503
10644
10645 036610 100403
10646 036612 001002
10647 036614 102401
10648 036616 103001
10649
10650 036620 104002
10651

3\$: ERROR 2 ;BIC FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TST542 ;:BR IF YES
5\$: ERROR 2 ;BIC DELIVERED THE WRONG RESULT
:*****
:*TEST 542 BIT TEST - SM1,DMO - N:C = 0111
:*****
TST542:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #542,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #100000,R4 ;RESULT S / B = 100000
MOV #DWTA+2,R5 ;SRC ADDR = DWTA+2
MOV R4,R3 ;[DEST] = 100000
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 TST543 ;:BR IF YES
5\$: ERROR 2 ;BIT DELIVERED A RESULT
:*****
:*TEST 543 BIT TEST - SM1,DMO - N:C = 1000
:*****
TST543:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #543,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,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 ;[DEST] = 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

10652 036622 020403
10653 036624 001401
10654
10655 036626 104002
10656
10657
10658
10659 036630
10660 036630 000004
10661 036632 012700 000544
10662 036636 013701 036660
10663 036642 012704 000001
10664 036646 012705 063322
10665 036652 010403
10666 036654 000257
10667 036656 000266
10668
10669 036660 021503
10670
10671 036662 100003
10672 036664 001402
10673 036666 102401
10674 036670 103401
10675
10676 036672 104002
10677
10678 036674 020403
10679 036676 001401
10680
10681 036700 104002
10682
10683
10684
10685
10686 036702
10687 036702 000004
10688 036704 012700 000545
10689 036710 013701 036732
10690 036714 012704 177777
10691 036720 012705 063324
10692 036724 010403
10693 036726 000257
10694 036730 000272
10695
10696 036732 021503
10697
10698 036734 100403
10699 036736 001002
10700 036740 102401
10701 036742 103001
10702
10703 036744 104002
10704
10705 036746 020403
10706 036750 001401
10707

4\$: CMP R4,R3 ;CORRECT RESULT ?
BEQ TS:544 ;:BR IF YES

5\$: ERROR 2 ;BIT DELIVERED A RESULT
:*****
:*TEST 544 CMP TEST - SM1,DMO - N:C = 0110
:*****
TST544:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #544,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #+1,R4 ;RESULT S / B = +1
MOV #DWTA,R5 ;SRC ADDR = DWTA
MOV R4,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 TST545 ;:BR IF YES

5\$: ERROR 2 ;CMP DELIVERED A RESULT
:*****
:*TEST 545 CMP TEST - SM1,DMO - N:C = 1010
:*****
TST545:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #545,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,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 ;[DEST] = 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 TST546 ;:BR IF YES

10708 036752 104002
10709
10710
10711
10712
10713 036754
10714 036754 000004
10715 036756 012700 000546
10716 036762 013701 037010
10717 036766 012704 000001
10718 036772 012705 063316
10719 036776 012703 000001
10720 037002 012715 100000
10721 037006 000257
10722
10723 037010 021503
10724
10725 037012 100403
10726 037014 001402
10727 037016 102001
10728 037020 103001
10729
10730 037022 104002
10731
10732 037024 020403
10733 037026 001401
10734
10735 037030 104002
10736
10737
10738
10739
10740 037032
10741 037032 000004
10742 037034 012700 000547
10743 037040 013701 037070
10744 037044 012702 063312
10745 037050 012704 177777
10746 037054 012705 063332
10747 037060 012712 052525
10748 037064 000257
10749 037066 000267
10750
10751 037070 051512
10752
10753 037072 100003
10754 037074 001402
10755 037076 102401
10756 037100 103401
10757
10758 037102 104001
10759
10760 037104 020412
10761 037106 001402
10762
10763 037110 011203

5\$: ERROR 2 ;CMP DELIVERED A RESULT
:*****
:*TEST 546 CMP TEST - SM1,DM0 - N:C = 0000
:*****
TST546:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #546,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #+1,R4 ;:RESULT S / B = +1
MOV #MBUF1,R5 ;:SRC ADDR = MBUF1
MOV #+1,R3 ;:[DEST] = +1
MOV #100000,(R5) ;:SRC OPR = 100000
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 TST547 ;:BR IF YES
5\$: ERROR 2 ;:CMP DELIVERED A RESULT
:*****
:*TEST 547 BIS SM1,DM1 TEST - N:C = 0111
:*****
TST547:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #547,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 = 177777
MOV #DWTA+10,R5 ;:SOURCE ADDR = DWTA+10
MOV #52525,(R2) ;:[DEST] = 052525
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 TST550 ;:BR IF YES
MOV (R2),R3 ;:GET THE WAS DATA

```

10764 037112 104001
10765
10766
10767
10768
10769 037114
10770 037114 000004
10771 037116 012700 000550
10772 037122 013701 037146
10773 037126 012702 063312
10774 037132 005004
10775 037134 012705 063322
10776 037140 005012
10777 037142 000257
10778 037144 000270
10779
10780 037146 051512
10781
10782 037150 100403
10783 037152 001002
10784 037154 102401
10785 037156 103001
10786
10787 037160 104001
10788
10789 037162 020412
10790 037164 001402
10791
10792 037166 011203
10793 037170 104001
10794
10795
10796
10797
10798 037172
10799 037172 000004
10800 037174 012700 000551
10801 037200 013701 037234
10802 037204 012702 063312
10803 037210 012704 100000
10804 037214 012705 063316
10805 037220 012715 077777
10806 037224 012712 177777
10807 037230 000257
10808 037232 000267
10809
10810 037234 041512
10811
10812 037236 100003
10813 037240 001402
10814 037242 102401
10815 037244 103401
10816
10817 037246 104001
10818
10819 037250 020412
  
```

```

5$: ERROR 1 ;BIS DELIVERED THE WRONG RESULT
;*****
;*TEST 550 BIS SM1,DM1 TEST - N:C = 1000
;*****
TST550:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #550,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,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) ;:[DEST] = 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

MOV (R2),R3 ;GET THE WAS DATA
5$: ERROR 1 ;BIS DELIVERED THE WRONG RESULT
;*****
;*TEST 551 BIC SM1,DM1 TEST - N:C = 0111
;*****
TST551:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #551,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,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 #77777,(R5) ;:[SOURCE] = 77777
MOV #-1,(R2) ;:[DEST] = 177777
CCC ;:CLEAR FLAGS
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 ?
  
```

10820	037252	001402		BEQ	TST552		::BR IF YES
10821							
10822	037254	011203		MOV	(R2),R3		:GET THE WAS DATA
10823	037256	104001		5\$:	ERROR 1		:BIC DELIVERED THE WRONG RESULT
10824							
10825							
10826							
10827							
10828	037260			TST552:			
10829	037260	000004		SCOPE			:CALL THE SCOPE LOOP UTILITY
10830	037262	012700	000552	MOV	#552,R0		::LOAD R0 WITH TEST NUMBER
10831	037266	013701	037314	MOV	@#2\$,R1		:LOAD R1 WITH TEST INSTRUCTION WORD
10832	037272	012702	063312	MOV	#MBUF0,R2		:DEST ADDR = MBUF0
10833	037276	005004		CLR	R4		:RESULT S / B = 000000
10834	037300	012705	063316	MOV	#MBUF1,R5		:SOURCE ADDR = MBUF1
10835	037304	005015		CLR	(R5)		: [SOURCE] = 000000
10836	037306	005012		CLR	(R2)		: [DEST] = 000000
10837	037310	000257		CCC			:CLEAR FLAGS
10838	037312	000270		SEN			:N:C = 1000
10839							
10840	037314	041512		2\$:	BIC (R5),(R2)		:TEST THE BIC
10841							
10842	037316	100403		BMI	3\$:N:C = 0100 ?
10843	037320	001002		BNE	3\$		
10844	037322	102401		BVS	3\$		
10845	037324	103001		BCC	4\$		
10846							
10847	037326	104001		3\$:	ERROR 1		:BIC FAILED TO ALTER CODES PROPERLY
10848							
10849	037330	020412		4\$:	CMP R4,(R2)		:CORRECT RESULT ?
10850	037332	001402		BEQ	TST553		::BR IF YES
10851							
10852	037334	011203		MOV	(R2),R3		:GET THE WAS DATA
10853	037336	104001		5\$:	ERROR 1		:BIC DELIVERED THE WRONG RESULT
10854							
10855							
10856							
10857							
10858	037340			TST553:			
10859	037340	000004		SCOPE			:CALL THE SCOPE LOOP UTILITY
10860	037342	012700	000553	MOV	#553,R0		::LOAD R0 WITH TEST NUMBER
10861	037346	013701	037402	MOV	@#2\$,R1		:LOAD R1 WITH TEST INSTRUCTION WORD
10862	037352	012702	063312	MOV	#MBUF0,R2		:DEST ADDR = MBUF0
10863	037356	012704	125252	MOV	#125252,R4		:RESULT S / B = 125252
10864	037362	012705	063316	MOV	#MBUF1,R5		:SOURCE ADDR = MBUF1
10865	037366	012715	052525	MOV	#52525,(R5)		: [SOURCE] = 052525
10866	037372	012712	125252	MOV	#125252,(R2)		: [DEST] = 125252
10867	037376	000257		CCC			:CLEAR FLAGS
10868	037400	000270		SEN			:N:C = 1000
10869							
10870	037402	031512		2\$:	BIT (R5),(R2)		:TEST THE BIT
10871							
10872	037404	100403		BMI	3\$:N:C = 0100 ?
10873	037406	001002		BNE	3\$		
10874	037410	102401		BVS	3\$		
10875	037412	103001		BCC	4\$		

10876
 10877 037414 104001
 10878
 10879 037416 020412
 10880 037420 001402
 10881
 10882 037422 011203
 10883 037424 104001
 10884
 10885
 10886
 10887
 10888 037426
 10889 037426 000004
 10890 037430 012700 000554
 10891 037434 013701 037502
 10892
 10893 037440 032737 004000 063234
 10894 037446 001401
 10895 037450 000000
 10896 037452 012702 063312
 10897 037456 012704 100000
 10898 037462 012705 063316
 10899 037466 012715 100000
 10900 037472 012712 100000
 10901 037476 000257
 10902 037500 000267
 10903
 10904 037502 031512
 10905
 10906 037504 100003
 10907 037506 001402
 10908 037510 102401
 10909 037512 103401
 10910
 10911 037514 104001
 10912
 10913 037516 020412
 10914 037520 001402
 10915
 10916 037522 011203
 10917 037524 104001
 10918
 10919
 10920
 10921
 10922 037526
 10923 037526 000004
 10924 037530 012700 000555
 10925 037534 013701 037566
 10926 037540 012702 063312
 10927 037544 012704 177777
 10928 037550 012705 063316
 10929 037554 012715 177777
 10930 037560 010412
 10931 037562 000257

```

3$: ERROR 1 ;BIT FAILED TO ALTER CODES PROPERLY
4$: CMP R4,(R2) ;CORRECT RESULT ?
   BEQ TST554 ;;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
*****
TST554:
  
```

```

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #554,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
.SBTTL USER CONTROLLED BREAKPOINT -- BIT11
BIT #BIT11,@#BPTLOC ;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 TST555 ;;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
*****
TST555:
  
```

```

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #555,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 = 177777
MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
MOV #-1,(R5) ;[SOURCE] = 177777
MOV R4,(R2) ;[DEST] = 177777
CCC ;CLEAR FLAGS
  
```

10932 037564 000272
10933
10934 037566 021512
10935
10936 037570 100403
10937 037572 001002
10938 037574 102401
10939 037576 103001
10940
10941 037600 104001
10942
10943 037602 020412
10944 037604 001402
10945
10946 037606 011203
10947 037610 104001
10948
10949
10950
10951
10952 037612
10953 037612 000004
10954 037614 012700 000556
10955 037620 013701 037652
10956 037624 012702 063312
10957 037630 012704 000001
10958 037634 012705 063316
10959 037640 005015
10960 037642 012712 000001
10961 037646 000257
10962 037650 000266
10963
10964 037652 021512
10965
10966 037654 100003
10967 037656 001402
10968 037660 102401
10969 037662 103401
10970
10971 037664 104001
10972
10973 037666 020412
10974 037670 001402
10975
10976 037672 011203
10977 037674 104001
10978
10979
10980
10981
10982 037676
10983 037676 000004
10984 037700 012700 000557
10985 037704 013701 037736
10986 037710 012702 063312
10987 037714 012704 000001

272 ;N:C = 1010
2\$: CMP (R5),(R2) ;TEST THE CMP
BMI 3\$;N:C = 0100 ?
BNE 3\$
BVS 3\$
BCC 4\$
3\$: ERROR 1 ;CMP FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST556 ;;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CMP DELIVERED A RESULT

;*TEST 556 CMP SM1,DM1 TEST - N:C = 0110

TST556:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #556,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 = 000001
MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
CLR (R5) ;[SOURCE] = 000000
MOV #+1,(R2) ;[DEST] = 000001
CCC ;CLEAR FLAGS
266 ;N:C = 0110

2\$: CMP (R5),(R2) ;TEST THE CMP
BPL 3\$;N:C = 1001 ?
BEQ 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 1 ;CMP FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST557 ;;BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CMP DELIVERED A RESULT

;*TEST 557 CMP SM1,DM1 TEST - N:C = 0000

TST557:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #557,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 = 000001

10988 037720 012705 063316
10989 037724 012715 100000
10990 037730 012712 000001
10991 037734 000257
10992
10993 037736 021512
10994
10995 037740 100403
10996 037742 001402
10997 037744 102001
10998 037746 103001
10999
11000 037750 104001
11001
11002 037752 020412
11003 037754 001402
11004
11005 037756 011203
11006 037760 104001
11007
11008
11009
11010
11011 037762
11012 037762 000004
11013 037764 012700 000560
11014 037770 013701 040010
11015 037774 012704 000377
11016 040000 012705 064631
11017 040004 005003
11018 040006 000257
11019
11020 040010 151503
11021
11022 040012 020403
11023 040014 001401
11024
11025 040016 104002
11026
11027
11028
11029
11030 040020
11031 040020 000004
11032 040022 012700 000561
11033 040026 013701 040052
11034 040032 012702 063312
11035 040036 012704 000377
11036 040042 012705 064631
11037 040046 005012
11038 040050 000257
11039
11040 040052 151512
11041
11042 040054 020412
11043 040056 001402

MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
MOV #100000,(R5) ;[SOURCE] = 000000
MOV #+1,(R2) ;[DEST] = 000001
CCC ;CLEAR FLAGS
2\$: CMP (R5),(R2) ;TEST THE CMP
BMI 3\$;N:C = 0010 ?
BEQ 3\$
BVC 3\$
BCC 4\$
3\$: ERROR 1 ;CMP FAILED TO ALTER CODES PROPERLY
4\$: CMP R4,(R2) ;CORRECT RESULT ?
BEQ TST560 ;:BR IF YES
5\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;CMP DELIVERED A RESULT
:*****
:*TEST 560 BISB SM1,DM0 TEST - SOURCE ADDR ODD
:*****
TST560:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #560,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #377,R4 ;RESULT S / B = 377
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
CLR R3 ;[DEST] = 000000
CCC ;SCOPE SYNC
2\$: BISB (R5),R3 ;TEST THE BISB
CMP R4,R3 ;RESULT CORRECT ?
BEQ TST561 ;:BR IF YES
3\$: ERROR 2 ;BISB DELIVERED THE WRONG RESULT
:*****
:*TEST 561 BISB SM1,DM1 TEST - SOURCE ADDR ODD
:*****
TST561:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #561,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,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
CCC ;SCOPE SYNC
2\$: BISB (R5),(R2) ;TEST THE BISB
CMP R4,(R2) ;CORRECT RESULT
BEQ TST562 ;:BR IF YES

```

11044
11045 040060 011203
11046 040062 104001
11047
11048
11049
11050
11051 040064
11052 040064 000004
11053 040066 012700 000562
11054 040072 013701 040120
11055 040076 012702 063312
11056 040102 012704 000377
11057 040106 012705 064631
11058 040112 005012
11059 040114 010203
11060 040116 000257
11061
11062 040120 151523
11063
11064 040122 020412
11065 040124 001402
11066
11067 040126 011203
11068 040130 104001
11069
11070
11071
11072
11073 040132
11074 040132 000004
11075 040134 012700 000563
11076 040140 013701 040170
11077 040144 012702 063312
11078 040150 012704 000377
11079 040154 012705 064631
11080 040160 005012
11081 040162 012703 063306
11082 040166 000257
11083
11084 040170 151533
11085
11086 040172 020412
11087 040174 001402
11088
11089 040176 011203
11090 040200 104001
11091
11092
11093
11094
11095 040202
11096 040202 000004
11097 040204 012700 000564
11098 040210 013701 040240
11099 040214 012702 063312
  
```

```

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

:*****
:*TEST 562 BISB SM1,DM2 TEST - SOURCE ADDR ODD
:*****
TST562:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #562,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
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$: BISB (R5),(R3)+ ;TEST THE BISB

CMP R4,(R2) ;CORRECT RESULT
BEQ TST563 ;:BR IF YES

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

:*****
:*TEST 563 BISB SM1,DM3 TEST - SOURCE ADDR ODD
:*****
TST563:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #563,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
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$: BISB (R5),@(R3)+ ;TEST THE BISB

CMP R4,(R2) ;CORRECT RESULT
BEQ TST564 ;:BR IF YES

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

:*****
:*TEST 564 BISB SM1,DM4 TEST - SOURCE ADDR ODD
:*****
TST564:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #564,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO
  
```

11100 040220 012704 177400
11101 040224 012705 064631
11102 040230 012703 063314
11103 040234 005012
11104 040236 000257
11105
11106 040240 151543
11107
11108 040242 020412
11109 040244 001402
11110
11111 040246 011203
11112 040250 104001
11113
11114
11115
11116
11117 040252
11118 040252 000004
11119 040254 012700 000565
11120 040260 013701 040310
11121 040264 012702 063312
11122 040270 012704 000377
11123 040274 012705 064631
11124 040300 012703 063310
11125 040304 005012
11126 040306 000257
11127
11128 040310 151553
11129
11130 040312 020412
11131 040314 001402
11132
11133 040316 011203
11134 040320 104001
11135
11136
11137
11138
11139 040322
11140 040322 000004
11141 040324 012700 000566
11142 040330 013701 040360
11143 040334 012702 063312
11144 040340 012704 000377
11145 040344 012705 064631
11146 040350 012703 063320
11147 040354 005012
11148 040356 000257
11149
11150 040360 151563 177772
11151
11152 040364 020412
11153 040366 001402
11154
11155 040370 011203

MOV #177400,R4 ;RESULT S / B = 177400
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
MOV #MBUFO+2,R3 ;BASE DEST ADDR = MBUFO+2
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2\$: BISB (R5),-(R3) ;TEST THE BISB
CMP R4,(R2) ;CORRECT RESULT
BEQ TST565 ;:BR IF YES
3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BISB DELIVERED THE WRONG RESULT
:*****
:*TEST 565 BISB SM1,DM5 TEST - SOURCE ADDR ODD
:*****
TST565:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #565,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
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\$: BISB (R5),@(R3) ;TEST THE BISB
CMP R4,(R2) ;CORRECT RESULT
BEQ TST566 ;:BR IF YES
3\$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;BISB DELIVERED THE WRONG RESULT
:*****
:*TEST 566 BISB SM1,DM6 TEST - SOURCE ADDR ODD
:*****
TST566:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #566,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #377,R4 ;RESULT S / B = 377
MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
MOV #MBUFO+6,R3 ;BASE DEST ADDR = MBUFO+6
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2\$: BISB (R5),-6(R3) ;TEST THE BISB
CMP R4,(R2) ;CORRECT RESULT
BEQ TST567 ;:BR IF YES
MOV (R2),R3 ;GET THE WAS DATA

11156 040372 104001
 11157
 11158
 11159
 11160
 11161 040374
 11162 040374 000004
 11163 040376 012700 000567
 11164 040402 013701 040432
 11165 040406 012702 063312
 11166 040412 012704 000377
 11167 040416 012705 064631
 11168 040422 012703 063276
 11169 040426 005012
 11170 040430 000257
 11171
 11172 040432 151573 000610
 11173
 11174 040436 020412
 11175 040440 001402
 11176
 11177 040442 011203
 11178 040444 104001
 11179
 11180
 11181
 11182
 11183 040446
 11184 040446 000004
 11185 040450 012700 000570
 11186 040454 013701 040476
 11187 040460 012702 063312
 11188 040464 012704 000377
 11189 040470 010203
 11190 040472 005012
 11191 040474 000257
 11192
 11193 040476 150423
 11194
 11195 040500 020412
 11196 040502 001402
 11197
 11198 040504 011203
 11199 040506 104001
 11200
 11201
 11202
 11203
 11204 040510
 11205 040510 000004
 11206 040512 012700 000571
 11207 040516 013701 040546
 11208 040522 012702 063312
 11209 040526 012704 177400
 11210 040532 012705 000377
 11211 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 @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
 MOV #377,R4 ;RESULT S / B = 377
 MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+!
 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 SMO,DM2 TEST - DEST ADDR EVEN
 :*****
 TST570:
 SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #570,R0 ;:LOAD R0 WITH TEST NUMBER
 MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
 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 SMO,DM1 TEST - DEST ADDR ODD
 :*****
 TST571:
 SCOPE ;CALL THE SCOPE LOOP UTILITY
 MOV #571,R0 ;:LOAD R0 WITH TEST NUMBER
 MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
 MOV #177400,R4 ;RESULT S / B = 177400
 MOV #377,R5 ;[R5]=SOURCE OPR = 377
 MOV #MBUFO+1,R3 ;ODD DEST ADDR IN R3

11212 040542 005012
11213 040544 000257
11214
11215 040546 150513
11216
11217 040550 020412
11218 040552 001402
11219
11220 040554 011203
11221 040556 104001
11222
11223
11224
11225
11226 040560
11227 040560 000004
11228 040562 012700 000572
11229 040566 013701 040610
11230 040572 012702 063312
11231 040576 012704 000377
11232 040602 010203
11233 040604 005012
11234 040606 000257
11235
11236 040610 150413
11237
11238 040612 020412
11239 040614 001402
11240
11241 040616 011203
11242 040620 104001
11243
11244
11245
11246
11247 040622
11248 040622 000004
11249 040624 012700 000573
11250 040630 013701 040660
11251 040634 012702 063312
11252 040640 012704 177400
11253 040644 012705 064631
11254 040650 012703 063313
11255 040654 005012
11256 040656 000257
11257
11258 040660 151513
11259
11260 040662 020412
11261 040664 001402
11262
11263 040666 011203
11264 040670 104001
11265
11266
11267

```
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2$: B1SB R5,(R3) ;TEST THE B1SB
CMP R4,(R2) ;CORRECT RESULT
BEQ TST572 ;;BR IF YES
3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;B1SB DELIVERED THE WRONG RESULT
:*****
:*TEST 572 B1SB SMO,DM1 TEST - DEST ADDR EVEN
:*****
TST572:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #572,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
MOV R2,R3 ;DEST ADDR IN R3
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
2$: B1SB R4,(R3) ;TEST THE B1SB
CMP R4,(R2) ;CORRECT RESULT
BEQ TST573 ;;BR IF YES
3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;B1SB DELIVERED THE WRONG RESULT
:*****
:*TEST 573 B1SB SM1,DM1 TEST - DEST ADDR ODD
:*****
TST573:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #573,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,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$: B1SB (R5),(R3) ;TEST THE B1SB
CMP R4,(R2) ;CORRECT RESULT
BEQ TST574 ;;BR IF YES
3$: MOV (R2),R3 ;GET THE WAS DATA
ERROR 1 ;B1SB DELIVERED THE WRONG RESULT
:*****
:*TEST 574 JMP MODE 1 TEST, FLAGS = 1111
```

11268
11269 040672
11270 040672 000004
11271 040674 012700 000574
11272 040700 013701 040712
11273 040704 012702 040720
11274 040710 000277
11275
11276 040712 000112
11277
11278 040714 104006
11279 040716 0004G5
11280
11281 040720 103003
11282 040722 102002
11283 040724 001001
11284 040726 100401
11285
11286 040730 104006
11287
11288
11289
11290
11291 040732
11292 040732 000004
11293 040734 012700 000575
11294 040740 013701 040752
11295 040744 012702 040760
11296 040750 000257
11297
11298 040752 000112
11299
11300 040754 104006
11301 040756 000405
11302
11303 040760 103403
11304 040762 102402
11305 040764 001401
11306 040766 100001
11307
11308 040770 104006
11309
11310
11311
11312
11313 040772
11314 040772 000004
11315 040774 012700 000576
11316 041000 013701 041012
11317 041004 012702 041020
11318 041010 000277
11319
11320 041012 000122
11321
11322 041014 104006
11323 041016 000411

```
*****  
TST574:  
      SCOPE                ;CALL THE SCOPE LOOP UTILITY  
      MOV      #574,R0      ;;LOAD R0 WITH TEST NUMBER  
      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
      MOV      #4$,R2      ;R2 CONTAINS JUMP ADDRESS  
      SCC                ;MAKE N:C = 1111  
  
2$:   JMP      (R2)        ;TEST THE JMP - GO TO 4$  
  
3$:   ERROR    6          ;JMP FAILED TO LOAD PC  
      BR      TST575      ;;GO CALL SCOPE  
  
4$:   BCC      5$         ;BR IF JMP CLEARED "C"  
      BVC      5$         ;BR IF JMP CLEARED "V"  
      BNE      5$         ;BR IF JMP CLEARED "Z"  
      BMI      TST575      ;;BR IF "N" STILL SET  
  
5$:   ERROR    6          ;JMP ALTERED CODES - CLEARED ONE
```

```
*****  
;*TEST 575      JMP MODE 1 TEST, FLAGS = 0000  
*****
```

```
TST575:  
      SCOPE                ;CALL THE SCOPE LOOP UTILITY  
      MOV      #575,R0      ;;LOAD R0 WITH TEST NUMBER  
      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
      MOV      #4$,R2      ;R2 CONTAINS JUMP ADDRESS  
      CCC                ;MAKE N:C = 0000  
  
2$:   JMP      (R2)        ;TEST THE JMP - GO TO 4$  
  
3$:   ERROR    6          ;JMP FAILED TO LOAD PC  
      BR      TST576      ;;GO CALL SCOPE  
  
4$:   BCS      5$         ;BR IF JMP SET "C"  
      BVS      5$         ;BR IF JMP SET "V"  
      BEQ      5$         ;BR IF JMP SET "Z"  
      BPL      TST576      ;;BR IF "N" STILL CLEAR  
  
5$:   ERROR    6          ;JMP ALTERED CODES - SET ONE
```

```
*****  
;*TEST 576      JMP MODE 2 TEST; FLAGS = 1111  
*****
```

```
TST576:  
      SCOPE                ;CALL THE SCOPE LOOP UTILITY  
      MOV      #576,R0      ;;LOAD R0 WITH TEST NUMBER  
      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
      MOV      #4$,R2      ;R2 CONTAINS JUMP ADDRESS  
      SCC                ;SET N:C = 1111  
  
2$:   JMP      (R2)+      ;TEST THE JMP - GO TO 4$  
  
3$:   ERROR    6          ;JMP FAILED TO LOAD PC  
      BR      TST577      ;;GO TO SCOPE EXIT
```

```

11324
11325 041020 103003
11326 041022 102002
11327 041024 001001
11328 041026 100401
11329
11330 041030 104006
11331
11332 041032 022702 041022
11333 041036 001401
11334
11335 041040 104006
11336
11337
11338
11339
11340 041042
11341 041042 000004
11342 041044 012700 000577
11343 041050 013701 041062
11344 041054 012702 041070
11345 041060 000257
11346
11347 041062 000122
11348
11349 041064 104006
11350 041066 000405
11351
11352 041070 103403
11353 041072 102402
11354 041074 001401
11355 041076 100001
11356
11357 041100 104006
11358
11359
11360
11361
11362 041102
11363 041102 000004
11364 041104 012700 000600
11365 041110 013701 041122
11366 041114 012702 041154
11367 041120 000277
11368
11369 041122 000132
11370
11371 041124 104006
11372 041126 000414
11373
11374 041130 103003
11375 041132 102002
11376 041134 001001
11377 041136 100401
11378
11379 041140 104006

4$: BCC 5$ :BR IF JMP CLEARED "C"
    BVC 5$ :BR IF JMP CLEARED "V"
    BNE 5$ :BR IF JMP CLEARED "Z"
    BMI 6$ :BR IF "N" STILL SET

5$: ERROR 6 :JMP ALTERED CODES - CLEARED

6$: CMP #4$+2,R2 :DID R2 GET AUTO-INCREMENTED?
    BEQ TST577 :;BR IF YES

7$: ERROR 6 :JMP FAILED TO UPDATE REGISTER (R2)

:*****
:*TEST 577 JMP MODE 2 TEST; FLAGS = 0000
:*****
TST577:
    SCOPE :CALL THE SCOPE LOOP UTILITY
    MOV #577,R0 :;LOAD R0 WITH TEST NUMBER
    MOV @#2$,R1 :LOAD R1 WITH TEST INSTRUCTION WORD
    MOV #4$,R2 :R2 CONTAINS JUMP ADDRESS
    CCC :MAKE N:C = 0000

2$: JMP (R2)+ :TEST THE JMP - GO TO 4$

3$: ERROR 6 :JMP FAILED TO LOAD PC
    BR TST600 :;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 TST600 :;BR IF "N" IS CLEAR

5$: ERROR 6 :JMP ALTERED CODES - SET

:*****
:*TEST 600 JMP TEST MODE 3; FLAGS = 1111
:*****
TST600:
    SCOPE :CALL THE SCOPE LOOP UTILITY
    MOV #600,R0 :;LOAD R0 WITH TEST NUMBER
    MOV @#2$,R1 :LOAD R1 WITH TEST INSTRUCTION WORD
    MOV #7$,R2 :R2 CONTAINS ADDRESS OF JUMP ADDRESS
    SCC :SET N:C = 1111

2$: JMP @ (R2)+ :TEST THE JMP - GO TO 4$

3$: ERROR 6 :JMP FAILED TO LOAD PC
    BR TST601 :;GO TO SCOPE EXIT

4$: BCC 5$ :BR IF JMP CLEARED "C"
    BVC 5$ :BR IF JMP CLEARED "V"
    BNE 5$ :BR IF JMP CLEARED "Z"
    BMI 6$ :BR IF "N" STILL SET

5$: ERROR 6 :JMP ALTERED CODES - CLEAR
  
```

CQKDA-D KD11-K BASIC LOGIC TESTS
CQKDA.D.P11 11-SEP-79 13:31

MACY11 30A(1052) 11-SEP-79^{E 1} 13:53 PAGE 212
T600 JMP TEST MODE 3; FLAGS = 1111

SEQ 0211

11380
11381 041142 022702 041156
11382 041146 001404
11383
11384 041150 104006

6S: CMP #7S+2,R2 ;DID JMP UPDATE R2?
 BEQ TST601 ;:BR IF YES

 ERROR 6 ;JMP FAILED TO UPDATE REGISTER

```

11385 041152 000402
11386 041154 041130
11387 041156 104006
11388
11389
11390
11391
11392
11393
11394 041160
11395 041160 000004
11396 041162 012700 000601
11397 041166 013701 041200
11398 041172 012702 041222
11399 041176 000257
11400
11401 041200 000132
11402
11403 041202 104006
11404 041204 000410
11405
11406 041206 103403
11407 041210 102402
11408 041212 001401
11409 041214 100004
11410
11411 041216 104006
11412 041220 000402
11413
11414 041222 041206
11415 041224 104006
11416
11417
11418
11419
11420 041226
11421 041226 000004
11422 041230 012700 000602
11423 041234 013701 041246
11424 041240 012702 041256
11425 041244 000277
11426
11427 041246 000142
11428
11429 041250 104006
11430 041252 000414
11431
11432 041254 000402
11433 041256 104006
11434 041260 000411
11435
11436 041262 103003
11437 041264 102002
11438 041266 001001
11439 041270 100401
11440
  
```

```

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
;:*****
TST601:
SCOPE ;:CALL THE SCOPE LOOP UTILITY
MOV #601,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #6$,R2 ;:R2 CONTAINS ADDRESS OF JUMP ADDRESS
CCC ;:MAKE N:C = 0000

2$: JMP @ (R2)+ ;:TEST THE JMP - GO TO 4$

3$: ERROR 6 ;:JMP FAILED TO LOAD THE PC
BR TST602 ;: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 TST602 ;:BR IF "N" STILL CLEAR

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
;:*****
TST602:
SCOPE ;:CALL THE SCOPE LOOP UTILITY
MOV #602,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #5$,R2 ;[R2] = JMP ADDRESS PLUS 2
SCC ;:MAKE N:C = 1111

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

5$: BR 4$ ;:GO TEST FLAGS - JMP LOADED PC OK
ERROR 6 ;:JMP FAILED TO AUTO-DECREMENT R2
BR TST603 ;:GO TO SCOPE EXIT

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
  
```

11441 041272 104006
 11442
 11443 041274 022702 041254
 11444 041300 001401
 11445
 11446 041302 104006
 11447
 11448
 11449
 11450
 11451 041304
 11452 041304 000004
 11453 041306 012700 000603
 11454 041312 013701 041324
 11455 041316 012702 041334
 11456 041322 000257
 11457
 11458 041324 000142
 11459
 11460 041326 104006
 11461 041330 000405
 11462
 11463 041332 103403
 11464 041334 102402
 11465 041336 001401
 11466 041340 100001
 11467
 11468 041342 104006
 11469
 11470
 11471
 11472
 11473 041344
 11474 041344 000004
 11475 041346 012700 000604
 11476 041352 013701 041364
 11477 041356 012702 041420
 11478 041362 000277
 11479
 11480 041364 000152
 11481
 11482 041366 104006
 11483 041370 000414
 11484
 11485 041372 103003
 11486 041374 102002
 11487 041376 001001
 11488 041400 100401
 11489
 11490 041402 104006
 11491
 11492 041404 022702 041416
 11493 041410 001404
 11494
 11495 041412 104006
 11496 041414 000402

7\$: ERROR 6 ;JMP ALTERED FLAGS
 6\$: CMP #5\$-2,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 @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #4\$+2,R2 ;[R2] = JUMP ADDRESS PLUS 2
 CCC ;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 @#2\$,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
 BR TST605 ;:GO TO SCOPE OXIT
 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

11497 041416 041372
 11498 041420 104006
 11499
 11500
 11501
 11502
 11503 041422
 11504 041422 000004
 11505 041424 012700 000605
 11506 041430 013701 041442
 11507 041434 012702 041466
 11508 041440 000257
 11509
 11510 041442 000152
 11511
 11512 041444 104006
 11513 041446 000410
 11514
 11515 041450 103403
 11516 041452 102402
 11517 041454 001401
 11518 041456 100004
 11519
 11520 041460 104006
 11521 041462 000402
 11522
 11523 041464 041450
 11524 041466 104006
 11525
 11526
 11527
 11528
 11529 041470
 11530 041470 000004
 11531 041472 012700 000606
 11532 041476 013701 041510
 11533 041502 012702 041534
 11534 041506 000277
 11535
 11536 041510 000162 177764
 11537
 11538 041514 104006
 11539 041516 000407
 11540
 11541 041520 103003
 11542 041522 102002
 11543 041524 001001
 11544 041526 100403
 11545
 11546 041530 104006
 11547 041532 000401
 11548
 11549 041534 104006
 11550
 11551
 11552

```

4$ ;THIS LOCATION CONTAINS JMP ADDRESS
JMP5: ERROR 6 ;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 @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #JMP5A,R2 ;:[R2] = ADDR +2 OF JUMP ADDRESS
CCC ;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

4$ ;THIS LOCATION CONTAINS JUMP ADDRESS
JMP5A: ERROR 6 ;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 @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #7$,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'
BVC 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]

*****

```


11553
 11554
 11555 041536
 11556 041536 000004
 11557 041540 012700 000607
 11558 041544 013701 041556
 11559 041550 012702 041602
 11560 041554 000257
 11561
 11562 041556 000162 177764
 11563
 11564 041562 104006
 11565 041564 000407
 11566
 11567 041566 103403
 11568 041570 102402
 11569 041572 001401
 11570 041574 100003
 11571
 11572 041576 104006
 11573 041600 000401
 11574
 11575 041602 104006
 11576
 11577
 11578
 11579
 11580
 11581 041604
 11582 041604 000004
 11583 041606 012700 000610
 11584 041612 013701 041624
 11585 041616 012702 041634
 11586 041622 000277
 11587
 11588 041624 000172 000020
 11589
 11590 041630 104006
 11591 041632 000412
 11592
 11593 041634 104006
 11594 041636 000410
 11595
 11596 041640 103003
 11597 041642 102002
 11598 041644 001001
 11599 041646 100404
 11600
 11601 041650 104006
 11602 041652 000402
 11603
 11604 041654 041640
 11605
 11606 041656 104006
 11607
 11608

```

;*TEST 607      JMP TEST MODE 6; FLAGS = 0000
:*****
TST607:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #607,R0      ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #7$,R2      ;[R2] = BASE ADDRESS FOR JUMP
      CCC                ;MAKE N:C = 0000

2$:   JMP      4$-7$(R2)    ;TEST THE JMP - GO TO 4$

3$:   ERROR   6            ;JMP FAILED TO LOAD PC
      BR      TST610      ;;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     TST610      ;;BR IF "N" STILL CLEAR

5$:   ERROR   6            ;JMP ALTERED CODES
      BR      TST610      ;;GO TO SCOPE EXIT

7$:   ERROR   6            ;JMP EXECUTED LIKE A MODE 1 OR 2, OR
                        ;FAILED TO INDEX [R2]

:*****
;*TEST 610      JMP TEST MODE 7; FLAGS = 1111
:*****
TST610:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #610,R0      ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #5$,R2      ;[R2] = BASE ADDRESS
      SCC                ;MAKE N:C = 1111

2$:   JMP      @8$-5$(R2)  ;TEST THE JMP - GO TO 4$

3$:   ERROR   6            ;JMP FAILED TO LOAD PC
      BR      TST611      ;;GO TO SCOPE EXIT

5$:   ERROR   6            ;JMP FAILED TO INDEX OR ACTED LIKE MODE 1 OR 2
      BR      TST611      ;;GO TO SCOPE EXIT

4$:   BCC     7$           ;BR IF JMP CLEARED "C"
      BVC     7$           ;BR IF JMP CLEARED "V"
      BNE     7$           ;BR IF JMP CLEARED "Z"
      BMI     TST611      ;;BR IF "N" STILL SET

7$:   ERROR   6            ;JMP ALTERED CODES - CLEARED
      BR      TST611      ;;GO TO SCOPE EXIT

8$:   4$                ;THIS LOCATION CONTAINS JMP ADDRESS

      ERROR   6            ;JMP EXECUTED LIKE MODE 6
:*****

```

11609
 11610
 11611 041660
 11612 041660 000004
 11613 041662 012700 000611
 11614 041666 013701 041700
 11615 041672 012702 041710
 11616 041676 000257
 11617
 11618 041700 000172 000020
 11619
 11620 041704 104006
 11621 041706 000412
 11622
 11623 041710 104006
 11624 041712 000410
 11625
 11626 041714 103403
 11627 041716 102402
 11628 041720 001401
 11629 041722 100004
 11630
 11631 041724 104006
 11632 041726 000402
 11633
 11634 041730 041714
 11635
 11636 041732 104006
 11637
 11638
 11639
 11640
 11641 041734
 11642 041734 000004
 11643 041736 012700 000612
 11644 041742 013701 041764
 11645 041746 010605
 11646 041750 010737 001010
 11647 041754 010506
 11648 041756 012702 041770
 11649 041762 000257
 11650
 11651 041764 004412
 11652
 11653 041766 104006
 11654
 11655 041770 005726
 11656 041772 020605
 11657 041774 001406
 11658
 11659 041776 005746
 11660 042000 010603
 11661 042002 010504
 11662 042004 005744
 11663 042006 104003
 11664

```

;*TEST 611      JMP TEST MODE 7; FLAGS = 0000
:*****
TST611:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #611,R0     ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #5$,R2     ;[R2] = BASE ADDRESS
      CCC                ;MAKE N:C = 0000

2$:   JMP      @8$-5$(R2)  ;TEST THE JMP - GO TO 4$

3$:   ERROR   6           ;JMP FAILED TO LOAD PC
      BR      TST612     ;;GO TO SCOPE EXIT

5$:   ERROR   6           ;JMP FAILED TO INDEX
      BR      TST612     ;;GO TO SCOPE EXIT

4$:   BCS     7$          ;BR IF JMP SET "C"
      BVS     7$          ;BR IF JMP SET "V"
      BEQ     7$          ;BR IF JMP SET "Z"
      BPL     TST612     ;;BR IF "N" STILL CLEAR

7$:   ERROR   6           ;JMP ALTERED CODES - SET
      BR      TST612     ;;GO TO SCOPE EXIT

8$:   4$                ;THIS LOCATION CONTAINS JUMP ADDRESS

      ERROR   6           ;JMP EXECUTED LIKE A MODE 6
:*****
;*TEST 612      JSR MODE 1 TEST - LOAD PC / PUSH SP
:*****
TST612:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #612,R0     ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      SP,R5       ;SAVE THE SP
      MOV      PC,@#5$LPERR ;SET ERROR LOOP ADDRESS
1$:   MOV      R5,SP       ;RESTORE SP FOR ERROR LOOPING
      MOV      #4$,R2     ;DEST ADDR = 4$
      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)+       ;POP THE SP
      CMP     SP,R5       ;DID JSR PUSH THE SP ?
      BEQ     TST613     ;;BR IF YES

      TST     -(SP)       ;RESTORE ERROR SP
      MOV     SP,R3       ;[R3]= WAS SP
      MOV     R5,R4
      TST     -(R4)       ;[R4]= S/B SP
5$:   ERROR   3           ;JSR FAILED TO PUSH THE SP

```

K 1

```

11665 042010 010506          MOV     R5,SP          ;RESTORE SP IN CASE OF ERROR
11666                          ;:*****
11667                          ;*TEST 613          JSR MODE 1 TEST - CHECK RN AND OLD PC
11668                          ;:*****
11669 042012          TST613:
11670 042012 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
11671 042014 012700 000613  MOV     #613,R0      ;:LOAD R0 WITH TEST NUMBER
11672 042020 013701 042052  MOV     @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
11673 042024 010605          MOV     SP,R5        ;SAVE THE SP
11674 042026 010737 001010  MOV     PC,@#5LPERR ;SET ERROR LOOP ADDRESS
11675 042032 010506          1$:  MOV     R5,SP      ;RESTORE SP FOR ERROR LOOPING
11676 042034 012702 042056  MOV     #4$,R2      ;DEST ADDR = 4$
11677 042040 005066 177776  CLR     -2(SP)      ;INIT STACK LOC TO GET [R4]
11678 042044 012704 125252  MOV     #125252,R4  ;INIT RN = 125252
11679 042050 000257          CCC              ;SCOPE SYNC
11680
11681 042052 004412          2$:  JSR     R4,(R2)   ;TEST THE JSR - GO TO 4$
11682
11683 042054 104006          3$:  ERROR  6         ;JSR FAILED TO LOAD THE PC
11684
11685 042056 022726 125252  4$:  CMP     #125252,(SP)+ ;DID JSR SAVE REG ON STACK
11686 042062 001401          BEQ     8$         ;BR IF IT DID
11687
11688 042064 104005          5$:  ERROR  5         ;JSR FAILED TO SAVE REG ON STACK
11689
11690 042066 022704 042054  8$:  CMP     #3$,R4     ;DID OLD PC GET SAVED ?
11691 042072 001401          BEQ     6$         ;BR IF YES
11692
11693 042074 104005          7$:  ERROR  5         ;JSR FAILED TO SAVE TH OLD PC
11694
11695 042076 010506          6$:  MOV     R5,SP      ;RESTORE SP IN CASE ERROR SCREWED IT UP
11696
11697                          ;:*****
11698                          ;*TEST 614          JSR MODE 1 TEST - N:C = 0000
11699                          ;:*****
11700 042100          TST614:
11701 042100 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
11702 042102 012700 000614  MOV     #614,R0      ;:LOAD R0 WITH TEST NUMBER
11703 042106 013701 042142  MOV     @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
11704                          .SBTTL USER CONTROLLED BREAKPOINT -- BIT12
11705 042112 032737 010000 063234  BIT     #BIT12,@#BPTLOC ;BREAKPOINT HALT SET ??
11706 042120 001401          BEQ     .+4        ;BR IF NOT
11707 042122 000000          HALT          ;BREAK-DEPRESS CONTINUE TO CONTINUE
11708 042124 010605          MOV     SP,R5        ;SAVE THE SP
11709 042126 010737 001010  MOV     PC,@#5LPERR ;SET ERROR LOOP ADDRESS
11710 042132 010506          1$:  MOV     R5,SP      ;RESTORE SP FOR ERROR LOOPING
11711 042134 012702 042146  MOV     #4$,R2      ;DEST ADDR = 4$
11712 042140 000257          CCC              ;N:C = 0000
11713
11714 042142 004412          2$:  JSR     R4,(R2)   ;TEST THE JSR - GO TO 4$
11715
11716 042144 104006          3$:  ERROR  6         ;JSR FAILED TO LOAD THE PC
11717
11718 042146 100403          4$:  BMI     5$         ;N:C = 0000 ?
11719 042150 001402          BEQ     5$
11720 042152 102401          BVS     5$

```

```

11721 042154 103001
11722
11723 042156 104005
11724
11725 042160 010506
11726
11727
11728
11729 042162
11730 042162 000004
11731 042164 012700 000615
11732 042170 013701 042212
11733 042174 010605
11734 042176 010737 001010
11735 042202 010506
11736 042204 012702 042216
11737 042210 000277
11738
11739 042212 004412
11740
11741 042214 104006
11742
11743 042216 100003
11744 042220 001002
11745 042222 102001
11746 042224 103401
11747 042226 104005
11748
11749 042230 010506
11750
11751
11752
11753
11754 042232
11755 042232 000004
11756 042234 012700 000616
11757 042240 013701 042262
11758 042244 010605
11759 042246 010737 001010
11760 042252 010506
11761 042254 012702 042266
11762 042260 000257
11763
11764 042262 004422
11765
11766 042264 104006
11767
11768 042266 005726
11769 042270 020605
11770 042272 001406
11771
11772 042274 005746
11773 042276 010603
11774 042300 010504
11775 042302 005744
11776 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
          MOV      #615,R0    ;CALL THE SCOPE LOOP UTILITY
          MOV      @#2$,R1   ;LOAD R0 WITH TEST NUMBER
          MOV      SP,R5     ;LOAD R1 WITH TEST INSTRUCTION WORD
          MOV      PC,@#$LPERR ;SAVE THE SP
1$:      MOV      R5,SP      ;SET ERROR LOOP ADDRESS
          MOV      #4$,R2    ;RESTORE SP FOR ERROR LOOPING
          SCC
          ;DEST ADDR = 4$
          ;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$
          BCS      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
          MOV      #616,R0    ;CALL THE SCOPE LOOP UTILITY
          MOV      @#2$,R1   ;LOAD R0 WITH TEST NUMBER
          MOV      SP,R5     ;LOAD R1 WITH TEST INSTRUCTION WORD
          MOV      PC,@#$LPERR ;SAVE THE SP
1$:      MOV      R5,SP      ;SET ERROR LOOP ADDRESS
          MOV      #4$,R2    ;RESET SP FOR ERROR LOOPS
          CCC               ;DEST ADDR = 4$
          ;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      R5,R4
          TST      -(R4)    ;S/B SP
5$:      ERROR    3          ;JSR FAILED TO PUSH SP
  
```

```

11777
11778 042306 010506          MOV    R5,SP          ;RESTORE SP JUST IN CASE
11779
11780
11781  ;*****
11782  ;*TEST 617      JSR MODE 3 TEST
11783  ;*****
11783  042310          TST617:
11784  042310 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
11785  042312 012700 000617  MOV    #617,R0      ;:LOAD R0 WITH TEST NUMBER
11786  042316 013701 042340  MOV    @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
11787  042322 010605          MOV    SP,R5        ;SAVE THE SP
11788  042324 010737 001010  MOV    PC,@#$$LPEXR ;SET ERROR LOOP ADDRESS
11789  042330 010506          1$:  MOV    R5,SP        ;RESET SP FOR ERROR LOOPS
11790  042332 012702 042366  MOV    #7$,R2       ;DEST ADDR = [7$]
11791  042336 000257          CCC              ;SCOPE SYNC
11792
11793  042340 004432          2$:  JSR    R4,@(R2)+  ;TEST THE JSR - GO TO 4$ VIA 7$
11794
11795  042342 104006          3$:  ERROR  6          ;JSR FAILED TO LOAD THE PC
11796
11797  042344 005726          4$:  TST    (SP)+      ;RESET SP
11798  042346 020605          CMP    SP,R5        ;DID JSR PUSH STACK ?
11799  042350 001411          BEQ    TST620       ;:BR IF YES
11800
11801  042352 005746          TST    -(SP)        ;RESET SP TO ERROR VALUE
11802  042354 010603          MOV    SP,R3        ;WAS SP
11803  042356 010504          MOV    R5,R4
11804  042360 005744          TST    -(R4)        ;S/B SP
11805  042362 104003          5$:  ERROR  3          ;JSR FAILED
11806  042364 000402          BR     6$           ;GO EXIT
11807
11808  042366 042344          7$:  4$          ;CONTAINS JUMP ADDR
11809  042370 104006          ERROR  6          ;JSR EXECUTED LIKE A MODE 1 OR 2
11810
11811  042372 010506          6$:  MOV    R5,SP        ;RESTORE SP JUST IN CASE
11812
11813  ;*****
11814  ;*TEST 620      JSR MODE 4 TEST
11815  ;*****
11816  042374          TST620:
11817  042374 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
11818  042376 012700 000620  MOV    #620,R0      ;:LOAD R0 WITH TEST NUMBER
11819  042402 013701 042424  MOV    @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
11820  042406 010605          MOV    SP,R5        ;SAVE THE SP
11821  042410 010737 001010  MOV    PC,@#$$LPERR ;SET ERROR LOOP ADDRESS
11822  042414 010506          1$:  MOV    R5,SP        ;RESET SP FOR ERROR LOOPS
11823  042416 012702 042432  MOV    #5$,R2       ;DEST ADDR = 4$+2
11824  042422 000257          CCC              ;SCOPE SYNC
11825
11826  042424 004442          2$:  JSR    R4,-(R2)  ;TEST THE JSR - GO TO 4$
11827
11828  042426 104006          3$:  ERROR  6          ;JSR FAILED TO LOAD THE PC
11829
11830  042430 000401          4$:  BR     6$          ;JUMPED OK - GO CHECK SP
11831  042432 104005          5$:  ERROR  5          ;JSR FAILED TO DECREMENT DEST REG
11832

```

11833 042434 005726
 11834 042436 020605
 11835 042440 001406
 11836
 11837 042442 005746
 11838 042444 010603
 11839 042446 010504
 11840 042450 005744
 11841 042452 104003
 11842
 11843 042454 010506
 11844
 11845
 11846
 11847
 11848 042456
 11849 042456 000004
 11850 042460 012700 000621
 11851 042464 013701 042506
 11852 042470 010605
 11853 042472 010737 001010
 11854 042476 010506
 11855 042500 012702 042536
 11856 042504 000257
 11857
 11858 042506 004452
 11859
 11860 042510 104006
 11861
 11862 042512 005726
 11863 042514 020605
 11864 042516 001411
 11865
 11866 042520 005746
 11867 042522 010603
 11868 042524 010504
 11869 042526 005744
 11870 042530 104003
 11871 042532 000402
 11872
 11873 042534 042512
 11874 042536 104005
 11875
 11876 042540 010506
 11877
 11878
 11879
 11880
 11881 042542
 11882 042542 000004
 11883 042544 012700 000622
 11884 042550 013701 042572
 11885 042554 010605
 11886 042556 010737 001010
 11887 042562 010506
 11888 042564 012702 042576

```

6$:  TST      (SP)+      ;RESET SP
      CMP      SP,R5      ;DID JSR PUSH STACK ?
      BEQ      TST621     ;;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      ;CALL THE SCOPE LOOP UTILITY
      MOV      #621,R0    ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      SP,R5      ;SAVE THE SP
      MOV      PC,@#$LPERR ;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      TST622     ;;BR IF YES

      TST      -(SP)      ;RESET SP TO ERROR VALUE
      MOV      SP,R3      ;WAS SP
      MOV      R5,R4
      TST      -(R4)      ;S/B SP
5$:  ERROR    3           ;JSR FAILED TO PUSH SP
      BR       6$         ;GO EXIT

      4$           ;CONTAINS JUMP ADDRESS
7$:  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      ;CALL THE SCOPE LOOP UTILITY
      MOV      #622,R0    ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1    ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      SP,R5      ;SAVE THE SP
      MOV      PC,@#$LPERR ;SET ERROR LOOP ADDRESS
1$:  MOV      R5,SP       ;RESET SP FOR ERROR LOOPS
      MOV      #3$,R2     ;[R2] = BASE DEST ADDR
  
```

```
11889 042570 000257          CCC          :SCOPE SYNC
11890
11891 042572 004462 000002    2$: JSR      R4,4$-3$(R2)  :TEST THE JSR - GO TO 4$
11892
11893 042576 104006          3$: ERROR    6          :JSR FAILED TO LOAD THE PC OR INDEX FAILED
11894
11895 042600 005726          4$: TST      (SP)+       :RESET SP
11896 042602 020605          CMP      SP,R5         :DID JSR PUSH STACK ?
11897 042604 001406          BEQ      TST623        :;BR IF YES
11898
11899 042606 005746          TST      -(SP)        :RESET SP TO ERROR VALUE
11900 042610 010603          MOV      SP,R3         :WAS SP
11901 042612 010504          MOV      R5,R4
11902 042614 005744          TST      -(R4)        :S/B SP
11903 042616 104003          5$: ERROR    3          :JSR FAILED TO PUSH STACK
11904 042620 010506          MOV      R5,SP        :RESET SP JUST IN CASE
```

```
::*****
:*TEST 623      JSR MODE 7 TEST
:*****
```

```
TST623:
          SCOPE          :CALL THE SCOPE LOOP UTILITY
11910 042622 000004          MOV      #623,R0       :;LOAD R0 WITH TEST NUMBER
11911 042624 012700 000623    MOV      @#2$,R1       :LOAD R1 WITH TEST INSTRUCTION WORD
11912 042630 013701 042652    MOV      SP,R5         :SAVE THE SP
11913 042634 010605          MOV      PC,@#5LPERR   :SET ERROR LOOP ADDRESS
11914 042636 010737 001010    MOV      R5,SP        :RESET SP FOR ERROR LOOPS
11915 042642 010506          MOV      #3$,R2        :BASE DEST ADDR = 3$
11916 042644 012702 042656    CCC                   :SCOPE SYNC
11917 042650 000257
11918
11919 042652 004472 000024    2$: JSR      R4,@7$-3$(R2) :TEST THE JSR - GO TO 4$ VIA 7$
11920
11921 042656 104006          3$: ERROR    6          :JSR FAILED TO LOAD THE PC
11922                                :OR THE INDEX FAILED
11923
11924 042660 005726          4$: TST      (SP)+       :RESET SP
11925 042662 020605          CMP      SP,R5         :DID JSR PUSH STACK ?
11926 042664 001411          BEQ      TST624        :;BR IF YES
11927
11928 042666 005746          TST      -(SP)        :RESET SP TO ERROR VALUE
11929 042670 010603          MOV      SP,R3         :WAS SP
11930 042672 010504          MOV      R5,R4
11931 042674 005744          TST      -(R4)        :S/B SP
11932 042676 104003          5$: ERROR    3          :JSR FAILED TO PUSH STACK
11933 042700 000402          BR       6$           :SKIP TO EXIT
11934
11935 042702 042660          7$: 4$           :CONTAINS JUMP ADDR
11936 042704 104005          ERROR    5           :JSR WORKED LIKE A MODE 1 OR 2
11937
11938 042706 010506          6$: MOV      R5,SP        :RESTORE SP JUST IN CASE
11939
```

```
::*****
:*TEST 624      SOB TEST, [R] = 1, NO BRANCH
:*****
```

```
TST624:
          SCOPE          :CALL THE SCOPE LOOP UTILITY
11943 042710
11944 042710 000004
```

11945 042712 012700 000624
 11946 042716 013701 042736
 11947 042722 012702 000001
 11948 042726 000402
 11949
 11950 042730 104006
 11951 042732 000402
 11952
 11953 042734 000257
 11954 042736 077204
 11955
 11956
 11957
 11958
 11959 042740
 11960 042740 000004
 11961 042742 012700 000625
 11962 042746 013701 043000
 11963 042752 012702 000005
 11964 042756 012705 177773
 11965 042762 000405
 11966
 11967 042764 000474
 11968 042766 000240
 11969 042770 000240
 11970
 11971 042772 005205
 11972 042774 001406
 11973
 11974 042776 000257
 11975 043000 077204
 11976 043002 005702
 11977 043004 001403
 11978
 11979 043006 104006
 11980 043010 000401
 11981 043012 104006
 11982
 11983
 11984
 11985
 11986 043014
 11987 043014 000004
 11988 043016 012700 000626
 11989 043022 013701 043034
 11990 043026 012702 000001
 11991 043032 000277
 11992
 11993 043034 077202
 11994
 11995 043036 103003
 11996 043040 102002
 11997 043042 001001
 11998 043044 100401
 11999
 12000 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 TST625      ;;GO TO SCOPE CALL

CCC            ;;SYNC INSTR.
SOB R2,3%     ;;TEST THE SOB

;*****
;*TEST 625     SOB TEST, [R] = 5, BRANCH 4 TIMES
;*****
TST625:
SCOPE         ;;CALL THE SCOPE LOOP UTILITY
MOV #625,R0   ;;LOAD R0 WITH TEST NUMBER
MOV @#SOB2,R1 ;;GET COPY OF 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
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 TST626   ;;BR IF IT DOES

ERROR 6      ;;SOB COUNTER NOT ZERO
BR TST626   ;;GO TO SCOPE CALL
SOBERR: ERROR 6 ;;SOB MADE TOO MANY BRANCHES

;*****
;*TEST 626     SOB TEST, [R] = 1, FLAGS = 1111
;*****
TST626:
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
BNE 3%       ;;BR IF Z = 0
BMI TST627   ;;BR IF N = 1

3$: ERROR 6  ;;SOB ALTERED CODES - CLEARED ONE
  
```


12001
12002
12003
12004
12005 043050
12006 043050 000004
12007 043052 012700 000627
12008 043056 013701 043070
12009 043062 012702 000001
12010 043066 000257
12011
12012 043070 077202
12013
12014 043072 103403
12015 043074 102402
12016 043076 001401
12017 043100 100001
12018
12019 043102 104006
12020
12021
12022
12023
12024 043104
12025 043104 000004
12026 043106 012700 000630
12027 043112 013701 043124
12028 043116 012702 000005
12029 043122 000277
12030
12031 043124 077201
12032
12033 043126 103003
12034 043130 102002
12035 043132 001001
12036 043134 100401
12037
12038 043136 104006
12039
12040
12041
12042
12043 043140
12044 043140 000004
12045 043142 012700 000631
12046 043146 013701 043160
12047 043152 012702 000005
12048 043156 000257
12049
12050 043160 077277
12051
12052 043162 103403
12053 043164 102402
12054 043166 001401
12055 043170 100001
12056

```
*****
*TEST 627 SOB TEST, [R] = 1, FLAGS = 0000
*****
TST627:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #627,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,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 TST630 ;: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 @#2$,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
BVC 3$ ;BR IF V = 0
BNE 3$ ;BR IF Z = 0
BMI TST631 ;: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 @#SOB4,R1 ;:GET COPY OF TEST INSTRUCTION WORD
MOV #5,R2 ;:SET SOB COUNTER = 5
SOB3: CCC ;MAKE N:C = 0000

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 TST632 ;:BR IF N = 0
```

12057 043172 104006
 12058
 12059
 12060
 12061
 12062 043174
 12063 043174 000004
 12064 043176 012700 000632
 12065 043202 013701 043234
 12066 043206 010605
 12067 043210 010737 001010
 12068 043214 012704 177777
 12069 043220 010506
 12070 043222 012703 043242
 12071 043226 012746 177777
 12072 043232 000257
 12073
 12074 043234 000203
 12075
 12076 043236 104005
 12077 043240 000415
 12078
 12079 043242 100403
 12080 043244 001402
 12081 043246 102401
 12082 043250 103001
 12083
 12084 043252 104005
 12085
 12086 043254 020403
 12087 043256 001401
 12088
 12089 043260 104002
 12090
 12091 043262 020506
 12092 043264 001404
 12093
 12094 043266 010504
 12095 043270 010603
 12096 043272 104003
 12097
 12098 043274 010506
 12099
 12100
 12101
 12102
 12103 043276
 12104 043276 000004
 12105 043300 012700 000633
 12106 043304 013701 043346
 12107 043310 012702 177776
 12108 043314 010605
 12109 043316 010737 001010
 12110 043322 010506
 12111 043324 012704 000340
 12112 043330 012746 000340

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 @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
 MOV SP,R5 ;:SAVE THE SP
 MOV PC,@#SLPERR ;:SET ERROR LOOP ADDRESS
 1\$: MOV #-1,R4 ;:R3 SHOULD GET 177777
 MOV R5,SP ;:RESET SP FOR ERROR LOOP
 MOV #4\$,R3 ;:RTS SHOULD LOAD PC FROM [R3]
 MOV #-1,-(SP) ;:RTS SHOULD LOAD R3 WITH 177777
 CCC ;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 TST633 ;:BR IF YES
 MOV R5,R4 ;:[R4] = S / B SP
 MOV SP,R3 ;:[R3] = WAS SP
 9\$: 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 @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
 MOV #PSW,R2 ;:DEST=PSW FOR 5\$ CALL
 MOV SP,R5 ;:SAVE THE SP
 MOV PC,@#SLPERR ;:SET ERROR LOOP ADDRESS
 1\$: MOV R5,SP ;:RESET SP FOR ERROR LOOP
 MOV #340,R4 ;:[R4] = S / B PSW AT HTIS POINT
 MOV #340,-(SP) ;:NEW PSW S / B = 340

```

12113 043334 012746 043354      MOV    #4$,-(SP)      ;NEW PC S / B = 4$
12114 043340 005037 177776      CLR    @#PSW          ;CLEAR THE PSW
12115 043344 000277              SCC                    ;N:C = 1111
12116
12117 043346 000006      2$:    RTT              ;TEST THE RTT - GO TO 4$
12118
12119 043350 104005      3$:    ERROR    5      ;RTT FAILED TO LOAD THE PC
12120 043352 000412      BR     8$             ;GO TO EXIT - SCHOOL'S OUT
12121
12122 043354 013703 177776      4$:    MOV    @#PSW,R3  ;SAVE THE PSW
12123 043360 020403      CMP    R4,R3         ;WAS PSW = 340 ?
12124 043362 001401      BEQ    6$             ;BR IF IT WAS
12125
12126 043364 104001      5$:    ERROR    1      ;RTT FAILED TO LOAD PSW PROPERLY
12127
12128 043366 020506      6$:    CMP    R5,SP    ;DID RTT UPDATE THE SP ?
12129 043370 001404      BEQ    TST634        ;:BR IF YES
12130
12131 043372 010504      MOV    R5,R4         ;[R4] = S / B SP
12132 043374 010603      MOV    SP,R3         ;[R3] = WAS SP
12133 043376 104003      7$:    ERROR    3      ;RTT FAILED TO UPDATE SP
12134
12135 043400 010506      8$:    MOV    R5,SP    ;FIX THE SP
12136
12137
12138      ;:*****
12139      ;:*TEST 634      RTT TEST - N:C = 0000
12140      ;:*****
12140 043402      TST634:
12141 043402 000004      SCOPE                ;CALL THE SCOPE LOOP UTILITY
12142 043404 012700 000634      MOV    #634,R0       ;:LOAD R0 WITH TEST NUMBER
12143 043410 013701 043454      MOV    @#2$,R1       ;:LOAD R1 WITH TEST INSTRUCTION WORD
12144 043414 012702 177776      MOV    #PSW,R2       ;:DEST=PSW FOR 5$ CALL
12145 043420 010605      MOV    SP,R5         ;:SAVE THE SP
12146 043422 010737 001010      MOV    PC,@#SLPERR   ;:SET ERROR LOOP ADDRESS
12147 043426 010506      1$:    MOV    R5,SP     ;:RESET SP FOR ERROR LOOP
12148 043430 012704 000017      MOV    #017,R4       ;:[R4] = S / B PSW AT HTIS POINT
12149 043434 012746 000017      MOV    #017,-(SP)    ;:NEW PSW S / B = 017
12150 043440 012746 043462      MOV    #4$,-(SP)    ;:NEW PC S / B = 4$
12151 043444 012737 000340 177776      MOV    #340,@#PSW   ;:MAKE [PSW] = 340
12152 043452 000257      CCC                    ;:N:C = 0000
12153
12154 043454 000006      2$:    RTT              ;TEST THE RTT - GO TO 4$
12155
12156 043456 104005      3$:    ERROR    5      ;RTT FAILED TO LOAD THE PC
12157 043460 000412      BR     8$             ;GO TO EXIT - SCHOOL'S OUT
12158
12159 043462 013703 177776      4$:    MOV    @#PSW,R3  ;SAVE THE PSW
12160 043466 020403      CMP    R4,R3         ;WAS PSW = 017 ?
12161 043470 001401      BEQ    6$             ;BR IF IT WAS
12162
12163 043472 104001      5$:    ERROR    1      ;RTT FAILED TO LOAD PSW PROPERLY
12164
12165 043474 020506      6$:    CMP    R5,SP    ;DID RTT UPDATE THE SP ?
12166 043476 001404      BEQ    TST635        ;:BR IF YES
12167
12168 043500 010504      MOV    R5,R4         ;[R4] = S / B SP

```

12169 043502 010603
 12170 043504 104003
 12171
 12172 043506 010506
 12173
 12174
 12175
 12176
 12177 043510
 12178 043510 000004
 12179 043512 012700 000635
 12180 043516 013701 043542
 12181 043522 010602
 12182 043524 012704 125252
 12183 043530 012705 043572
 12184 043534 010437 043556
 12185 043540 000257
 12186
 12187 043542 006405
 12188
 12189 043544 010637 001074
 12190 043550 010206
 12191 043552 104005
 12192
 12193 043554 000444
 12194
 12195 043556 125252
 12196
 12197 043560 010637 001074
 12198 043564 010206
 12199 043566 104005
 12200
 12201 043570 000436
 12202
 12203 043572 100403
 12204 043574 001402
 12205 043576 102401
 12206 043600 103011
 12207
 12208 043602 013703 177776
 12209 043606 010637 001074
 12210 043612 010206
 12211 043614 012702 177776
 12212 043620 104007
 12213 043622 000421
 12214
 12215 043624 020627 043560
 12216 043630 001406
 12217 043632 010603
 12218 043634 012704 043560
 12219 043640 010206
 12220 043642 104003
 12221
 12222 043644 000410
 12223
 12224 043646 020504

```

MOV SP,R3          ;[R3] = WAS SP
7$: ERROR 3        ;RTT FAILED TO UPDATE SP

MOV R5,SP          ;FIX THE SP
8$:

*****
;*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$,R1        ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R2          ;:SAVE SP
MOV #125252,R4     ;:[R5] SHOULD BE 125252
MOV #4$,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,@#5REG5    ;: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,@#5REG5    ;: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$: MOV @#PSW,R3  ;:SAVE FLAGS IN R3
      MOV SP,@#5REG5 ;:SAVE BAD SP FOR PRINTING
      MOV R2,SP     ;:RESET SP
      MCV #PSW,R2   ;:DEST=PSW
7$: ERROR 7        ;:MARK SET A FLAG
      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$+2,R4 ;:S/B SP
      MOV R2,SP   ;:RESET SP
9$: ERROR 3       ;:MARK FAILED TO RESET SP

BR TST636        ;:GO TO SCOPE EXIT

11$: CMP R5,R4    ;:DID MARK RESTORE OLD R5

```

```

12225 043650 001405          BEQ      12$          ;BR IF YES
12226
12227 043652 010637 001074    MOV      SP,@#$REG5    ;SAVE BAD SP FOR PRINTING
12228 043656 010503          MOV      R5,R3         ;WAS DEST
12229 043660 010206          MOV      R2,SP         ;RESET SP
12230 043662 104004          ERROR    4             ;MARK FAILED TO RESET R5
12231
12232 043664 010206          12$:    MOV      R2,SP         ;RESET SP
12233
12234
12235
12236
12237 043666
12238 043666 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12239 043670 012700 000636    MOV      #636,R0       ;LOAD R0 WITH TEST NUMBER
12240 043674 013701 043720    MOV      @#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
12241 043700 010602          MOV      SP,R2         ;SAVE SP
12242 043702 012704 125252    MOV      #125252,R4    ;[R5] SHOULD BE 125252
12243 043706 012705 043750    MOV      #4$,R5        ;MARK GOES TO 4$ VIA [R5]
12244 043712 010437 043734    MOV      R4,@#6$      ;INITIALIZE WORD LOADED INTO R5
12245 043716 000277          SCC                   ;N:C=1111
12246
12247 043720 006405          2$:    MARK+5          ;TEST THE MARK
12248
12249 043722 010637 001074    MOV      SP,@#$REG5    ;SAVE BAD SP FOR PRINTING
12250 043726 010206          MOV      R2,SP         ;RESET SP
12251 043730 104005          3$:    ERROR    5             ;MARK FAILED TO EXECUTE
12252
12253 043732 000444          BR      TST637         ;GO TO SCOPE EXIT
12254
12255 043734 125252          6$:    125252          ;THIS WORD SHOULD GET LOADED INTO R5
12256
12257 043736 010637 001074    MOV      SP,@#$REG5    ;SAVE BAD SP FOR PRINTING
12258 043742 010206          MOV      R2,SP         ;RESET SP
12259 043744 104005          5$:    ERROR    5             ;MARK FAILED TO LOAD RC FROM [R5]
12260
12261 043746 000436          BR      TST637         ;GO TO SCOPE EXIT
12262
12263 043750 100003          4$:    BPL      7$          ;N:C=1111
12264 043752 001002          BNE     7$
12265 043754 102001          BVC     7$
12266 043756 103411          BCS     8$
12267
12268 043760 013703 177776    7$:    MOV      @#PSW,R3     ;SAVE FLAGS IN R3
12269 043764 010637 001074    MOV      SP,@#$REG5    ;SAVE BAD SP FOR PRINTING
12270 043770 010206          MOV      R2,SP         ;RESET SP
12271 043772 012702 177776    MOV      #PSW,R2       ;DEST=PSW
12272 043776 104007          ERROR    7             ;MARK SET A FLAG
12273 044000 000421          BR      TST637         ;GO TO SCOPE EXIT
12274
12275 044002 020627 043736    8$:    CMP      SP,#6$+2     ;DID MARK RESET SP?
12276 044006 001406          BEQ     9$             ;BR IF YES
12277 044010 010603          MOV      SP,R3         ;PUT BAD SP IN R3
12278 044012 012704 043736    MOV      #6$+2,R4     ;S/B SP
12279 044016 010206          MOV      R2,SP         ;RESET SP
12280 044020 104003          ERROR    3             ;MARK FAILED TO RESET SP

```

```
12281
12282 044022 000410          BR      TST637          ;;GO TO SCOPE EXIT
12283
12284 044024 020504          9$:    CMP      R5,R4          ;DID MARK RESTORE OLD R5
12285 044026 001405          BEQ     10$          ;BR IF YES
12286
12287 044030 010637 001074      MOV     SP,@#SREG5      ;SAVE BAD SP FOR PRINTING
12288 044034 010503          MOV     R5,R3          ;WAS DEST
12289 044036 010206          MOV     R2,SP          ;RESET SP
12290 044040 104004          ERROR  4              ;MARK FAILED TO RESET R5
12291
12292 044042 010206          10$:   MOV     R2,SP          ;RESET SP
12293
12294          ;;*****
12295          ;*TEST 637      BASIC LINE CLOCK RESPONSE TEST
12296          ;;*****
12297          TST637:
12298 044044 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12299 044046 012700 000637      MOV     #637,R0        ;LOAD R0 WITH TEST NUMBER
12300 044052 013701 044102      MOV     @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
12301 044056 010605          MOV     SP,R5          ;SAVE SP
12302 044060 012702 177546      MOV     #LKCSR,R2      ;[R2] = LINE CLOCK ADDRESS
12303 044064 010737 001010      MOV     PC,@#SLPERR    ;SET ERROR LOOP ADDRESS
12304 044070 010506          1$:    MOV     R5,SP          ;RESET SP FOR ERROR LOOP
12305 044072 012737 044106 000004  MOV     #4$,@#4        ;GO TO 4$ IF BUS TIMEOUT
12306 044100 000257          CCC          ;SCOPE SYNC
12307
12308 044102 005712          2$:    TST     (R2)        ;REFERENCE LKCSR ADDR
12309
12310 044104 000404          BR      6$            ;GO TO EXIT
12311
12312 044106 012737 061220 000004  4$:    MOV     #BERR,@#4    ;RESTORE TIMEOUT VECTOR
12313 044114 104006          3$:    ERROR  6          ;LKCSR FAILED TO RESPOND
12314
12315 044116 010506          6$:    MOV     R5,SP          ;RESET SP
12316 044120 012737 061220 000004  MOV     #BERR,@#4    ;RESTORE TIMEOUT VECTOR
12317
12318          ;;*****
12319          ;*TEST 640      LINE CLOCK TEST - LKCSR BIT 7 SET
12320          ;;*****
12321          TST640:
12322 044126 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12323 044130 012700 000640      MOV     #640,R0        ;LOAD R0 WITH TEST NUMBER
12324 044134 013701 044152      MOV     @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
12325 044140 012702 177546      MOV     #LKCSR,R2      ;DEST ADDR = 177546
12326 044144 012704 000200      MOV     #200,R4        ;[LKCSR] S / B = 200
12327 044150 000257          CCC          ;SCOPE SYNC
12328
12329 044152 030412          2$:    BIT     R4,(R2)      ;TEST BIT 7 IN LKCSR
12330
12331 044154 001002          BNE     TST641        ;;BR IF IT'S SET
12332
12333 044156 011203          3$:    MOV     (R2),R3      ;GET WAS DATA
12334 044160 104001          ERROR  1          ;BIT 7 NOT SET IN LKCSR
12335
12336          ;;*****
```

```

12337
12338
12339 044162
12340 044162 000004
12341 044164 012700 000641
12342 044170 013701 044206
12343 044174 012702 177546
12344 044200 012704 000200
12345 044204 000257
12346
12347 044206 032712 000100
12348
12349 044212 001402
12350
12351 044214 011203
12352 044216 104001
12353
12354
12355
12356
12357 044220
12358 044220 000004
12359 044222 012700 000642
12360 044226 013701 044276
12361 044232 010605
12362 044234 012702 177546
12363 044240 012704 000300
12364 044244 010737 001010
12365 044250 012737 044312 000100
12366 044256 012737 000340 000102
12367 044264 010506
12368 044266 012737 000340 177776
12369 044274 000257
12370
12371 044276 052712 000100
12372
12373 044302 020412
12374 044304 001402
12375
12376 044306 011203
12377 044310 104001
12378
12379 044312 042737 000102 000100
12380 044320 005037 000102
12381 044324 042712 000100
12382 044330 010506
12383
12384
12385
12386
12387 044332
12388 044332 000004
12389 044334 012700 000643
12390 044340 013701 044406
12391 044344 010605
12392 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 @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #LKCSR,R2 ;:R2 POINTS TO LKCSR
MOV #200,R4 ;:[LKCSR] S / B = 200
CCC ;SCOPE SYNC

2$: BIT #100,(R2) ;:TEST BIT 6 IN LKCSR
BEQ TST642 ;:BR IF CLEAR

MOV (R2),R3 ;:GET WAS DATA
3$: 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 @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;:SAVE SP
MOV #LKCSR,R2 ;:R2 POINTS TO LKCSR
MOV #300,R4 ;:[LKCSR] S / B = 300
MOV PC,@#SLPERR ;:SET ERROR LOOP ADDRESS
1$: MOV #4$,@#100 ;:SET UP LCLK VECTOR IN CASE LOGIC
MOV #340,@#102 ;:FAULT CAUSES ATL INTERRUPT
MOV R5,SP ;:RESET SP FOR ERROR LOOP
MOV #340,@#PSW ;:SET PRIORITY TO LEVEL 7
CCC ;SCOPE SYNC

2$: BIS #100,(R2) ;:SET BIT 6 IN LKCSR

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

MOV (R2),R3 ;:GET WAS DATA
3$: ERROR 1 ;:BIT 6 FAILED TO SET IN LKCSR

4$: BIC #102,@#100 ;:RESTORE TRAP CATCHER IN LINE CLOCK VECTOR
CLR @#102
BIC #100,(R2) ;:TURN OF 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 @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;:SAVE SP
MOV #LKCSR,R2 ;:R2 POINTS TO LKCSR
  
```

```

12393 044352 010737 001010          MOV    PC,@#SLPERR      ;SET ERROR LOOP ADDRESS
12394 044356 010506          1$:   MOV    R5,SP          ;RESET SP FOR ERROR LOOP
12395 044360 005004          CLR    R4              ;INITIALIZE TIMER
12396 044362 012737 044424 000100    MOV    #4,@#100        ;SET UP LINE CLOCK VECTOR TO TO
12397 044370 012737 000340 000102    MOV    #340,@#102      ;TO 4$ WITH PROCESSOR PRIORITY = 7
12398 044376 005012          CLR    (R2)           ;CLEAR LKCSR
12399 044400 005037 177776          CLR    @#PSW          ;SET PRIORITY TO LEVEL 000
12400 044404 000257          CCC                   ;SCOPE SYNC
12401
12402 044406 052712 000100          2$:   BIS    #100,(R2)    ;ENABLE LINE CLK INTERRUPT
12403
12404 044412 005304          DEC    R4              ;WAIT FOR INTR - REPORT ERROR IF
12405 044414 001376          BNE    .-2            ;R4 GOES TO 000000
12406
12407 044416 042712 000100          BIC    #100,(R2)      ;TURN OFF INTR. ENAB.
12408 044422 104006          3$:   ERROR  6          ;LINE CLK FAILED TO INTERRUPT
12409
12410 044424 042712 000100          4$:   BIC    #100,(R2)  ;TURN OFF INTR. ENAB.
12411 044430 012737 000102 000100    MOV    #102,@#100     ;RESTORE TRAP CATCHER IN LINE CLK VECTOR
12412 044436 005037 000102          CLR    @#102
12413 044442 010506          MOV    R5,SP          ;RESET SP
12414 044444 005037 177776          CLR    @#PSW          ;RESET PRIORITY TO LEVEL 0
12415
12416          ;*****
12417          ;*TEST 644      RESET TEST - N:C = 1111
12418          ;*****
12419          TST644:
12420 044450 000004          SCOPE                   ;CALL THE SCOPE LOOP UTILITY
12421 044452 012700 000644          MOV    #644,R0        ;LOAD R0 WITH TEST NUMBER
12422 044456 013701 044510          MOV    @#2$,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD
12423 044462 012737 000001 001110    MOV    #1,@#$TIMES    ;NO ITERATIONS ON THIS TEST
12424 044470 012702 177564          MOV    #XCSR,R2       ;R2 POINTS TO DL11 XCSR
12425 044474 012737 000340 177776          MOV    #340,@#PSW     ;MAKE PRTY. BITS ALL 1'S
12426 044502 052712 000004          BIS    #4,(R2)        ;SET THE DL11 MAINT. BIT
12427 044506 000277          SCC                   ;N:C = 1111
12428
12429 044510 000005          2$:   RESET                   ;TEST THE RESET - IT SHOULD CLEAR THE DL11 MAINT BIT
12430
12431 044512 013705 177776          MOV    @#PSW,R5       ;SAVE THE PSW
12432 044516 032712 000004          BIT    #4,(R2)        ;DID MAINT. BIT CLEAR ??
12433 044522 001403          BEQ    4$             ;BR IF YES
12434
12435 044524 042712 000004          3$:   BIC    #4,(R2)        ;MAKE SURE TO TURN OFF MAINT. BIT
12436 044530 104006          ERROR  6          ;RESET FAILED TO CLEAR MAINT BIT
12437
12438 044532 022705 000357          4$:   CMP    #357,R5     ;DID RESET ALTER THE PSW ??
12439 044536 001406          BEQ    6$             ;BR IF NOT
12440
12441 044540 012704 000357          MOV    #357,R4        ;[R4] = S/B PSW
12442 044544 010503          MOV    R5,R3          ;[R3] = WAS PSW
12443 044546 012702 177776          MOV    #PSW,R2        ;DEST = PSW
12444 044552 104001          5$:   ERROR  1          ;RESET ALTERED THE PSW
12445
12446 044554 005037 177776          6$:   CLR    @#PSW       ;CLEAR OUT THE PSW
12447 044560 042737 000004 177564    BIC    #4,@#XCSR      ;MAKE SURE MAINT BIT IS OFF
12448

```


12449
12450
12451
12452 044566
12453 044566 000004
12454 044570 012700 000645
12455 044574 013701 044624
12456 044600 012737 000001 001110
12457 044606 012702 177564
12458 044612 005037 177776
12459 044616 052712 000004
12460 044622 000257
12461
12462 044624 000005
12463
12464 044626 013705 177776
12465 044632 032712 000004
12466 044636 001403
12467
12468 044640 042712 000004
12469 044644 104006
12470
12471 044646 022705 000000
12472 044652 001406
12473
12474 044654 012704 000357
12475 044660 010503
12476 044662 012702 177776
12477 044666 104001
12478
12479 044670 005037 177776
12480 044674 042737 000004 177564
12481
12482
12483
12484
12485 044702
12486 044702 000004
12487 044704 012700 000646
12488 044710 013701 044776
12489 044714 010605
12490 044716 010737 001010
12491 044722 012702 177564
12492 044726 012737 045014 000064
12493 044734 012737 000200 000066
12494 044742 010506
12495 044744 005012
12496 044746 005003
12497
12498 044750 105712
12499 044752 100403
12500 044754 005303
12501 044756 001374
12502 044760 000440
12503
12504 044762 012737 000140 177776

*TEST 645 RESET TEST - N:C = 0000

TST645:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #645,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #1,@#\$TIMES ;NO ITERATIONS ON THIS TEST
MOV #XCSR,R2 ;R2 POINTS TO DL11 XCSR
CLR @#PSW ;MAKE PRY. BITS ALL 0'S
BIS #4,(R2) ;SET THE DL11 MAINT. BIT
CCC ;N:C = 0000
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 #0,R5 ;DID RESET ALTER THE PSW ??
BEQ 6\$;BR IF NOT
5\$: MOV #357,R4 ;[R4] = S/B PSW
MOV R5,R3 ;[R3] = WAS PSW
MOV #PSW,R2 ;DEST = PSW
ERROR 1 ;RESET ALTERED THE PSW
6\$: CLR @#PSW ;CLEAR OUT THE PSW
BIC #4,@#XCSR ;MAKE SURE MAINT BIT IS OFF

*TEST 646 WAIT INSTRUCTION TEST - [PSW] = 151

TST646:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #646,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV PC,@#\$LPERR ;SET ERROR LOOP ADDRESS
1\$: MOV #XCSR,R2 ;R2 POINT TO DL11 XCSR
MOV #4,@#64 ;GO TO 4\$ ON DL11 INTR.
MOV #200,@#66 ;AT LEVEL 4
MOV R5,SP ;RESET SP FOR ERROR LOOP
CLR (R2) ;INIT DL11 XCSR
CLR R3 ;INIT TIMER
3\$: TSTB (R2) ;DL11 XMIT READY SET ??
BMI 5\$;BR IF YES
DEC R3 ;COUNT THE TIMER
BNE 3\$;BR IF NO TIMEOUT
BR 9\$;GO REPORT TIMEOUT
5\$: MOV #140,@#PSW ;SET PSW PRY BITS TO LEVEL 3

```

12505 044770 000277          SCC          ;N:C=1111
12506 044772 152712 000100  BISB      #100,(R2) ;ENAB. DL11 INTR - N:C=1001
12507
12508 044776 000001          2$: WAIT          ;TEST THE WAIT-GO TO 4$ ON INTR
12509
12510 045000 012737 000340 177776  MOV      #340,@#PSW ;LOCK OUT INTR
12511 045006 005012          CLR      (R2)      ;TURN OFF DL11 INTR ENAB
12512 045010 104006          ERROR    6        ;WAIT FAILED TO EXECUTE PROPERLY
12513 045012 000424          BR       8$        ;GO EXIT THIS TEST
12514
12515 045014 042712 000100          4$: BIC      #100,(R2) ;TURN OFF DL11 INTR ENAB
12516 045020 022716 045000          CMP      #2$+2,(SP) ;DID WAIT GET FETCHED ??
12517 045024 001402          BEQ     6$        ;BR IF YES
12518
12519 045026 104006          ERROR    6        ;WAIT NOT FETCHED PROPERLY
12520 045030 000415          BR       8$        ;GO EXIT THE TEST
12521
12522 045032 022766 000151 000002  6$: CMP      #151,2(SP) ;DID "WAIT" ALTER THE PSW ??
12523 045040 001411          BEQ     8$        ;BR IF YES
12524
12525 045042 012704 000151          MOV      #151,R4   ;[R4] = S/B PSW
12526 045046 016603 000002          MOV      2(SP),R3  ;[R3] = WAS PSW
12527 045052 012702 177776          MOV      #PSW,R2   ;DEST = PSW
12528 045056 104001          7$: ERROR    1        ;"WAIT" ALTERED THE PSW
12529 045060 000401          BR       8$        ;GOT TO EXIT TEST
12530
12531 045062 104006          9$: ERROR    6        ;DL11 FAILED TO SET READY ON TIME
12532
12533 045064 010506          8$: MOV      R5,SP   ;RESET THE SP
12534 045066 005037 177776          CLR      @#PSW     ;CLEAR OUT THE PSW
12535 045072 005012          CLR      (R2)      ;TURN OFF DL11 INTR.
12536 045074 012737 000066 000064  MOV      #66,@#64  ;RESTORE DL11 VECTOR WITH TRAPCATCHER
12537 045102 005037 000066          CLR      @#66
12538
12539
12540
12541
12542 045106
12543 045106 000004          :*****
12544 045110 012700 000647          :*TEST 647 WAIT INSTRUCTION TEST - [PSW] = 010
12545 045114 013701 045200          :*****
12546 045120 010605          TST647:
12547 045122 010737 001010          SCOPE          ;CALL THE SCOPE LOOP UTILITY
12548 045126 012702 177564          MOV      #647,R0   ;:LOAD R0 WITH TEST NUMBER
12549 045132 012737 045216 000064  MOV      @#2$,R1   ;LOAD R1 WITH TEST INSTRUCTION WORD
12550 045140 012737 000200 000066  MOV      SP,R5     ;SAVE THE SP
12551 045146 010506          MOV      PC,@#SLPERR ;SET ERROR LOOP ADDRESS
12552 045150 005012          1$: MOV      #XCSR,R2 ;R2 POINT TO DL11 XCSR
12553 045152 005003          MOV      #4$,@#64  ;GO TO 4$ ON DL11 INTR.
12554          MOV      #200,@#66 ;AT LEVEL 4
12555 045154 105712          3$: MOV      R5,SP   ;RESET SP FOR ERROR LOOP
12556 045156 100403          CLR      (R2)      ;INIT DL11 XCSR
12557 045160 005303          CLR      R3        ;INIT TIMER
12558 045162 001374          ;DL11 XMIT READY SET ??
12559 045164 000437          BMI     5$        ;BR IF YES
12560          DEC     R3        ;COUNT THE TIMER
          BNE    3$        ;BR IF NO TIMEOUT
          BR     9$        ;GO REPORT TIMEOUT

```

```
12561 045166 005037 177776 5$: CLR @#PSW ;SET PSW PRY BITS TO LEVEL 0
12562 045172 000257 CCC ;N:C=0000
12563 045174 152712 000100 BISB #100,(R2) ;ENAB. DL11 INTR - N:C=1000
12564
12565 045200 000001 2$: WAIT ;TEST THE WAIT-GO TO 4$ ON INTR
12566
12567 045202 012737 000340 177776 MOV #340,@#PSW ;LOCK OUT INTR
12568 045210 005012 (R2) ;TURN OFF DL11 INTR ENAB
12569 045212 104006 ERROR 6 ;WAIT FAILED TO EXECUTE PROPERLY
12570 045214 000424 BR 8$ ;GO EXIT THIS TEST
12571
12572 045216 042712 000100 4$: BIC #100,(R2) ;TURN OFF DL11 INTR ENAB
12573 045222 022716 045202 CMP #2$+2,(SP) ;DID WAIT GET FETCHED ??
12574 045226 001402 BEQ 6$ ;BR IF YES
12575
12576 045230 104006 ERROR 6 ;WAIT NOT FETCHED PROPERLY
12577 045232 000415 BR 8$ ;GO EXIT THE TEST
12578
12579 045234 022766 000010 000002 6$: CMP #010,2(SP) ;DID "WAIT" ALTER THE PSW ??
12580 045242 001411 BEQ 8$ ;BR IF NO
12581
12582 045244 012704 000010 MOV #010,R4 ;[R4] = S/B PSW
12583 045250 016603 000002 MOV 2(SP),R3 ;[R3] = WAS PSW
12584 045254 012702 177776 MOV #PSW,R2 ;DEST = PSW
12585 045260 104001 7$: ERROR 1 ;"WAIT" ALTERED THE PSW
12586 045262 000401 BR 8$ ;GOT TO EXIT TEST
12587
12588 045264 104006 9$: ERROR 6 ;DL11 FAILED TO SET READY ON TIME
12589
12590 045266 010506 8$: MOV R5,SP ;RESET THE SP
12591 045270 005037 177776 CLR @#PSW ;CLEAR OUT THE PSW
12592 045274 005012 (R2) ;TURN OFF DL11 INTR.
12593 045276 012737 000066 000064 MOV #66,@#64 ;RESTORE DL11 VECTOR WITH TRAPCATCHER
12594 045304 005037 000066 CLR @#66
12595
12596 ::*****
12597 :*TEST 650 BR PRIORITY ARBITRATION TEST - LEVEL 1 USING LINE CLK
12598 :*****
12599 TST650:
12600 045310 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
12601 045312 012700 000650 MOV #650,R0 ;:LOAD R0 WITH TEST NUMBER
12602 045316 013701 045364 MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
12603 045322 010605 MOV SP,R5 ;SAVE THE SP
12604 045324 010737 001010 MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
12605 045330 012702 177546 1$: MOV #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
12606 045334 012737 045402 000100 MOV #4$,@#100 ;IF INTR OCCURS - GO TO 4$
12607 045342 012737 000340 000102 MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
12608 045350 010506 MOV R5,SP ;RESET SP FOR ERROR LOOPING
12609 045352 005004 CLR R4 ;INITIALIZE R4 AS TIMER
12610 045354 012737 000040 177776 MOV #40,@#PSW ;SET CPU PRIORITY TO LEVEL 1
12611 045362 000257 CCC ;SCOPE SYNC
12612
12613 045364 052712 000100 2$: BIS #100,(R2) ;ENABLE LINE CLK INTERRUPTS
12614
12615 045370 005304 DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
12616 045372 001376 BNE .-2 ;TIMER FROM GETTING BACK TO 000000
```

CQKDA-D KD11-K BASIC LOGIC TESTS
CQKDA.D.P11 11-SEP-79 13:31

MACY11 30A(1052) 11-SEP-79 ^{B 3} 13:53 PAGE 235
T650 BR PRIORITY ARBITRATION TEST - LEVEL 1 USING LINE CLK

SEQ 0234

12617
12618 045374 042712 000100
12619 045400 104006
12620
12621 045402 042712 000100

38: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
ERROR 6 ;LINE CLK FAILED TO INTR AT LEVEL 1
48: BIC #100,(R2) ;TURN OFF INTR. ENABLE

12622 045406 012737 000102 000100
12623 045414 005037 000102
12624 045420 010506
12625 045422 005037 177776

MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
CLR @#102
MOV R5,SP ;RESET THE SP
CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0

12626
12627
12628
12629

*TEST 651 BR PRIORITY ARBITRATION TEST - LEVEL 2 USING LINE CLK

12630 045426
12631 045426 000004
12632 045430 012700 000651
12633 045434 013701 045502
12634 045440 010605
12635 045442 010737 001010
12636 045446 012702 177546
12637 045452 012737 045520 000100
12638 045460 012737 000340 000102
12639 045466 010506
12640 045470 005004
12641 045472 012737 000100 177776
12642 045500 000257

TST651:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #651,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
1\$: MOV #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
MOV #4\$,@#100 ;IF INTR OCCURS - GO TO 4\$
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

12643
12644 045502 052712 000100
12645
12646 045506 005304
12647 045510 001376
12648
12649 045512 042712 000100
12650 045516 104006

2\$: BIS #100,(R2) ;ENABLE LINE CLK INTERRUPTS
DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
BNE .-2 ;TIMER FROM GETTING BACK TO 000000
3\$: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
ERROR 6 ;LINE CLK FAILED TO INTR AT LEVEL 2

12651
12652 045520 042712 000100
12653 045524 012737 000102 000100
12654 045532 005037 000102
12655 045536 010506
12656 045540 005037 177776

4\$: BIC #100,(R2) ;TURN OFF INTR. ENABLE
MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
CLR @#102
MOV R5,SP ;RESET THE SP
CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0

12657
12658
12659
12660

*TEST 652 BR PRIORITY ARBITRATION TEST - LEVEL 3 USING LINE CLK

12661 045544
12662 045544 000004
12663 045546 012700 000652
12664 045552 013701 045620
12665 045556 010605
12666 045560 010737 001010
12667 045564 012702 177546
12668 045570 012737 045636 000100
12669 045576 012737 000340 000102
12670 045604 010506
12671 045606 005004
12672 045610 012737 000140 177776
12673 045616 000257

TST652:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #652,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
1\$: MOV #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
MOV #4\$,@#100 ;IF INTR OCCURS - GO TO 4\$
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

12674
12675 045620 052712 000100
12676
12677 045624 005304

2\$: BIS #100,(R2) ;ENABLE LINE CLK INTERRUPTS
DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT

12678 045626 001376
12679
12680 045630 042712 000100
12681 045634 104006
12682
12683 045636 042712 000100
12684 045642 012737 000102 000100
12685 045650 005037 000102
12686 045654 010506
12687 045656 005037 177776
12688
12689

BNE -2 ;TIMER FROM GETTING BACK TO 000000
3\$: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
ERROR 6 ;LINE CLK FAILED TO INTR AT LEVEL 3
4\$: BIC #100,(R2) ;TURN OFF INTR. ENABLE
MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
CLR @#102
MOV R5,SP ;RESET THE SP
CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0

*TEST 653 BR PRIORITY ARBITRATION TEST - LEVEL 4 USING LINE CLK

12692 045662
12693 045662 000004
12694 045664 012700 000653
12695 045670 013701 045736
12696 045674 010605
12697 045676 010737 001010
12698 045702 012702 177546
12699 045706 012737 045754 000100
12700 045714 012737 000340 000102
12701 045722 010506
12702 045724 005004
12703 045726 012737 000200 177776
12704 045734 000257
12705

TST653:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #653,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
1\$: MOV #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
MOV #4\$,@#100 ;IF INTR OCCURS - GO TO 4\$
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

12706 045736 052712 000100
12707
12708 045742 005304
12709 045744 001376
12710
12711 045746 042712 000100
12712 045752 104006
12713
12714 045754 042712 000100
12715 045760 012737 000102 000100
12716 045766 005037 000102
12717 045772 010506
12718 045774 005037 177776
12719

2\$: BIS #100,(R2) ;ENABLE LINE CLK INTERRUPTS
DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
BNE -2 ;TIMER FROM GETTING BACK TO 000000
3\$: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
ERROR 6 ;LINE CLK FAILED TO INTR AT LEVEL 4
4\$: BIC #100,(R2) ;TURN OFF INTR. ENABLE
MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
CLR @#102
MOV R5,SP ;RESET THE SP
CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0

*TEST 654 BR PRIORITY ARBITRATION TEST - LEVEL 5 USING LINE CLK

12723 046000
12724 046000 000004
12725 046002 012700 000654
12726 046006 013701 046054
12727 046012 010605
12728 046014 010737 001010
12729 046020 012702 177546
12730 046024 012737 046072 000100
12731 046032 012737 000340 000102
12732 046040 010506
12733 046042 005004

TST654:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #654,R0 ;LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS
1\$: MOV #LKCSR,R2 ;R2 POINTS TO LINE CLK CSR
MOV #4\$,@#100 ;IF INTR OCCURS - GO TO 4\$
MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
MOV R5,SP ;RESET SP FOR ERROR LOOPING
CLR R4 ;INITIALIZE R4 AS TIMER

```
12734 046044 012737 000240 177776      MOV    #240,@#PSW      ;SET CPU PRIORITY TO LEVEL 5
12735 046052 000257                      CCC                    ;SCOPE SYNC
12736
12737 046054 052712 000100      2$:   BIS    #100,(R2)  ;ENABLE LINE CLK INTERRUPTS
12738
12739 046060 005304                      DEC    R4              ;COUNT THE TIMER - LCLK SHOULD PREVENT
12740 046062 001376                      BNE   .-2             ;TIMER FROM GETTING BACK TO 000000
12741
12742 046064 042712 000100      3$:   BIC    #100,(R2)  ;TURN OFF THE INTERRUPT ENABLE
12743 046070 104006                      ERROR  6              ;LINE CLK FAILED TO INTR AT LEVEL 5
12744
12745 046072 042712 000100      4$:   BIC    #100,(R2)  ;TURN OFF INTR. ENABLE
12746 046076 012737 000102 000100      MOV    #102,@#100    ;RESTORE TRAP CATCHER IN THE VECTOR
12747 046104 005037 000102                      CLR    @#102
12748 046110 010506                      MOV    R5,SP         ;RESET THE SP
12749 046112 005037 177776                      CLR    @#PSW         ;SET CPU PRIORITY BACK TO LEVEL 0
12750
```

```
:::*****
:*TEST 655      BR PRIORITY ARBITRATION TEST - LEVEL 6 USING LINE CLK
:::*****
```

```
12751
12752
12753
12754 046116      TST655:
12755 046116 000004      SCOPE                ;CALL THE SCOPE LOOP UTILITY
12756 046120 012700 000655      MOV    #655,R0       ;LOAD R0 WITH TEST NUMBER
12757 046124 013701 046204      MOV    @#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
12758
12759 046130 032737 020000 063234 .SBTTL USER CONTROLLED BREAKPOINT -- BIT13
12760 046136 001401      BIT    #BIT13,@#BPTLOC ;BREAKPOINT HALT SET ??
12761 046140 000000      BEQ   .+4            ;BR IF NOT
12762 046142 010605      HALT                    ;BREAK-DEPRESS CONTINUE TO CONTINUE
12763 046144 010737 001010      MOV    SP,R5         ;SAVE THE SP
12764 046150 012702 177546      MOV    PC,@#$LPERR   ;SET ERROR LOOP ADDRESS
12765 046154 012737 046216 000100 1$:   MOV    #LKCSR,R2     ;R2 POINTS TO LINE CLK CSR
12766 046162 012737 000340 000102      MOV    #4$,@#100     ;IF INTR OCCURS - GO TO 4$
12767 046170 010506      MOV    #340,@#102    ;WITH CPU PRIORITY AT LEVEL 7
12768 046172 005004      MOV    R5,SP         ;RESET SP FOR ERROR LOOP
12769 046174 012737 000300 177776      CLR    R4            ;INITIALIZE R4 AS TIMER
12770 046202 000257      MOV    #300,@#PSW    ;SET CPU PRIORITY TO LEVEL 6
12771                      CCC                    ;SCOPE SYNC
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$:   BIC    #100,(R2)  ;TURN OFF THE INTR ENABLE
12779 046222 104006      3$:   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 000100      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
:::*****
```

12787
12788
12789

```

12790 046250
12791 046250 000004
12792 046252 012700 000656
12793 046256 013701 046324
12794 046262 010605
12795 046264 010737 001010
12796 046270 012702 177564
12797 046274 012737 046336 000064
12798 046302 012737 000340 000066
12799 046310 010506
12800 046312 005004
12801 046314 012737 000340 177776
12802 046322 000257
12803
12804 046324 052712 000100
12805
12806 046330 005304
12807 046332 001376
12808 046334 000403
12809
12810 046336 042712 000100
12811 046342 104006
12812
12813 046344 042712 000100
12814 046350 012737 000066 000064
12815 046356 005037 000066
12816 046362 010506
12817 046364 005037 177776
12818
12819
12820
12821
12822
12823
12824
12825 046370
12826 046370 000004
12827 046372 012700 000657
12828 046376 013701 046470
12829 046402 012702 177546
12830 046406 010605
12831 046410 010737 001010
12832 046414 012737 046476 000100
12833 046422 012737 000300 000102
12834 046430 010506
12835 046432 005004
12836 046434 005003
12837 046436 012737 000340 177776
12838 046444 052712 000100
12839 046450 042712 000200
12840 046454 105712
12841 046456 100403
12842 046460 005304
12843 046462 001374
12844 046464 000411
12845 046466 000257
  
```

```

TST656:
SCOPE                                :CALL THE SCOPE LOOP UTILITY
MOV #656,R0                          :;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1                          :;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5                             :;SAVE THE SP
MOV PC,@#$LPERR                      :;SET ERROR LOOP ADDRESS
1$: MOV #XCSR,R2                      :;R2 POINTS TO DL11 XCSR
MOV #4$,@#64                          :;IF INTR OCCURS - GO TO 4$
MOV #340,@#66                         :;WITH CPU PRIORITY AT LEVEL 7
MOV R5,SP                             :;RESET SP FOR ERROR LOOP
CLR R4                                :;INITIALIZE R4 AS TIMER
MOV #340,@#PSW                        :;SET CPU PRIORITY TO LEVEL 7
CCC                                    :;SCOPE SYNC

2$: BIS #100,(R2)                     :;ENABLE INTERRUPTS

DEC R4                                :;COUNT UNTIL [R4] = 000000 - THEN
BNE .-2                               :;CONTINUE - NO INTERRUPT SHOULD OCCUR
BR 6$                                  :;GO TO EXIT - ALL OK

4$: BIC #100,(R2)                     :;TURN OFF THE INTR ENABLE
3$: ERROR 6                           :;INTR OCCURRED WITH CPU AT LEVEL 7

6$: BIC #100,(R2)                     :;TURN OFF INTR ENABLE
MOV #66,@#64                          :;RESET THE TRAP CATCHER IN THE VECTOR
CLR @#66
MOV R5,SP                             :;RESET SP JUST IN CASE
CLR @#PSW                              :;SET CPU PRIORITY BACK TO LEVEL 0
  
```

```

:*****
:*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"
:*****
  
```

```

TST657:
SCOPE                                :CALL THE SCOPE LOOP UTILITY
MOV #657,R0                          :;LOAD R0 WITH TEST NUMBER
MOV @#2$,R1                          :;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #LKCSR,R2                        :;R2 POINTS TO LINE CLK CSR
MOV SP,R5                             :;SAVE THE SP
MOV PC,@#$LPERR                      :;SET ERROR LOOP ADDRESS
1$: MOV #4$,@#100                     :;SET UP LCLK VECTOR TO GO TO 4$
MOV #300,@#102
MOV R5,SP                             :;RESET THE SP FOR ERROR LOOPING
CLR R4                                :;INITIALIZE TIMER FO KW
CLR R3                                :;CLEAR SOFTWARE FLAG
MOV #340,@#PSW                        :;LOCK OUT ALL INTRs
BIS #100,(R2)                         :;ENABLE LCLK INTRs
BIC #200,(R2)                         :;CLEAR LINE CLOCK READY
11$: TSTB (R2)                        :;LCLK READY TO INTR ??
BMI 12$                               :;BR IF YES
DEC R4                                :;COUNT THE TIMER
BNE 11$                               :;BR IF NO TIMEOUT
BR 6$                                  :;GO REPORT TIMEOUT
12$: CCC                              :;SCOPE SYNC
  
```


12846										
12847	046470	005037	177776		2\$:	CLR	@#PSW			:ALLOW INTRS - LCLK SHOULD INTERRUPT
12848										:BEFORE FETCHING NEXT INSTRUCTION
12849	046474	005103				COM	R3			:SHOULD NOT BE FETCHED
12850	046476	005012			4\$:	CLR	(R2)			:DISABLE THE LCLK INTR
12851	046500	005703				TST	R3			:DID SOFTWARE FLAG GET SET ??
12852	046502	001404				BEQ	8\$:BR IF NOT - IT WORKED OK
12853	046504	104006			3\$:	ERROR	6			:LCLK FAILED TO INTR ONTIME
12854	046506	000402				BR	8\$:GO EXIT
12855										
12856	046510	005012			6\$:	CLR	(R2)			:DISABLE LCLK INTR
12857	046512	104006			5\$:	ERROR	6			:LINE CLK TIMED OUT
12858										
12859	046514	010506			8\$:	MOV	R5,SP			:RESET THE SP
12860	046516	012737	000102	000100		MOV	#102,@#100			:RESTORE THE LINE CLK TRAPCATCHER
12861	046524	005037	000102			CLR	@#102			
12862										
12863										
12864										
12865										
12866										
12867										
12868										
12869										
12870	046530									
12871	046530	000004								
12872	046532	012700	000660							
12873	046536	013701	046676							
12874	046542	010605								
12875	046544	010737	001010							
12876	046550	012702	177546							
12877	046554	012703	177564							
12878	046560	012737	046706	000100						
12879	046566	012737	000300	000102						
12880	046574	012737	046740	000064						
12881	046602	012737	000200	000066						
12882	046610	010506								
12883	046612	012737	000340	177776						
12884	046620	005037	063312							
12885	046624	005037	063316							
12886	046630	005004								
12887	046632	052713	000100							
12888	046636	105713			11\$:	TSTB	(R3)			:XMIT READY SET ??
12889	046640	100403				BMI	12\$:BR IF YES
12890	046642	005304				DEC	R4			:COUNT THE TIMER
12891	046644	001374				BNE	11\$:BR IF NO TIMEOUT
12892	046646	000443				BR	5\$:GO REPORT TIMEOUT FOR DL11
12893										
12894	046650	005004			12\$:	CLR	R4			:INIT THE TIMER AGAIN
12895	046652	052712	000100			BIS	#100,(R2)			:ENABLE LCLK INTRS
12896	046656	042712	000200			BIC	#200,(R2)			:CLEAR THE LINE CLOCK READY BIT
12897	046662	105712			13\$:	TSTB	(R2)			:LCLK READY TO INTR
12898	046664	100403				BMI	14\$:BR IF YES
12899	046666	005304				DEC	R4			:COUNT THE TIMER
12900	046670	001374				BNE	13\$:BR IF NO TIMEOUT
12901	046672	000436				BR	7\$:GO REPORT LINE CLK TIMEOUT

```

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

```

```

TST660:
SCOPE
MOV #660,R0 ;CALL THE SCOPE LOOP UTILITY
MOV @#2$,R1 ;:LOAD R0 WITH TEST NUMBER
MOV SP,R5 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV PC,@#SLPERR ;SAVE THE SP
MOV #LKCSR,R2 ;SET ERROR LOOP ADDRESS
MOV #XCSR,R3 ;R2 POINTS TO LINE CLK CSR
MOV #4$,@#100 ;R3 POINTS TO DL11 XCSR
MOV #300,@#102 ;SET UP THE LCLK VECTOR - GO TO 4$
MOV #8$,@#64 ;SET UP THE DL11 VECTOR - GO TO 8$
MOV #200,@#66
MOV R5,SP ;RESET SP FOR ERROR LOOPING
MOV #340,@#PSW ;LOCK OUT ALL INTRS
CLR @#MBOF0 ;INIT TIMER
CLR @#MBOF1 ;CLEAR DL11 INTR FLAG
CLR R4 ;INIT TIMER
BIS #100,(R3) ;ENABLE DL11 XMIT INTR
11$: TSTB (R3) ;XMIT READY SET ??
BMI 12$ ;BR IF YES
DEC R4 ;COUNT THE TIMER
BNE 11$ ;BR IF NO TIMEOUT
BR 5$ ;GO REPORT TIMEOUT FOR DL11

12$: CLR R4 ;INIT THE TIMER AGAIN
BIS #100,(R2) ;ENABLE LCLK INTRS
BIC #200,(R2) ;CLEAR THE LINE CLOCK READY BIT
13$: TSTB (R2) ;LCLK READY TO INTR
BMI 14$ ;BR IF YES
DEC R4 ;COUNT THE TIMER
BNE 13$ ;BR IF NO TIMEOUT
BR 7$ ;GO REPORT LINE CLK TIMEOUT

```

```
12902 046674 000257          14$:   CCC                ;SCOPE SYNC
12903
12904 046676 005037 177776    2$:   CLR      @#PSW        ;ALLOW INTRs - KW SHOULD INTR FIRST
12905
12906 046702 005137 063312    4$:   COM      @#MBUFO      ;SET SOFTWARE FLAG IF FETCHED
12907 046706 005013          CLR      (R3)            ;DISABLE BOTH INTERRUPTS
12908 046710 005012          CLR      (R2)
12909 046712 005737 063312    TST      @#MBUFO        ;DID SOFTWARE FLAG GET SET ??
12910 046716 001402          BEQ      6$              ;BR IF NOT
12911
12912 046720 104006          3$:   ERROR    6          ;LINE CLK INTR OCCURRED TOO LATE
12913 046722 000425          BR      9$              ;GO TO EXIT
12914
12915 046724 005737 063316    6$:   TST      @#MBUF1     ;DID DL11 SOFTWARE FLAG SET ??
12916 046730 001422          BEQ      9$              ;BR IF NOT
12917
12918 046732 010302          MOV      R3,R2          ;FOR CORRECT DESTINATION TYP0UT
12919 046734 104006          ERROR    6              ;DL11 INTERRUPTED THE KW11
12920 046736 000417          BR      9$              ;GO TO EXIT TEST
12921
12922 046740 005137 063316    8$:   COM      @#MBUF1     ;FLAG THE DL11 INTR
12923 046744 005013          CLR      (R3)            ;DISABLE BOTH INTR ENABLES
12924 046746 005012          CLR      (R2)
12925 046750 010302          MOV      R3,R2          ;FOR CORRECT DESTINATION TYP0UT
12926 046752 104006          ERROR    6              ;DL11 SHOULD NOT HAVE INTERRUPTED
12927 046754 000410          BR      9$              ;GO EXIT TEST
12928
12929 046756 005012          5$:   CLR      (R2)          ;DISABLE THE INTR ENABLES
12930 046760 005013          CLR      (R3)
12931 046762 010302          MOV      R3,R2          ;FOR CORRECT DESTINATION TYP0UT
12932 046764 104006          ERROR    6              ;DL11 TIMEOUT
12933 046766 000403          BR      9$              ;GO TO EXIT
12934
12935 046770 005012          7$:   CLR      (R2)          ;DISABLE INTR ENABLES
12936 046772 005013          CLR      (R3)
12937 046774 104006          ERROR    6              ;KW11 TIMEOUT
12938
12939 046776 010506          9$:   MOV      R5,SP         ;RESET THE SP
12940 047000 005037 177776    CLR      @#PSW          ;RESET THE CPU PRIORITY
12941 047004 012737 000102 000100  MOV      #102,@#100     ;RESTORE LCLK VECTOR
12942 047012 005037 000102    CLR      @#102
12943 047016 012737 000066 000064  MOV      #66,@#64      ;RESTORE THE DL11 XMIT VECTOR
12944 047024 005037 000066    CLR      @#66
12945
12946          ; *****
12947          ; ///////////////////////////////////////////////////COMBINED INSTRUCTION EXERCISER SECTION ///////////////////////////////////////////////////
12948          ; *****
12949
12950          ; *****
12951          ; *TEST 661      "BPT" TRAP LINKAGE TEST
12952          ; *****
12953          TST661:
12954          SCOPE                ;CALL THE SCOPE LOOP UTILITY
12955          MOV      #661,R0      ;:LOAD R0 WITH TEST NUMBER
12956          MOV      @#2$,R1     ;:LOAD R1 WITH TEST INSTRUCTION WORD
12957          MOV      SP,R5        ;SAVE THE SP
```

```

12958 047044 010737 001010      MOV      PC,@#SLPERR      ;SET ERROR LOOP ADDRESS
12959 047050 012737 047066 000014 1$:      MOV      #4,@#14          ;GO TO 4$ ON "BPT" TRAP
12960 047056 010506                MOV      R5,SP            ;RESET THE SP FOR ERROR LOOPING
12961 047060 000257                CCC                       ;SCOPE SYNC
12962
12963 047062 000003                2$:      BPT                    ;TEST THE "BPT" - GO TO 4$
12964
12965 047064 104005                3$:      ERROR      5          ;BPT FAILED TO TRAP
12966
12967 047066 010506                4$:      MOV      R5,SP            ;RESET THE SP
12968 047070 012737 000016 000014      MOV      #16,@#14        ;RESTORE THE VECTOR
12969
12970
12971
12972
12973 047076
12974 047076 000004
12975 047100 012700 000662
12976 047104 013701 047144
12977 047110 010605
12978 047112 013704 000004
12979 047116 013703 000336
12980 047122 012737 047162 000004      MOV      #4,@#4 ;GO TO 4$ ON OVFLW
12981 047130 012737 125252 000336      MOV      #125252,@#336  ;INIT. [336]
12982 047136 012706 000340      MOV      #340,SP        ;SET SP TO CAUSE RED ZONE TRAP
12983 047142 000257                CCC                       ;SCOPE SYNC
12984
12985 047144 010046                2$:      MOV      R0,-(SP)        ;FORCE RED ZONE TRAP - GO TO 4$
12986
12987 047146 010437 000004      MOV      R4,@#4          ;RESTORE T.O. VECTOR
12988 047152 010637 001074      MOV      SP,@#REG5       ;SAVE BAD SP FOR PRINTING
12989 047156 010506                MOV      R5,SP            ;RESET SP FOR ERROR CALL
12990 047160 104005                3$:      ERROR      5          ;MOV FAILED TO CAUSE TRAP
12991
12992 047162 010437 000004      4$:      MOV      R4,@#4          ;RESTORE T.O. VECTOR
12993 047166 022706 000000      CMP      #0,SP           ;[SP]=0?
12994 047172 001404                BEQ      6$              ;BE IF YES
12995
12996 047174 010637 001074      MOV      SP,@#REG5       ;SAVE BAD SP FOR PRINTING
12997 047200 010506                MOV      R5,SP            ;RESET SP FOR ERROR CALL
12998 047202 104005                5$:      ERROR      5          ;SP NOT BEING JAMMED TO 4
12999
13000 047204 022737 125252 000336 6$:      CMP      #125252,@#336  ;DID PUSH OCCUR IN YELLOW ZONE?
13001 047212 001404                BEQ      8$              ;BR IF NOT
13002
13003 047214 010637 001074      MOV      SP,@#REG5       ;SAVE BAD SP FOR PRINTING
13004 047220 010506                MOV      R5,SP            ;RESET SP FOR ERROR CALL
13005 047222 104005                7$:      ERROR      5          ;MOV PUSHED INTO YELLOW ZONE
13006
13007 047224 010337 000336      8$:      MOV      R3,@#336        ;RESTORE VECTOR 336
13008 047230 010506                MOV      R5,SP            ;RESET SP
13009
13010
13011
13012
13013 047232

```

```

:*****
:*TEST 662      RED ZONE OVERFLOW TEST - MOV R,-(SP)
:*****
TST662:

```

```

SCOPE                ;CALL THE SCOPE LOOP UTILITY
MOV      #662,R0      ;:LOAD R0 WITH TEST NUMBER
MOV      @#2$,R1      ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV      SP,R5        ;:SAVE SP
MCV      @#4,R4       ;:SAVE T.O. VECTOR
MOV      @#336,R3     ;:SAVE VECTOR AT 336
MOV      #4,@#4 ;GO TO 4$ ON OVFLW
MOV      #125252,@#336 ;:INIT. [336]
MOV      #340,SP     ;:SET SP TO CAUSE RED ZONE TRAP
CCC                ;:SCOPE SYNC

```

```

:*****
:*TEST 663      YELLOW ZONE OVERFLOW TEST - MOV R,-(SP)
:*****
TST663:

```

```

13014 047232 000004          SCOPE          :CALL THE SCOPE LOOP UTILITY
13015 047234 012700 000663  MOV      #663,R0      :;LOAD R0 WITH TEST NUMBER
13016 047240 013701 047272  MOV      @#2$,R1     :LOAD R1 WITH TEST INSTRUCTION WORD
13017 047244 010605          MOV      SP,R5       :SAVE SP
13018 047246 012702 000376  MOV      #376,R2     :R2 POINTS TO STACK
13019 047252 013704 000004  MOV      @#4,R4       :SAVE T.O. VECTOR
13020 047256 012737 047310 000004  MOV      #4$,@#4 ;ON OVFLW - GO TO 4$
13021 047264 012706 000400  MOV      #400,SP     :SET SP TO CAUSE OVFLW
13022 047270 000257          CCC              :SCOPE SYNC
13023
13024 047272 010046          2$:  MOV      R0,-(SP)  :FORCE STACK OVFLW - GO TO 4$
13025
13026 047274 010437 000004  MOV      R4,@#4       :RESTORE T.O. VECTOR
13027 047300 010637 001074  MOV      SP,@#$REG5   :SAVE BAD SP FOR PRINTING
13028 047304 010506          MOV      R5,SP       :RESET SP FOR ERROR CALL
13029 047306 104005          3$:  ERROR    5       :STACK OVFLW FAILED TO TRAP
13030
13031 047310 010437 000004  4$:  MOV      R4,@#4       :RESTORE T.O. VECTOR
13032 047314 020012          CMP      R0,(R2)     :DID [R0] GET PUSHED?
13033 047316 001404          BEQ      6$         :BR IF YES
13034
13035 047320 010637 001074  MOV      SP,@#$REG5   :SAVE BAD SP FOR PRINTING
13036 047324 010506          MOV      R5,SP       :RESET SP FOR ERROR CALL
13037 047326 104005          5$:  ERROR    5       :MOV FAILED TO PUSH IN YELLOW ZONE
13038
13039 047330 005706          6$:  TST      SP        :[SP]=0?
13040 047332 001004          BNE      8$         :BR IF NOT
13041
13042 047334 010637 001074  MOV      SP,@#$REG5   :SAVE BAD SP FOR PRINTING
13043 047340 010506          MOV      R5,SP       :RESET SP FOR ERROR CALL
13044 047342 104005          7$:  ERROR    5       :RED ZONE INSTEAD OF YELLOW ZONE
13045
13046 047344 010506          8$:  MOV      R5,SP     :RESET SP
13047
13048
13049
13050
13051 047346          :*****
13052 047346 000004          :*TEST 664  YELLOW ZONE OVERFLOW TEST - (CMP R0,-(SP))
13053 047350 012700 000664  :*****
13054 047354 013701 047402  TST664:
13055 047360 010605          SCOPE          :CALL THE SCOPE LOOP UTILITY
13056 047362 013704 000004  MOV      #664,R0      :;LOAD R0 WITH TEST NUMBER
13057 047366 012737 047406 000004  MOV      @#2$,R1     :LOAD R1 WITH TEST INSTRUCTION WORD
13058 047374 012706 000400  MOV      SP,R5       :SAVE THE SP
13059 047400 000257          MOV      @#4,R4       :SAVE TRAP VECTOR
13060          MOV      #4$,@#4 ;GO TO 4$ IF TRAP SPRUNG
13061          MOV      #400,SP   :SET SP TO PUSH INTO "YELLOW ZONE"
13062          CCC              :SCOPE SYNC
13063
13064          2$:  CMP      R0,-(SP)  :TEST THE CMP - NO TRAP SHOULD OCCUR
13065
13066          BR      6$         :GO TO EXIT TEST
13067
13068          4$:  MOV      R4,@#4       :RESTORE TRAP VECTOR
13069          MOV      SP,@#$REG5   :SAVE BAD SP FOR PRINTING
13070          MOV      R5,SP       :RESET THE SP
13071          3$:  ERROR    5       :CMP CAUSED OVERFLOW IRAP

```

```

13070 047422 010437 000004
13071 047426 010506
13072
13073
13074
13075
13076 047430
13077 047430 000004
13078 047432 012700 000665
13079 047436 013701 047464
13080 047442 010605
13081 047444 013704 000004
13082 047450 012737 047470 000004
13083 047456 012706 000400
13084 047462 000257
13085
13086 047464 030046
13087
13088 047466 000406
13089
13090 047470 010437 000004
13091 047474 010637 001074
13092 047500 010506
13093 047502 104005
13094
13095 047504 010437 000004
13096 047510 010506
13097
13098
13099
13100
13101 047512
13102 047512 000004
13103 047514 012700 000666
13104 047520 013701 047546
13105 047524 010605
13106 047526 013704 000004
13107 047532 012737 047552 000004
13108 047540 012706 000400
13109 047544 000257
13110
13111 047546 005746
13112
13113 047550 000406
13114
13115 047552 010437 000004
13116 047556 010637 001074
13117 047562 010506
13118 047564 104006
13119
13120 047566 010437 000004
13121 047572 010506
13122
13123
13124
13125
  
```

```

6$:  MOV R4,@#4 ;RESTORE THE VECTOR
     MOV R5,SP ;RESET THE SP

*****
:*TEST 665 YELLOW ZONE OVERFLOW TEST - (BIT R0,-(SP))
*****
TST665:
      SCOPE ;CALL THE SCOPE LOOP UTILITY
      MOV #665,R0 ;:LOAD R0 WITH TEST NUMBER
      MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV SP,R5 ;SAVE THE SP
      MOV @#4,R4 ;SAVE TRAP VECTOR
      MOV #4$,@#4 ;GO TO 4$ IF TRAP SPRUNG
      MOV #400,SP ;SET SP TO PUSH INTO "YELLOW ZONE"
      CCC ;SCOPE SYNC

2$:  BIT R0,-(SP) ;TEST THE BIT - NO TRAP SHOULD OCCUR

      BR 6$ ;GO TO EXIT TEST

4$:  MOV R4,@#4 ;RESTORE TRAP VECTOR
     MOV SP,@#$REG5 ;SAVE BAD SP FOR PRINTING
     MOV R5,SP ;RESET THE SP

3$:  ERROR 5 ;BIT CAUSED OVERFLOW TRAP

6$:  MOV R4,@#4 ;RESTORE THE VECTOR
     MOV R5,SP ;RESET THE SP

*****
:*TEST 666 YELLOW ZONE OVERFLOW TEST - (TST -(SP))
*****
TST666:
      SCOPE ;CALL THE SCOPE LOOP UTILITY
      MOV #666,R0 ;:LOAD R0 WITH TEST NUMBER
      MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV SP,R5 ;SAVE THE SP
      MOV @#4,R4 ;SAVE TRAP VECTOR
      MOV #4$,@#4 ;GO TO 4$ IF TRAP SPRUNG
      MOV #400,SP ;SET SP TO PUSH INTO "YELLOW ZONE"
      CCC ;SCOPE SYNC

2$:  TST -(SP) ;TEST THE TST - NO TRAP SHOULD OCCUR

      BR 6$ ;GO TO EXIT TEST

4$:  MOV R4,@#4 ;RESTORE TRAP VECTOR
     MOV SP,@#$REG5 ;SAVE BAD SP FOR PRINTING
     MOV R5,SP ;RESET THE SP

3$:  ERROR 6 ;TST CAUSED OVERFLOW TRAP

6$:  MOV R4,@#4 ;RESTORE THE VECTOR
     MOV R5,SP ;RESET THE SP

*****
:*TEST 667 ODD ADDRESS ERROR TEST - SUB RA,(RB) - (RB) = ODD
*****
  
```

```

13126 047574
13127 047574 000004
13128 047576 012700 000667
13129 047602 013701 047636
13130 047606 010605
13131 047610 010737 001010
13132 047614 013704 000004
13133 047620 012737 047646 000004
13134 047626 010506
13135 047630 012702 000001
13136 047634 000257
13137
13138 047636 160012
13139
13140 047640 010437 000004
13141 047644 104006
13142
13143 047646 010437 000004
13144 047652 010506
13145 047654 005037 000000
13146
13147
13148
13149
13150 047660
13151 047660 000004
13152 047662 012700 000670
13153 047666 013701 047710
13154 047672 012702 063317
13155 047676 012737 047760 000004
13156
13157 047704 010205
13158 047706 000257
13159
13160 047710 105435
13161
13162 047712 104006
13163
13164 047714 012705 063321
13165 047720 013701 047726
13166 047724 000257
13167
13168 047726 105455
13169
13170 047730 104006
13171
13172 047732 010205
13173 047734 013701 047742
13174 047740 000257
13175
13176 047742 105475 000000
13177
13178 047746 104006
13179
13180 047750 012737 061220 000004
13181 047756 000403

```

```

TST667:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #667,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;:SAVE SP
MOV PC,@#SLPERR ;:SET ERROR LOOP ADDRESS
1$: MOV @#4,R4 ;:SAVE T.O. VECTOR
MOV #4$,@#4 ;ON ODD ADDR ERROR - GO TO 4$
MOV R5,SP ;:RESET SP FOR ERROR LOOP
MOV #1,R2 ;:R2 GETS ODD ADDRESS
CCC ;SCOPE SYNC

2$: SUB R0,(R2) ;:FORCE ODD ADDR ERROR - GO TO 4$

3$: MOV R4,@#4 ;:RESTORE T.O. VECTOR
ERROR 6 ;:ODD ADDR FAILED TO TRAP

4$: MOV R4,@#4 ;:RESTORE T.O. VECTOR
MOV R5,SP ;:RESET SP
CLR @#0 ;:CLR LOC. 0 JUST IN CASE

;*****
;*TEST 670 TEST FOR ODD ADDR. ERROR TRAP FOR DEST. DEFERRED MODES
;*****
TST670:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #670,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF1+1,R2 ;:DEST ADDR=MBUF1+1 (ODD)
MOV #4$,@#4 ;GO TO 4$ ON ODA TRAP

MOV R2,R5 ;:[R5] = DEST. ADDR
CCC ;SCOPE SYNC

2$: NEGB @ (R5)+ ;:TEST DM=3 TRAP

3$: ERROR 6 ;:ODA TRAP NOT SPRUNG

MOV #MBUF1+3,R5 ;:[R5] = DEST. ADDR
MOV @#20$,R1 ;:[R1] = TEST INSTR
CCC ;SCOPE SYNC

20$: NEGB @-(R5) ;:TEST DM=5 TRAP

5$: ERROR 6 ;:ODA TRAP NOT SPRUNG

MOV R2,R5 ;:[R5] = DEST ADDR
MOV @#21$,R1 ;:[R1] = TEST INSTR
CCC ;SCOPE SYNC

21$: NEGB @0(R5) ;:TEST DM=7 TRAP

7$: ERROR 6 ;:ODA TRAP NOT SPRUNG

MOV #BERR,@#4 ;:RESET T.O. VECTOR
BR TST671 ;:GO TO SCOPE EXIT

```

13182
13183 047760 062716 000002
13184 047764 000002
13185
13186
13187
13188
13189 047766
13190 047766 000004
13191 047770 012700 000671
13192 047774 013701 050016
13193 050000 012702 063317
13194 050004 012737 050066 000004
13195
13196 050012 010205
13197 050014 000257
13198
13199 050016 113504
13200
13201 050020 104006
13202
13203 050022 012705 063321
13204 050026 013701 050034
13205 050032 000257
13206
13207 050034 115504
13208
13209 050036 104006
13210 050040 010205
13211 050042 013701 050050
13212 050046 000257
13213
13214 050050 117504 000000
13215
13216 050054 104006
13217
13218 050056 012737 061220 000004
13219 050064 000403
13220
13221 050066 062716 000002
13222 050072 000002
13223
13224
13225
13226
13227 050074
13228 050074 000004
13229 050076 012700 000672
13230 050102 013701 050124
13231 050106 012702 050177
13232 050112 012737 050202 000004
13233
13234 050120 010205
13235 050122 000257
13236
13237 050124 000135

```
4$: ADD #2,(SP) ;MOV RETURN PC AROUND ERROR CALL
RTI ;RETURN TO NEXT SUB-TEST

;*****
;*TEST 671 TEST FOR ODD ADDR ERROR TRAP FOR SOURCE DEFERRED MODES
;*****
TST671:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #671,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF1+1,R2 ;[R2] = SOURCE ADDR. (ODD)
MOV #4$,@#4 ;GO TO 4$ ON TRAP

MOV R2,R5 ;[R5] = SOURCE ADDR.
CCC ;SCOPE SYNC

2$: MOVB @(R5)+,R4 ;TEST SM=3

3$: ERROR 6 ;ODA TRAP NOT SPRUNG

MOV #MBUF1+3,R5 ;[R5] = SOURCE ADDR
MOV @#20$,R1 ;[R1] = TEST INSTR
CCC ;SCOPE SYNC

20$: MOVB @-(R5),R4 ;TEST SM=5

5$: ERROR 6 ;ODA TRAP NOT SPRUNG
MOV R2,R5 ;[R5] = SOURCE ADDR
MOV @#21$,R1 ;[R1] = TEST INSTR
CCC ;SCOPE SYNC

21$: MOVB @0(R5),R4 ;TEST SM=7

7$: ERROR 6 ;ODA TRAP NOT SPRUNG

MOV #BERR,@#4 ;RESET T.O. VECTOR
BR TST672 ;:GO TO SCOPE EXIT

4$: ADD #2,(SP) ;MOVE RETURN PC AROUND ERROR CALL
RTI ;RETURN TO NEXT SUB-TEST

;*****
;*TEST 672 TEST FOR ODD ADDR ERROR TRAP FOR JMP DEST DEFERRED MODES
;*****
TST672:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #672,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #6$+3,R2 ;DEST ADDR = 6$+3 (ODD)
MOV #4$,@#4 ;GO TO 4$ ON ODA TRAP

MOV R2,R5 ;[R5] = DEST ADDR
CCC ;SCOPE SYNC

2$: JMP @(R5)+ ;TEST JMP DM=3
```

```

13238
13239 050126 104006 3$: ERROR 6 ;ODA TRAP NOT SPRUNG IN ROM LOC 153
13240
13241 050130 012705 050177 MOV #6$+3,R5 ;[R5] = DEST ADDR
13242 050134 013701 050142 MOV @#20$,R1 ;[R1] = TEST INSTR
13243 050140 000257 CCC ;SCOPE SYNC
13244
13245 050142 000155 20$: JMP @-(R5) ;TEST JMP DM=5
13246
13247 050144 104006 5$: ERROR 6 ;ODA TRAP NOT SPRUNG IN ROM LOC 155
13248
13249 050146 010205 MOV R2,R5 ;[R5] = DEST ADDR
13250 050150 013701 050156 MOV @#21$,R1 ;[R1] = TEST INSTR
13251 050154 000257 CCC ;SCOPE SYNC
13252
13253 050156 000175 000000 21$: JMP @0(R5) ;TEST JMP DM=7
13254
13255 050162 104006 7$: ERROR 6 ;ODA TRAP NOT SPRUNG
13256
13257 050164 012737 061220 000004 MOV #BERR,@#4 ;RESET BUS T.O. VECTOR
13258 050172 000420 BR TST673 ;GO TO SCOPE EXIT
13259
13260 050174 000000 6$: HALT ;CATASTOPHIC ERROR - [PC] QUESTIONABLE.
13261 050176 000000 HALT ;RESTART PROGRAM - DO NOT CONTINUE.
13262 050200 000000 HALT
13263
13264 050202 032716 000001 4$: BIT #1,(SP) ;TRAP DUE TO ODD PC?
13265 050206 001003 BNE 8$ ;BR IF YES
13266 050210 062716 000002 ADD #2,(SP) ;MOV RETURN PC AROUND ERROR CALL
13267 050214 000002 RTI ;RETURN TO NEXT SUB TEST
13268
13269 050216 011603 8$: MOV (SP),R3 ;GET ODD PC OFF STACK INTO R3
13270 050220 062706 000004 ADD #4,SP ;FIX SP
13271
13272 050224 104007 9$: ERROR 7 ;PC TRAPPED WITH ODD ADDRESS
13273
13274 050226 012737 061220 000004 MOV #BERR,@#4 ;RESET T.O. VECTOR
13275
13276
13277
13278
13279 050234 TST673:
13280 050234 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
13281 050236 012700 000673 MOV #673,R0 ;LOAD R0 WITH TEST NUMBER
13282 050242 013701 050266 MOV @#2$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13283 050246 012737 050400 000004 MOV #4$,@#4 ;GO TO 4$ ON OVFLW TRAP
13284 050254 010605 MOV SP,R5 ;SAVE SP
13285 050256 012702 000376 MOV #376,R2 ;USE R2 TO SET UP SP TO CAUSE TRAP
13286
13287 050262 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
13288 050264 000257 CCC ;SCOPE SYNC
13289
13290 050266 005016 2$: CLR (SP) ;TEST DM1 - SHOULD SPRING TRAP
13291
13292 050270 010637 001074 MOV SP,@#5REG5 ;SAVE BAD SP FOR PRINTING
13293 050274 010506 MCV R5,SP ;RESET SP
  
```



```

13294 050276 104006      3$:  ERROR 6          ;DM1 FAILED TO CAUSE OVERFLOW TRAP
13295
13296 050300 013701 050310      MOV @#20$,R1       ;[R1] = TEST INSTR.
13297 050304 010206      MOV R2,SP          ;SET UP SP TO CAUSE OVERFLOW
13298 050306 000257      CCC               ;SCOPE SYNC
13299
13300 050310 005026      20$: CLR (SP)+      ;TEST DM2 - SHOULD SPRING TRAP
13301
13302 050312 010637 001074      MOV SP,@#$REG5    ;SAVE BAD SP FOR PRINTING
13303 050316 010506      MOV R5,SP         ;RESET SP
13304 050320 104006      5$:  ERROR 6          ;DM2 FAILED TO CAUSE OVERFLOW TRAP
13305
13306 050322 013701 050332      MOV @#21$,R1       ;[R1] = TEST INSTR.
13307 050326 010206      MOV R2,SP          ;SET UP SP TO CAUSE OVERFLOW
13308 050330 000257      CCC               ;SCOPE SYNC
13309
13310 050332 005046      21$: CLR -(SP)      ;TEST DM4 - SHOULD SPRING TRAP
13311
13312 050334 010637 001074      MOV SP,@#$REG5    ;SAVE BAD SP FOR PRINTING
13313 050340 010506      MOV R5,SP         ;RESET SP
13314 050342 104006      7$:  ERROR 6          ;DM4 FAILED TO CAUSE OVERFLOW TRAP
13315
13316 050344 013701 050354      MOV @#22$,R1       ;[R1] = TEST INSTR.
13317 050350 010206      MOV R2,SP          ;SET SP TO CAUSE ERROR
13318 050352 000257      CCC               ;SCOPE SYNC
13319
13320 050354 005066 000000      22$: CLR 0(SP)     ;TEST DM6 - SHOULD SPRING TRAP
13321
13322 050360 010637 001074      MOV SP,@#$REG5    ;SAVE BAD SP FOR PRINTING
13323 050364 010506      MOV R5,SP         ;RESET SP
13324 050366 104006      9$:  ERROR 6          ;DM6 FAILED TO CAUSE OVERFLOW TRAP
13325
13326 050370 012737 061220 000004      MOV #BERR,@#4     ;RESET BUS T.O. VECTOR
13327 050376 000407      BR TST674         ;:GO TO SCOPE EXIT
13328
13329 050400 011604      4$:  MOV (SP),R4     ;GET RETURN PC OFF STACK
13330 050402 062704 000010      ADD #10,R4        ;MOVE RETURN PC AROUND ERROR CALL
13331 050406 010506      MOV R5,SP         ;RESET SP
13332 050410 005046      CLR -(SP)         ;PUSH NEW PS ON STACK
13333 050412 010446      MOV R4,-(SP)     ;PUSH RETURN PC ON STACK
13334 050414 000002      RTI              ;RETURN TO NEXT SUB-TEST
13335

```

```

:*****
:*TEST 674 TEST FOR STACK OVFLW FOR MOV DEST MODES 1,2,4, AND 6.
:*****
TST674:

```

```

13339 050416      SCOPE              ;CALL THE SCOPE LOOP UTILITY
13340 050416 000004      MOV #674,R0       ;:LOAD R0 WITH TEST NUMBER
13341 050420 012700 000674      MOV @#2$,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
13342 050424 013701 050450      MOV #4$,@#4 ;GO TO 4$ ON STACK OVFLW TRAP
13343 050430 012737 050562 000004      MOV SP,R5         ;SAVE SP
13344 050436 010605      MOV #376,R2       ;USE R2 TO SET UP SP TO CAUSE TRAP
13345 050440 012702 000376
13346
13347 050444 010206      MOV R2,SP         ;SET UP SP TO CAUSE OVERFLOW
13348 050446 000257      CCC               ;SCOPE SYNC
13349

```

```

13350 050450 010016          2$:  MOV      R0,(SP)          ;TEST MOV DM1 - SHOULD SPRING TRAP
13351
13352 050452 010637 001074    MOV      SP,@#5REG5        ;SAVE BAD SP FOR PRINTING
13353 050456 010506          MOV      R5,SP             ;RESET SP
13354 050460 104006          3$:  ERROR    6              ;MOV DM1 FAILED TO SPRING TRAP
13355
13356 050462 013701 050472    MOV      @#20$,R1          ;[R1] = TEST INSTR.
13357 050466 010206          MOV      R2,SP             ;SET UP SP TO CAUSE OVERFLOW
13358 050470 000257          CCC                       ;SCOPE SYNC
13359
13360 050472 010026          20$: MOV      R0,(SP)+       ;TEST MOV DM2 - SHOULD SPRING TRAP
13361
13362 050474 010637 001074    MOV      SP,@#5REG5        ;SAVE BAD SP FOR PRINTING
13363 050500 010506          MOV      R5,SP             ;RESET SP
13364 050502 104006          5$:  ERROR    6              ;MOV DM2 FAILED TO SPRING TRAP
13365
13366 050504 013701 050514    MOV      @#21$,R1          ;[R1] = TEST INSTR.
13367 050510 010206          MOV      R2,SP             ;SET UP SP TO CAUSE OVERFLOW
13368 050512 000257          CCC                       ;SCOPE SYNC
13369
13370 050514 010046          21$: MOV      R0,-(SP)       ;TEST MOV DM4 - SHOULD SPRING TRAP
13371
13372 050516 010637 001074    MOV      SP,@#5REG5        ;SAVE BAD SP FOR PRINTING
13373 050522 010506          MOV      R5,SP             ;RESET SP
13374 050524 104006          7$:  ERROR    6              ;MOV DM4 FAILED TO SPRING TRAP
13375
13376 050526 013701 050536    MOV      @#22$,R1          ;[R1] = TEST INSTR.
13377 050532 010206          MOV      R2,SP             ;SET UP SP TO CAUSE OVERFLOW
13378 050534 000257          CCC                       ;SCOPE SYNC
13379
13380 050536 010066 000000    22$: MOV      R0,0(SP)      ;TEST MOV DM6 - SHOULD SPRING TRAP
13381
13382 050542 010637 001074    MOV      SP,@#5REG5        ;SAVE BAD SP FOR PRINTING
13383 050546 010506          MOV      R5,SP             ;RESET SP
13384 050550 104006          9$:  ERROR    6              ;MOV DM6 FAILED TO CAUSE OVFLW TRAP
13385
13386 050552 012737 061220 000004 MOV      #BERR,@#4         ;RESET T.O. VECTOR
13387 050560 000407          BR       T675              ;GO TO SCOPE EXIT
13388
13389 050562 011604          4$:  MOV      (SP),R4        ;GET RETURN PC
13390 050564 062704 000010    ADD     #10,R4             ;MOVE RETURN PC AROUND ERROR CALL
13391 050570 010506          MOV      R5,SP             ;RESET SP
13392 050572 005046          CLR     -(SP)              ;PUSH NEW PSW
13393 050574 010446          MOV     R4,-(SP)          ;PUSH RETURN PC
13394 050576 000002          RTI                       ;RETURN TO NEXT SUB-TEST

```

```

:*****
:TEST 675      TEST THAT JSR CAN CAUSE OVERFLOW TRAP
:*****

```

```

13395
13396
13397
13398
13399 050600          T675:  SCOPE                ;CALL THE SCOPE LOOP UTILITY
13400 050600 000004          MOV      #675,R0           ;LOAD R0 WITH TEST NUMBER
13401 050602 012700 000675    MOV      @#2$,R1           ;LOAD R1 WITH TEST INSTRUCTION WORD
13402 050606 013701 050630    MOV      #4$,@#4          ;GO TO 4$ ON OVERFLOW ERROR
13403 050612 012737 050652 000004 MOV      SP,R5             ;SAVE SP
13404 050620 010605          MOV      #400,SP          ;SET THE SP TO CAUSE TRAP
13405 050622 012706 000400

```

```

13406 050626 000257          CCC          ;SCOPE SYNC
13407
13408 050630 004737 050656  2$:  JSR      PC,@#6$ ;TEST JSR - SHOULD SPRING TRAP
13409
13410 050634 010637 001074          MOV      SP,@#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
13411 050640 010506          MOV      R5,SP      ;RESET SP
13412 050642 104005  3$:  ERROR    5      ;JSR PUSH DID NOT SPRING OVFL TRAP
13413
13414 050644 000410          BR       8$        ;GO TO SCOPE EXIT
13415
13416 050646 010637 001074          MOV      SP,@#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
13417 050652 010506          MOV      R5,SP      ;RESET SP
13418 050654 000404          BR       8$        ;GO EXIT TEST - ALL OK
13419
13420 050656 010637 001074          MOV      SP,@#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
13421 050662 010506          MOV      R5,SP      ;RESET SP
13422 050664 104005  5$:  ERROR    5      ;JSR PUSH FAILED TO SPRING OVFLW TRAP
13423
13424 050666 012737 061220 000004 8$:  MOV      #BERR,@#4 ;RESET BUS T.O. VECTOR
13425
13426          ::*****
13427          :*TEST 676      TEST THAT 1ST PUSH IN TRAP MICROROUTINE CAUSES OVFLW TRAP
13428          :*****
13429          TST676:
13430          SCOPE          ;CALL THE SCOPE LOOP UTILITY
13431 050674 000004          MOV      #676,R0    ;:LOAD R0 WITH TEST NUMBER
13432 050676 012700 000676          MOV      @#2$,R1    ;:LOAD R1 WITH TEST INSTRUCTION WORD
13433 050702 013701 050736          MOV      @#14,R4    ;:SAVE BREAK POINT TRAP VECTOR
13434 050706 013704 000014          MOV      SP,R5      ;:SAVE SP
13435 050712 010605          MOV      #4$,@#4 ;GO TO 4$ ON OVFLW TRAP
13436 050714 012737 050752 000004          MOV      #6$,@#14 ;GO TO 6$ IF BPT SERVICED
13437 050722 012737 050756 000014          MOV      #400,SP    ;SET UP SP TO CAUSE OVFLW ON 1ST PUSH
13438 050730 012706 000400          MOV      #400,SP    ;SCOPE SYNC
13439 050734 000257          CCC
13440 050736 000003  2$:  BPT          ;TEST THE BPT - SHOULD CAUSE OVERFLOW TRAP
13441
13442 050740 010637 001074          MOV      SP,@#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
13443 050744 010506          MOV      R5,SP      ;RESET SP
13444 050746 104005  3$:  ERROR    5      ;BPT FAILED TO TRAP
13445
13446 050750 000406          BR       8$        ;GO TO SCOPE EXIT
13447
13448 050752 010506  4$:  MOV      R5,SP      ;RESET SP
13449 050754 000404          BR       8$        ;GO EXIT - ALL OK
13450
13451 050756 010637 001074          MOV      SP,@#$REG5 ;SAVE BAD SP FOR ERROR PRINTOUT
13452 050762 010506          MOV      R5,SP      ;RESET SP
13453 050764 104005  5$:  ERROR    5      ;OVFLW TRAP FAILED TO BUMP BPT SERVICE
13454
13455 050766 012737 061220 000004 8$:  MOV      #BERR,@#4 ;RESET VECTORS
13456 050774 010437 000014          MOV      R4,@#14
13457
13458          ::*****
13459          :*TEST 677      TEST THAT 2ND PUSH IN TRAP MICROUTINE CAUSES OVFLW TRAP
13460          :*****
13461 051000          TST677:

```

```
13462 051000 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
13463 051002 012700 000677    MOV      #677,R0    ;:LOAD R0 WITH TEST NUMBER
13464 051006 013701 051042    MOV      @#2$,R1    ;:LOAD R1 WITH TEST INSTRUCTION WORD
13465 051012 013704 000014    MOV      @#14,R4    ;:SAVE BPT VECTOR
13466 051016 010605          MOV      SP,R5      ;:SAVE SP
13467 051020 012737 051056 000004  MOV      #4$,@#4 ;GO TO 4$ ON STACK OVFLOW
13468 051026 012737 051062 000014  MOV      #6$,@#14 ;GO TO 6$ IF BPT SERVICED
13469 051034 012706 000402    MOV      #402,SP    ;:SET SP TO CAUSE TRAP ON 2ND PUSH
13470 051040 000257          CCC              ;:SCOPE SYNC
13471
13472 051042 000003          2$:  BPT              ;:TEST THE BPT - SHOULD CAUSE OVERFLOW TRAP
13473
13474 051044 010637 001074    MOV      SP,@#$REG5 ;:SAVE BAD SP FOR PRINTING
13475 051050 010506          MOV      R5,SP     ;:RESET SP
13476 051052 104005          3$:  ERROR 5        ;:BPT FAILED TO TRAP
13477
13478 051054 000406          BR      8$         ;:GO TO SCOPE EXIT
13479
13480 051056 010506          4$:  MOV      R5,SP ;:RESET SP
13481 051060 000404          BR      8$         ;:GO EXIT - ALL OK
13482
13483 051062 010637 001074    6$:  MOV      SP,@#$REG5 ;:SAVE BAD SP FOR PRINTING
13484 051066 010506          MOV      R5,SP     ;:RESET SP
13485 051070 104005          5$:  ERROR 5        ;:OVFLW TRAP FAILED TO BUMP BPT SERVICE
13486
13487 051072 012737 061220 000004  8$:  MOV      #BERR,@#4 ;:RESET VECTORS
13488 051100 010437 000014    MOV      R4,@#14
13489
13490          ;:*****
13491          ;:*TEST 700      ILLEGAL INSTRUCTION TEST - JSR RN,%R
13492          ;:*****
13493          TST700:
13494 051104 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
13495 051106 012700 000700    MOV      #700,R0    ;:LOAD R0 WITH TEST NUMBER
13496 051112 013701 051146    MOV      @#2$,R1    ;:LOAD R1 WITH TEST INSTRUCTION WORD
13497 051116 010605          MOV      SP,R5      ;:SAVE SP
13498 051120 010737 001010    MOV      PC,@#$LPERR ;:SET ERROR LOOP ADDRESS
13499 051124 013704 000004    1$:  MOV      @#4,R4    ;:SAVE T.O. VECTOR
13500 051130 012737 051156 000004  MOV      #4$,@#4 ;ILLEGAL INSTR. TRAP GOES TO 4$
13501 051136 010506          MOV      R5,SP     ;:RESET SP FOR ERROR LOOP
13502 051140 012702 051154    MOV      #3$,R2    ;:IN CASE JSR JUMPS TO [R2]
13503 051144 000257          CCC              ;:SCOPE SYNC
13504
13505 051146 004302          2$:  JSR      R3,R2    ;:JSR MODE 0 FORCES TRAP - GO TO 4$
13506
13507 051150 010437 000004    MOV      R4,@#4    ;:RESTORE T.O. VECTOR
13508 051154 104005          3$:  ERROR 5        ;:JSR FAILED TO SPRING TRAP
13509
13510 051156 010437 000004    4$:  MOV      R4,@#4 ;:RESTORE VECTOR
13511 051162 010506          MOV      R5,SP     ;:RESET SP
13512
13513          ;:*****
13514          ;:*TEST 701      ILLEGAL INSTRUCTION TEST - JMP %R
13515          ;:*****
13516 051164          TST701:
13517 051164 000004          SCOPE          ;CALL THE SCOPE LOOP UTILITY
```

```
13518 051166 012700 000701      MOV      #701,R0      ;;LOAD R0 WITH TEST NUMBER
13519 051172 013701 051226      MOV      @#2$,R1     ;;LOAD R1 WITH TEST INSTRUCTION WORD
13520 051176 010605                MOV      SP,R5       ;;SAVE SP
13521 051200 010737 001010      MOV      PC,@#5LPERR ;;SET ERROR LOOP ADDRESS
13522 051204 013704 000004      MOV      @#4,R4     ;;SAVE VECTOR POINTER AT LOC. 4
13523 051210 012737 051236 000004 1$:  MOV      #4$,@#4 ;ON TRAP - GO TO 4$
13524 051216 010506                MOV      R5,SP      ;;RESET SP FOR ERROR LOOP
13525 051220 012702 051234      MOV      #3$,R2     ;;IN CASE IT JUMPS TO ADDR IN RN
13526 051224 000257                CCC                ;;SCOPE SYNC
13527
13528 051226 000102                2$:  JMP      R2        ;;JMP MODE 0 FORCES TRAP - GO TO 4$
13529
13530 051230 010437 000004      MOV      R4,@#4     ;;RESTORE VECTOR POINTER AT LOC. 4
13531 051234 104005                3$:  ERROR   5        ;;ILLEGAL INSTR TRAP FAILED
13532
13533 051236 010437 000004      4$:  MOV      R4,@#4     ;;RESTORE VECTOR POINTER AT LOC. 4
13534 051242 010506                MOV      R5,SP      ;;RESET SP
13535
```

: *TEST 702 BUS TIMEOUT TRAP TEST - TST (R)
: *****

```
TST702:
13539 051244                SCOPE                ;;CALL THE SCOPE LOOP UTILITY
13540 051244 000004      MOV      #702,R0   ;;LOAD R0 WITH TEST NUMBER
13541 051246 012700 000702      MOV      @#2$,R1   ;;LOAD R1 WITH TEST INSTRUCTION WORD
13542 051252 013701 051306      MOV      SP,R5     ;;SAVE SP
13543 051256 010605                MOV      PC,@#5LPERR ;;SET ERROR LOOP ADDRESS
13544 051260 010737 001010      MOV      @#4,R4     ;;SAVE ORIGINAL T.O. VECTOR POINTER
13545 051264 013704 000004      MOV      #4$,@#4 ;ON T.O. TRAP - GO TO 4$
13546 051270 012737 051316 000004 1$:  MOV      #160000,R2 ;;ADDRESS CAUSES T.O.
13547 051276 012702 160000      MOV      R5,SP     ;;RESET SP FOR ERROR LOOP
13548 051302 010506                CCC                ;;SCOPE SYNC
13549 051304 000257
13550
13551 051306 005712                2$:  TST      (R2)    ;;FORCE T.O. TRAP - GO TO 4$
13552
13553 051310 010437 000004      MOV      R4,@#4     ;;RESTORE T.O. VECTOR
13554 051314 104005                3$:  ERROR   5        ;;TIMEOUT TRAP FAILED
13555 051316 010437 000004      4$:  MOV      R4,@#4     ;;RESTORE T.O. VECTOR
13556 051322 010506                MOV      R5,SP     ;;RESET SP
13557
```

: *TEST 703 "T" BIT TRAP TEST
: *****

```
TST703:
13561 051324                SCOPE                ;;CALL THE SCOPE LOOP UTILITY
13562 051324 000004      MOV      #703,R0   ;;LOAD R0 WITH TEST NUMBER
13563 051326 012700 000703      MOV      @#2$,R1   ;;LOAD R1 WITH TEST INSTRUCTION WORD
13564 051332 013701 051370      MOV      SP,R5     ;;SAVE SP
13565 051336 010605                MOV      PC,@#5LPERR ;;SET ERROR LOOP ADDRESS
13566 051340 010737 001010      MOV      R5,SP     ;;RESET SP FOR ERROR LOOP
13567 051344 010506                1$:  MOV      #4$,@#14  ;;GO TO 4$ WHEN "T" TRAP SPRUNG
13568 051346 012737 051376 000014  MOV      #20,-(SP)  ;;SET "T" BIT ON STACK
13569 051354 012746 000020      MOV      #2$,-(SP) ;;SET UP NEW PC ON STACK
13570 051360 012746 051370      CCC                ;;SCOPE SYNC
13571 051364 000257                RTT                ;;TURN ON "T" BIT - GO TO 2$
13572 051366 000006
13573
```

```

13574 051370 005700      2$:   TST     R0           ;SPRING "T" BIT TRAP - GO TO 4$
13575
13576 051372 104005      3$:   ERROR   5           ;NO "T" BIT TRAP OCCURRED
13577
13578 051374 000405           BR     6$           ;GO EXIT
13579
13580 051376 032766 000020 000002 4$:   BIT     #20,2(SP)      ;"T" BIT SET IN OLD PSW?
13581 051404 001001           BNE    6$           ;BR IF YES
13582
13583 051406 104001      5$:   ERROR   1           ;#T# BIT NOT SAVED ON STACK
13584
13585 051410 012737 000016 000014 6$:   MOV     #16,@#14      ;RESTORE "T" BIT TRAP CATCHER
13586 051416 005037 000016           CLR     @#16
13587 051422 010506           MOV     R5,SP        ;RESET SP
13588
13589
13590
13591
13592
13593
13594 051424
13595 051424 000004
13596 051426 012700 000704
13597 051432 013701 051456
13598 051436 010605
13599 051440 013704 000004
13600 051444 012737 051500 000004
13601 051452 005006
13602 051454 000257
13603
13604 051456 012746 007777      2$:   MOV     #7777,-(SP)  ;ATTEMPT PUSH INTO PSW - SHOULD CAUSE
13605                                     ;"RED ZONE" TRAP TO BE SPRUNG
13606
13607 051462 010437 000004           MOV     R4,@#4       ;RESTORE BUS ERROR VECTOR
13608 051466 005004           CLR     R4           ;[R4] = S / B SP
13609 051470 010603           MOV     SP,R3        ;[R3] = WAS SP
13610 051472 010506           MOV     R5,SP        ;RESET THE SP
13611 051474 104003      3$:   ERROR   3           ;TRAP NOT SPRUNG
13612 051476 000414           BR     TST705        ;GO TO SCOPE EXIT - SCHOOL'S OUT
13613
13614 051500 022706 000000      4$:   CMP     #0,SP        ;WAS IT A RED ZONE TRAP ?
13615 051504 001406           BEQ    6$           ;BR IF YES
13616
13617 051506 010437 000004           MOV     R4,@#4       ;RESTORE BUS ERROR VECTOR
13618 051512 005004           CLR     R4           ;[R4]= S / B SP
13619 051514 010603           MOV     SP,R3        ;[R3] = WAS SP
13620 051516 010506           MOV     R5,SP        ;RESET THE SP
13621 051520 104003      5$:   ERROR   3           ;TRAP SPRUNG BUT NOT RED ZONE
13622
13623 051522 010506      6$:   MOV     R5,SP        ;FIX UP THE SP
13624 051524 010437 000004           MOV     R4,@#4       ;RESTORE BERR VECTOR
13625
13626
13627
13628
13629 051530

```

 :*TEST 704 TEST PUSH INTO PSW WITH [SP] = 000000
 :THESE NEXT TWO TESTS VERIFY THAT A "RED ZONE" TRAP OCCURS IF A
 :PUSH IS ATTEMPTED WITH THE [SP] INITIALLY EQUAL TO 000000,177572.
 :*****
 TST704:

SCOPE
 MOV #704,R0 ;CALL THE SCOPE LOOP UTILITY
 MOV @#2\$,R1 ;LOAD R0 WITH TEST NUMBER
 MOV SP,R5 ;LOAD R1 WITH TEST INSTRUCTION WORD
 MOV @#4,R4 ;SAVE THE SP
 MOV #4\$,@#4 ;SAVE THE BUS ERROR VECTOR
 CLR SP ;"RED ZONE" TRAP GOES TO 4\$
 CCC ;MAKE SP = 000000
 ;SCOPE SYNC

 :*TEST 705 TEST PUSH INTO SR WITH [SP] = 177572
 :*****
 TST705:

```

13630 051530 000004          SCOPE          :CALL THE SCOPE LOOP UTILITY
13631 051532 012700 000705  MOV      #705,R0          :;LOAD R0 WITH TEST NUMBER
13632 051536 013701 051564  MOV      @#2$,R1         :LOAD R1 WITH TEST INSTRUCTION WORD
13633 051542 010605          MOV      SP,R5          :SAVE THE SP
13634 051544 013704 000004  MOV      @#4,R4          :SAVE THE BUS ERROR VECTOR
13635 051550 012737 051606 000004  MOV      #4$,@#4 ;"RED ZONE" TRAP GOES TO 4$
13636 051556 012706 177572          MOV      #177572,SP     :MAKE SP=177572
13637 051562 000257          CCC                   :SCOPE SYNC
13638
13639 051564 012746 177777  2$:      MOV      #-1,-(SP) :ATTEMPT PUSH INTO SR - SHOULD CAUSE
13640                                     :;"RED ZONE" TRAP TO BE SPRUNG
13641
13642 051570 010437 000004          MOV      R4,@#4         :RESTORE BUS ERROR VECTOR
13643 051574 005004          CLR      R4             :[R4] = S / B SP
13644 051576 010603          MOV      SP,R3         :[R3] = WAS SP
13645 051600 010506          MOV      R5,SP        :RESET THE SP
13646 051602 104003  3$:      ERROR    3           :TRAP NOT SPRUNG
13647 051604 000414          BR       TST706        :;GO TO SCOPE EXIT - SCHOOL'S OUT
13648
13649 051606 022706 000000  4$:      CMP      #0,SP       :WAS IT A RED ZONE TRAP ?
13650 051612 001406          BEQ     6$             :BR IF YES
13651
13652 051614 010437 000004          MOV      R4,@#4         :RESTORE BUS ERROR VECTOR
13653 051620 005004          CLR      R4             :[R4]= S / B SP
13654 051622 010603          MOV      SP,R3         :[R3] = WAS SP
13655 051624 010506          MOV      R5,SP        :RESET THE SP
13656 051626 104003  5$:      ERROR    3           :TRAP SPRUNG BUT NOT RED ZONE
13657
13658 051630 010506  6$:      MOV      R5,SP       :FIX UP THE SP
13659 051632 010437 000004  MOV      R4,@#4         :RESTORE BUS ERROR VECTOR
13660
13661
13662
13663
13664 051636          :;*****
13665          :*TEST 706      TEST PUSH INTO SLR WITH [SP] = 177776
13666          :;*****
13667          TST706:
13668          SCOPE          :CALL THE SCOPE LOOP UTILITY
13669 051636 000004          MOV      #706,R0          :;LOAD R0 WITH TEST NUMBER
13670 051640 012700 000706  MOV      @#2$,R1         :LOAD R1 WITH TEST INSTRUCTION WORD
13671 051644 013701 051672  MOV      SP,R5          :SAVE THE SP
13672 051650 010605          MOV      @#4,R4          :SAVE THE BUS ERROR VECTOR
13673 051652 013704 000004  MOV      #4$,@#4 ;"RED ZONE" TRAP GOES TO 4$
13674 051656 012737 051714 000004  MOV      #177776,SP     :MAKE SP=177776
13675 051664 012706 177776          CCC                   :SCOPE SYNC
13676 051670 000257
13677 051672 012746 000200  2$:      MOV      #200,-(SP) :ATTEMPT PUSH INTO SLR - SHOULD CAUSE
13678                                     :;"RED ZONE" TRAP TO BE SPRUNG
13679
13680 051676 010437 000004          MOV      R4,@#4         :RESTORE BUS ERROR VECTOR
13681 051702 005004          CLR      R4             :[R4] = S / B SP
13682 051704 010603          MOV      SP,R3         :[R3] = WAS SP
13683 051706 010506          MOV      R5,SP        :RESET THE SP
13684 051710 104003  3$:      ERROR    3           :TRAP NOT SPRUNG
13685 051712 000414          BR       TST707        :;GO TO SCOPE EXIT - SCHOOL'S OUT
13686
13687 051714 022706 000000  4$:      CMP      #0,SP       :WAS IT A RED ZONE TRAP ?
13688 051720 001406          BEQ     6$             :BR IF YES

```

```
13686
13687 051722 010437 000004      MOV     R4,@#4      ;RESTORE BUS ERROR VECTOR
13688 051726 005004              CLR     R4          ;[R4]= S / B SP
13689 051730 010603              MOV     SP,R3      ;[R3] = WAS SP
13690 051732 010506              MOV     R5,SP      ;RESET THE SP
13691 051734 104003      5$:    ERROR    3      ;TRAP SPRUNG BUT NOT RED ZONE
13692
13693 051736 010506      6$:    MOV     R5,SP      ;FIX UP THE SP
13694 051740 010437 000004      MOV     R4,@#4      ;RESTORE BUS ERROR VECTOR
13695
13696      ;*****
13697      ;*TEST 707      RSVD INSTRUCTION TEST - 000007 THRU 000077
13698      ;*****
13699 051744      TST707:
13700 051744 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
13701 051746 012700 000707      MOV     #707,R0    ;:LOAD R0 WITH TEST NUMBER
13702 051752 010605      5$:    MOV     SP,R5      ;SAVE THE SP
13703 051754 012737 052012 000010      MOV     #4$,@#10   ;SET UP RSVD INSTR. TRAP VECTOR
13704 051762 005037 000012      CLR     @#12
13705 051766 012701 000007      MOV     #7,R1      ;SET UP FIRST ONE IN GROUP
13706 051772 010737 001010      MOV     PC,@#SLPERR ;ONLY LOOP ON BAD OP CODE
13707 051776 010506      1$:    MOV     R5,SP      ;RESET SP FOR ERROR LOOP AND NEW INSTR
13708 052000 010137 052006      MOV     R1,@#2$ ;LOAD NEW INSTR
13709 052004 000257      CCC          ;SCOPE SYNC
13710
13711 052006 000007      2$:    000007      ;TEST THE RSVD INSTR - THIS LOCATION
13712      ;GETS CHANGED EACH PASS THROUGH
13713
13714 052010 104005      3$:    ERROR    5      ;RSVD INSTR. IN R1 FAILED TO TRAP
13715
13716 052012 005201      4$:    INC     R1          ;GENERATE NEW RSVD INSTR
13717 052014 022701 000100      CMP     #100,R1    ;AT END OF THIS GROUP ??
13718 052020 001366      BNE    1$          ;BR IF NOT
13719
13720 052022 010506      MOV     R5,SP      ;MAKE SURE TO RESET THE SP
13721 052024 012737 051752 001010      MOV     #5$,@#SLPERR ;LOOP FROM BEGINNING ON ERROR
13722      ;*****
13723      ;*TEST 710      RSVD INSTRUCTION TEST - 000210 THRU 000237
13724      ;*****
13725 052032      TST710:
13726 052032 000004      SCOPE      ;CALL THE SCOPE LOOP UTILITY
13727 052034 012700 000710      MOV     #710,R0    ;:LOAD R0 WITH TEST NUMBER
13728 052040 010605      5$:    MOV     SP,R5      ;SAVE THE SP
13729 052042 012737 052100 000010      MOV     #4$,@#10   ;SET UP RSVD INSTR. TRAP VECTOR
13730 052050 005037 000012      CLR     @#12
13731 052054 012701 000210      MOV     #210,R1    ;SET UP FIRST ONE IN GROUP
13732 052060 010737 001010      MOV     PC,@#SLPERR ;SET ERROR LOOP ADDRESS
13733 052064 010506      1$:    MOV     R5,SP      ;RESET SP FOR ERROR LOOP AND NEW INSTR
13734 052066 010137 052074      MOV     R1,@#2$ ;LOAD NEW INSTR
13735 052072 000257      CCC          ;SCOPE SYNC
13736
13737 052074 000210      2$:    000210      ;TEST THE RSVD INSTR - THIS LOCATION
13738      ;GETS CHANGED EACH PASS THROUGH
13739
13740 052076 104005      3$:    ERROR    5      ;RSVD INSTR. IN R1 FAILED TO TRAP
13741
```



```

13742 052100 005201          4$:  INC    R1          ;GENERATE NEW RSVD INSTR
13743 052102 022701 000240    CMP    #240,R1       ;AT END OF THIS GROUP ??
13744 052106 001366          BNE    1$           ;BR IF NOT
13745
13746 052110 010506          MOV    R5,SP        ;MAKE SURE TO RESET THE SP
13747 052112 012737 052040 001010  MOV    #5$,@#SLPERR ;LOOP FROM BEGINNING ON ERROR
13748
13749
13750
13751
13752 052120          ::*****
13753 052120 000004          :*TEST 711  RSVD INSTRUCTION TEST - 007000 THRU 007777
13754 052122 012700 000711    :*****
13755 052126 010605          TST711:
13756 052130 012737 052166 000010  SCOPE          ;CALL THE SCOPE LOOP UTILITY
13757 052136 005037 000012    MOV    #711,R0     ;:LOAD R0 WITH TEST NUMBER
13758 052142 012701 007000    5$:  MOV    SP,R5      ;SAVE THE SP
13759 052146 010737 001010    MOV    #4$,@#10   ;SET UP RSVD INSTR. TRAP VECTOR
13760 052152 010506          CLR    @#12
13761 052154 010137 052162    MOV    #7000,R1   ;SET UP FIRST ONE IN GROUP
13762 052160 000257          MOV    PC,@#SLPERR ;SET ERROR LOOP ADDRESS
13763
13764 052162 007000          1$:  MOV    R5,SP      ;RESET SP FOR ERROR LOOP AND NEW INSTR
13765
13766
13767 052164 104005          MOV    R1,@#2$ ;LOAD NEW INSTR
13768
13769 052166 005201          CCC          ;SCOPE SYNC
13770 052170 022701 010000    2$:  007000       ;TEST THE RSVD INSTR - THIS LOCATION
13771 052174 001366          ;GETS CHANGED EACH PASS THROUGH
13772
13773 052176 010506          3$:  ERROR    5     ;RSVD INSTR. IN R1 FAILED TO TRAP
13774 052200 012737 052126 001010  4$:  INC    R1          ;GENERATE NEW RSVD INSTR
13775
13776
13777
13778
13779 052206          :*TEST 712  RSVD INSTRUCTION TEST - 075000 THRU 076777
13780 052206 000004          :*****
13781 052210 012700 000712    TST712:
13782 052214 010605          SCOPE          ;CALL THE SCOPE LOOP UTILITY
13783 052216 012737 052256 000010  MOV    #712,R0     ;:LOAD R0 WITH TEST NUMBER
13784 052224 005037 000012    5$:  MOV    SP,R5      ;SAVE THE SP
13785 052230 012701 075000    MOV    #4$,@#10   ;SET UP RSVD INSTR. TRAP VECTOR
13786 052234 010737 001010    CLR    @#12
13787 052240 010506          MOV    #75000,R1  ;SET UP FIRST ONE IN GROUP
13788 052242 010137 052250    MOV    PC,@#SLPERR ;SET ERROR LOOP ADDRESS
13789 052246 000257          1$:  MOV    R5,SP      ;RESET SP FOR ERROR LOOP AND NEW INSTR
13790
13791 052250 075000          MOV    R1,@#2$ ;LOAD NEW INSTR
13792
13793
13794 052252 000240          CCC          ;SCOPE SYNC
13795 052254 104005          2$:  75000       ;TEST THE RSVD INSTR - THIS LOCATION
13796
13797 052256 005201          ;GETS CHANGED EACH PASS THROUGH
13798
13799
13800
13801
13802
13803
13804
13805
13806
13807
13808
13809
13810
13811
13812
13813
13814
13815
13816
13817
13818
13819
13820
13821
13822
13823
13824
13825
13826
13827
13828
13829
13830
13831
13832
13833
13834
13835
13836
13837
13838
13839
13840
13841
13842
13843
13844
13845
13846
13847
13848
13849
13850
13851
13852
13853
13854
13855
13856
13857
13858
13859
13860
13861
13862
13863
13864
13865
13866
13867
13868
13869
13870
13871
13872
13873
13874
13875
13876
13877
13878
13879
13880
13881
13882
13883
13884
13885
13886
13887
13888
13889
13890
13891
13892
13893
13894
13895
13896
13897
13898
13899
13900
13901
13902
13903
13904
13905
13906
13907
13908
13909
13910
13911
13912
13913
13914
13915
13916
13917
13918
13919
13920
13921
13922
13923
13924
13925
13926
13927
13928
13929
13930
13931
13932
13933
13934
13935
13936
13937
13938
13939
13940
13941
13942
13943
13944
13945
13946
13947
13948
13949
13950
13951
13952
13953
13954
13955
13956
13957
13958
13959
13960
13961
13962
13963
13964
13965
13966
13967
13968
13969
13970
13971
13972
13973
13974
13975
13976
13977
13978
13979
13980
13981
13982
13983
13984
13985
13986
13987
13988
13989
13990
13991
13992
13993
13994
13995
13996
13997
13998
13999

```

13798 052260 022701 076600
13799 052264 001774
13800 052266 022701 077000
13801 052272 001362
13802
13803 052274 010506
13804 052276 012737 052214 001010
13805
13806
13807
13808
13809 052304
13810 052304 000004
13811 052306 012700 000713
13812 052312 010605
13813 052314 012737 052352 000010
13814 052322 005037 000012
13815 052326 012701 106400
13816 052332 010737 001010
13817 052336 010506
13818 052340 010137 052346
13819 052344 000257

```
CMP #MED,R1 ;MED INSTRUCTION?  
BEQ 4$ ;BR IF YES--SKIP IT.  
CMP #077000,R1 ;AT END OF THIS GROUP ??  
BNE 1$ ;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  
5$: MOV SP,R5 ;SAVE THE SP  
MOV #4$,@#10 ;SET UP RSVD INSTR. TRAP VECTOR  
CLR @#12  
MOV #106400,R1 ;SET UP FIRST ONE IN GROUP  
MOV PC,@#SLPERR ;SET ERROR LOOP ADDRESS  
1$: MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR  
MOV R1,@#2$ ;LOAD NEW INSTR  
CCC ;SCOPE SYNC
```

```

13820
13821 052346 106400 2$: 106400 ;TEST THE RSVD INSTR - THIS LOCATION
13822 ;GETS CHANGED EACH PASS THROUGH
13823
13824 052350 104005 3$: ERROR 5 ;RSVD INSTR. IN R1 FAILED TO TRAP
13825
13826 052352 005201 4$: INC R1 ;GENERATE NEW RSVD INSTR
13827 052354 022701 106500 CMP #106500,R1 ;MFPD INSTRUCTION ??
13828 052360 001002 BNE 10$ ;BR IF NOT
13829 052362 012701 106700 MOV #106700,R1 ;SKIP MFPD AND MTPD INSTRUCTIONS
13830 052366 022701 110000 10$: CMP #110000,R1 ;AT END OF THIS GROUP ??
13831 052372 001361 BNE 1$ ;BR IF NOT
13832
13833 052374 010506 MOV R5,SP ;MAKE SURE TO RESET THE SP
13834 052376 012737 052312 001010 MOV #5,@#$LPERR ;LOOP FROM BEGINNING ON ERROR
13835 052404 012737 061122 000010 MOV #RSERR,@#10 ;RESTORE RSVD INSTR VECTOR
13836 052412 012737 000340 000012 MOV #340,@#12
13837 052420 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
13838
13839 ;THIS NEXT GROUP OF SEQUENTIAL TESTS VERIFIES THAT A "T" BIT
13840 ;TRAP CAN BE SERVICED IN EACH MICROWORD THAT DOES A "BUT SERVICE"
13841 ;EACH ROUTINE ENTERS THE TRAP MICROUTINE WHEN THE TRAP IS SPRUNG
13842
13843 052422 012737 061070 000014 TSET: MOV #TBSER,@#14 ;SET UP THE "T" BIT TRAP VECTOR
13844 052430 012737 000340 000016 MOV #340,@#16 ;PRIORITY 7
13845
13846 ;*****
13847 ;*TEST 714 BUT SERVICE -- ONE WORD INSTRUCTIONS--ALL MODES -- FROM TABLE
13848 ;"INSTAB" (INSTRUCTION TABLE) CONTAINS ALL ONE WORD INSTRUCTIONS
13849 ;THAT TEST A "BUT SERVICE" IN A UNIQUE ROM LOCATION. THE TABLE MUST
13850 ;BE TERMINATED WITH A 0 ENTRY.
13851 ;*****
13852 052436 TST714:
13853 052436 012700 000714 MOV #714,R0 ;:LOAD R0 WITH TEST NUMBER
13854 052442 010605 6$: MOV SP,R5 ;:SAVE THE SP
13855 052444 012704 063636 MOV #INSTAB,R4 ;:PUT POINTER TO TABLE IN R4
13856 052450 012401 4$: MOV (R4)+,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
13857 052452 001422 BEQ 5$ ;:EXIT TEST IF END OF TABLE
13858 052454 010737 001010 MOV PC,@#$LPERR ;:LOOP ON FAILING INSTRUCTION ONLY
13859 052460 010137 052512 1$: MOV R1,@#2$ ;:STORE TEST INSTRUCTION TO BE EXECUTED
13860 052464 012702 063312 MOV #MBUF0,R2 ;:IN CASE DM1 DEST--(R2)
13861 052470 012703 063316 MOV #MBUF1,R3 ;:IN CASE SM1--(R3)
13862 052474 010506 MOV R5,SP ;:RESTORE SP FOR ERROR LOOPING
13863 052476 012746 000020 MOV #20,-(SP) ;:SET "T" BIT IN THE NEW PSW
13864 052502 012746 052512 MOV #2$,-(SP) ;:MAKE NEW PC = 2$
13865 052506 000257 CCC ;:SCOPE SYNC
13866 052510 000006 RTT ;:SET "T" BIT - GO TO 2$
13867
13868 052512 000240 2$: NOP ;:INSTRUCTION FROM TABLE IS STORED HERE AND
13869 ;:SHOULD SPRING TRAP
13870
13871 052514 104005 3$: ERROR 5 ;:BUT SERVICE FAILED
13872
13873 052516 000754 BR 4$ ;:GET NEXT INSTRUCTION FOR BUT SERVICE TEST
13874 052520 012737 052442 001010 5$: MOV #6$,@#$LPERR ;:LOOP FROM BEGINNING ON ERROR
13875

```

13876
13877
13878
13879 052526
13880 052526 000004
13881 052530 012700 000715
13882 052534 013701 052552
13883 052540 012746 000020
13884 052544 012746 052554
13885 052550 000257
13886
13887 052552 000002
13888
13889 052554 104005
13890
13891
13892
13893
13894 052556
13895 052556 000004
13896 052560 012700 000716
13897 052564 013701 052634
13898
13899 052570 032737 040000 063234
13900 052576 001401
13901 052600 000000
13902 052602 010605
13903 052604 010737 001010
13904 052610 010506
13905 052612 012737 052640 063316
13906 052620 012746 000020
13907 052624 012746 052634
13908 052630 000257
13909 052632 000006
13910
13911 052634 004777 010456
13912
13913 052640 104005
13914
13915 052642 010506
13916
13917
13918
13919 052644
13920 052644 000004
13921 052646 012700 000717
13922 052652 013701 052672
13923 052656 012746 000020
13924 052662 012746 052672
13925 052666 000257
13926 052670 000006
13927
13928 052672 000167 000000
13929
13930 052676 104005
13931

```

*****
:*TEST 715      BUT SERVICE TEST - (RTI)
*****
TST715:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #715,R0     ;:LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1    ;:LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #20,-(SP)  ;:SET 'T' BIT IN THE NEW PSW
      MOV      #3$,-(SP) ;:MAKE NEW PC = 3$
      CCC                ;SCOPE SYNC
2$:   RTI                  ;INSTRUCTION SHOULD SPRING TRAP
3$:   ERROR 5              ;BUT SERVICE IN XXX FAILED

*****
:*TEST 716      BUT SERVICE TEST - (JSR %R,@A)
*****
TST716:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #716,R0     ;:LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1    ;:LOAD R1 WITH TEST INSTRUCTION WORD
      .SBTTL USER CONTROLLED BREAKPOINT -- BIT14
      BIT      #BIT14,@#BPTLOC ;BREAKPOINT HALT SET ??
      BEQ      .+4         ;BR IF NOT
      HALT                    ;BREAK-DEPRESS CONTINUE TO CONTINUE
      MOV      SP,R5        ;SAVE THE SP
      MOV      PC,@#SLPERR ;:FOR PROPER SP RESETTING ON ERROR LOOP
1$:   MOV      R5,SP        ;RESTORE SP FOR ERROR LOOPING
      MOV      #3$,@#MBUF1 ;:SET UP POINTER--DEST ADDR = 3$ FOR JSR
      MOV      #20,-(SP)   ;:SET 'T' BIT IN THE NEW PSW
      MOV      #2$,-(SP)   ;:MAKE NEW PC = 2$
      CCC                ;SCOPE SYNC
      <TT                ;SET 'T' BIT - GO TO 2$
2$:   JSR      PC,@#MBUF1  ;INSTRUCTION SHOULD SPRING TRAP
3$:   ERROR 5              ;BUT SERVICE IN XXX FAILED
      MOV      R5,SP        ;RESTORE SP IF ALL OK OR NOT LOOPING

*****
:*TEST 717      BUT SERVICE TEST - (JMP A)
*****
TST717:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #717,R0     ;:LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1    ;:LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #20,-(SP)  ;:SET 'T' BIT IN THE NEW PSW
      MOV      #2$,-(SP)  ;:MAKE NEW PC = 2$
      CCC                ;SCOPE SYNC
      RTI                  ;SET 'T' BIT - GO TO 2$
2$:   JMP      3$          ;JMP INSTRUCTION SHOULD SPRING TRAP
3$:   ERROR 5              ;BUT SERVICE IN XXX FAILED

```

13932
 13933
 13934
 13935 052700
 13936 052700 000004
 13937 052702 012700 000720
 13938 052706 013701 052734
 13939 052712 012737 052740 063312
 13940 052720 012746 000020
 13941 052724 012746 052734
 13942 052730 000257
 13943 052732 000006
 13944
 13945 052734 000177 010352
 13946
 13947 052740 104005
 13948
 13949
 13950
 13951
 13952 052742
 13953 052742 000004
 13954 052744 012700 000721
 13955 052750 013701 053004
 13956 052754 010605
 13957 052756 010737 001010
 13958 052762 010506
 13959 052764 012746 053006
 13960 052770 012746 000020
 13961 052774 012746 053004
 13962 053000 000257
 13963 053002 000006
 13964
 13965 053004 000207
 13966
 13967 053006 104005
 13968
 13969
 13970
 13971
 13972
 13973
 13974
 13975
 13976
 13977
 13978
 13979
 13980
 13981
 13982
 13983
 13984
 13985
 13986
 13987

```

:*****
:*TEST 720      BUT SERVICE TEST - (JMP @A)
:*****
TST720:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #720,R0      ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      #3$,@#M$BUFO ;SET UP POINTER--DEST ADDR = 3$ FOR JMP
      MOV      #20,-(SP)    ;SET "T" BIT IN THE NEW PSW
      MOV      #2$,-(SP)    ;MAKE NEW PC = 2$
      CCC
      RTT                  ;SCOPE SYNC
                          ;SET "T" BIT - GO TO 2$
2$:   JMP      @M$BUFO      ;JMP INSTRUCTION SHOULD SPRING TRAP
3$:   ERROR   5              ;BUT SERVICE IN XXX FAILED

:*****
:*TEST 721      BUT SERVICE TEST - (RTS PC)
:*****
TST721:
      SCOPE                ;CALL THE SCOPE LOOP UTILITY
      MOV      #721,R0      ;;LOAD R0 WITH TEST NUMBER
      MOV      @#2$,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
      MOV      SP,R5        ;SAVE THE SP
      MOV      PC,@#5$LPERR ;FOR PROPER SP RESETTING ON ERROR LOOP
1$:   MOV      R5,SP        ;RESTORE SP FOR ERROR LOOPING
      MOV      #3$,-(SP)    ;RTS WILL LOAD PC WITH 3$
      MOV      #20,-(SP)    ;SET "T" BIT IN THE NEW PSW
      MOV      #2$,-(SP)    ;MAKE NEW PC = 2$
      CCC
      RTT                  ;SCOPE SYNC
                          ;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:

```

13988
 13989
 13990
 13991
 13992
 13993
 13994
 13995
 13996
 13997
 13998
 13999
 14000
 14001 053010
 14002 053010 000004
 14003 053012 012700 000722
 14004 053016 012705 063340
 14005 053022 010737 001010
 14006 053026 024545
 14007
 14008 053030 005725
 14009 053032 022705 063416
 14010 053036 001413
 14011 053040 012501
 14012 053042 012503
 14013 053044 000257
 14014
 14015 053046 060103
 14016
 14017 053050 021503
 14018 053052 001766
 14019
 14020 053054 011504
 14021 053056 014502
 14022 053060 104010
 14023
 14024 053062 005725
 14025 053064 000761
 14026
 14027 053066 012737 053016 001010
 14028
 14029
 14030
 14031
 14032
 14033
 14034
 14035
 14036
 14037
 14038
 14039
 14040
 14041
 14042
 14043

```

: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          ;CALL THE SCOPE LOOP UTILITY
MOV #722,R0    ;:LOAD R0 WITH TEST NUMBER
1$: MOV #ALUADD+4,R5 ;R5 POINTS TO TABLE OF NO.S
MOV PC,@#SLPERR ;:LOOP ONLY ON FAILING PAIR OF #'S
CMP -(R5),-(R5) ;:RESET R5 TO POINT TO BAD GUYS
                ;:(OR NULL ENTRY FIRST TIME THROUGH)
4$: TST (R5)+    ;POINT TO A SRC OP
CMP #ALUADD+62,R5 ;:DONE ALL NO.C 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

MOV (R5),R4    ;:GET S / B SUM
MOV -(R5),R2   ;:GET DEST OP
3$: ERROR 10    ;:ALU ADD OPERATION FAILED

TST (R5)+      ;:CORRECT R5 POINTER
BR 4$         ;:GO DO NEXT PAIR

5$: MOV #1$,@#SLPERR ;: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

14044
 14045
 14046
 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

```

:      1      1      1
;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          ;CALL THE SCOPE LOOP UTILITY
MOV      #723,R0        ;:LOAD R0 WITH TEST NUMBER
1$:      MOV      #ALUSUB+4,R5 ;:R5 POINTS TO TABLE OF NO.S
          MOV      PC,@#SLPERR ;:LOOP ONLY ON FAILING PAIR OF #'S
          CMP      -(R5),-(R5) ;:RESET R5 TO POINT TO BAD GUYS
          ;(OR NULL ENTRY FIRST TIME THROUGH)
4$:      TST      (R5)+        ;:POINT TO A SRC OP
          CMP      #ALUSUB+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$:      SUB      R1,R3        ;:TEST THE SUB FUNCTION
          CMP      (R5),R3     ;:CORRECT DIFF. ?
          BEQ      4$          ;:GO SUB NEXT PAIR IF YES
          MOV      (R5),R4     ;:GET S / B DIFF
          MOV      -(R5),R2    ;:GET DEST OP
3$:      ERROR   10          ;:ALU SUB OPERATION FAILED
          TST      (R5)+        ;:CORRECT R5 POINTER
          BR      4$          ;:GO DO NEXT PAIR
5$:      MOV      #1$,@#SLPERR ;: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

```

14100	:000000	177777	177777
14101	:177777	000000	000000
14102	:125252	125252	000000
14103	:052525	052525	000000
14104	:125252	052525	052525
14105	:052525	125252	125252

: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

14120

14121	053160		
14122	053160	000004	
14123	053162	012700	000724
14124	053166	012705	063420
14125	053172	010737	001010
14126	053176	024545	
14127			
14128	053200	005725	
14129	053202	022705	063476
14130	053206	001413	
14131	053210	012501	
14132	053212	012503	
14133	053214	000257	
14134			
14135	053216	040103	
14136			
14137	053220	020315	
14138	053222	001766	
14139			
14140	053224	011504	
14141	053226	014502	
14142	053230	104010	
14143			
14144	053232	005725	
14145	053234	000761	
14146			
14147	053236	012737	053166 001010
14148			
14149			
14150			
14151			
14152			
14153			
14154			
14155			

```

TST1724:
          SCOPE                ;CALL THE SCOPE LOOP UTILITY
1$:      MOV      #724,R0      ;:LOAD R0 WITH TEST NUMBER
          MOV      #ANDTAB+4,R5 ;:R5 POINTS TO TABLE OF TEST NO.S
          MOV      PC,@#SLPERR ;:LOOP ONLY ON FAILING PAIR OF #'S
          CMP      -(R5),-(R5) ;:RESET R5 TO POINT TO BAD GUYS
          ;(OR NULL ENTRY FIRST TIME THROUGH)
4$:      TST      (R5)+        ;:POINT TO A SOURCE OPR
          CMP      #ANDTAB+62,R5 ;:DONE ALL COMBINATIONS ?
          BEQ      5$          ;:BR IF YES
          MOV      (R5)+,R1     ;:LOAD THE SRC OP
          MOV      (R5)+,R3     ;:LOAD THE DEST OP
          CCC                          ;:SCOPE SYNC
2$:      BIC      R1,R3        ;:TEST THE "AND"
          ;
          CMP      R3,(R5)     ;:RESULT CORRECT ?
          BEQ      4$          ;:BR IF YES - GET THE NEXT PAIR
          ;
          MOV      (R5),R4     ;:GET THE S / B DATA
          MOV      -(R5),R2    ;:GET DEST OP
3$:      ERROR   10           ;:ALU "AND" FAILED
          ;
          TST      (R5)+        ;:CORRECT R5 POINTER
          BR       4$          ;:GO GET NEXT PAIR
5$:      MOV      #1$,@#SLPERR ;: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:

	:SOURCE OP	DEST. OP	RESULT
14156			
14157			
14158	:000000	000000	000000
14159	:177777	177777	177777
14160	:000000	177777	177777
14161	:177777	000000	177777
14162	:125252	125252	125252
14163	:052525	052525	052525
14164	:125252	052525	177777
14165	:052525	125252	177777
14166			
14167			
14168			
14169			
14170			
14171			
14172			
14173			
14174			
14175			
14176			
14177			
14178			
14179			
14180			

:THE 8 PAIRS OF NO.S AND THE ANSWERS ARE STORED IN A TABLE TAGGED
:"ORTAB" IN THE FOLLOWING PATTERN:

:ORTAB: 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

14181	053244			
14182	053244	000004		
14183	053246	012700	000725	
14184	053252	012705	063500	
14185	053256	010737	001010	
14186	053262	024545		
14187				
14188	053264	005725		
14189	053266	022705	063556	
14190	053272	001413		
14191	053274	012501		
14192	053276	012503		
14193	053300	000257		
14194				
14195	053302	050103		
14196				
14197	053304	020315		
14198	053306	001766		
14199				
14200	053310	011504		
14201	053312	014502		
14202	053314	104010		
14203				
14204	053316	005725		
14205	053320	000761		
14206				
14207	053322	012737	053252	001010
14208				
14209				
14210				
14211				

```
TST725:
SCOPE                                     ;CALL THE SCOPE LOOP UTILITY
MOV #725,R0                               ;:LOAD R0 WITH TEST NUMBER
1$: MOV #ORTAB+4,R5                       ;:R5 POINTS TO TABLE OF TEST NO.S
MOV PC,@#SLPERR                           ;:LOOP ONLY ON FAILING PAIR OF #'S
CMP -(R5),-(R5)                           ;:RESET R5 TO POINT TO BAD GUYS
                                           ;:(OR NULL ENTRY FIRST TIME THROUGH)
4$: TST (R5)+                             ;:POINT TO A SOURCE OPR
CMP #ORTAB+62,R5                          ;:DONE ALL COMBINATIONS ?
BEQ 5$                                     ;:BR IF YES
MOV (R5)+,R1                              ;:LOAD THE SRC OP
MOV (R5)+,R3                              ;:LOAD THE DEST OP
CCC                                         ;:SCOPE SYNC
2$: BIS R1,R3                             ;:TEST THE "OR"
                                           ;:RESULT CORRECT ?
CMP R3,(R5)                               ;:BR IF YES - GET THE NEXT PAIR
BEQ 4$
                                           ;:GET THE S / B DATA
MOV (R5),R4                               ;:GET DEST OP
MOV -(R5),R2                              ;:ALU "OR" FAILED
3$: ERROR 10
                                           ;:CORRECT R5 POINTER
TST (R5)+                                  ;:GO GET NEXT PAIR
BR 4$
5$: MOV #1$,@#SLPERR                       ;:LOOP FROM BEGINNING ON ERROR
```

:*TEST 726 INC / DEC / ADD TEST - CYCLE NO.S 000000-077777
:THIS TEST COMBINES THE INC / DEC / ADD INSTRUCTIONS IN THE FOLLOWING

14212
14213
14214
14215
14216
14217
14218
14219
14220
14221
14222
14223
14224
14225
14226
14227
14228
14229
14230
14231
14232
14233
14234
14235
14236
14237
14238
14239
14240
14241
14242
14243
14244
14245
14246
14247
14248
14249
14250
14251
14252
14253
14254
14255
14256
14257
14258
14259
14260
14261
14262
14263
14264
14265
14266
14267

053330
053330 000004
053332 012700 000726
053336 005001
053340 005002
053342 005004
053344 010737 001010
053350 010203
053352 000257

053354 060103

053356 020403
053360 001402

053362 104010

053364 000407

053366 005201
053370 100402
053372 005302
053374 000765

053376 012737 053336 001010

;TEST SEQUENCE:

- :1. BOTH SOURCE AND DEST OPS ARE ZEROED
- :2. THE TWO NO.S ARE ADDED AND THE RESULT COMPARED WITH 000000
- :3. THE SOURCE OP IS INCREMENTED
- :4. THE DEST OP IS DECREMENTED
- :5. STEPS 2,3, AND 4 ARE REPEATED UNTIL THE SOURCE OP GOES
: NEGATIVE

;ON DETECTION OF A NON-ZERO RESULT THE ERROR IS REPORTED AND THEN IF:

- : 1. SW09=0 THE TEST IS EXITED
- : 2. SW09=1 THE ROUTINE LOCKS ON THE FAILING PAIR OF OPERANDS

TST726:

```
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #726,R0 ;:LOAD R0 WITH TEST NUMBER
10$: CLR R1 ;INITIALIZE REGS TO 000000
CLR R2
CLR R4
MOV PC,@#SLPERR ;LOOP ONLY ON FAILING PAIR OF #'S
1$: MOV R2,R3 ;LOAD DEST OPERAND
CCC ;SCOPE SYNC

2$: ADD R1,R3 ;ADD THE TWO TEST NO.S
;RESULT S / B = 000000

CMP R4,R3 ;RESULT = 000000 ?
BEQ 4$ ;BR IF YES

3$: ERROR 10 ;INCORRECT RESULT IN R3

BR TST727 ;:EXIT TO NEXT TEST

4$: INC R1 ;ADD 1 TO SOURCE OP
BMI 5$ ;GET OUT IF IT WENT NEGATIVE
DEC R2 ;SUB 1 FROM THE DEST OP
BR 1$ ;GO ADD THE TWO NO.S

5$: MOV #10,@#SLPERR ;LOOP FROM BEGINNING ON ERROR
```

*TEST 727 INC / DEC / ADD TEST - CYCLE NO.S 077777-000000
:THIS TEST COMBINES THE INC / DEC / ADD INSTRUCTIONS IN THE FOLLOWING
:TEST SEQUENCE:

- :1. BOTH SOURCE AND DEST OPS ARE ZEROED
- :2. THE TWO NO.S ARE ADDED AND THE RESULT COMPARED WITH 000000
- :3. THE SOURCE OP IS DECREMENTED
- :4. THE DEST OP IS INCREMENTED
- :5. STEPS 2,3, AND 4 ARE REPEATED UNTIL THE DEST. OP GOES
: NEGATIVE

;ON DETECTION OF A NON-ZERO RESULT THE ERROR IS REPORTED AND THEN IF:

14268
 14269
 14270
 14271 053404
 14272 053404 000004
 14273 053406 012700 000727
 14274 053412 005001
 14275 053414 005002
 14276 053416 005004
 14277 053420 010737 001010
 14278 053424 010203
 14279 053426 000257
 14280
 14281 053430 060103
 14282
 14283
 14284 053432 020403
 14285 053434 001402
 14286
 14287 053436 104010
 14288
 14289 053440 000407
 14290
 14291 053442 005202
 14292 053444 100402
 14293 053446 005301
 14294 053450 000765
 14295
 14296 053452 012737 053412 001010
 14297
 14298
 14299
 14300
 14301 053460
 14302 053460 000004
 14303 053462 012700 000730
 14304 053466 013737 053516 001076
 14305 053474 005001
 14306 053476 012704 000006
 14307 053502 012702 000002
 14308 053506 005003
 14309 053510 012705 000003
 14310 053514 000277
 14311
 14312 053516 070205
 14313
 14314 053520 100403
 14315 053522 001402
 14316 053524 102401
 14317 053526 103001
 14318
 14319 053530 104044
 14320
 14321 053532 020304
 14322 053534 001002
 14323 053536 020102

```

:      1. SW09=0 THE TEST IS EXITED
:      2. SW09=1 THE ROUTINE LOCKS ON THE FAILING PAIR OF OPERANDS
:.....
TST727:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #727,R0                      ;;LOAD R0 WITH TEST NUMBER
10$:    CLR      R1                      ;INITIALIZE REGS TO 000000
        CLR      R2
        CLR      R4
MOV      PC,@#$LPERR                  ;LOOP ONLY ON FAILING PAIR OF #'S
1$:     MOV      R2,R3                  ;LOAD DEST OPERAND
        CCC
        ;SCOPE SYNC
2$:     ADD      R1,R3                  ;ADD THE TWO TEST NO.S
        ;RESULT S / B = 000000
        ;
        CMP      R4,R3                  ;RESULT = 000000 ?
        BEQ      4$                      ;BR IF YES
3$:     ERROR    10                      ;INCORRECT RESULT IN R3
        BR      TST730                  ;;GO TO SCOPE EXIT
4$:     INC      R2                      ;ADD 1 TO DEST. OP
        BMI      5$                      ;GET OUT IF IT WENT NEGATIVE
        DEC      R1                      ;SUB 1 FROM THE SOURCE OP
        BR      1$                      ;GO ADD THE TWO NO.S
5$:     MOV      #10$,@#$LPERR          ;LOOP FROM BEGINNING ON ERROR
:.....
;*TEST 730      MUL RA,RB TEST ; N:C = 1111
:.....
TST730:
SCOPE                                ;CALL THE SCOPE LOOP UTILITY
MOV      #730,R0                      ;;LOAD R0 WITH TEST NUMBER
MOV      @#2$,@#$TMP0                 ;GET TEST INSTRUCTION WORD
CLR      R1                              ;S/B RESULT IN R2
MOV      #6,R4                          ;S/B RESULT IN R3
MOV      #2,R2                          ;INITIALIZE REG
CLR      R3                              ;INITIALIZE REG + 1
MOV      #3,R5                          ;INITIALIZE SRC
        SCC
        ;SCOPE SYNC
2$:     MUL      R5,R2                  ;TEST THE MUL
        ;N:C=0000?
        BMI      3$
        BEQ      3$
        BVS      3$
        BCC      4$
3$:     ERROR    44                      ;COND CODES SET IMPROPERLY
4$:     CMP      R3,R4                  ;REG+1 CORRECT?
        BNE      5$                      ;BR IF NOT
        CMP      R1,R2                  ;REG CORRECT?
  
```

14324 053540 001401
14325
14326 053542 104045
14327
14328
14329
14330
14331 053544
14332 053544 000004
14333 053546 012700 000731
14334 053552 013737 053606 001076
14335 053560 005001
14336 053562 012704 123450
14337 053566 012702 012345
14338 053572 005003
14339 053574 012705 063312
14340 053600 012715 000010
14341 053604 000257
14342
14343 053606 070215
14344
14345 053610 100403
14346 053612 001402
14347 053614 102401
14348 053616 103401
14349
14350 053620 104044
14351
14352 053622 020304
14353 053624 001002
14354 053626 020102
14355 053630 001401
14356
14357 053632 104045
14358
14359
14360
14361
14362 053634
14363 053634 000004
14364 053636 012700 000732
14365 053642 013737 053674 001076
14366 053650 005001
14367 053652 005004
14368 053654 005002
14369 053656 012703 177777
14370 053662 012705 063312
14371 053666 012715 000010
14372 053672 000257
14373
14374 053674 070225
14375
14376 053676 100403
14377 053700 001002
14378 053702 102401
14379 053704 103001

BEQ TST731 ;:BR IF YES
5\$: ERROR 45 ;MUL DELIVERED WRONG RESULT
:*****
:*TEST 731 MUL (RA),RB TEST ; N:C = 0000-SET C
:*****
TST731:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #731,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,@#\$TMP0 ;GET TEST INSTRUCTION WORD
CLR R1 ;S/B RESULT IN R2
MOV #123450,R4 ;S/B RESULT IN R3
MOV #012345,R2 ;INITIALIZE REG
CLR R3 ;INITIALIZE REG + 1
MOV #MBUFO,R5 ;SET UP POINTER TO SRC
MOV #10,(R5) ;INITIALIZE SRC
CCC ;SCOPE SYNC
2\$: MUL (R5),R2 ;TEST THE MUL
BMI 3\$;N:C=0001?
BEQ 3\$
BVS 3\$
BCS 4\$
3\$: ERROR 44 ;COND CODES SET IMPROPERLY
4\$: CMP R3,R4 ;REG+1 CORRECT?
BNE 5\$;BR IF NOT
CMP R1,R2 ;REG CORRECT?
BEQ TST732 ;:BR IF YES
5\$: ERROR 45 ;MUL DELIVERED WRONG RESULT
:*****
:*TEST 732 MUL (RA)+,RB TEST ; N:C = 0000-SET Z
:*****
TST732:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #732,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,@#\$TMP0 ;GET TEST INSTRUCTION WORD
CLR R1 ;S/B RESULT IN R2
CLR R4 ;S/B RESULT IN R3
CLR R2 ;INITIALIZE REG
MOV #-1,R3 ;INITIALIZE REG + 1
MOV #MBUFO,R5 ;SET UP POINTER TO SRC
MOV #10,(R5) ;INITIALIZE SRC
CCC ;SCOPE SYNC
2\$: MUL (R5)+,R2 ;TEST THE MUL
BMI 3\$;N:C=0100?
BNE 3\$
BVS 3\$
BCC 4\$

```

14380
14381 053706 104044      3$:  ERROR  44      ;COND CODES SET IMPROPERLY
14382
14383 053710 020304      4$:  CMP     R3,R4      ;REG+1 CORRECT?
14384 053712 001002      BNE     5$           ;BR IF NOT
14385 053714 020102      CMP     R1,R2      ;REG CORRECT?
14386 053716 001401      BEQ     6$           ;BR IF YES
14387
14388 053720 104045      5$:  ERROR  45      ;MUL DELIVERED WRONG RESULT
14389
14390 053722 022705 063314 6$:  CMP     #MBUF0+2,R5 ;DID R5 GET AUTO-INCREMENTED?
14391 053726 001401      BEQ     TST733      ;:BR IF YES
14392
14393 053730 104046      ERROR  46      ;AUTO INCREMENT DID NOT OCCUR
14394
14395
14396
14397
14398 053732
14399 053732 000004
14400 053734 012700 000733      SCOPE           ;CALL THE SCOPE LOOP UTILITY
14401 053740 013737 054000 001076  MOV     #733,R0      ;:LOAD R0 WITH TEST NUMBER
14402 053746 012701 177777      MOV     @#2$,@#STMPO ;GET TEST INSTRUCTION WORD
14403 053752 012704 177770      MOV     #-1,R1      ;S/B RESULT IN R2
14404 053756 012702 000001      MOV     #-10,R4     ;S/B RESULT IN R3
14405 053762 005003      MOV     #1,R2       ;INITIALIZE REG
14406 053764 012705 063306      CLR     R3          ;INITIALIZE REG + 1
14407 053770 012737 177770 063312  MOV     #ATA+10,R5   ;SET UP POINTER TO POINTER TO MBUF0
14408 053776 000257      MOV     #-10,@#MBUF0 ;INITIALIZE SRC
14409
14410 054000 070235      CCC           ;SCOPE SYNC
14411
14412 054002 100003      2$:  MUL     @ (R5)+,R2 ;TEST THE MUL
14413 054004 001402      BPL     3$           ;N:C=1000?
14414 054006 102401      BEQ     3$
14415 054010 103001      BVS     3$
14416
14417 054012 104044      BCC     4$
14418
14419 054014 020304      3$:  ERROR  44      ;COND CODES SET IMPROPERLY
14420 054016 001002      4$:  CMP     R3,R4      ;REG+1 CORRECT?
14421 054020 020102      BNE     5$           ;BR IF NOT
14422 054022 001401      CMP     R1,R2      ;REG CORRECT?
14423
14424 054024 104045      BEQ     6$           ;BR IF YES
14425
14426 054026 022705 063310 5$:  ERROR  45      ;MUL DELIVERED WRONG RESULT
14427 054032 001401      6$:  CMP     #ATA+12,R5 ;DID R5 GET AUTO-INCREMENTED?
14428
14429 054034 104046      BEQ     TST734      ;:BR IF YES
14430
14431
14432
14433
14434 054036
14435 054036 000004      ERROR  46      ;AUTO INCREMENT DID NOT OCCUR
14436
14437
14438
14439
14440
14441
14442
14443
14444
14445
14446
14447
14448
14449
14450
14451
14452
14453
14454
14455
14456
14457
14458
14459
14460
14461
14462
14463
14464
14465
14466
14467
14468
14469
14470
14471
14472
14473
14474
14475
14476
14477
14478
14479
14480
14481
14482
14483
14484
14485
14486
14487
14488
14489
14490
14491
14492
14493
14494
14495
14496
14497
14498
14499
14500
14501
14502
14503
14504
14505
14506
14507
14508
14509
14510
14511
14512
14513
14514
14515
14516
14517
14518
14519
14520
14521
14522
14523
14524
14525
14526
14527
14528
14529
14530
14531
14532
14533
14534
14535
14536
14537
14538
14539
14540
14541
14542
14543
14544
14545
14546
14547
14548
14549
14550
14551
14552
14553
14554
14555
14556
14557
14558
14559
14560
14561
14562
14563
14564
14565
14566
14567
14568
14569
14570
14571
14572
14573
14574
14575
14576
14577
14578
14579
14580
14581
14582
14583
14584
14585
14586
14587
14588
14589
14590
14591
14592
14593
14594
14595
14596
14597
14598
14599
14600
14601
14602
14603
14604
14605
14606
14607
14608
14609
14610
14611
14612
14613
14614
14615
14616
14617
14618
14619
14620
14621
14622
14623
14624
14625
14626
14627
14628
14629
14630
14631
14632
14633
14634
14635
14636
14637
14638
14639
14640
14641
14642
14643
14644
14645
14646
14647
14648
14649
14650
14651
14652
14653
14654
14655
14656
14657
14658
14659
14660
14661
14662
14663
14664
14665
14666
14667
14668
14669
14670
14671
14672
14673
14674
14675
14676
14677
14678
14679
14680
14681
14682
14683
14684
14685
14686
14687
14688
14689
14690
14691
14692
14693
14694
14695
14696
14697
14698
14699
14700
14701
14702
14703
14704
14705
14706
14707
14708
14709
14710
14711
14712
14713
14714
14715
14716
14717
14718
14719
14720
14721
14722
14723
14724
14725
14726
14727
14728
14729
14730
14731
14732
14733
14734
14735
14736
14737
14738
14739
14740
14741
14742
14743
14744
14745
14746
14747
14748
14749
14750
14751
14752
14753
14754
14755
14756
14757
14758
14759
14760
14761
14762
14763
14764
14765
14766
14767
14768
14769
14770
14771
14772
14773
14774
14775
14776
14777
14778
14779
14780
14781
14782
14783
14784
14785
14786
14787
14788
14789
14790
14791
14792
14793
14794
14795
14796
14797
14798
14799
14800
14801
14802
14803
14804
14805
14806
14807
14808
14809
14810
14811
14812
14813
14814
14815
14816
14817
14818
14819
14820
14821
14822
14823
14824
14825
14826
14827
14828
14829
14830
14831
14832
14833
14834
14835
14836
14837
14838
14839
14840
14841
14842
14843
14844
14845
14846
14847
14848
14849
14850
14851
14852
14853
14854
14855
14856
14857
14858
14859
14860
14861
14862
14863
14864
14865
14866
14867
14868
14869
14870
14871
14872
14873
14874
14875
14876
14877
14878
14879
14880
14881
14882
14883
14884
14885
14886
14887
14888
14889
14890
14891
14892
14893
14894
14895
14896
14897
14898
14899
14900
14901
14902
14903
14904
14905
14906
14907
14908
14909
14910
14911
14912
14913
14914
14915
14916
14917
14918
14919
14920
14921
14922
14923
14924
14925
14926
14927
14928
14929
14930
14931
14932
14933
14934
14935
14936
14937
14938
14939
14940
14941
14942
14943
14944
14945
14946
14947
14948
14949
14950
14951
14952
14953
14954
14955
14956
14957
14958
14959
14960
14961
14962
14963
14964
14965
14966
14967
14968
14969
14970
14971
14972
14973
14974
14975
14976
14977
14978
14979
14980
14981
14982
14983
14984
14985
14986
14987
14988
14989
14990
14991
14992
14993
14994
14995
14996
14997
14998
14999
15000
15001
15002
15003
15004
15005
15006
15007
15008
15009
15010
15011
15012
15013
15014
15015
15016
15017
15018
15019
15020
15021
15022
15023
15024
15025
15026
15027
15028
15029
15030
15031
15032
15033
15034
15035
15036
15037
15038
15039
15040
15041
15042
15043
15044
15045
15046
15047
15048
15049
15050
15051
15052
15053
15054
15055
15056
15057
15058
15059
15060
15061
15062
15063
15064
15065
15066
15067
15068
15069
15070
15071
15072
15073
15074
15075
15076
15077
15078
15079
15080
15081
15082
15083
15084
15085
15086
15087
15088
15089
15090
15091
15092
15093
15094
15095
15096
15097
15098
15099
15100
15101
15102
15103
15104
15105
15106
15107
15108
15109
15110
15111
15112
15113
15114
15115
15116
15117
15118
15119
15120
15121
15122
15123
15124
15125
15126
15127
15128
15129
15130
15131
15132
15133
15134
15135
15136
15137
15138
15139
15140
15141
15142
15143
15144
15145
15146
15147
15148
15149
15150
15151
15152
15153
15154
15155
15156
15157
15158
15159
15160
15161
15162
15163
15164
15165
15166
15167
15168
15169
15170
15171
15172
15173
15174
15175
15176
15177
15178
15179
15180
15181
15182
15183
15184
15185
15186
15187
15188
15189
15190
15191
15192
15193
15194
15195
15196
15197
15198
15199
15200
15201
15202
15203
15204
15205
15206
15207
15208
15209
15210
15211
15212
15213
15214
15215
15216
15217
15218
15219
15220
15221
15222
15223
15224
15225
15226
15227
15228
15229
15230
15231
15232
15233
15234
15235
15236
15237
15238
15239
15240
15241
15242
15243
15244
15245
15246
15247
15248
15249
15250
15251
15252
15253
15254
15255
15256
15257
15258
15259
15260
15261
15262
15263
15264
15265
15266
15267
15268
15269
15270
15271
15272
15273
15274
15275
15276
15277
15278
15279
15280
15281
15282
15283
15284
15285
15286
15287
15288
15289
15290
15291
15292
15293
15294
15295
15296
15297
15298
15299
15300
15301
15302
15303
15304
15305
15306
15307
15308
15309
15310
15311
15312
15313
15314
15315
15316
15317
15318
15319
15320
15321
15322
15323
15324
15325
15326
15327
15328
15329
15330
15331
15332
15333
15334
15335
15336
15337
15338
15339
15340
15341
15342
15343
15344
15345
15346
15347
15348
15349
15350
15351
15352
15353
15354
15355
15356
15357
15358
15359
15360
15361
15362
15363
15364
15365
15366
15367
15368
15369
15370
15371
15372
15373
15374
15375
15376
15377
15378
15379
15380
15381
15382
15383
15384
15385
15386
15387
15388
15389
15390
15391
15392
15393
15394
15395
15396
15397
15398
15399
15400
15401
15402
15403
15404
15405
15406
15407
15408
15409
15410
15411
15412
15413
15414
15415
15416
15417
15418
15419
15420
15421
15422
15423
15424
15425
15426
15427
15428
15429
15430
15431
15432
15433
15434
15435
15436
15437
15438
15439
15440
15441
15442
15443
15444
15445
15446
15447
15448
15449
15450
15451
15452
15453
15454
15455
15456
15457
15458
15459
15460
15461
15462
15463
15464
15465
15466
15467
15468
15469
15470
15471
15472
15473
15474
15475
15476
15477
15478
15479
15480
15481
15482
15483
15484
15485
15486
15487
15488
15489
15490
15491
15492
15493
15494
15495
15496
15497
15498
15499
15500
15501
15502
15503
15504
15505
15506
15507
15508
15509
15510
15511
15512
15513
15514
15515
15516
15517
15518
15519
15520
15521
15522
15523
15524
15525
15526
15527
15528
15529
15530
15531
15532
15533
15534
15535
15536
15537
15538
15539
15540
15541
15542
15543
15544
15545
15546
15547
15548
15549
15550
15551
15552
15553
15554
15555
15556
15557
15558
15559
15560
15561
15562
15563
15564
15565
15566
15567
15568
15569
15570
15571
15572
15573
15574
15575
15576
15577
15578
15579
15580
15581
15582
15583
15584
15585
15586
15587
15588
15589
15590
15591
15592
15593
15594
15595
15596
15597
15598
15599
15600
15601
15602
15603
15604
15605
15606
15607
15608
15609
15610
15611
15612
15613
15614
15615
15616
15617
15618
15619
15620
15621
15622
15623
15624
15625
15626
15627
15628
15629
15630
15631
15632
15633
15634
15635
15636
15637
15638
15639
15640
15641
15642
15643
15644
15645
15646
15647
15648
15649
15650
15651
15652
15653
15654
15655
15656
15657
15658
15659
15660
15661
15662
15663
15664
15665
15666
15667
15668
15669
15670
15671
15672
15673
15674
15675
15676
15677
15678
15679
15680
15681
15682
15683
15684
15685
15686
15687
15688
15689
15690
15691
15692
15693
15694
15695
15696
15697
15698
15699
15700
15701
15702
15703
15704
15705
15706
15707
15708
15709
15710
15711
15712
15713
15714
15715
15716
15717
15718
15719
15720
15721
15722
15723
15724
15725
15726
15727
15728
15729
15730
15731
15732
15733
15734
15735
15736
15737
15738
15739
15740
15741
15742
15743
15744
15745
15746
15747
15748
15749
15750
15751
15752
15753
15754
15755
15756
15757
15758
15759
15760
15761
15762
15763
15764
15765
15766
15767
15768
15769
15770
15771
15772
15773
15774
15775
15776
15777
15778
15779
15780
15781
15782
15783
15784
15785
15786
15787
15788
15789
15790
15791
15792
15793
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
15832
15833
15834
15835
15836
15837
15838
15839
15840
15841
15842
15843
15844
15845
15846
15847
15848
15849
15850
15851
15852
15853
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
15888
15889
15890
15891
15892
15893
15894
15895
15896
15897
15898
15899
15900
15901
15902
15903
15904
15905
15906
15907
15908
15909
15910
15911
15912
15913
15914
15915
15916
15917
15918
15919
15920
15921
15922
15923
15924
15925
15926
15927
15928
15929
15930
15931
15932
15933
15934
15935
15936
15937
15938
15939
15940
15941
15942
15943
15944
15945
15946
15947
15948
15949
15950
15951
15952
15953
15954
15955
15956
15957
15958
15959
15960
15961
15962
15963
15964
15965
15966
15967
15968
15969
15970
15971
15972
15973
15974
15975
15976
15977
15978
15979
15980
15981
15982
15983
15984
15985
15986
15987
15988
15989
15990
15991
15992
15993
15994
15995
15996
15997
15998
15999
16000
16001
16002
16003
16004
16005
16006
16007
16008
16009
16010
16011
16012
16013
16014
16015
16016
16017
16018
16019
16020
16021
16022
16023
16024
16025
16026
16027
16028
16029
16030
16031
16032
16033
16034
16035
16036
16037
16038
16039
16040
16041
16042
16043
16044
16045
16046
16047
16048
16049
16050
16051
16052
16053
16054
16055
16056
16057
16058
16059
16060
16061
16062
16063
16064
16065
16066
16067
16068
16069
16070
16071
16072
16073
16074
16075
16076
16077
16078
16079
16080
16081
16082
16083
16084
16085
16086
16087
16088
16089
16090
16091
16092
16093
16094
16095
16096
16097
16098
16099
16100
16101
16102
16103
16104
16105
16106
16107
16108
16109
16110
16111
16112
16113
16114
16115
16116
16117
16118
16119
16120
16121
16122
16123
16124
16125
16126
16127
16128
16129
16130
16131
16132
16133
16134
16135
16136
16137
16138
16139
16140
16141
16142
16143
16144
16145
16146
16147
16148
16149
16150
16151
16152
16153
16154
16155
16156
16157
16158
16159
16160
16161
16162
16163
16164
16165
16166
16167
16168
16169
16170
16171
16172
16173
16174
16175
16176
16177
16178
16179
16180
16181
16182
16183
16184
16185
16186
16187
16188
16189
16190
16191
16192
16193
16194
16195
16196
16197
16198
16199
16200
16201
16202
16203
16204
16205
16206
16207
16208
16209
16210
16211
16212
16213
16214
16215
16216
16217
16218
16219
16220
16221
16222
16223
1
```

```
14436 054040 012700 000734      MOV      #734,R0      ;;LOAD R0 WITH TEST NUMBER
14437 054044 013737 054104 001076  MOV      @#2$,@#$TMPO  ;GET TEST INSTRUCTION WORD
14438 054052 012701 177777      MOV      #-1,R1      ;S/B RESULT IN R2
14439 054056 012704 177770      MOV      #-10,R4     ;S/B RESULT IN R3
14440 054062 012702 177777      MOV      #-1,R2     ;INITIALIZE REG
14441 054066 005003      CLR      R3         ;INITIALIZE REG + 1
14442 054070 012705 063314      MOV      #MBUF0+2,R5 ;SET UP POINTER TO SRC
14443 054074 012737 000010 063312  MOV      #10,@#MBUF0 ;INITIALIZE SRC
14444 054102 000277      SCC                    ;SCOPE SYNC
14445
14446 054104 070245      2$:      MUL      -(R5),R2  ;TEST THE MUL
14447
14448 054106 100003      BPL      3$         ;N:C=1000?
14449 054110 001402      BEQ      3$
14450 054112 102401      BVS      3$
14451 054114 103001      BCC      4$
14452
14453 054116 104044      3$:      ERROR    44      ;COND CODES SET IMPROPERLY
14454
14455 054120 020304      4$:      CMP      R3,R4     ;REG+1 CORRECT?
14456 054122 001002      BNE      5$         ;BR IF NOT
14457 054124 020102      CMP      R1,R2     ;REG CORRECT?
14458 054126 001401      BEQ      6$         ;BR IF YES
14459
14460 054130 104045      5$:      ERROR    45      ;MUL DELIVERED WRONG RESULT
14461
14462 054132 022705 063312      6$:      CMP      #MBUF0,R5 ;DID SRC REG GET AUTO-DECREMENTED?
14463 054136 001401      BEQ      TST735    ;;BR IF YES
14464
14465 054140 104046      ERROR    46      ;AUTO DECREMENT DID NOT OCCUR
14466
14467
14468
14469
14470
14471
14472
14473
14474
14475
14476
14477
14478
14479
14480
14481
14482
14483
14484
14485
14486
14487
14488
14489
14490
14491
```

*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      @#2$,@#$TMPO ;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
SCC                    ;SCOPE SYNC
2$:      MUL      @-(R5),R2  ;TEST THE MUL
BMI      3$         ;N:C=0001?
BEQ      3$
BVS      3$
BCS      4$
3$:      ERROR    44      ;COND CODES SET IMPROPERLY
4$:      CMP      R3,R4     ;REG+1 CORRECT?
```

14492 054226 001002
14493 054230 020102
14494 054232 001401
14495
14496 054234 104045
14497
14498 054236 022705 063306
14499 054242 001401
14500
14501 054244 104046
14502

BNE 5\$;BR IF NOT
CMP R1,R2 ;REG CORRECT?
BEQ 6\$;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

14503
14504
14505

: *TEST 736 MUL X(RA),RB TEST ; N:C = 1111 TO 0100

14506 054246
14507 054246 000004
14508 054250 012700 000736
14509 054254 013737 054310 001076
14510 054262 005001
14511 054264 005004
14512 054266 012702 012345
14513 054272 012703 177777
14514 054276 012705 063312
14515 054302 005065 000002
14516 054306 000277

TST736:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #736,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,@#\$TMP0 ;GET TEST INSTRUCTION WORD
CLR R1 ;S/B RESULT IN R2
CLR R4 ;S/B RESULT IN R3
MOV #012345,R2 ;INITIALIZE REG
MOV #-1,R3 ;INITIALIZE REG + 1
MOV #MBUFO,R5 ;SET UP POINTER TO SRC
CLR 2(R5) ;INITIALIZE SRC
SCC ;SCOPE SYNC

14517
14518 054310 070265 000002
14519
14520 054314 100403
14521 054316 001002
14522 054320 102401
14523 054322 103001
14524

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

14525 054324 104044
14526
14527 054326 020304
14528 054330 001002
14529 054332 020102
14530 054334 001401
14531

3\$: ERROR 44 ;COND CODES SET IMPROPERLY
4\$: CMP R3,R4 ;REG+1 CORRECT?
BNE 5\$;BR IF NOT
CMP R1,R2 ;REG CORRECT?
BEQ TST737 ;:BR IF YES

14532 054336 104045
14533
14534

5\$: ERROR 45 ;MUL DELIVERED WRONG RESULT

14535
14536

: *TEST 737 MUL @X(RA),RB TEST

14537 054340
14538 054340 000004
14539 054342 012700 000737
14540 054346 013737 054404 001076
14541 054354 005001
14542 054356 012704 000100
14543 054362 012702 000010
14544 054366 005003
14545 054370 012705 063276
14546 054374 012737 000010 063312
14547 054402 000257

TST737:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #737,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,@#\$TMP0 ;GET TEST INSTRUCTION WORD
CLR R1 ;S/B RESULT IN R2
MOV #100,R4 ;S/B RESULT IN R3
MOV #10,R2 ;INITIALIZE REG
CLR R3 ;INITIALIZE REG + 1
MOV #ATA,R5 ;GET POINTER TO TABLE OF POINTERS
MOV #10,@#MBUFO ;INITIALIZE SRC
CCC ;SCOPE SYNC

14548							
14549	054404	070275	000010		2\$:	MUL @10(R5),R2	:TEST THE MUL
14550							
14551	054410	020304				CMP R3,R4	:REG+1 CORRECT?
14552	054412	001002				BNE 3\$:BR IF NOT
14553	054414	020102				CMP R1,R2	:REG CORRECT?
14554	054416	001401				BEQ TST740	::BR IF YES
14555							
14556	054420	104045			3\$:	ERROR 45	:MUL DELIVERED WRONG RESULT
14557							
14558							
14559							
14560							
14561	054422						
14562	054422	000004					
14563	054424	012700	000740				
14564	054430	013737	054456	001076			
14565	054436	012701	010000				
14566	054442	012704	000001				
14567	054446	005002					
14568	054450	012703	020001				
14569	054454	000277					
14570							
14571	054456	071227	000002		2\$:	DIV #2,R2	:TEST DIV
14572							
14573	054462	100403				BMI 3\$:N:C=0000?
14574	054464	001402				BEQ 3\$	
14575	054466	102401				BVS 3\$	
14576	054470	103001				BCC 4\$	
14577							
14578	054472	104044			3\$:	ERROR 44	:COND CODES SET IMPROPERLY
14579							
14580	054474	020304			4\$:	CMP R3,R4	:CORRECT RESULT IN REG+1?
14581	054476	001002				BNE 5\$:BR IF NOT
14582	054500	020102				CMP R1,R2	:CORRECT RESULT IN REG?
14583	054502	001401				BEQ TST741	::BR IF YES
14584							
14585	054504	104045			5\$:	ERROR 45	:DIV DELIVERED WRONG RESULT
14586							
14587							
14588							
14589							
14590	054506						
14591	054506	000004					
14592	054510	012700	000741				
14593	054514	013737	054544	001076			
14594	054522	012701	177775				
14595	054526	012704	177776				
14596	054532	012702	177777				
14597	054536	012703	177762				
14598	054542	000257					
14599							
14600	054544	071227	000004		2\$:	DIV #4,R2	:TEST DIV
14601							
14602	054550	100003				BPL 3\$:N:C=1000?
14603	054552	001402				BEQ 3\$	

```

:*****
:*TEST 740 DIV #N,RA TEST ; N:C = 1111
:*****
TST740:

```

```

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #740,R0 ;;LOAD R0 WITH TEST NUMBER
MOV @#2$,@#$TMP0 ;GET COPY OF TEST INSTRUCTION
MOV #010000,R1 ;S/B RES IN R2
MOV #1,R4 ;S/B RES IN R3
CLR R2 ;SET UP REG OPERAND
MOV #020001,R3 ;SET UP REG+1 OP
SCC ;SCOPE SYNC

```

```

:*****
:*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
MOV @#2$,@#$TMP0 ;GET COPY OF TEST INSTRUCTION
MOV #-3,R1 ;S/B RES IN R2
MOV #-2,R4 ;S/B RES IN R3
MOV #-1,R2 ;SET UP REG OPERAND
MOV #-14.,R3 ;SET UP REG+1 OP
CCC ;SCOPE SYNC

```


14604 054554 102401
14605 054556 103001
14606
14607 054560 104044
14608
14609 054562 020304
14610 054564 001002
14611 054566 020102
14612 054570 001401
14613
14614 054572 104045
14615
14616
14617
14618
14619 054574

BVS 3\$
BCC 4\$
3\$: ERROR 44 ;COND CODES SET IMPROPERLY
4\$: CMP R3,R4 ;CORRECT RESULT IN REG+1?
BNE 5\$;BR IF NOT
CMP R1,R2 ;CORRECT RESULT IN REG?
BEQ TST742 ;:BR IF YES
5\$: ERROR 45 ;DIV DELIVERED WRONG RESULT

*TEST 742 DIV #N,RA TEST ; N:C = 0000 TO 0100

TST742:

14620 054574 000004
14621 054576 012700 000742
14622 054602 013737 054626 001076
14623 054610 005001
14624 054612 012704 000001
14625 054616 005002
14626 054620 012703 000001
14627 054624 000257
14628

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #742,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,@#STMPO ;GET COPY OF TEST INSTRUCTION
CLR R1 ;S/B RES IN R2
MOV #1,R4 ;S/B RES IN R3
CLR R2 ;SET UP REG OPERAND
MOV #1,R3 ;SET UP REG+1 OP
CCC ;SCOPE SYNC

14629 054626 071227 000002
14630
14631 054632 100403
14632 054634 001002
14633 054636 102401
14634 054640 103001
14635

2\$: DIV #2,R2 ;TEST DIV
BMI 3\$;N:C=0100?
BNE 3\$
BVS 3\$
BCC 4\$

14636 054642 104044
14637
14638 054644 020304
14639 054646 001002
14640 054650 020102
14641 054652 001401
14642

3\$: ERROR 44 ;COND CODES SET IMPROPERLY
4\$: CMP R3,R4 ;CORRECT RESULT IN REG+1?
BNE 5\$;BR IF NOT
CMP R1,R2 ;CORRECT RESULT IN REG?
BEQ TST743 ;:BR IF YES

14643 054654 104045
14644
14645

5\$: ERROR 45 ;DIV DELIVERED WRONG RESULT

*TEST 743 DIV #-N,RA TEST ; RA POS

TST743:

14648 054656
14649 054656 000004
14650 054660 012700 000743
14651 054664 013737 054712 001076
14652 054672 012701 177775
14653 054676 012704 000002
14654 054702 005002
14655 054704 012703 000016
14656 054710 000257
14657

SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #743,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,@#STMPO ;GET COPY OF TEST INSTRUCTION
MOV #-3,R1 ;S/B RES IN R2
MOV #2,R4 ;S/B RES IN R3
CLR R2 ;SET UP REG OPERAND
MOV #14.,R3 ;SET UP REG+1 OP
CCC ;SCOPE SYNC

14658 054712 071227 177774
14659

2\$: DIV #-4,R2 ;TEST DIV

14660 054716 020304
14661 054720 001002
14662 054722 020102
14663 054724 001401
14664
14665 054726 104045
14666
14667
14668
14669
14670
14671
14672
14673
14674 054730
14675 054730 000004
14676 054732 012700 000744
14677 054736 013701 054756
14678 054742 012704 000002
14679 054746 005037 177776
14680 054752 012702 000050
14681
14682 054756 071227 000005
14683
14684 054762 100424
14685 054764 001423
14686 054766 102022
14687 054770 103421
14688
14689 054772 012702 177777
14690 054776 005003
14691
14692 055000 071227 177776
14693
14694 055004 100413
14695 055006 001412
14696 055010 102011
14697 055012 103410
14698
14699 055014 012704 000003
14700
14701 055020 071227 000000
14702
14703 055024 100403
14704 055026 001402
14705 055030 102001
14706 055032 103405
14707
14708 055034 013703 177776
14709 055040 012702 177776
14710
14711 055044 104001
14712
14713
14714
14715

CMP R3,R4 ;CORRECT RESULT IN REG+1?
BNE 3\$;BR IF NOT
CMP R1,R2 ;CORRECT RESULT IN REG?
BEQ TST744 ;:BR IF YES

3\$: ERROR 45 ;DIV DELIVERED WRONG RESULT

:*****
:*TEST 744 DIV TEST - V BIT GETS SET
:* THIS TEST TESTS THAT THE V BIT CAN BE SET IN ALL THE
:* POSSIBLE WAYS. SINCE THE INSTRUCTION SHOULD BE ABORTED, THE
:* RESULTS CANNOT BE GUARANTEED. FOR THIS REASON, ONLY
:* THE CONDITION CODES ARE CHECKED.
:*****
TST744:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #744,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #2,R4 ;S/B PSW
CLR @#PSW ;CLEAR OUT OTHER PSW BITS
MOV #50,R2 ;SET UP REG OP

2\$: DIV #5,R2 ;TEST DIV -- SHOULD ABORT

BMI 3\$;N:C=0010?
BEQ 3\$
BVC 3\$
BCS 3\$

MOV #-1,R2 ;INITIALIZE REG OP
CLR R3 ;INITIALIZE REG+1 OP

DIV #-2,R2 ;TEST DIV -- SHOULD ABORT

BMI 3\$;N:C=0010?
BEQ 3\$
BVC 3\$
BCS 3\$

MOV #3,R4 ;S/B PSW

DIV #0,R2 ;TEST DIV BY 0 -- SHOULD ABORT

BMI 3\$;N:C=0010?
BEQ 3\$
BVC 3\$
BCS TST745 ;:IF ALL OK, THEN EXIT TEST

3\$: MOV @#PSW,R3 ;GET WAS PSW
MOV #PSW,R2 ;DESTINATION IS PSW

ERROR 1 ;CONDITION CODES SET WRONG

:*****
:*TEST 745 ASH #N,RA TEST ; SHIFT LEFT ; N:C = 0000 TO 1010
:*****

14716 055046
14717 055046 000004
14718 055050 012700 000745
14719 055054 013701 055072
14720 055060 012704 123450
14721 055064 012703 112345
14722 055070 000257
14723
14724 055072 072327 000003
14725
14726 055076 100003
14727 055100 001402
14728 055102 102001
14729 055104 103001
14730
14731 055106 104002
14732
14733 055110 020304
14734 055112 001401
14735 055114 104002
14736
14737
14738
14739
14740 055116
14741 055116 000004
14742 055120 012700 000746
14743 055124 013701 055142
14744 055130 005004
14745 055132 012703 000004
14746 055136 000257
14747 055140 000270
14748
14749 055142 072327 177775
14750
14751 055146 100403
14752 055150 001002
14753 055152 102401
14754 055154 103401
14755
14756 055156 104002
14757
14758 055160 020304
14759 055162 001401
14760 055164 104002
14761
14762
14763
14764
14765 055166
14766 055166 000004
14767 055170 012700 000747
14768 055174 013701 055212
14769 055200 012704 177234
14770 055204 012703 123432
14771 055210 000277

TST745:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #745,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH 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 R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;: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 R0 WITH TEST NUMBER
MOV @#2\$,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177234,R4 ;:S/B RESULT
MOV #123432,R3 ;:INITIAL REG
SCC ;:SCOPE SYNC

14772
14773 055212 072327 177772
14774
14775 055216 100003
14776 055220 001402
14777 055222 102401
14778 055224 103001
14779
14780 055226 104002
14781
14782 055230 020304
14783 055232 001401
14784 055234 104002
14785

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
:::*****

14789 055236
14790 055236 000004
14791 055240 012700 000750
14792 055244 013737 055274 001076
14793 055252 012701 123456
14794 055256 012704 076530
14795 055262 012702 112345
14796 055266 012703 147653
14797 055272 000257
14798

TST750:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #750,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,@#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

14799 055274 073227 000003
14800
14801 055300 100003
14802 055302 001402
14803 055304 102001
14804 055306 103001
14805
14806 055310 104044
14807
14808 055312 020102
14809 055314 001002
14810 055316 020403
14811 055320 001401
14812 055322 104045
14813

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
5\$: ERROR 45 ;ASHC DELIVERED WRONG RES

:::*****
:*TEST 751 ASHC #N,RA TEST ; SHIFT RIGHT ; N:C = 1000 TO 0101
:::*****

14817 055324
14818 055324 000004
14819 055326 012700 000751
14820 055332 013737 055356 001076
14821 055340 005001
14822 055342 005004
14823 055344 005002
14824 055346 012703 000005
14825 055352 000257
14826 055354 000270
14827

TST751:
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV #751,R0 ;:LOAD R0 WITH TEST NUMBER
MOV @#2\$,@#STMP0 ;GET TEST INSTRUCTION WORD
CLR R1 ;S/B RES IN R2
CLR R4 ;S/B RES IN R3
CLR R2 ;INITIALIZE COMBINED
MOV #5,R3 ; REGISTERS
CCC ;SCOPE SYNC
SEN ;CODES = 1000

14828 055356 073227 177775
14829
14830 055362 100403
14831 055364 001002
14832 055366 102401
14833 055370 103401
14834
14835 055372 104044
14836
14837 055374 020102
14838 055376 001002
14839 055400 020403
14840 055402 001401
14841 055404 104045
14842
14843
14844
14845
14846 055406
14847 055406 000004
14848 055410 012700 000752
14849
14850 055414 032737 100000 063234
14851 055422 001401
14852 055424 000000
14853 055426 013737 055456 001076
14854 055434 012701 177234
14855 055440 012704 135275
14856 055444 012702 123456
14857 055450 012703 127542
14858 055454 000257
14859
14860 055456 073227 177772
14861
14862 055462 100003
14863 055464 001402
14864 055466 102401
14865 055470 103401
14866
14867 055472 104044
14868
14869 055474 020102
14870 055476 001002
14871 055500 020403
14872 055502 001401
14873 055504 104045
14874
14875
14876
14877
14878
14879
14880
14881
14882
14883

2\$: ASHC #-3,R2 ;TEST ASHC
BMI 3\$;N:C=0101?
BNE 3\$
BVS 3\$
BCS 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 TST752 ;:BR IF YES
5\$: 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 @#2\$,@#\$TMP0 ;GET TEST INSTRUCTION WORD
MOV #177234,R1 ;S/B RES IN R2
MOV #135275,R4 ;S/B RES IN R3
MOV #123456,R2 ;INITIALIZE COMBINED
MOV #127542,R3 ; REGISTERS
CCC ;SCOPE SYNC
2\$: ASHC #-6,R2 ;TEST ASHC
BPL 3\$;N:C=1000?
BEQ 3\$
BVS 3\$
BCS 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 TST753 ;:BR IF YES
5\$: 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 !!! BE USED DURING THESE MED *
: * TESTS. NO INTERRUPTS OR TRAPS CAN BE ALLOWED EITHER *
: *

14940	055640	000004				SCOPE			;CALL THE SCOPE LOOP UTILITY
14941	055642	012737	000340	177776	MEDT1:	MOV	#340,@#PSW		;KERNEL MODE-PRIORITY 7
14942	055650	012701	064166			MOV	#TBL2,R1		;INITIALIZE ADDRESS POINTER
14943	055654	012737	125252	001102	1\$:	MOV	#125252,@#STMP2		
14944	055662	111137	055730			MOVB	(R1),@#11\$;PUT WRITE CODE BY 'WRITE-MED'S'
14945	055666	112137	055752			MOVB	(R1)+,@#13\$;AND POINT R1 TO READ CODE
14946	055672	111137	055710			MOVB	(R1),@#10\$;PUT READ CODE BY 'READ-MED'S'
14947	055676	112137	055736			MOVB	(R1)+,@#12\$;R1 NOW POINTS TO NEXT REG.
14948	055702	005037	001106			CLR	@#STMP4		;CLEAR ERROR COUNTER
14949	055706	076600			2\$:	MED			;MED-READ THE INTERNAL REG.
14950	055710	000000			10\$:	.WORD	0		;MED-READ CODE
14951	055712	010037	001076			MOV	R0,@#STMP0		;SAVE ITS ORIGINAL CONTENTS
14952	055716	010137	001100			MOV	R1,@#STMP1		;SAVE ADDR. PTR. VALUE
14953	055722	013700	001102			MOV	@#STMP2,R0		;LOAD R0 WITH DATA TO BE WRITTEN
14954	055726	076600				MED			;MED-WRITE THE TEST DATA
14955	055730	000000			11\$:	.WORD	0		;MED-WRITE CODE
14956	055732	005000				CLR	R0		;CLEAR R0
14957	055734	076600				MED			;MED-READ THE DATA BACK
14958	055736	000000			12\$:	.WORD	0		;MED-READ CODE
14959	055740	010037	001104			MOV	R0,@#STMP3		;SAVE DATA READ FOR COMPARISON
14960	055744	013700	001076			MOV	@#STMP0,R0		;LOAD ORIGINAL DATA IN R0
14961	055750	076600				MED			;MED-WRITE ORG. DATA TO REG.
14962	055752	000000			13\$:	.WORD	0		;MED-WRITE CODE
14963	055754	023737	001102	001104		CMP	@#STMP2,@#STMP3		;DID DATA READ=DATA WRITTEN?
14964	055762	001412				BEQ	3\$;BRANCH IF YES
14965	055764	013737	055736	001100		MOV	@#12\$,@#STMP1		;SAVE MED-CODE FOR ERROR
14966	055772	022737	000003	001106		CMP	#3,@#STMP4		;MAX. ERROR REPORTS YET?
14967	056000	002401				BLT	14\$;BRANCH IF YES
14968	056002	104022				ERROR	22		;INT. REG. READ BACK WRONG DATA
14969	056004	005237	001106		14\$:	INC	@#STMP4		;INCREMENT ERROR COUNTER
14970	056010	005137	001102		3\$:	COM	@#STMP2		;CHANGE DATA PATTERN
14971	056014	013701	001100			MOV	@#STMP1,R1		;RESTORE ADDR. POINTER

14972 056020 022737 125252 001102
14973 056026 001327
14974 056030 005711
14975 056032 001310

CMP #125252,@#STMP2 ;BOTH DATA PATTERNS BEEN USED?
BNE 2\$;BRANCH IF NO
TST (R1) ;END OF ADDR. TABLE?
BNE 1\$;BRANCH IF NO

14976
14977
14978
14979
14980
14981
14982
14983
14984
14985
14986
14987
14988

:TEST 755 MED TEST - VERIFY NOPS; READ R7 IN A & B SP
:*****
: THIS TEST CHECKS ALL OF THE "NOP" OPERATION CODES
: TO ENSURE THEY WILL EXECUTE AS NOP'S AND
: NOT RESULT IN A PROCESSOR HANG. THE "NOPS"
: TABLE (TABLE III) HOLDS THESE CODES.
: THIS TEST ALSO READS THE PROGRAM COUNTER (R7) VALUES
: STORED IN A & B SCRATCH PADS TO SEE THAT THEY
: READ PROPERLY. THE R7 ADDRESSES ARE IN TABLE IV.
:*****

14989 056034
14990 056034 012700 000754
14991 056040 000004
14992 056042 012701 064404
14993 056046 112137 056054
14994
14995 056052 076600
14996 056054 000000
14997 056056 123711 056054
14998
14999 056062 103003
15000 056064 005237 056054
15001 056070 000770
15002 056072 105721
15003 056074 005711
15004 056076 001363
15005

TST755:
MOV #754,R0 ;:SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;:CALL THE SCOPE LOOP UTILITY
MEDT3: MOV #TBL3,R1 ;:INITIALIZE NOP TABLE PTR. (R1)
1\$: MOV (R1)+,@#10\$;:PLACE FIRST "NOP-CODE" AFTER MED
;:AND POINT R1 TO LAST CODE IN GROUP
5\$: MED ;:EXECUTE MED WITH NOP OP-CODE
10\$: .WORD 0
CMPB @#10\$,(R1) ;:HAVE ALL NOPS IN THAT GROUP
;:BEEN TESTED?
;:BRANCH IF YES
6\$: BHS 6\$;:NEXT NOP IN GROUP
TSTB (R1)+ ;:POINT R1 TO NEXT NOP GROUP
TST (R1) ;:HAVE ALL GROUPS BEEN TESTED
BNE 1\$;:BRANCH IF NO

15006 056100 113737 064425 056112
15007 056106 005000
15008 056110 076600
15009 056112 000000
15010 056114 020027 056114
15011 056120 001411
15012 056122 013737 056112 001100
15013 056130 012737 056114 001102
15014 056136 010037 001104
15015 056142 104022
15016 056144 023727 056112 000047
15017 056152 001404
15018 056154 113737 064431 056112
15019 056162 000751
15020 056164

MEDT4: MOV @#R7A+1,@#5\$;:LOAD R7A READ CODE AFTER MED
4\$: CLR R0 ;:CLEAR R0
MED ;:MED READ R7 IN THE ASP
5\$: .WORD 0 ;:READ CODE FOR R7A
CMP R0,#5\$+2 ;:DID R7A READ CORRECTLY?
BEQ 6\$;:BRANCH IF YES
MOV @#5\$,@#STMP1 ;:SAVE MED-CODE FOR ERROR
MOV #5\$+2,@#STMP2 ;:SAVE DATA EXPECTED
MOV R0,@#STMP3 ;:SAVE DATA RECEIVED
ERROR 22 ;:R7A DID NOT READ THE RIGHT VALUE
6\$: CMP @#5\$,#47 ;:HAS R7B BEEN CHECKED?
BEQ 8\$;:BRANCH IF YES
MOV @#R7B+1,@#5\$;:LOAD R7B READ CODE AFTER MED
BR 4\$;:TEST R7 BSP
8\$:

15021
15022
15023
15024
15025
15026
15027

:TEST 756 MED TEST - CSP CONSTANTS CHECK
:*****
: THIS TEST CHECKS THE CONSTANT VALUES LOCATED
: IN THE C SCRATCH PAD. THE CONSTANTS ARE READ

15028
15029
15030
15031
15032
15033 056164
15034 056164 012700 000755
15035 056170 000004
15036
15037 056172 076600
15038 056174 000144
15039 056176 052700 004000
15040 056202 076600
15041 056204 000344
15042 056206 170000
15043
15044 056210 012701 064536
15045 056214 012167 000006
15046 056220 001414
15047 056222 005000
15048 056224 076600
15049 056226 000000
15050 056230 020021
15051 056232 001770
15052 056234 013737 056226 001100
15053 056242 016137 177776 001102
15054 056250 104021
15055 056252
15056
15057
15058
15059
15060
15061
15062
15063
15064
15065
15066
15067
15068
15069
15070
15071
15072
15073
15074
15075 056252
15076 056252 012700 000756
15077 056256 000004
15078 056260 012737 000071 177770
15079 056266 012737 061104 000004
15080 056274 012737 000340 000006
15081 056302 005037 061112
15082 056306 076600
15083 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.

TST756: MOV #755,R0 ;;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;CALL THE SCOPE LOOP UTILITY

MED RFLAG
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
10\$: MOV #TBL7,R1 ;SETUP TABLE POINTER
MOV (R1)+,1\$;LOAD MED READ CODE AT 1\$
BEQ 11\$;BR IF END OF TABLE
CLR R0
MED ;READ INTERNAL CONTENTS INTO R0
1\$: .WORD 0
CMP R0,(R1)+ ;WAS THE CONSTANT READ THE ONE EXPECTED
BEQ 10\$;BRANCH IF YES
MOV @#1\$,@#STMP1 ;SAVE MEDCODE FOR ERROR
MOV -2(R1),@#STMP2 ;SAVE CONSTANT VALUE EXPECTED
ERROR 21 ;CSP LOCATION HELD WRONG VALUE
11\$:

*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

TST757: MOV #756,R0 ;;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;CALL THE SCOPE LOOP UTILITY
MEDT11: MOV #SWB01,@#UBREAK ;LOAD MICROBK. REG. WITH AN MICRO ADDR.
MOV #BKROUT,@#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

15084	056312	052700	001000		BIS	#BIT9,R0	;SET BIT 9
15085	056316	076600			MED		;MED-WRITE THE WHAMI REG TO
15086	056320	000222		10\$:	WRWHAMI		;ENABLE MICROBK-TRAP-TO-4
15087	056322	076600			MED		;GET FLAG REGISTER
15088	056324	000144			RDFLAG		
15089	056326	052700	100000		BIS	#BIT15,R0	;SET BIT 15 IN R0
15090	056332	076600			MED		;MED-WRITE THE FLAG REG TO
15091	056334	000344		11\$:	WRFLAG		;ENABLE MICROBK TRAPPING
15092	056336	000300			SWAB	R0	;MICROBK TRAP SHOULD OCCUR ON SWAB
15093	056340	005737	061112		TST	@#BKFLAG	;DID TRAP TO 4 OCCUR?
15094	056344	001007			BNE	1\$;BRANCH IF YES
15095	056346	005037	001076		CLR	@#\$TMP0	
15096	056352	016737	121513	001100	MOV	SWB01,@#\$TMP1	;SAVE EXPECTED UBREAK ADDR
15097	056360	104015			ERROR	15	;MICROBREAK TRAP DIDN'T WORK
15098	056362	000453			BR	50\$;SKIP TO END OF TEST
15099							
15100	056364	012701	000710	1\$:	MOV	#SWB01*10,R1	;GET CORRECT U-ADDR
15101	056370	076600			MED		;GET LOG CUA REG
15102	056372	000103			RDLCUA		
15103	056374	042700	100007		BIC	#100007,R0	;GET RID OF IRRELEVANT BITS
15104	056400	020001			CMP	R0,R1	;WAS CORRECT UADDR LOGGED?
15105	056402	001401			BEQ	3\$;BR IF YES
15106	056404	104025			ERROR	25	;CUA CONTAINS INCORRECT U-ADDR
15107	056406	012701	064436	3\$:	MOV	#TBL5,R1	;INITIALIZE TABLE PTR. (R1)
15108	056412	012702	064464		MOV	#TBL6,R2	
15109	056416	010737	001010		MOV	PC,@#\$LPERR	;SET ERROR LOOP RETURN TO 2\$
15110	056422	111137	056460	2\$:	MOVB	(R1),@#12\$;LOAD WRITE CODE AFTER MED
15111	056426	001431			BEQ	50\$;BR IF END OF TABLE
15112	056430	011237	177770	4\$:	MOV	(R2),@#UBREAK	;LOAD MICROBK REG. WITH MICROADDR.
15113	056434	005037	061112		CLR	@#BKFLAG	;CLEAR MICROBK TRAP-TO-4 FLAG
15114	056440	076600			MED		;GET FLAG REGISTER
15115	056442	000144			RDFLAG		
15116	056444	052700	100000		BIS	#BIT15,R0	;SET BIT 15 IN R0
15117	056450	076600			MED		;MED WRITE TO FLAG REG TO
15118	056452	000344		15\$:	WRFLAG		;ENABLE MICROBK TRAPPING
15119	056454	005000			CLR	R0	;IN CASE U-BREAK TRAP DOESN'T OCCOR
15120							;USUALLY BETTER TO WRITE 0'S
15121	056456	076600			MED		
15122	056460	000000		12\$:	.WORD	0	
15123	056462	005737	061112		TST	@#BKFLAG	;DID WE TRAP-TO-4? (FLAG NOT = 0)
15124	056466	001006			BNE	20\$;BRANCH IF YES TO NEXT ENTRY
15125	056470	013737	056460	001076	MOV	@#12\$,@#\$TMP0	;SAVE MED-CODE FOR ERROR
15126	056476	011237	001100		MOV	(R2),@#\$TMP1	;SAVE EXPECTED U-ADDR FOR ERROR
15127	056502	104015			ERROR	15	;MICROBK. TRAP-TO-4 DID NOT OCCUR
15128							
15129	056504	105721		20\$:	TSTB	(R1)+	;INCREMENT TO NEXT TABLE
15130	056506	005722			TST	(R2)+	;ENTRIES AND
15131	056510	000744			BR	2\$;CONTINUE
15132							
15133	056512	076600		50\$:	MED		;GET WHAMI INTO R0
15134	056514	000022			RDWHAMI		
15135	056516	042700	001000		BIC	#BIT9,R0	; CLEAR BIT 9
15136	056522	076600			MED		;CLEAR THE FLAG REG. TO
15137	056524	000344		13\$:	WRFLAG		; DISABLE MICROBK. TRAPPING
15138	056526	076600			MED		;CLEAR THE WHAMI REG. TO
15139	056530	000222		14\$:	WRWHAMI		; DISABLE MICROBK. TRAP-TO-4

15140 056532 012737 056260 001010
15141 056540 012737 061220 000004
15142 056546 012737 000304 177770

MOV #MEDT11,@#SLPERR ;RESET LOOP ON ERROR POINTER
MOV #BERR,@#4 ;RESTORE NORMAL ERROR ROUTINE
MOV #304,@#UBREAK ;GENERATE SYNC PULSE ON MED INSTR

15143
15144
15145
15146
15147
15148
15149
15150
15151

: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.

15152
15153 056554
15154 056554 012700 000757
15155 056560 000004
15156 056562 012737 056622 000004
15157 056570 012737 000340 000006
15158 056576 012700 100001
15159 056602 076600
15160 056604 000222
15161 056606 012702 056563
15162 056612 005767 177745

TST760:
MOV #757,R0 ;:SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
SCOPE ;CALL THE SCOPE LOOP UTILITY
1\$: MOV #2\$,@#4 ;:SETUP PC FOR ODD ADDR SERVICE
MOV #340,@#6
MOV #BIT15+BIT0,R0 ;:SETUP 'LOG FIRST' MODE
MED
WRWHAMI
MOV #1\$+1,R2 ;:SAVE ADDRESS OF ODD ADDR. INSTRUCTION
TST 1\$+1 ;:DO ODD ADDRESS INSTRUCTION TO FORCE
;A JAMUPP & TRAP TO 4

15163
15164 056616 104023
15165 056620 000441
15166 056622 022626
15167 056624 012737 061220 000004
15168 056632 076600
15169 056634 000100
15170 056636 013701 177766
15171 056642 032701 000100

ERROR 23 ;*** ODD ADDR. TRAP DID NOT OCCUR
BR 10\$;:EXIT TEST
2\$: CMP (SP)+,(SP)+ ;:RESTORE STACK
MOV #BERR,@#4 ;:RESTORE OLD PC & PSW
MED
RDLJAM
MOV @#CPUERR,R1
BIT #BIT6,R1 ;:WAS ODD ADDR. ERROR RECORDED BY
;THE CPU ERROR REGISTER?

15172
15173 056646 001001
15174 056650 104024
15175
15176
15177 056652 032700 100004
15178 056656 001001
15179 056660 104024
15180
15181

BNE 3\$;:BRANCH IF YES
ERROR 24 ;*** CPU ERROR REG. DID NOT
;REPORT ODD ADDRESS ERROR
;READ THE LOG JAM REGISTER
3\$: BIT #BIT15+BIT2,R0 ;:WAS ODD ADDR. ERROR LOGGED BY LOG JAM
BNE 4\$;:BRANCH IF YES
ERROR 24 ;*** LOG JAM REG. DID NOT LOG
;ODD ADDRESS ERROR CORRECTLY

15182 056662 005005
15183 056664 076600
15184 056666 000102
15185 056670 010003
15186 056672 020002
15187

4\$: CLR R5 ;:CLR ERROR FLAG
MED ;:READ THE LOG PBA REGISTER
RDLPBA
MOV R0,R3 ;:SAVE RECEIVED PHYS ADDR <15:0>
CMP R0,R2 ;:WERE BITS <15:00> OF THE PHYSICAL
;BUS ADDR. LOGGED CORRECTLY?

15188 056674 001401
15189 056676 005205
15190 056700 076600
15191 056702 000101
15192 056704 000300
15193 056706 042700 177774
15194 056712 001002
15195 056714 005705

BEQ 5\$;:BRANCH IF YES
INC R5 ;:SET ERROR FLAG
5\$: MED ;:READ THE LOG SERVICE REGISTER
RDLSERVICE
SWAB R0 ;:GET 'PBA 17&16' DOWN TO BIT POSITION 0&1
BIC #177774,R0
BNE 11\$;:BR IF PHYS ADDR BITS <17:16> LOGGED CORRECTLY
TST R5 ;:PREVIOUS ERROR?

15196 056716 001402
15197 056720 005001
15198 056722 104026
15199
15200
15201 056724 005000
15202 056726 076600
15203 056730 000222
15204
15205
15206
15207
15208
15209
15210
15211
15212
15213
15214
15215

11\$: BEQ 10\$:BR IF NOT
CLR R1 :SET UP EXPECTED PA<17:16>
ERROR 26 :*** PHYSICAL BUS ADDR. <17:00>
:NOT LOGGED CORRECTLY WHEN
:ODD ADDRESS TRAP OCCURRED
10\$: CLR R0 :DISABLE "LOG FIRST" MODE
MED
WRWHAMI

: *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.

15216 056732
15217 056732 012700 000760
15218 056736 000004
15219
15220 056740 012701 064046
15221 056744 005711
15222 056746 012737 000100 177746
15223 056754 012711 125252
15224 056760 012737 000001 177746
15225
15226 056766 012737 057026 000114
15227 056774 012737 000340 000116
15228 057002 005000
15229 057004 076600
15230 057006 000302
15231 057010 076600
15232 057012 000306
15233 057014 076600
15234 057016 000307
15235 057020 005767 005022
15236 057024 000406
15237 057026 012700 000200
15238 057032 076600
15239 057034 000352
15240 057036 022626
15241 057040 104030
15242
15243 057042 012700 000200
15244 057046 076600
15245 057050 000352
15246 057052 012711 125252
15247 057056 012737 000116 000114
15248 057064 005037 000116
15249 057070 005005
15250 057072 076600
15251 057074 000102

TST761:
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 #DPTRP,@#CCR :DISABLE PARITY ERROR TRAPS
:AND CLEAR WWP
MOV #1\$,@#114 :SETUP PARITY ERROR VECTOR
MOV #340,@#116
CLR R0
MED :CLEAR LOG PBA REGISTER
WRLPBA :CLEAR LOG CACHE DATA REGISTER
MED :CLEAR LOG CACHE TAG REGISTER
WRLDATA :CLEAR LOG CACHE TAG REGISTER
MED :CLEAR LOG CACHE TAG REGISTER
WRLTAG :CLEAR LOG CACHE TAG REGISTER
TST TLOC1 :READ TEST LOCO TO FORCE PARITY ERROR
BR 2\$:BRANCH IF NO TRAP OCCURS
1\$: MOV #200,R0
MED :CLEAN UP THE CACHE
352 :INITIALIZATION CODE
CMP (SP)+,(SP)+ :CLEAN UP STACK
ERROR 30 :*** PARITY TRAP TO 114 OCCURRED
:WHEN IT SHOULD HAVE BEEN DISABLED
2\$: MOV #200,R0
MED :CLEAN UP THE CACHE
352 :INITIALIZATION CODE
MOV #125252,(R1) :WRITE BAK GOOD PARITY IN TST LOC
MOV #116,@#114 :RESTORE ORIGINAL PARITY HANDLER & PSW
CLR @#116
CLR R5 :CLEAR ERROR FLAG
MED :READ LOG PBA REGISTER
RDLPBA

```
15252 057076 010003      MOV      R0,R3      :SAVE COPY
15253                    :LOG PBA REG. STILL CLEAR?
15254 057100 001401      BEQ      3$         :BRANCH IF YES
15255 057102 005205      INC      R5         :OTHERWISE SET ERROR FLAG
15256 057104 076600      3$:  MED          :READ LOG CACHE DATA REG.
15257 057106 000106      RDLDATA
15258 057110 010001      MOV      R0,R1      :SAVE COPY
15259                    :LOG CACHE DATA REG. STILL CLEAR?
15260 057112 001401      BEQ      4$         :BRANCH IF YES
15261 057114 005205      INC      R5         :OTHERWISE SET ERROR FLAG
15262 057116 076600      4$:  MED          :READ LOG CACHE TAG REG.
15263 057120 000107      RDLTAG
15264 057122 010002      MOV      R0,R2      :SAVE COPY
15265                    :LOG CACHE TAG REG. STILL CLEAR?
15266 057124 001401      BEQ      5$         :BRANCH IF YES
15267 057126 005205      INC      R5         :OTHERWISE SET ERROR FLAG
15268 057130 005705      5$:  TST      R5     :WERE ANY OF LOG REGISTERS CHANGED
15269 057132 001401      BEQ      6$         :BRANCH IF NO
15270 057134 104027      ERROR   27         :*** ONE OF LOG REGISTERS CHANGED
15271                    :WHEN ERROR SHOULD NOT HAVE BEEN LOGGED
15272                    :LOG PBA, LOG DATA & LOG TAG
15273                    :REGISTER SHOULD BE CLEAR.
15274 057136 005037 177746 6$:  CLR      @#CCR    :ENABLE PARITY ERROR TRAPS
15275
15276                    :*****
15277                    :*TEST 762      CHECK PARITY ERROR BITS IN MEMERR REG. IN BACKUP MODE OF CACHE (TRAP)
15278
15279                    :*THIS TEST CHECKS THAT ALL OF THE PARITY ERROR BITS (5,6,7)
15280                    :*OF THE MEMORY ERROR REGISTER ARE SET TO '1' WHEN A CACHE
15281                    :*PARITY ERROR OCCURS IN THE BACKUP MODE.
15282                    :*****
15283 057142
15284 057142 012700 000761  TST762:  MOV      #761,R0    ;;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
15285 057146 000004          SCOPE          :CALL THE SCOPE LOOP UTILITY
15286 057150 012701 064046  MOV      #TLOC1,R1  :GET POINTER TO TEST LOCATION
15287 057154 005711          TST      (R1)      :MAKE IT A HIT
15288 057156 012737 000100 177746  MOV      #WWP,@#CCR :SET WRITE WRONG PARITY BIT
15289 057164 012711 125252  MOV      #125252,(R1) :WRITE TO TEST LOC. WITH WRONG PARITY
15290 057170 042737 000100 177746  BIC      #WWP,@#CCR  :CLEAR WWP
15291 057176 012737 057232 000114  MOV      #1$,@#114  :SETUP NEW TEST HANDLER AT PARITY VECTOR
15292 057204 012737 000340 000116  MOV      #340,@#116
15293 057212 005737 064046          TST      @#TLOC1   :READ TEST LOC. TO FORCE PARITY ERROR
15294 057216 012700 000200          MOV      #200,R0
15295 057222 076600          MED          :CLEAN UP THE CACHE
15296 057224 000352          352         :INITIALIZATION CODE
15297 057226 104031          ERROR   31      :*** PARITY ERROR DID NOT CAUSE TRAP
15298 057230 000405          BR      2$       :BRANCH TO 2$
15299 057232 012700 000200      1$:  MOV      #200,R0
15300 057236 076600          MED          :CLEAN UP THE CACHE
15301 057240 000352          352         :INITIALIZATION CODE
15302 057242 022626          CMP      (SP)+,(SP)+ :CLEAN UP STACK
15303 057244 022737 000340 177744 2$:  CMP      #000340,@#MEMERR :WERE PARITY ERROR BITS (5,6,7) SET
15304                    :AND CPU ABORT BIT (15) LEFT CLEAR
15305                    :IN MEMORY ERROR REGISTER?
15306 057252 001403          BEQ      3$       :BRANCH IF YES
15307 057254 013700 177744          MOV      @#MEMERR,R0
```

```

15308 057260 104032          ERROR 32          :*** MEMORY ERROR REGISTER BITS
15309                                     :WERE SET INCORRECTLY
15310 057262 012737 000116 000114 3$:  MOV    #116,@#114      :RESTORE OLD PARITY HANDLER PC & PSW
15311 057270 005037 000116          CLR    @#116
15312
15313 ::*****
15314 :*TEST 763      CHECK UNIBUS TIMEOUT, ODD ADDRESS AND LOG CONTINUOUS MODE
15315
15316 :*THIS TEST CHECKS THAT THE 'UNIBUS TIMEOUT' BIT (BIT4)
15317 :*GETS SET IN THE CPU ERROR REGISTER WHEN A TIMEOUT OCCURS.
15318 :*A TIMEOUT TRAP IS FORCED BY REFERENCING BUS ADDRESS 760000.
15319 :*THEN AN ODD ADDRESS ERROR IS FORCED AND IT
15320 :*IS CHECKED IF ONLY BIT (6)-ODD ADDRESS ERROR IS SET
15321 :*(IN CPUERR). THIS CHECKS THAT THE ERROR LOG IS
15322 :*CONTINUOUSLY UPDATED IN THE 'LOG CONTINUOUS' MODE.
15323 ::*****
15324 057274          TST763:
15325 057274 012700 000762          MOV    #762,R0          ;;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
15326 057300 000004          SCOPE          ;;CALL THE SCOPE LOOP UTILITY
15327 057302 012737 057324 000004  MOV    #1$,@#4          ;;SETUP NEW PC & PSW FOR THE
15328 057310 012737 000340 000006  MOV    #340,@#6        ;;TIMEOUT SERVICE ROUTINE
15329 057316 005737 160000          TST    @#160000        ;;FORCE A TIMEOUT TRAP TO 4 BY
15330                                     ;;REFERENCING NON-EXISTENT ADDRESS
15331 057322 000461          BR     6$
15332 057324 022626          1$:  CMP    (SP)+,(SP)+      ;;RESTORE STACK
15333 057326 012737 061220 000004  MOV    #BERR,@#4       ;;RESTORE OLD PC & PSW FOR TIMEOUT
15334 057334 076600          MED
15335 057336 000100          RDLJAM
15336 057340 013701 177766          MOV    @#CPUERR,R1     ;;SAVE CPU ERR REG
15337 057344 022701 000020          CMP    #BIT4,R1        ;;DID 'UNIBUS TIMEOUT' BIT IN CPU ERROR
15338                                     ;;REGISTER GET SET?
15339 057350 001401          BEQ   2$             ;;BRANCH IF YES
15340 057352 104033          ERROR 33            ;*** 'UNIBUS TIMEOUT' BIT (BIT4) IN CPU
15341                                     ;;ERROR REG. DID NOT SET WHEN A
15342                                     ;;TIMEOUT WAS FORCED
15343                                     ;;READ THE LOG JAM REGISTER
15344 057354 022700 021200          2$:  CMP    #BIT13+BIT9+BIT7,R0  ;;DID 'UNIBUS TIMEOUT' BIT (BIT7) SET?
15345                                     ;;BRANCH IF YES
15346 057360 001401          BEQ   3$             ;*** 'UNIBUS TIMEOUT' BIT (BIT7)
15347 057362 104033          ERROR 33            ;;DID NOT SET IN LOG JAM REGISTER
15348                                     ;;WHEN UNIBUS TIMEOUT WAS FORCED
15349                                     ;;READ LOG PBA
15350 057364 076600          3$:  MED
15351 057366 000102          RDLPBA
15352 057370 020027 160000          CMP    R0,#160000      ;;WAS PHYS BA LOGGED CORRECTLY?
15353 057374 001403          BEQ   5$
15354 057376 012701 160000          MOV    #160000,R1
15355 057402 104020          ERROR 20            ;;PHYSICAL BUS ADDRESS WAS
15356                                     ;;LOGGED WRONG ON A UNIBUS
15357                                     ;;TIMEOUT
15358 057404 012737 057426 000004  5$:  MOV    #4$,@#4          ;;SET UP PC,PSW FOR ODD ADDRESS
15359 057412 012737 000340 000006  MOV    #340,@#6
15360 057420 005767 177741          TST   3$+1           ;;FORCE ODD ADDRESS ERROR
15361 057424 000420          BR    6$
15362 057426 022626          4$:  CMP    (SP)+,(SP)+      ;;RESTORE STACK
15363 057430 012737 061220 000004  MOV    #BERR,@#4

```

(REV.DO)

15364 057436 076600
15365 057440 000100
15366 057442 013701 177766
15367 057446 022701 000100
15368 057452 001401
15369 057454 104024

MED
RDLJAM
MOV @#CPUERR,R1
CMP #BIT6,R1
BEQ 7\$
ERROR 24

:ODD ADDR. BUT SET 3
:ODD ADDRESS BIT WAS
:NOT SET IN THE CPU
:ERROR REGISTER. IN LOG
:CONTINUOUS MADE THE
:LATEST ERROR SHOULD
:BE LOGGED

15370
15371
15372
15373
15374
15375 057456 032700 000004
15376 057462 001001
15377 057464 104024

7\$: BIT #BIT2,R0
BNE 6\$
ERROR 24

:ODD ADR. BIT SET IN
:LOG JAM?
:ODD ADDRESS BIT WAS
:NOT SET IN THE LOG
:JAM REGISTER ON A
:ODD ADDRESS ERROR
:CHECK IF LAST INTERRUPT VECTOR
:WAS LOGGED?

15378
15379
15380
15381 057466 076600
15382 057470 000104
15383 057472 120027 000004
15384 057476 001401
15385 057500 104036

6\$: MED
RDLFGINT
CMPB R0,#4
BEQ 8\$
ERROR 36

:LAST ERROR VECTOPR WS NOT LOGGED

15386
15387 057502

8\$:

15388
15389
15390
15391
15392
15393
15394
15395
15396
15397
15398
15399

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

15400 057502
15401 057502 012700 000763
15402 057506 000004
15403 057510 012737 057540 000004
15404 057516 012737 000340 000006
15405 057524 005037 177746
15406 057530 012707 177746
15407 057534 104034

IST764:

MOV #763,R0
SCOPE
MOV #1\$,@#4
MOV #340,@#6
CLR @#CCR
MOV #CCR,PC
ERROR 34

:::SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
:CALL THE SCOPE LOOP UTILITY
:SETUP NEW HANDLER PC & PSW
:
:ILLEGAL INTERNAL ADDRESS TRAP SHOULD OCCUR
:*** ILLEGAL INTERNAL ADDRESS
:DID NOT RESULT IN A TRAP
:BRANCH TO EXIT IF NO TRAP
:RESTORE STACK
:RESTORE OLD HANDLER PC & PSW

15408
15409 057536 000420
15410 057540 022626
15411 057542 012737 061220 000004
15412 057550 076600
15413 057552 000100
15414 057554 013701 177766
15415 057560 032701 000001

1\$: BR 3\$
CMP (SP)+,(SP)+
MOV #BERR,@#4
MED
RDLJAM
MOV @#CPUERR,R1
BIT #BIT0,R1

:DID "ILLEGAL INTERNAL ADDRESS" BIT (0)
:IN CPU ERROR REGISTER GET SET?
:BRANCH IF YES
:*** ILLEGAL INTERNAL ADDRESS
:BIT DID NOT SET IN CPU ERROR REG.

15416
15417 057564 001001
15418 057566 104035
15419

BNE 2\$
ERROR 35

15420
15421 057570 032700 000040
15422
15423 057574 001001
15424 057576 104035
15425
15426 057600
15427
15428
15429
15430
15431
15432
15433
15434
15435
15436
15437
15438
15439
15440
15441 057600
15442 057600 012700 000764
15443 057604 000004
15444
15445 057606 012737 000201 177746
15446 057614 005037 001062
15447 057620 012701 064046
15448 057624 005711
15449 057626 052737 000100 177746
15450 057634 012711 125252
15451 057640 042737 000100 177746
15452 057646 012700 100001
15453 057652 076600
15454 057654 000222
15455 057656 042737 000001 177746
15456 057664 012737 057712 000114
15457 057672 016737 004150 001062
15458 057700 012700 000200
15459 057704 076600
15460 057706 000352
15461 057710 104031
15462
15463
15464
15465
15466
15467 057712 012700 000200
15468 057716 076600
15469 057720 000352
15470 057722 012737 000001 177746
15471 057730 012737 000116 000114
15472 057736 005037 000116
15473 057742 022626
15474 057744 005737 001062
15475

28: BIT #BIT5,R0 ;READ THE LOG JAM REG.
;DID "ILLEGAL INTERNAL ADDRESS" BIT (5)
;IN LOG JAM REG. GET SET
BNE 38 ;BRANCH IF YES
ERROR 35 ;*** ILLEGAL INTERNAL ADDRESS BIT
;DID NOT SET IN LOG JAM REG.

38:

: *TEST 765 CHECK LOG SERVICE & MEMERR LOGS LO-HI BYTE & TAG, IN CACHE ABORT MODE
;TEST CHECKS THAT "LO BYTE PARITY" "HI 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 @#SREGO
MOV #TLOC1,R1 ;GET POINTER TO TEST LOC.
TST (R1) ;MAKE IT A HIT
BIS #WWP,@#CCR ;WRITE WRONG PARITY SET
MOV #125252,(R1) ;WRITE TEST LOCATION WITH WRONG PARITY
BIC #WWP,@#CCR ;CLEAR WWP
MOV #BIT15+BIT0,R0
MED ;ENABLE "LOG FIRST" MODE, AND
WRWHAMI ;ERROR LOGGING
BIC #DPTRP,@#CCR ;ENABLE CACHE PARITY TRAPS
MOV #PTRP1,@#114 ;NEW PARITY TRAP SERVICE
MOV TLOC1,@#SREGO ;READ TEST LOC, FORCE PARITY ERROR
MOV #200,R0
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

PTRP1: MOV #200,R0
MED ;CLEAN UP THE CACHE
352 ;INITIALIZATION CODE
MOV #DPTRP,@#CCR ;DISABLE CACHE PARITY ERROR TRAPS
MOV #116,@#114 ;REESTABLISH OLD SERVICE VECTORS
CLR @#116
CMP (SP)+,(SP)+
TST @#SREGO ;WAS THE INSTRUCTION ABORTED ON
;CACHE PARITY ERROR (ABORT MODE)?

15476	057750	001401		BEQ	1\$:YES
15477	057752	10404i		ERROR	41	:INSTRUCTION HAVING CACHE PARITY
15478						:ERROR WAS NOT ABORTED, IN THE
15479						:CACHE ABORT MODE.
15480	057754	076600		1\$: MED		:READ THE "LOG SERVICE" REGISTER
15481	057756	000101		RDLSERVICE		
15482	057760	010004		MOV	R0,R4	:COPY
15483	057762	042704	177435	BIC	#^C<LO+HI+TAG+BIT11>,R4	:MASK ALL BUT LO,HI,TAG BITS
15484	057766	022704	000342	CMP	#342,R4	:LO,HI ,TAG, CACHE PARITY BITS SET? IN "SERVICE"
15485	057772	001401		BEQ	2\$:YES
15486	057774	104042		ERROR	42	:*** "LO BYTE" PARITY ERROR
15487						:AND "TAG" PARITY ERROR BITS
15488						:WERE NOT LOGGED CORRECTLY IN "LOG
15489						:SERVICE" REGISTER, WHEN PARITY
15490						:ERROR TRAP WAS FORCED.
15491						:CLEAR BITS ARE ACTIVE.
15492	057776	013700	177744	2\$: MOV	@#MEMERR,R0	:GET MEM ERR REG
15493	060002	022700	100340	CMP	#HI+LO+TAG+BIT15,R0	:DID "LO BYTE" "HI BYTE" AND "TAG"
15494						:PARITY ERROR BITS SET IN
15495						:THE MEMORY ERROR REGISTER?
15496	060006	001401		BEQ	3\$:YES
15497	060010	104043		ERROR	43	:*** "LO BYTE" "HI BYTE" AND "TAG" PARITY
15498						:ERROR BITS DID NOT SET
15499						:CORRECTLY IN THE MEMORY
15500						:ERROR REGISTER
15501	060012	076600		3\$: MED		:READ "LOG PBA" REGISTER
15502	060014	000102		RDLPBA		
15503	060016	020027	064046	CMP	R0,#TLOC1	:DID "LOG PBA" CONTAIN CORRECT
15504						:PHYSICAL BUS ADDRESS-WHERE
15505						:THE PARITY ERROR OCCURRED?
15506	060022	001403		BEQ	4\$:YES
15507	060024	012701	064046	MOV	#TLOC1,R1	:EXPECTED PBA
15508	060030	104020		ERROR	20	:*** PHYSICAL BUS ADDRESS
15509						:(WHERE PARITY ERROR OCCURRED)
15510						:WAS NOT LOGGED CORRECTLY
15511						:WHEN CACHE PARITY ERROR WAS FORCED
15512	060032	076600		4\$: MED		:READ "LOG CACHE TAG" REGISTER
15513	060034	000107		RDLTAG		
15514	060036	000300		SWAB	R0	
15515	060040	012701	064046	MOV	#TLOC1,R1	:SHIFT RIGHT (3 TIMES) THE 16 BIT
15516	060044	000301		SWAB	R1	
15517	060046	106201		ASRB	R1	:PHYSICAL BUS ADDRESS OF THE
15518	060050	106201		ASRB	R1	:TEST LOCATION
15519	060052	106201		ASRB	R1	
15520	060054	052701	000200	BIS	#BIT7,R1	:FUDGE TAGE BIT
15521	060060	120100		CMPB	R1,R0	:WAS THE CORRECT TAG LOGGED?
15522	060062	001401		BEQ	5\$:YES
15523	060064	104017		ERROR	17	:TAG BITS WERE NOT LOGGED
15524						:CORRECTLY, WHEN CACHE
15525						:PARITY ERROR WAS FORCED
15526	060066	076600		5\$: MED		:READ CACHE DATA
15527	060070	000106		RDLDATA		
15528	060072	020027	125252	CMP	R0,#125252	:CACHE DATA LOGGED CORRECTLY?
15529	060076	001403		BEQ	6\$	
15530	060100	012701	125252	MOV	#125252,R1	:EXPECTED DATA
15531	060104	104016		ERROR	16	

```

15532
15533 060106 012700 000001      6$:  MOV    #BIT0,R0          ;SET UP LOG CONTINUOUS
15534 060112 076600
15535 060114 000222      WRWHAMI
15536 060116 012737 060130 000004      MOV    #7$,@#4          ;SETUP CPU VECTOR
15537 060124 005737 160000      TST   @#160000         ;FORCE TIMEOUT & TRAP TO 7$
15538 060130 022626      7$:  CMP    (SP)+,(SP)+
15539 060132 012737 061220 000004      MOV    #BERR,@#4       ;RESTORE CPU VECTOR
15540 060140 076600      MED   ;READ LOG FLAG/INTERRUPT REGISTER
15541 060142 000104      RDLFGINT
15542 060144 120027 000114      CMPB  R0,#114         ;DID LO BYTE CONTAIN VECTOR 114?
15543 060150 001403      BEQ   8$
15544 060152 010037 001062      MOV    R0,@#$REGO
15545 060156 104036      ERROR 36             ;LAST INTERRUPT VECTOR WAS NOT
15546 ;LOGGED CORRECTLY IN FLAG REGISTER
15547 ;WHEN A CACHE PARITY ERROR WAS
15548 ;FORCED.
15549 060160      8$:
15550
15551 ;*****
15552 ;*TEST 766 CHECK "LOG FIRST" MODE OF ERROR LOGGING
15553 ;*THIS TEST CHECKS THE "LOG FIRST" MODE OF ERROR LOGGING.
15554 ;*THE "LOG FIRST" MODE IS ENABLED. THEN A TIME-OUT TRAP
15555 ;*IS FORCED, BIT 4 OF CPU ERROR REGISTER SHOULD BE SET.
15556 ;*THEN AN ODD ADDRESS TRAP IS FORCED. HOWEVER, THIS
15557 ;*TIME THE ERROR SHOULD NOT BE LOGGED; BIT 6 (ODD
15558 ;*ADDRESS) SHOULD NOT BE SET BECAUSE THE ERROR LOG
15559 ;*IS LOCKED UP AFTER THE FIRST ERROR.
15560
15561 ;*THEN, THE ERROR LOG IS ENABLED (BY SETTING BIT 0 OF
15562 ;*WHAMI). 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 060160      TST766:
15567 060160 012700 000765      MOV    #765,R0        ;;SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
15568 060164 000004      SCOPE ;CALL THE SCOPE LOOP UTILITY
15569
15570 060166 012700 100001      MOV    #BIT15+BIT0,R0 ;SET UP "LOG FIRST MODE
15571 060172 076600      MED
15572 060174 000222      WRWHAMI
15573 060176 012737 060220 000004      MOV    #1$,@#4        ;SET UP NEW PC & PSW FOR
15574 060204 012737 000340 000006      MOV    #340,@#6      ;TIMEOUT
15575 060212 005737 160000      TST   @#160000       ;FORCE A TIMEOUT
15576 060216 000462      BR    5$             ;SKIP TEST IF NO TIMEOUT
15577
15578 060220 022626      1$:  CMP    (SP)+,(SP)+    ;RESTORE STACK
15579 ;BIT 4 OF CPU ERROR REGISTER
15580 ;SHOULD HAVE SET
15581 060222 012737 060236 000004      MOV    #2$,@#4        ;SET UP NEW PC FOR ODD ADDRESS
15582 060230 005767 177765      TST   1$+1           ;FORCE ODD ADDRESS TRAP
15583 060234 000453      BR    5$             ;SKIP TEST IF NO ODD ADDRESS TRAP
15584
15585 060236 022626      2$:  CMP    (SP)+,(SP)+    ;RESTORE STACK
15586 060240 012737 061220 000004      MOV    #BERR,@#4
15587 060246 076600      MED
  
```

```
15588 060250 000100 RDLJAM
15589 060252 013701 177766 MOV @#CPUERR,R1
15590 060256 022701 000020 CMP #BIT4,R1 ;"TIMEOUT" BIT SHOULD BE STILL
15591 ;SET, CHECK?
15592 060262 001402 BEQ 3$
15593 060264 104033 ERROR 33 ;*** SECOND ERROR (ODD ADDRESS)
15594 ;UPDATED THE ERROR LOG IN
15595 ;THE LOG FIRST MODE. BIT 4
15596 ;(UNIBUS TIMEOUT) SHOULD BE
15597 ;STILL SET FROM THE FIRST
15598 ;ERROR
15599 060266 000436 BR 5$ ;SKIP THE REST
15600 060270 032700 100004 3$: BIT #BIT2+BIT15,R0 ;CHECK THAT ODD ADRES ERROR BITS NOT
15601 060274 001401 BEQ 6$ ;SET IN LOG JAM. NOTE LOG FIRST
15602 ;MODE SHOULD INHIBIT FURTHER
15603 ;ERROR LOGGING
15604 060276 104037 ERROR 37 ;ODD ADDRESS ERROR BITS GOT SET IN LOG JAM
15605 ;THEY SHOULD NOT BE SINCE LOG FIRST MODE
15606 ;INHIBITS ERROR LOGGING AFTER THE FIRST ERROR
15607 060300 012700 100001 6$: MOV #BIT15+BIT0,R0 ;ENABLE ERROR LOG AGAIN IN
15608 ;LOG FIRST MODE
15609 060304 076600 MED
15610 060306 000222 WRWHAMI
15611 060310 012737 060332 000004 MOV #4$,@#4 ;SET UP NEW PC & PSW FOR
15612 060316 012737 000340 000006 MOV #340,@#6 ;ODD ADDRESS ERROR
15613 060324 005767 177741 TST 3$+1 ;FORCE ODD ADDRESS TRAP
15614 060330 000415 BR 5$ ;SKIP IF NO TRAP
15615 060332 022626 4$: CMP (SP)+,(SP)+ ;RESTORE STACK
15616 ;RESTORE OLD PC(4), PSW(6)
15617 060334 012737 061220 000004 MOV #BERR,@#4
15618 060342 022737 000100 177766 CMP #BIT6,@#CPUERR ;THE ERROR LOG FROM PREVIOUS
15619 ;ERROR SHOULD BE OVER WRITTEN.
15620 ;ODD ADDRESS BIT SHOULD
15621 ;BE SET, BECAUSE THE ERROR
15622 060350 001405 BEQ 5$ ;LOG WAS ENABLED.
15623 ;OK, IF YES
15624 060352 076600 MED
15625 060354 000100 RDLJAM
15626 060356 013701 177766 MOV @#CPUERR,R1
15627 060362 104040 ERROR 40 ;THE ERROR LOG WAS NOT UPDATED
15628 ;(UPON AN ODD ADDRESS ERROR)
15629 ;AFTER THE LOG WAS ENABLED.
15630 ;AT THIS FORMAT BIT 6 OF
15631 ;CPU ERROR REGISTER SHOULD
15632 ;BE SET. IT WAS NOT.
15633 060364 012737 061220 000004 5$: MOV #BERR,@#4 ;RESTORE OLD PC(4), PSW(6)
15634 060372 012700 000001 MOV #BIT0,R0
15635 060376 076600 MED
15636 060400 000222 WRWHAMI ;PUT THE LOGGING BACK INTO
15637 ;"CONTINUOUS" MODE
15638 ;*****
15639 ;*TEST 767 CHECK LAST INTRRUPT VECTOR IS LOGGED IN FLAG REG.
15640 ;*****
15641 060402 TST767:
15642 060402 012700 000766 MOV #766,R0 ;:SETUP MISSED TEST & FULL WRD TEST # PRIOR TO SCOPE
15643 060406 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
```

```
15644
15645 060410 012737 060420 000030      MOV      #1$,@#30      ;LOAD EMT VECTOR WITH 1$
15646 060416 104000                      EMT                    ;FIRST INTERRUPT -- EMT
15647 060420 022626                      CMP      (SP)+,(SP)+   ;CLEAN UP STACK
15648 060422 012737 061620 000030      MOV      #ERROR,@#30  ;RESTORE VECTOR
15649 060430 012737 060442 000004      MOV      #2$,@#4      ;SET UP CPU VECTOR
15650 060436 005737 160000                      TST      @#160000     ;FORCE TIMEOUT
15651 060442 022626                      CMP      (SP)+,(SP)+   ;CLEAN UP STACK
15652 060444 012737 061220 000004      MOV      #BERR,@#4    ;RESTORE BUS ERROR VECTOR
15653 060452 076600                      MED                      ;CHECK FLAG
15654 060454 000104                      RDLFGINT
15655 060456 120027 000030      CMPB     RO,#30        ;EMT VECTOR LAST LOGGED?
15656 060462 001401                      BEQ      3$            ;BR IF YES
15657 060464 104036                      ERROR    36           ;LOG FLAG/INT REG DID NOT LOG VECTOR
15658
15659
15660 060466 012737 060476 000020      3$:      MOV      #4$,@#20      ;LOAD IOT VECTOR WITH 4$
15661 060474 000004                      IOT                    ;SECOND INTERRUPT-SHOULD LOAD LOG FLAG REG
15662 060476 022626                      4$:      CMP      (SP)+,(SP)+   ;CLEANUP STACK
15663 060500 012737 061260 000020      MOV      #SCOPE,@#20  ;RESTORE IOT VECTOR
15664 060506 012737 060520 000004      MOV      #5$,@#4      ;SET UP CPU VECTOR
15665 060514 005737 160000                      TST      @#160000     ;FORCE TIMEOUT
15666 060520 022626                      5$:      CMP      (SP)+,(SP)+   ;CLEAN UP STACK
15667 060522 012737 061220 000004      MOV      #BERR,@#4    ;RESTORE BUS ERROR VECTOR
15668 060530 076600                      MED                      ;CHECK FLAG
15669 060532 000104                      RDLFGINT
15670 060534 120027 000020      CMPB     RO,#20        ;IOT VECTOR LAST LOGGED?
15671 060540 001401                      BEQ      6$            ;BR IF YES
15672 060542 104036                      ERROR    36           ;LOG FLAG/INT REG DID NOT LOG VECTOR
15673
15674
15675 060544 012700 000767      6$:      MOV      #STN-1,RO     ;SET UP FOR MISSED TEST CHECK AND
15676
15677
15678                      .ENABLE AMA
15679
15680
15681                      .SBTTL  END OF PASS ROUTINE
15682
15683                      ;*****
15684                      ;*INCREMENT THE PASS NUMBER ($PASS)
15685                      ;*IF THERESA MONITOR GO TO IT
15686                      ;*IF THERE ISN'T JUMP TO INIT
15687
15688                      $EOP:
15689                      SCOPE
15690                      CLR      $TIMES      ;;ZERO THE NUMBER OF ITERATIONS
15691                      INC      $PASS      ;;INCREMENT THE PASS NUMBER
15692                      BIC      #100000,$PASS ;;DON'T ALLOW A NEG. NUMBER
15693                      DEC      (PC)+      ;;LOOP?
15694                      $EOPCT: .WORD    1
15695                      BGT      $DOAGN     ;;YES
15696                      MOV      (PC)+,@(PC)+ ;;RESTORE COUNTER
15697                      $ENDCT: .WORD    1
15698                      $EOPCT
15699                      TYPE      ,EOP1      ;TYPE 'END PASS #'
```

15700	060610	013746	001126		MOV	\$PASS,-(SP)		::SAVE \$PASS FOR TYP0UT
15701	060614	104402			TYPOC			::TYPE PASS NUMBER IN OCTAL
15702	060616	104401	065121		TYPE	,EOP2		::TYPE "ERROR COUNT ="
15703	060622	013746	001012		MOV	\$ERTTL,-(SP)		::SAVE ERROR TOTAL FOR TYP0UT
15704	060626	104402			TYPOC			::TYPE ERROR TOTAL
15705	060630	104401	001115		TYPE	, \$CRLF		
15706	060634	013700	000042	\$GET42:	MOV	@#42,R0		::GET MONITOR ADDRESS
15707	060640	001405			BEQ	\$DOAGN		::BRANCH IF NO MONITOR
15708	060642	000005			RESET			::CLEAR THE WORLD
15709	060644	004710		\$ENDAD:	JSR	PC,(R0)		::GO TO MONITOR
15710	060646	000240			NOP			::SAVE ROOM
15711	060650	000240			NOP			::FOR
15712	060652	000240			NOP			::ACT11
15713	060654			\$DOAGN:				
15714	060654	000137			JMP	@(PC)+		::RETURN
15715	060656	003262		\$RTNAD:	.WORD	INIT		
15716	060660	377	377	\$ENULL:	.BYTE	-1,-1,0		::NULL CHARACTER STRING
15717		060664			.EVEN			

15718
15719
15720
15721
15722
15723
15724
15725
15726 060664 012737 061036 000024
15727 060672 012737 000340 000026
15728 060700 010046
15729 060702 010146
15730 060704 010246
15731 060706 010346
15732 060710 010446
15733 060712 010546
15734 060714 017746 120120
15735 060720 010637 061042
15736 060724 012737 060736 000024
15737 060732 000000
15738 060734 000776
15739
15740
15741
15742 060736 012737 061036 000024
15743 060744 013706 061042
15744 060750 005037 061042
15745 060754 005237 061042
15746 060760 001375
15747 060762 011600
15748 060764 076600
15749 060766 000226
15750 060770 012677 120044
15751 060774 012605
15752 060776 012604
15753 061000 012603
15754 061002 012602
15755 061004 012601
15756 061006 012600
15757 061010 012737 060664 000024
15758 061016 012737 000340 000026
15759 061024 104401
15760 061026 061044
15761 061030 012716
15762 061032 061054
15763 061034 000002
15764 061036 000000
15765 061040 000776
15766 061042 000000
15767 061044 005015 047520 042527
15768 061052 000122
15769
15770
15771 061054 012706 001000
15772 061060 005037 177776
15773 061064 000137 001630

;XX
; .SBTTL / / / / / / / UTILITIES / / / / / / /
;XX

.SBTTL POWER DOWN AND UP ROUTINES

:::*****

:POWER DOWN ROUTINE

\$PWRDN: MOV #SILLUP,@#PWRVEC ;;SET FOR FAST UP
MOV #340,@#PWRVEC+2 ;;PRIO:7
MOV R0,-(SP) ;;PUSH R0 ON STACK
MOV R1,-(SP) ;;PUSH R1 ON STACK
MOV R2,-(SP) ;;PUSH R2 ON STACK
MOV R3,-(SP) ;;PUSH R3 ON STACK
MOV R4,-(SP) ;;PUSH R4 ON STACK
MOV R5,-(SP) ;;PUSH R5 ON STACK
MOV @SWR,-(SP) ;;PUSH @SWR ON STACK
MOV SP,\$SAVR6 ;;SAVE SP
MOV #PWRUP,@#PWRVEC ;;SET UP VECTOR
HALT
BR -2 ;;HANG UP

:::*****

:POWER UP ROUTINE

\$PWRUP: MOV #SILLUP,@#PWRVEC ;;SET FOR FAST DOWN
MOV \$SAVR6,SP ;;GET SP
CLR \$SAVR6 ;;WAIT LOOP FOR THE TTY
1\$: INC \$SAVR6 ;;WAIT FOR THE INC
BNE 1\$;;OF WORD
MOV (SP),R0 ;;GET OLD SWR VALUE
MED ;;WRITE BACK ORIGINAL SWR VALUE
WCNSSW ;;INTO HARDWARE SWITCH REGISTER
MOV (SP)+,@SWR ;;POP STACK INTO @SWR
MOV (SP)+,R5 ;;POP STACK INTO R5
MOV (SP)+,R4 ;;POP STACK INTO R4
MOV (SP)+,R3 ;;POP STACK INTO R3
MOV (SP)+,R2 ;;POP STACK INTO R2
MOV (SP)+,R1 ;;POP STACK INTO R1
MOV (SP)+,R0 ;;POP STACK INTO R0
MOV #PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
MOV #340,@#PWRVEC+2 ;;PRIO:7
TYPE ;;REPORT THE POWER FAILURE

\$PWRMG: .WORD \$POWER ;;POWER FAIL MESSAGE POINTER

MOV (PC)+,(SP) ;;RESTART AT PWRUP

\$PWRAD: .WORD PWRUP ;;RESTART ADDRESS

RTI

\$SILLUP: HALT ;;THE POWER UP SEQUENCE WAS STARTED

BR -2 ;; BEFORE THE POWER DOWN WAS COMPLETE

\$SAVR6: 0 ;;PUT THE SP HERE

\$POWER: .ASCIZ <15><12>'POWER'

.EVEN

PWRUP: MOV #STACK,SP ;;RESET SP

CLR @#PSW ;;PRIORITY 0 -- CLEAR CODES

JMP @#START ;;RESTART PROGRAM

15774
15775
15776
15777
15778
15779
15780
15781
15782
15783
15784
15785
15786
15787
15788
15789
15790
15791
15792
15793
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

061070 062716 000002
061074 042766 000020 000002
061102 000006

061104 005237 061112
061110 000002
061112 000000

```
; *****  
; .SBTTL 'T' BIT SERVICE ROUTINE  
; *****  
TBSER: ADD #2,(SP) ;MOVE RETURN PC AROUND ERROR CALL  
BIC #20,2(SP) ;TURN OFF THE 'T' BIT  
RTT ;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
```

15830 061114 005137 063246
 15831 061120 000002
 15832
 15833 061122 005737 063252
 15834 061126 001025
 15835 061130 105237 063253
 15836 061134 032777 010000 117676
 15837 061142 001015
 15838 061144 104401
 15839 061146 065233
 15840 061150 011646
 15841 061152 104402
 15842 061154 104401
 15843 061156 001115
 15844 061160 005237 001012
 15845 061164 032777 100000 117646
 15846 061172 001401
 15847 061174 000000
 15848 061176 000137 003262
 15849 061202 105237 063253
 15850 061206 000000
 15851 061210 000772

```

RSVTST: 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     RESTAR     ;GO TYPE "TRAPPED TO 10 PC="

RSBERT: MOV      (SP),-(SP)  ;GET ERROR PC ON STACK FOR PRINTING
          TYPOC    ;TYPE THE ERROR PC
          TYPE     ;OUTPUT CR / LF
          $CRLF
          INC      @#SBTTL    ;COUNT THE ERROR
          BIT      #BIT15,@SWR ;HALT ON ERROR?
          BEQ      RESTAR     ;BR IF NOT
          HALT     ;HALT ON ERROR--PRESS CONTINUE TO RESTART.

RESTAR: 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.
  
```

15852
 15853
 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

```

; *****
; .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 LOLLOWING
;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
  
```


15886
15887
15888
15889
15890
15891
15892
15893
15894
15895
15896
15897
15898 061212 005137 063250
15899 061216 000002
15900
15901 061220 005737 063252
15902 061224 001011
15903 061226 105237 063252
15904 061232 032777 010000 117600
15905 061240 001356
15906 061242 104401
15907 061244 065206
15908 061246 000740
15909
15910 061250 105237 063252
15911 061254 000000
15912 061256 000747
15913
15914
15915
15916
15917
15918
15919
15920
15921
15922
15923
15924
15925
15926
15927
15928
15929 061260
15930 061260 020037 001124
15931 061264 001406
15932 061266 012737 061276 001112
15933 061274 104011
15934 061276 005037 001112
15935 061302 110037 001002
15936 061306 032777 002000 117524
15937 061314 001411
15938 061316 017737 117516 063242
15939 061324 042737 177000 063242
15940 061332 020037 063242
15941 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?
2\$;BR IF YES
BNE 2\$
INCB @#CATERR ;SET CATASTROPHIC ERROR FLAG
BIT #SW12,@SWR ;INHIBIT ERROR PRINT
BNE RESTAR ;BR IF YES
TYPE ;PRINT "TRAP TO 4" MESSAGE
BEMSG
BR RSBERT ;TYPE REST OF BUS ERROR MESSAGE

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(\$TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
;AND LOAD THE ERROR FLAG (\$ERFLG) INTO DISPLAY<15:08>
;THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
;SW14=1 LOOP ON TEST
;SW11=1 INHIBIT ITERATIONS
;SW09=1 LOOP ON ERROR
;CALL
;* SCOPE ;:SCOPE=10T

\$SCOPE:
CMP R0,@#STESTN ;ANY MISSED TESTS ?
BEQ 10\$;BR IF NOT
MOV #12\$,@#\$ESCAPE ;NO ERROR LOOPING
ERROR 11 ;MISSED TESTS ERROR CALL
12\$: CLR @#\$ESCAPE ;NORMAL ERROR LOOPING
10\$: MOVB R0,@#STSTNM ;INSURE TSTNUM IS CORRECT
BIT #SW10,@SWR ;LOOP ON SELECTED TEST?
BEQ 11\$;BR IF NO
MOV @SWR,@#SELTST ;GET CONTENTS OF SWITCHES
BIC #177000,@#SELTST ;MASK OUT SWR<15:9>
CMP R0,@#SELTST ;IS THIS THE SELECTED TEST?
BEQ \$OVER ;BR IF YES

```

15942 061340 11$:
15943 061340 032777 040000 117472 1$: BIT #BIT14,@SWR ;;LOOP ON PRESENT TEST?
15944 061346 001104 BNE $OVER ;;YES IF SW14=1
15945 :#####START OF CODE FOR THE XOR TESTER#####
15946 061350 000416 $XTSTR: BR 6$ ;;IF RUNNING ON THE "XOR" TESTER CHANGE
15947 ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
15948 061352 013746 000004 MOV @#ERRVEC,-(SP) ;;SAVE THE CONTENTS OF THE ERROR VECTOR
15949 061356 012737 061376 000004 MOV #5$,@#ERRVEC ;;SET FOR TIMEOUT
15950 061364 005737 177060 TST @#177060 ;;TIME OUT ON XOR?
15951 061370 012637 000004 MOV (SP)+,@#ERRVEC ;;RESTORE THE ERROR VECTOR
15952 061374 000453 BR $SVLAD ;;GO TO THE NEXT TEST
15953 061376 022626 5$: CMP (SP)+,(SP)+ ;;CLEAR THE STACK AFTER A TIME OUT
15954 061400 012637 000004 MOV (SP)+,@#ERRVEC ;;RESTORE THE ERROR VECTOR
15955 061404 000413 BR 7$ ;;LOOP ON THE PRESENT TEST
15956 061406 6$:;#####END OF CODE FOR THE XOR TESTER#####
15957 061406 105737 001003 2$: TSTB $ERFLG ;;HAS AN ERROR OCCURRED?
15958 061412 001421 BEQ 3$ ;;BR IF NO
15959 061414 123737 001015 001003 CMPB $ERMAX,$ERFLG ;;MAX. ERRORS FOR THIS TEST OCCURRED?
15960 061422 101015 BHI 3$ ;;BR IF NO
15961 061424 032777 001000 117406 BIT #BIT09,@SWR ;;LOOP ON ERROR?
15962 061432 001404 BEQ 4$ ;;BR IF NO
15963 061434 013737 001010 001006 7$: MOV $LPERR,$LPADR ;;SET LOOP ADDRESS TO LAST SCOPE
15964 061442 000446 BR $OVER
15965 061444 105037 001003 4$: CLRB $ERFLG ;;ZERO THE ERROR FLAG
15966 061450 005037 001110 CLR $TIMES ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
15967 061454 000415 BR 1$ ;;ESCAPE TO THE NEXT TEST
15968 061456 032777 004000 117354 3$: BIT #BIT11,@SWR ;;INHIBIT ITERATIONS?
15969 061464 001011 BNE 1$ ;;BR IF YES
15970 061466 005737 001126 TST $PASS ;;IF FIRST PASS OF PROGRAM
15971 061472 001406 BEQ 1$ ;; INHIBIT ITERATIONS
15972 061474 005237 001004 INC $ICNT ;;INCREMENT ITERATION COUNT
15973 061500 023737 001110 001004 CMP $TIMES,$ICNT ;;CHECK THE NUMBER OF ITERATIONS MADE
15974 061506 002024 BGE $OVER ;;BR IF MORE ITERATION REQUIRED
15975 061510 012737 000001 001004 1$: MOV #1,$ICNT ;;REINITIALIZE THE ITERATION COUNTER
15976 061516 013737 061610 001110 MOV $MXCNT,$TIMES ;;SET NUMBER OF ITERATIONS TO DO
15977 061524 105237 001002 $SVLAD: INCB $TSTNM ;;COUNT TEST NUMBERS
15978 061530 113737 001002 001124 MOV $TSTNM,$TESTN ;;SET TEST NUMBER IN APT MAILBOX
15979 061536 011637 001006 MOV (SP),$LPADR ;;SAVE SCOPE LOOP ADDRESS
15980 061542 011637 001010 MOV (SP),$LPERR ;;SAVE ERROR LOOP ADDRESS
15981 061546 005037 001112 CLR $ESCAPE ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
15982 061552 112737 000001 001015 MOV #1,$ERMAX ;;ONLY ALLOW ONE(i) ERROR ON NEXT TEST
15983 061560 013777 001002 117254 $OVER: MOV $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
15984 061566 013716 001006 MOV $LPADR,(SP) ;;FUDGE RETURN ADDRESS
15985 061572 120037 001002 CMPB R0,@#$TSTNM ;;WAS $TSTNM INCREMENTED?
15986 061576 001401 BEQ 10$ ;;BR IF NOT
15987 061600 005200 INC R0 ;;INCREMENT TEST NUMBER
15988 061602 010037 001124 10$: MOV R0,@#$TESTN ;;FIX $TESTN TO BE WORD COUNT, NOT BYTE
15989 061606 000002 RTI
15990 061610 000200 $MXCNT: 200 ;;MAX. NUMBER OF ITERATIONS
15991
15992 061612 005137 063244 SCOPEA: COM @#SCOFLG ;;THESE TWO INSTRUCTIONS ARE
15993 061616 000002 RTI ;;USED IN THE BASIC TESTS TO
15994 ;;VERIFY THE IOT LINKAGE
15995
15996
15997

```

```

15998
15999
16000
16001
16002
16003
16004
16005
16006
16007
16008
16009
16010
16011 061620
16012 061620 010546
16013 061622 012705 001060
16014 061626 016625 000004
16015 061632 010025
16016 061634 010125
16017 061636 010225
16018 061640 010325
16019 061642 010425
16020 061644 022715 177777
16021 061650 001001
16022 061652 010615
16023 061654 012605
16024 061656 105237 001003
16025 061662 001775
16026 061664 013777 001002 117150
16027 061672 005237 001012
16028 061676 011637 001016
16029 061702 162737 000002 001016
16030 061710 117737 117102 001014
16031 061716 032777 020000 117114
16032 061724 001004
16033 061726 004737 062046
16034 061732 104401 001115
16035 061736
16036 061736 122737 000001 001140
16037 061744 001007
16038 061746 113737 001014 061760
16039 061754 004737 062736
16040 061760 000
16041 061761 000
16042 061762 000777
16043 061764 005777 117050
16044 061770 100001
16045 061772 000000
16046 061774 032777 001000 117036
16047 062002 001402
16048 062004 013716 001010
16049 062010 005737 001112
16050 062014 001402
16051 062016 013716 001112
16052 062022
16053 062022 012737 177777 001074
  
```

```

.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 $ERRTYP 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 #REGAD,R5 ;GET POINTER
MOV 4(SP),(R5)+ ;SAVE ERROR PSW IN $REGAD FOR TYP0UT
MOV R0,(R5)+ ;SAVE R0 FOR TYPEOUTS
MOV R1,(R5)+ ;SAVE R1 IN $REG1
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
MOV SP,(R5) ;PUT SP IN $REG5 FOR TYP0UT
10$: MOV (SP)+,R5 ;RESTORE R5
7$: INCB $ERFLG ;;SET THE ERROR FLAG
BEQ 7$ ;;DON'T LET THE FLAG GO TO ZERO
MOV $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
INC $ERTTL ;;INC THE ERROR COUNT
MOV (SP),$ERRPC ;;GET ADDRESS OF ERROR INSTRUCTION
SUB #2,$ERRPC
MOVB @ERRPC,$ITEMB ;;STRIP AND SAVE THE ERROR ITEM CODE
BIT #BIT13,@SWR ;;SKIP TYPEOUT IF SET
BNE 20$ ;;SKIP TYPEOUTS
JSR PC,$ERRTYP ;;GO TO USER ERROR ROUTINE
TYPE ,SRLF

20$: CMPB #APTENV,$ENV ;;RUNNING IN APT MODE
BNE 2$ ;;NO,SKIP APT ERROR REPORT
MOVB $ITEMB,21$ ;;SET ITEM NUMBER AS ERROR NUMBER
JSR PC,$ATY4 ;;REPORT FATAL ERROR TO APT

21$: .BYTE 0
.BYTE 0
22$: BR 22$ ;;APT ERROR LOOP
2$: TST @SWR ;;HALT ON ERROR
BPL 3$ ;;SKIP IF CONTINUE
HALT ;;HALT ON ERROR!
3$: BIT #BIT09,@SWR ;;LOOP ON ERROR SWITCH SET?
BEQ 4$ ;;BR IF NO
MOV $LPERR,(SP) ;;FUDGE RETURN FOR LOOPING
4$: TST $ESCAPE ;;CHECK FOR AN ESCAPE ADDRESS
BEQ 5$ ;;BR IF NONE
MOV $ESCAPE,(SP) ;;FUDGE RETURN ADDRESS FOR ESCAPE

5$: MOV #-1,@REG5 ;FLAG CURRENT STACK POINTER TO BE TYPED
  
```

16054 062030 042766 000020 000002
 16055
 16056 062036 000002
 16057
 16058 062040 005137 063240
 16059 062044 000002
 16060
 16061
 16062
 16063
 16064
 16065
 16066
 16067
 16068 062046
 16069 062046 104401 001115
 16070 062052 010046
 16071 062054 005000
 16072 062056 153700 001014
 16073 062062 001004
 16074
 16075 062064 013746 001016
 16076
 16077 062070 104402
 16078 062072 000426
 16079 062074 005300
 16080 062076 006300
 16081 062100 006300
 16082 062102 006300
 16083 062104 062700 001150
 16084 062110 012037 062120
 16085 062114 001404
 16086 062116 104401
 16087 062120 000000
 16088 062122 104401 001115
 16089 062126 012037 062136
 16090 062132 001404
 16091 062134 104401
 16092 062136 000000
 16093 062140 104401 001115
 16094 062144 011000
 16095 062146 001004
 16096 062150 012600
 16097 062152 104401 001115
 16098 062156 000207
 16099 062160
 16100 062160 013046
 16101 062162 104402
 16102 062164 005710
 16103 062166 001770
 16104 062170 104401 062176
 16105 062174 000771
 16106 062176 020040 000
 16107 062202
 16108
 16109

BIC #20,2(SP) ;CLEAR T BIT IN CASE ERROR OCCURED
 ;IN T BIT TESTS
 RTI
 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" (\$ITEMB) TO DETERMINE WHICH
 ;*ERROR IS TO BE REPORTED. IT THEN OBTAINS, FROM THE "ERROR TABLE" (\$ERRTB),
 ;*AND REPORTS THE APPROPRIATE INFORMATION CONCERNING THE ERROR.

\$ERRTYP:

TYPE ,%CRLF ;:"CARRIAGE RETURN" & "LINE FEED"
 MOV RO,-(SP) ;:SAVE RO
 CLR RO ;:PICKUP THE ITEM INDEX
 BISB @#\$ITEMB,RO
 BNE 1\$;:IF ITEM NUMBER IS ZERO, JUST
 ;:TYPE THE PC OF THE ERROR
 ;:SAVE \$ERRPC FOR TYPEOUT
 ;:ERROR ADDRESS
 ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
 ;:GET OUT
 ;:ADJUST THE INDEX SO THAT IT WILL
 ;:WORK FOR THE ERROR TABLE
 MOV \$ERRPC,-(SP)
 TYPOC
 BR 6\$
 1\$: DEC RO
 ASL RO
 ASL RO
 ASL RO
 ADD #\$ERRTB,RO ;:FORM TABLE POINTER
 MOV (RO)+,2\$;:PICKUP "ERROR MESSAGE" POINTER
 BEQ 3\$;:SKIP TYPEOUT IF NO POINTER
 TYPE ;:TYPE THE "ERROR MESSAGE"
 ;:"ERROR MESSAGE" POINTER GOES HERE
 2\$: .WORD 0
 TYPE ,%CRLF ;:"CARRIAGE RETURN" & "LINE FEED"
 3\$: MOV (RO)+,4\$;:PICKUP "DATA HEADER" POINTER
 BEQ 5\$;:SKIP TYPEOUT IF 0
 TYPE ;:TYPE THE "DATA HEADER"
 ;:"DATA HEADER" POINTER GOES HERE
 4\$: .WORD 0
 TYPE ,%CRLF ;:"CARRIAGE RETURN" & "LINE FEED"
 5\$: MOV (RO),RO ;:PICKUP "DATA TABLE" POINTER
 BNE 7\$;:GO TYPE THE DATA
 ;:RESTORE RO
 6\$: MOV (SP)+,RO
 TYPE ,%CRLF ;:"CARRIAGE RETURN" & "LINE FEED"
 RTS PC ;:RETURN
 7\$: MOV @ (RO)+,-(SP) ;:SAVE @ (RO)+ FOR TYPEOUT
 TYPOC ;:GO TYPE--OCTAL ASCII(ALL DIGITS)
 TST (RO) ;:IS THERE ANOTHER NUMBER?
 BEQ 6\$;:BR IF NO
 TYPE ,8\$;:TYPE TWO(2) SPACES
 BR 7\$;:LOOP
 8\$: .ASCIZ / / ;:TWO(2) SPACES
 .EVEN

16110
 16111
 16112
 16113 062202 005137 063236
 16114 062206 000002
 16115
 16116
 16117
 16118
 16119
 16120
 16121
 16122
 16123
 16124
 16125
 16126
 16127
 16128
 16129
 16130
 16131
 16132
 16133
 16134 062210 105737 001057
 16135 062214 100002
 16136 062216 000000
 16137 062220 000430
 16138 062222 010046
 16139 062224 017600 000002
 16140 062230 122737 000001 001140
 16141 062236 001011
 16142 062240 132737 000100 001141
 16143 062246 001405
 16144 062250 010037 062260
 16145 062254 004737 062726
 16146 062260 000000
 16147 062262 132737 000040 001141
 16148 062270 001003
 16149 062272 112046
 16150 062274 001005
 16151 062276 005726
 16152 062300 012600
 16153 062302 062716 000002
 16154 062306 000002
 16155 062310 122716 000011
 16156 062314 001430
 16157 062316 122716 000200
 16158 062322 001006
 16159 062324 005726
 16160 062326 104401
 16161 062330 001115
 16162 062332 105037 062466
 16163 062336 000755
 16164 062340 004737 062422
 16165 062344 123726 001056

```

.SBTTL PRINT ROUTINES
; *****
PRINA: COM @#PRIFLG ;THESE TWO INSTRUCTIONS ARE
RTI ;USED BY THE BASIC TESTS TO VERIFY
;THE TRAP INSTRUCTION

.SBTTL TYPE ROUTINE
;*****
;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
;*NOTE1: $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
;*NOTE2: $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
;*NOTE3: $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
;
;*CALL:
;*1) USING A TRAP INSTRUCTION
;* TYPE ,MESADR ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
;*OR
;* TYPE
;* MESADR
;
$TYPE: TSTB $TPFLG ;;IS THERE A TERMINAL?
BPL 1$ ;;BR IF YES
HALT ;;HALT HERE IF NO TERMINAL
BR 3$ ;;LEAVE
1$: MOV RO,-(SP) ;;SAVE RO
MOV @2(SP),RO ;;GET ADDRESS OF ASCIZ STRING
CMPB #APTENV,$ENV ;;RUNNING IN APT MODE
BNE 62$ ;;NO,GO CHECK FOR APT CONSOLE
BITB #APTPOOL,$ENVM ;;SPOOL MESSAGE TO APT
BEQ 62$ ;;NO,GO CHECK FOR CONSOLE
MOV RO,61$ ;;SETUP MESSAGE ADDRESS FOR APT
JSR PC,$ATY3 ;;SPOOL MESSAGE TO APT
61$: .WORD 0 ;;MESSAGE ADDRESS
62$: BITB #APTCSUP,$ENVM ;;APT CONSOLE SUPPRESSED
BNE 60$ ;;YES,SKIP TYPE OUT
2$: MOVB (RO)+,-(SP) ;;PUSH CHARACTER TO BE TYPED ONTO STACK
BNE 4$ ;;BR IF IT ISN'T THE TERMINATOR
TST (SP)+ ;;IF TERMINATOR POP IT OFF THE STACK
60$: MOV (SP)+,RO ;;RSTORE RO
3$: ADD #2,(SP) ;;ADJUST RETURN PC
RTI ;;RETURN
4$: CMPB #HT,(SP) ;;BRANCH IF <HT>
BEQ 8$
CMPB #CRLF,(SP) ;;BRANCH IF NOT <CRLF>
BNE 5$
TST (SP)+ ;;POP <CR><LF> EQUIV
TYPE ;;TYPE A CR AND LF
$CRLF
CLRB $CHARCNT ;;CLEAR CHARACTER COUNT
BR 2$ ;;GET NEXT CHARACTER
5$: JSR PC,$TYPEC ;;GO TYPE THIS CHARACTER
6$: CMPB $FILLC,(SP)+ ;;IS IT TIME FOR FILLER CHARS.?

```

```
16166 062350 001350          BNE 2$          ;; IF NO GO GET NEXT CHAR.  
16167 062352 013746 0U1054  MOV $NULL,-(SP) ;; GET # OF FILLER CHARS. NEEDED  
16168                                ;; AND THE NULL CHAR.  
16169 062356 105366 000001 7$: DECB 1(SP)    ;; DOES A NULL NEED TO BE TYPED?  
16170 062362 002770          BLT 6$          ;; BR IF NO--GO POP THE NULL OFF OF STACK  
16171 062364 004737 062422  JSR PC,$TYPEC  ;; GO TYPE A NULL  
16172 062370 105337 062466  DECB $CHARCNT  ;; DO NOT COUNT AS A COUNT  
16173 062374 000770          BR 7$          ;; LOOP
```

;HORIZONTAL TAB PROCESSOR

```
16174  
16175  
16176  
16177 062376 112716 000040 8$: MOVB #' ,(SP)  ;; REPLACE TAB WITH SPACE  
16178 062402 004737 062422 9$: JSR PC,$TYPEC  ;; TYPE A SPACE  
16179 062406 132737 000007 062466 BITB #7,$CHARCNT ;; BRANCH IF NOT AT  
16180 062414 001372          BNE 9$          ;; TAB STOP  
16181 062416 005726          TST (SP)+      ;; POP SPACE OFF STACK  
16182 062420 000724          BR 2$          ;; GET NEXT CHARACTER  
16183 062422 105777 116422 $TYPEC: TSTB @STPS ;; WAIT UNTIL PRINTER IS READY  
16184 062426 100375          BPL $TYPEC  
16185 062430 116677 000002 116414 MOVB 2(SP),@STPB ;; LOAD CHAR TO BE TYPED INTO DATA REG.  
16186 062436 122766 000015 000002 CMPB #CR,2(SP)  ;; IS CHARACTER A CARRIAGE RETURN?  
16187 062444 001003          BNE 1$          ;; BRANCH IF NO  
16188 062446 105037 062466 CLRB $CHARCNT  ;; YES--CLEAR CHARACTER COUNT  
16189 062452 000406          BR $TYPEX      ;; EXIT  
16190 062454 122766 000012 000002 1$: CMPB #LF,2(SP) ;; IS CHARACTER A LINE FEED?  
16191 062462 001402          BEQ $TYPEX      ;; BRANCH IF YES  
16192 062464 105227          INCB (PC)+    ;; COUNT THE CHARACTER  
16193 062466 000000 $CHARCNT: .WORD 0 ;; CHARACTER COUNT STORAGE  
16194 062470 000207 $TYPEX: RTS PC
```

.SBTTL BINARY TO OCTAL (ASCII) AND TYPE

```
16195  
16196  
16197  
16198  
16199  
16200  
16201  
16202  
16203  
16204  
16205  
16206  
16207  
16208  
16209  
16210  
16211  
16212  
16213  
16214  
16215  
16216  
16217  
16218  
16219  
16220  
16221
```

```
*****  
*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT  
*OCTAL (ASCII) NUMBER AND TYPE IT.  
*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE  
*CALL:  
*   MOV   NUM,-(SP)      ;;NUMBER TO BE TYPED  
*   TYPOS                ;;CALL FOR TYPEOUT  
*   .BYTE N              ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE  
*   .BYTE M              ;;M=1 OR 0  
*                               ;;1=TYPE LEADING ZEROS  
*                               ;;0=SUPPRESS LEADING ZEROS  
*$TYPON---ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST  
*$TYPOS OR $TYPOC  
*CALL:  
*   MOV   NUM,-(SP)      ;;NUMBER TO BE TYPED  
*   TYPON                ;;CALL FOR TYPEOUT  
*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER  
*CALL:  
*   MOV   NUM,-(SP)      ;;NUMBER TO BE TYPED  
*   TYPOC                ;;CALL FOR TYPEOUT
```

```
16222 062472 017646 000000
16223 062476 116637 000001 062715
16224 062504 112637 062717
16225 062510 062716 000002
16226 062514 000406
16227 062516 112737 000001 062715
16228 062524 112737 000006 062717
16229 062532 112737 000005 062714
16230 062540 010346
16231 062542 010446
16232 062544 010546
16233 062546 113704 062717
16234 062552 005404
16235 062554 062704 000006
16236 062560 110437 062716
16237 062564 113704 062715
16238 062570 016605 000012
16239 062574 005003
16240 062576 006105
16241 062600 000404
16242 062602 006105
16243 062604 006105
16244 062606 006105
16245 062610 010503
16246 062612 006103
16247 062614 105337 062716
16248 062620 100016
16249 062622 042703 177770
16250 062626 001002
16251 062630 005704
16252 062632 001403
16253 062634 005204
16254 062636 052703 000060
16255 062642 052703 000040
16256 062646 110337 062712
16257 062652 104401 062712
16258 062656 105337 062714
16259 062662 003347
16260 062664 002402
16261 062666 005204
16262 062670 000744
16263 062672 012605
16264 062674 012604
16265 062676 012603
16266 062700 016666 000002 000004
16267 062706 012616
16268 062710 000002
16269 062712 000
16270 062713 000
16271 062714 000
16272 062715 000
16273 062716 000000
16274
16275
16276
16277

$TYPOS: MOV @ (SP), -(SP) :: PICKUP THE MODE
MOV 1 (SP), $OFILL :: LOAD ZERO FILL SWITCH
MOV (SP)+, $SOMODE+1 :: NUMBER OF DIGITS TO TYPE
ADD #2, (SP) :: ADJUST RETURN ADDRESS
BR $TYPON
$TYPOC: MOV #1, $OFILL :: SET THE ZERO FILL SWITCH
MOV #6, $SOMODE+1 :: SET FOR SIX(6) DIGITS
$TYPON: MOV #5, $SOCNT :: SET THE ITERATION COUNT
MOV R3, -(SP) :: SAVE R3
MOV R4, -(SP) :: SAVE R4
MOV R5, -(SP) :: SAVE R5
MOV $SOMODE+1, R4 :: GET THE NUMBER OF DIGITS TO TYPE
NEG R4
ADD #6, R4 :: SUBTRACT IT FOR MAX. ALLOWED
MOV R4, $SOMODE :: SAVE IT FOR USE
MOV $OFILL, R4 :: GET THE ZERO FILL SWITCH
MOV 12 (SP), R5 :: PICKUP THE INPUT NUMBER
CLR R3 :: CLEAR THE OUTPUT WORD
1$: ROL R5 :: ROTATE MSB INTO 'C'
BR 3$ :: GO DO MSB
2$: ROL R5 :: FORM THIS DIGIT
ROL R5
MOV R5, R3
3$: ROL R3 :: GET LSB OF THIS DIGIT
DECB $SOMODE :: TYPE THIS DIGIT?
BPL 7$ :: BR IF NO
BIC #177770, R3 :: GET RID OF JUNK
BNE 4$ :: TEST FOR 0
TST R4 :: SUPPRESS THIS 0?
BEQ 5$ :: BR IF YES
4$: INC R4 :: DON'T SUPPRESS ANYMORE 0'S
BIS #'0, R3 :: MAKE THIS DIGIT ASCII
5$: BIS #' , R3 :: MAKE ASCII IF NOT ALREADY
MOV R3, 8$ :: SAVE FOR TYPING
TYPE , 8$ :: GO TYPE THIS DIGIT
7$: DECB $SOCNT :: COUNT BY 1
BGT 2$ :: BR IF MORE TO DO
BLT 6$ :: BR IF DONE
INC R4 :: INSURE LAST DIGIT ISN'T A BLANK
BR 2$ :: GO DO THE LAST DIGIT
6$: MOV (SP)+, R5 :: RESTORE R5
MOV (SP)+, R4 :: RESTORE R4
MOV (SP)+, R3 :: RESTORE R3
MOV 2 (SP), 4 (SP) :: SET THE STACK FOR RETURNING
MOV (SP)+, (SP)
RTI :: RETURN
8$: .BYTE 0 :: STORAGE FOR ASCII DIGIT
.BYTE 0 :: TERMINATOR FOR TYPE ROUTINE
$SOCNT: .BYTE 0 :: OCTAL DIGIT COUNTER
$OFILL: .BYTE 0 :: ZERO FILL SWITCH
$SOMODE: .WORD 0 :: NUMBER OF DIGITS TO TYPE

.SBTTL APT COMMUNICATIONS ROUTINE
;*****
```

16278	062720	112737	000001	063164	\$ATY1:	MOVB	#1,\$FFLG	::	TO REPORT FATAL ERROR
16279	062726	112737	000001	063162	\$ATY3:	MOVB	#1,\$MFLG	::	TO TYPE A MESSAGE
16280	062734	000403				BR	\$ATYC		
16281	062736	112737	000001	063164	\$ATY4:	MOVB	#1,\$FFLG	::	TO ONLY REPORT FATAL ERROR
16282	062744				\$ATYC:				
16283	062744	010046				MOV	R0,-(SP)	::	PUSH R0 ON STACK
16284	062746	010146				MOV	R1,-(SP)	::	PUSH R1 ON STACK
16285	062750	105737	063162			TSTB	\$MFLG	::	SHOULD TYPE A MESSAGE?
16286	062754	001450				BEQ	5\$::	IF NOT: BR
16287	062756	122737	000001	001140		CMPB	#APTENV,\$ENV	::	OPERATING UNDER APT?
16288	062764	001031				BNE	3\$::	IF NOT: BR
16289	062766	132737	000100	001141		BITB	#APTSPOOL,\$ENVM	::	SHOULD SPOOL MESSAGES?
16290	062774	001425				BEQ	3\$::	IF NOT: BR
16291	062776	017600	000004			MOV	@4(SP),R0	::	GET MESSAGE ADDR.
16292	063002	062766	000002	000004		ADD	#2,4(SP)	::	BUMP RETURN ADDR.
16293	063010	005737	001120		1\$:	TST	\$MSGTYPE	::	SEE IF DONE W/ LAST XMISSION?
16294	063014	001375				BNE	1\$::	IF NOT: WAIT
16295	063016	010037	001134			MOV	R0,\$MSGAD	::	PUT ADDR IN MAILBOX
16296	063022	105720			2\$:	TSTB	(R0)+	::	FIND END OF MESSAGE
16297	063024	001376				BNE	2\$		
16298	063026	163700	001134			SUB	\$MSGAD,R0	::	SUB START OF MESSAGE
16299	063032	006200				ASR	R0	::	GET MESSAGE LNTH IN WORDS
16300	063034	010037	001136			MOV	R0,\$MSGGLT	::	PUT LENGTH IN MAILBOX
16301	063040	012737	000004	001120		MOV	#4,\$MSGTYPE	::	TELL APT TO TAKE MSG.
16302	063046	000413				BR	5\$		
16303	063050	017637	000004	063074	3\$:	MOV	@4(SP),4\$::	PUT MSG ADDR IN JSR LINKAGE
16304	063056	062766	000002	000004		ADD	#2,4(SP)	::	BUMP RETURN ADDRESS
16305	063064	013746	177776			MOV	177776,-(SP)	::	PUSH 177776 ON STACK
16306	063070	004737	062210			JSR	PC,\$TYPE	::	CALL TYPE MACRO
16307	063074	000000			4\$:	.WORD	0		
16308	063076				5\$:				
16309	063076	105737	063164		10\$:	TSTB	\$FFLG	::	SHOULD REPORT FATAL ERROR?
16310	063102	001416				BEQ	12\$::	IF NOT: BR
16311	063104	005737	001140			TST	\$ENV	::	RUNNING UNDER APT?
16312	063110	001413				BEQ	12\$::	IF NOT: BR
16313	063112	005737	001120		11\$:	TST	\$MSGTYPE	::	FINISHED LAST MESSAGE?
16314	063116	001375				BNE	11\$::	IF NOT: WAIT
16315	063120	017637	000004	001122		MOV	@4(SP),\$FATAL	::	GET ERROR #
16316	063126	062766	000002	000004		ADD	#2,4(SP)	::	BUMP RETURN ADDR.
16317	063134	005237	001120			INC	\$MSGTYPE	::	TELL APT TO TAKE ERROR
16318	063140	105037	063164		12\$:	CLRB	\$FFLG	::	CLEAR FATAL FLAG
16319	063144	105037	063163			CLRB	\$LFLG	::	CLEAR LOG FLAG
16320	063150	105037	063162			CLRB	\$MFLG	::	CLEAR MESSAGE FLAG
16321	063154	012601				MOV	(SP)+,R1	::	POP STACK INTO R1
16322	063156	012600				MOV	(SP)+,R0	::	POP STACK INTO R0
16323	063160	000207				RTS	PC	::	RETURN
16324	063162	000				\$MFLG:	.BYTE 0	::	MESSG. FLAG
16325	063163	000				\$LFLG:	.BYTE 0	::	LOG FLAG
16326	063164	000				\$FFLG:	.BYTE 0	::	FATAL FLAG
16327		063166				.EVEN			
16328		000200				APTSIZE=200			
16329		000001				APTENV=001			
16330		000100				APTSPOOL=100			
16331		000040				APTCSUP=040			
16332									
16333						.SBTTL	TRAP DECODER		

16334
16335
16336
16337
16338
16339
16340
16341 063166 010046
16342 063170 016600 000002
16343 063174 005740
16344 063176 111000
16345 063200 006300
16346 063202 016000 063222
16347 063206 000200
16348
16349
16350
16351
16352 063210 011646
16353 063212 016666 000004 000002
16354 063220 000002
16355
16356
16357
16358
16359
16360
16361
16362
16363 063222 063210
16364 063224 062210
16365 063226 062516
16366 063230 062472
16367 063232 062532
16368
16369
16370
16371
16372
16373 063234 000000
16374
16375 063236 000000
16376 063240 000000
16377 063242 000000
16378 063244 000000
16379 063246 000000
16380 063250 000000
16381 063252 000000
16382
16383 063254 000000
16384
16385
16386 063256 177400
16387 063260 177400
16388 063262 177400
16389 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.

```
$TRAP:  MOV    RO,-(SP)      ;;SAVE RO
        MOV    2(SP),RO    ;;GET TRAP ADDRESS
        TST   -(RO)       ;;BACKUP BY 2
        MOVB  (RO),RO     ;;GET RIGHT BYTE OF TRAP
        ASL   RO          ;;POSITION FOR INDEXING
        MOV   $TRPAD(RO),RO ;;INDEX TO TABLE
        RTS   RO          ;;GO TO ROUTINE
```

;;THIS IS USE TO HANDLE THE "GETPRI" MACRO

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

;;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
SELTST: 0 ;;STORES SR<8:0> FOR LOOP ON SELECTED TEST
SCOFFLG: 0 ;;USED BY BASIC TESTS FOR IOT TEST
RSVFLG: 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
```

16390
 16391 063266 000004
 16392
 16393 063276 063322
 16394 063300 064032
 16395 063302 064630
 16396 063304 064634
 16397 063306 063312
 16398 063310 063316
 16399
 16400 063312 000000
 16401 063314 000000
 16402 063316 000000
 16403 063320 000000
 16404 063322 000000
 16405 063324 177777
 16406 063326 177400
 16407 063330 000377
 16408 063332 125252
 16409 063334 052525
 16410
 16411
 16412
 16413
 16414 063336 000000
 16415 063340 000000
 16416 063342 000000
 16417 063344 177777
 16418 063346 177777
 16419 063350 177776
 16420 063352 125252
 16421 063354 052525
 16422 063356 177777
 16423 063360 052525
 16424 063362 125252
 16425 063364 177777
 16426 063366 125252
 16427 063370 125252
 16428 063372 052524
 16429 063374 052525
 16430 063376 052525
 16431 063400 125252
 16432 063402 052525
 16433 063404 125253
 16434 063406 000000
 16435 063410 125253
 16436 063412 052525
 16437 063414 000000
 16438
 16439
 16440
 16441
 16442 063416 000000
 16443 063420 000000
 16444 063422 000000
 16445 063424 177777

IBUF: .BLKW 4 ;DL11 INPUT TEST BUFFER
 ATA: DWTA
 DWTB
 DBTA
 DBTB
 MBUFO
 MBUF1
 MBUFO: 0
 0
 MBUF1: 0
 0
 DWTA: 0
 -1
 177400
 377
 125252
 ALUADD: 052525 ;ALSO SERVES AS NULL ENTRY FOR ALUADD

;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
 177777 ;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
 125252 ;DST OP5
 052524 ;ANS5
 052525 ;SRC OP6
 052525 ;DST OP6
 125252 ;ANS6
 052525 ;SRC OP7
 125253 ;DST OP7
 000000 ;ANS7
 125253 ;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

16446	063426	177777	177777	:DST OP2
16447	063430	000000	000000	:ANS2
16448	063432	000000	000000	:SRC OP3
16449	063434	177777	177777	:DST OP3
16450	063436	177777	177777	:ANS3
16451	063440	177777	177777	:SRC OP4
16452	063442	000000	000000	:DST OP4
16453	063444	000000	000000	:ANS4
16454	063446	125252	125252	:SRC OP5
16455	063450	125252	125252	:DST OP5

16456 063452 000000
16457 063454 052525
16458 063456 052525
16459 063460 000000
16460 063462 125252
16461 063464 052525
16462 063466 052525
16463 063470 052525
16464 063472 125252
16465 063474 125252

000000
052525
052525
000000
125252
052525
052525
052525
125252
125252
ORTAB: 125252

:ANS5
:SRC OP6
:DST OP6
:ANS6
:SRC OP7
:DST OP7
:ANS7
:SRC OP8
:DST OP8
:ANS8 -- ALSO NULL ENTRY FOR ORTAB

:THIS TABLE OF 8 ENTRIES IS USED BY THE ALU "OR" TEST IN THE
:COMBINED INSTRUCTION EXERCISER TEST

16470 063476 000000
16471 063500 000000
16472 063502 000000
16473 063504 177777
16474 063506 177777
16475 063510 177777
16476 063512 000000
16477 063514 177777
16478 063516 177777
16479 063520 177777
16480 063522 000000
16481 063524 177777
16482 063526 125252
16483 063530 125252
16484 063532 125252
16485 063534 052525
16486 063536 052525
16487 063540 052525
16488 063542 125252
16489 063544 052525
16490 063546 177777
16491 063550 052525
16492 063552 125252
16493 063554 177777

000000
000000
000000
177777
177777
177777
000000
177777
177777
177777
000000
177777
125252
125252
125252
052525
052525
052525
125252
052525
177777
052525
125252
177777
ALUSUB: 177777

:SRC OP1
:DST OP1
:ANS1
:SRC OP2
:DST OP2
:ANS2
:SRC OP3
:DST OP3
:ANS3
:SRC OP4
:DST OP4
:ANS4
:SRC OP5
:DST OP5
:ANS5
:SRC OP6
:DST OP6
:ANS6
:SRC OP7
:DST OP7
:ANS7
:SRC OP8
:DST OP8
:ANS8 -- ALSO NULL ENTRY FOR ALUSUB

:THIS TABLE OF 8 ENTRIES IS USED BY THE ALU SUB TEST IN THE
:COMBINED INSTRUCTION EXERCISER TESTS

16494
16495
16496
16497
16498 063556 000000
16499 063560 000000
16500 063562 000000
16501 063564 177777
16502 063566 177777
16503 063570 000000
16504 063572 125252
16505 063574 052525
16506 063576 125253
16507 063600 052525
16508 063602 125252
16509 063604 052525
16510 063606 125252
16511 063610 125252

000000
000000
000000
177777
177777
000000
125252
052525
125253
052525
125252
052525
125252
125252

:SRC OP1
:DST OP1
:ANS1
:SRC OP2
:DST OP2
:ANS2
:SRC OP3
:DST OP3
:ANS3
:SRC OP4
:DST OP4
:ANS4
:SRC OP5
:DST OP5

16512	063612	000000	000000	:ANS5
16513	063614	052525	052525	:SRC OP6
16514	063616	052525	052525	:DST OP6
16515	063620	000000	000000	:ANS6
16516	063622	052525	052525	:SRC OP7
16517	063624	125253	125253	:DST OP7
16518	063626	052526	052526	:ANS7
16519	063630	125253	125253	:SRC OP8
16520	063632	052525	052525	:DST OP8
16521	063634	125252	125252	:ANS8

16522				
16523	063636	005702		
16524	063640	005002		
16525	063642	005102		
16526	063644	005202		
16527	063646	005302		
16528	063650	005502		
16529	063652	005602		
16530	063654	006202		
16531	063656	006302		
16532	063660	105002		
16533	063662	105102		
16534	063664	105202		
16535	063666	105302		
16536	063670	105502		
16537	063672	105502		
16538	063674	105602		
16539	063676	105702		
16540	063700	106202		
16541	063702	106302		
16542	063704	151302		
16543	063706	074302		
16544	063710	121302		
16545	063712	131302		
16546	063714	141302		
16547	063716	111302		
16548	063720	021302		
16549	063722	031302		
16550	063724	041302		
16551	063726	051302		
16552	063730	006702		
16553	063732	005402		
16554	063734	161302		
16555	063736	020312		
16556	063740	030312		
16557	063742	120312		
16558	063744	131302		
16559	063746	005712		
16560	063750	105712		
16561	063752	021312		
16562	063754	031312		
16563	063756	121312		
16564	063760	131312		
16565	063762	061302		
16566	063764	000302		
16567	063766	160302		

INSTAB:	TST	R2	:BEGINNING OF INSTRUCTION TABLE OF INSTRUCTIONS
	CLR	R2	:THAT TEST BUT SERVICE IN VARIOUS ROM LOCATIONS
	COM	R2	
	INC	R2	
	DEC	R2	
	ADC	R2	
	SBC	R2	
	ASR	R2	
	ASL	R2	
	CLRB	R2	
	COMB	R2	
	INCB	R2	
	DECB	R2	
	ADCB	R2	
	ADCB	R2	
	SBCB	R2	
	TSTB	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	R2	
	SUB	(R3),R2	
	CMP	R3,(R2)	
	BIT	R3,(R2)	
	CMPB	R3,(R2)	
	BITB	(R3),R2	
	TST	(R2)	
	TSTB	(R2)	
	CMP	(R3),(R2)	
	BIT	(R3),(R2)	
	CMPB	(R3),(R2)	
	BITB	(R3),(R2)	
	ADD	(R3),R2	
	SWAB	R2	
	SUB	R3,R2	

16568	063770	060302
16569	063772	010302
16570	063774	011302
16571	063776	110302
16572	064000	006102
16573	064002	106102
16574	064004	105402
16575	064006	102400
16576	064010	102000
16577	064012	000005
16578	064014	020302
16579	064016	030302
16580	064020	040302
16581	064022	120302
16582	064024	130302
16583	064026	140302
16584	064030	150302
16585		
16586	064032	000000
16587	064034	000001
16588	064036	000400
16589	064040	177401
16590	064042	052526
16591	064044	125253
16592		
16593		
16594	064046	000000
16595	064050	000000
16596	064052	000000
16597	064054	000000
16598	064056	000040
16599	064156	000000
16600	064160	000000
16601	064162	000000
16602	064164	000000
16603		
16604		
16605		
16606		
16607		
16608		
16609		
16610		
16611		
16612		
16613		
16614		
16615		
16616		
16617	064166	
16618		
16619	064166	
16620	064166	201 001
16621	064170	202 002
16622	064172	203 003
16623	064174	204 004

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
 52526
 125253
 ;* MED TEST TABLES

TLOC1: .WORD 0
 PSWHOL: .WORD 0
 TABBEG: .WORD 0
 TABEND: .WORD 0
 STGBLK: .BLKW 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: ;A SCRATCH PAD - LO
 R1A: .BYTE 201,001 ;LOBYTE, HIBYTE=WRITE CODE, READ CODE
 R2A: .BYTE 202,002
 R3A: .BYTE 203,003
 R4A: .BYTE 204,004

16624	064176	205	005
16625	064200	206	006
16626	064202	210	010
16627	064204	211	011
16628	064206	212	012
16629	064210	213	013
16630	064212	214	014
16631	064214	215	015
16632	064216	216	016
16633	064220	217	017
16634	064222	220	020
16635	064224	221	021
16636	064226	222	022
16637	064230	223	023
16638	064232	226	026
16639	064234	227	027
16640	064236	230	030
16641	064240	231	031
16642	064242	232	032
16643	064244	233	033
16644	064246	234	034
16645	064250	235	035
16646	064252	236	036
16647	064254	237	037
16648			
16649	064256		
16650	064256	241	041
16651	064260	242	042
16652	064262	243	043
16653	064264	244	044
16654	064266	245	045
16655	064270	246	046
16656	064272	250	050
16657	064274	251	051
16658	064276	252	052
16659	064300	253	053
16660	064302	254	054
16661	064304	255	055
16662	064306	256	056
16663	064310	257	057
16664	064312	260	060
16665	064314	261	061
16666	064316	262	062
16667	064320	263	063
16668	064322	266	066
16669	064324	270	070
16670	064326	272	072
16671	064330	273	073
16672	064332	274	074
16673	064334	275	075
16674	064336	276	076
16675	064340	277	077
16676			
16677	064342		
16678	064342	300	100
16679	064344	301	101

R5A:	.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	:A SCRATCH PAD-HI
WCSA.1:	.BYTE	221,021	
GNWHAM:	.BYTE	222,022	
CNSTSW:	.BYTE	223,023	
CNSSW:	.BYTE	226,026	
CNSCDR:	.BYTE	227,027	
FAC1.0:	.BYTE	230,030	
FAC1.1:	.BYTE	231,031	
FAC1.2:	.BYTE	232,032	
FAC1.3:	.BYTE	233,033	
FAC1.4:	.BYTE	234,034	
FAC1.5:	.BYTE	235,035	
FPSHI:	.BYTE	236,036	
ASP2:	FDST1:	.BYTE	237,037
BSP1:			
R1B:	.BYTE	241,041	:B SCRATCH PAD - LO
R2B:	.BYTE	242,042	
R3B:	.BYTE	243,043	
R4B:	.BYTE	244,044	
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	
WCSB.0:	.BYTE	260,060	:B SCRATCH PAD - HI
WCSB.1:	.BYTE	261,061	
WCSADR:	.BYTE	262,062	
RZERO:	.BYTE	263,063	
RVECT:	.BYTE	266,066	
FACO.0:	.BYTE	270,070	
FACO.1:	.BYTE	272,072	
FACO.2:	.BYTE	273,073	
FACO.4:	.BYTE	274,074	
FACO.5:	.BYTE	275,075	
FEA:	.BYTE	276,076	
BSP2:	FDST0:	.BYTE	277,077
CSP1:			
LJAM:	.BYTE	300,100	:C SCRATCH PAD
LSERV:	.BYTE	301,101	

16680	064346	302	102
16681	064350	303	103
16682	064352	304	104
16683	064354	305	105
16684	064356	307	107
16685	064360	310	110
16686	064362	311	111
16687	064364	312	112
16688	064366	313	113
16689	064370	316	116
16690	064372	224	024
16691	064374	225	025
16692	064376	264	064
16693	064400	265	065
16694	064402	000000	

LPBA:	.BYTE	302,102	
LCUA:	.BYTE	303,103	
LFGIN:	.BYTE	304,104	
LWHAM:	.BYTE	305,105	
LTAG:	.BYTE	307,107	
CNSCO:	.BYTE	310,110	
CNSC1:	.BYTE	311,111	
CNSC2:	.BYTE	312,112	
CST200:	.BYTE	313,113	
CSP2:	CNSTO:	.BYTE	316,116
RT1A:	.BYTE	224,024	
RT2A:	.BYTE	225,025	
RT1B:	.BYTE	264,064	
RT2B:	.BYTE	265,065	
	.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.

```

16703	064404		
16704	064404	120	137
16705	064406	145	145
16706	064410	150	151
16707	064412	156	177
16708	064414	320	343
16709	064416	353	357
16710	064420	000000	

TBL3:			
NOPS:	.BYTE	120,137	:GROUP A
	.BYTE	145,145	:GROUP B
	.BYTE	150,151	:GROUP C
	.BYTE	156,177	:GROUP D
	.BYTE	320,343	:GROUP E
	.BYTE	353,357	:FROUP 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. . .

```

16720	064422		
16721	064422	200	000
16722	064424	207	007
16723	064426	240	040
16724	064430	247	047
16725	064432	314	114
16726	064434	317	117

TBL4:			
ROA:	.BYTE	200,000	:LOBYTE, HYBYTE - WRITE CODE, READ CODE
R7A:	.BYTE	207,007	:0 REPLACES ANY NON EXSISTENT CODES
ROB:	.BYTE	240,040	:EXCEPT IN THE CASE OF ROA
R7B:	.BYTE	247,047	
CNST2:	.BYTE	314,114	
CNST1:	.BYTE	317,117	

```

:*      TABLE V

```

16729	064436		
16730			
16731	064436	306	
16732	064437	106	
16733	064440	315	
16734	064441	115	
16735	064442	267	

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

16736 064443 067
 16737 064444 140
 16738 064445 141
 16739 064446 142
 16740 064447 143
 16741 064450 344
 16742 064451 144
 16743 064452 345
 16744 064453 146
 16745 064454 346
 16746 064455 147
 16747 064456 347
 16748 064457 351
 16749 064460 152
 16750 064461 352
 16751 064462 153
 16752 064463 000

JAM: .BYTE 067
 SERV: .BYTE 140
 PBA: .BYTE 141
 CUA: .BYTE 142
 FLAG: .BYTE 143
 DREG: .BYTE 144
 REV: .BYTE 146
 SREG: .BYTE 346
 COUNT: .BYTE 147
 NUA: .BYTE 347
 RES: .BYTE 351
 DCS0: .BYTE 152
 DCS1: .BYTE 153
 .EVEN 0

;THE NEXT TABLE

;INIT REG

;TABLE TERMINATOR

16753
 16754
 16755
 16756
 16757 064464
 16758
 16759 064464 003330
 16760 064466 003150
 16761 064470 003375
 16762 064472 003271
 16763 064474 003240
 16764 064476 003224
 16765 064500 003160
 16766 064502 003161
 16767 064504 003170
 16768 064506 003171
 16769 064510 003344
 16770 064512 003320
 16771 064514 003345
 16772 064516 003340
 16773 064520 003350
 16774 064522 003341
 16775 064524 003351
 16776 064526 003355
 16777 064530 003720
 16778 064532 003724
 16779 064534 003721

;* TABLE VI

;*
 TBL6:

ULCDTA: .WORD 3330
 UMD: .WORD 3150
 UCNSCTL: .WORD 3375
 .WORD 3271
 .WORD 3240
 .WORD 3224
 UJAM: .WORD 3160
 USERV: .WORD 3161
 UPBA: .WORD 3170
 UCUA: .WORD 3171
 UFLAG: .WORD 3344
 .WORD 3320
 UDREG: .WORD 3345
 UREV: .WORD 3340
 USREG: .WORD 3350
 UCOUNT: .WORD 3341
 UNUA: .WORD 3351
 URES: .WORD 3355
 UDCS0: .WORD 3720
 UINIT: .WORD 3724
 UDCS1: .WORD 3721

;THIS TABLE CONTAINS THE MICRO-ADDRESSES
 ;WHICH ARE LOADED INTO THE MICROBREAK
 ;REG. TO TEST THE OPERATION CODES
 ;CONTAINED IN THE PRECEEDING TABLE.

16780
 16781
 16782
 16783
 16784
 16785 064536
 16786
 16787 064536 000100 077600
 16788 064542 000101 000010
 16789 064546 000102 020000
 16790 064552 000103 000004
 16791 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

16792 064562 000105 054000
16793 064566 000107 024000
16794 064572 000110 177400
16795 064576 000111 177600
16796 064602 000112 100000
16797 064606 000113 000200
16798 064612 000114 000002
16799 064616 000116 000000
16800 064622 000117 000001
16801 064626 000000
16802
16803
16804 064630
16805 064630 000 377 252
16806 064633 125
16807 064634
16808 064634 000 001 120
16809 064637 253
16810
16811
16812
16813 064640
16814 064640
16815 064640 027523 020102 051504
16816 064646 020124
16817 064650 040527 020123 051504
16818 064656 020124
16819 064660 042040 051505 004524
16820 064666 024040 051111 004451
16821 064674 052040 051505 004524
16822 064702 024040 041520 004451
16823 064710 024040 050123 004451
16824 064716 050050 053523 000051
16825 064724 027523 020102 042522
16826 064732 020123 040527 020123
16827 064740 042522 020123 051504
16828 064746 020124 050117 020040
16829 064754 051123 020103 050117
16830 064762 020040 042524 052123
16831 064770 020011 050050 024503
16832 064776 020011 051450 024520
16833 065004 024011 051520 024527
16834 065012 000
16835 065013 123 041057 051440
16836 065020 004520 040527 020123
16837 065026 050123 020011 044450
16838 065034 024522 020011 042524
16839 065042 052123 020011 050050
16840 065050 024503 024011 051520
16841 065056 024527 000
16842 065061 011 020011 051511
16843 065066 051040 000063
16844 065072 004411 044440 020123
16845 065100 032522 000
16846 065103 015 042412 042116
16847 065110 050040 051501 020123

CLWHAM: .WORD 105,54000
CLTAG: .WORD 107,24000
CCNSCO: .WORD 110,177400
CCNSC1: .WORD 111,177600
CCNSC2: .WORD 112,100000
CCST200: .WORD 113,200
CCNST2: .WORD 114,2
CCNST0: .WORD 116,0
CCNST1: .WORD 117,1
.WORD 0

.EVEN
DBTA: .BYTE 000,377,252,125
DBTB: .BYTE 000,001,120,253

;MESSAGE TABLES

EM1:
EM2:
EM4: .ASCII 'S/B DST '
EM7: .ASCII 'WAS DST '
EM6: .ASCII ' DEST'<HT>
EM5: .ASCIZ ' (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'
EOP1: .ASCIZ <15><12>'END PASS # '

16848	065116	020043	000		
16849	065121	011	051105	047522	EOP2: .ASCIZ <HT>'ERROR COUNT = '
16850	065126	020122	047503	047125	
16851	065134	020124	020075	000	
16852	065141	015	041412	045521	IDENT1: .ASCIZ <15><12>'CQKDADO KD11-K BASIC LOGIC TESTS'<15><12>
16853	065146	040504	030104	045440	
16854	065154	030504	026461	020113	
16855	065162	040502	044523	020103	
16856	065170	047514	044507	020103	
16857	065176	042524	052123	006523	
16858	065204	000012			
16859	065206	005015	051124	050101	BEMSG: .ASCIZ <CR><LF>'TRAPPED TO 4 PC = '
16860	065214	042520	020104	047524	
16861	065222	032040	050040	020103	
16862	065230	020075	000		
16863	065233	015	052012	040522	RSMSG: .ASCIZ <CR><LF>'TRAPPED TO 10 PC = '
16864	065240	050120	042105	052040	
16865	065246	020117	030061	050040	
16866	065254	020103	020075	000	
16867	065261	124	051505	051524	EM11: .ASCIZ 'TESTS SKIPPED'
16868	065266	051440	044513	050120	
16869	065274	042105	000		
16870	065277	040	050040	004503	DH11: .ASCIZ '' PC''<HT>'EXPCTD''<HT>'ACTUAL''<HT>'(TEST #'S)''
16871	065304	054105	041520	042124	
16872	065312	040411	052103	040525	
16873	065320	004514	052050	051505	
16874	065326	020124	023443	024523	
16875	065334	000			
16876	065335	115	042105	042040	EM12: .ASCIZ /MED DID NOT ABORT IN USER MODE/
16877	065342	042111	047040	052117	
16878	065350	040440	047502	052122	
16879	065356	044440	020116	051525	
16880	065364	051105	046440	042117	
16881	065372	000105			
16882	065374	042515	020104	054105	EM13: .ASCIZ /MED EXECUTED IN USER MODE/
16883	065402	041505	052125	042105	
16884	065410	044440	020116	051525	
16885	065416	051105	046440	042117	
16886	065424	000105			
16887	065426	042515	020104	044103	EM14: .ASCIZ /MED CHANGED PSW/
16888	065434	047101	042507	020104	
16889	065442	051520	000127		
16890	065446	044515	051103	041117	EM15: .ASCIZ /MICROBREAK TRAP-TO-4 DID NOT OCCUR/
16891	065454	042522	045501	052040	
16892	065462	040522	026520	047524	
16893	065470	032055	042040	042111	
16894	065476	047040	052117	047440	
16895	065504	041503	051125	000	
16896	065511	114	043517	052503	EM17: .ASCIZ /LOGCUA LOGGED WRONG/
16897	065516	020101	047514	043507	
16898	065524	042105	053440	047522	
16899	065532	043516	000		
16900	065535	103	050123	041440	EM21: .ASCIZ /CSP CONSTANT WRONG/
16901	065542	047117	052123	047101	
16902	065550	020124	051127	047117	
16903	065556	000107			

16904	065560	040502	020104	040504	EM22:	.ASCIZ	/BAD DATA READ BY A MED/
16905	065566	040524	051040	040505			
16906	065574	020104	054502	040440			
16907	065602	046440	042105	000			
16908	065607	116	020117	042117	EM23:	.ASCIZ	/NO ODD PC TRAP/
16909	065614	020104	041520	052040			
16910	065622	040522	000120				
16911	065626	042117	020104	042101	EM24:	.ASCIZ	/ODD ADR. BIT NOT SET IN CPU ERR REG OR LOG JAM/
16912	065634	027122	041040	052111			
16913	065642	047040	052117	051440			
16914	065650	052105	044440	020116			
16915	065656	050103	020125	051105			
16916	065664	020122	042522	020107			
16917	065672	051117	046040	043517			
16918	065700	045040	046501	000			
16919	065705	120	054510	020123	EM26:	.ASCIZ	/PHYS BA LOGGED WRONG/
16920	065712	040502	046040	043517			
16921	065720	042507	020104	051127			
16922	065726	047117	000107				
16923	065732	040503	044103	020105	EM27:	.ASCIZ	/CACHE PARITY ERROR LOGGED IN BAKUP MODE/
16924	065740	040520	044522	054524			
16925	065746	042440	051122	051117			
16926	065754	046040	043517	042507			
16927	065762	020104	047111	041040			
16928	065770	045501	050125	046440			
16929	065776	042117	000105				
16930	066002	040503	044103	020105	EM30:	.ASCIZ	/CACHE PARITY TRAPPED WHEN DISABLED/
16931	066010	040520	044522	054524			
16932	066016	052040	040522	050120			
16933	066024	042105	053440	042510			
16934	066032	020116	044504	040523			
16935	066040	046102	042105	000			
16936	066045	111	051516	051124	EM41:	.ASCIZ	/INSTR. NOT ABORTED IN CACHE ABORT MODE/
16937	066052	020056	047516	020124			
16938	066060	041101	051117	042524			
16939	066066	020104	047111	041440			
16940	066074	041501	042510	040440			
16941	066102	047502	052122	046440			
16942	066110	042117	000105				
16943	066114	042515	047515	054522	EM32:	.ASCIZ	/MEMORY ERR REG INCORRECT/
16944	066122	042440	051122	051040			
16945	066130	043505	044440	041516			
16946	066136	051117	042522	052103			
16947	066144	000					
16948	066145	124	046511	047505	EM33:	.ASCIZ	/TIMEOUT BIT NOT SET IN CPU ERR REG OR LOG JAM/
16949	066152	052125	041040	052111			
16950	066160	047040	052117	051440			
16951	066166	052105	044440	020116			
16952	066174	050103	020125	051105			
16953	066202	020122	042522	020107			
16954	066210	051117	046040	043517			
16955	066216	045040	046501	000			
16956	066223	116	020117	046111	EM34:	.ASCIZ	/NO ILLEGAL INTERNAL ADR TRAP/
16957	066230	042514	040507	020114			
16958	066236	047111	042524	047122			
16959	066244	046101	040440	051104			

16960	066252	052040	040522	000120
16961	066260	047111	051124	040516
16962	066266	020114	042101	020122
16963	066274	051105	020122	044502
16964	066302	020124	047516	020124
16965	066310	042523	020124	047111
16966	066316	041440	052520	042440
16967	066324	051122	051040	043505
16968	066332	047440	020122	047514
16969	066340	020107	040512	000115
16970	066346	040514	052123	044440
16971	066354	052116	027522	051124
16972	066362	050101	053040	041505
16973	066370	047524	020122	047516
16974	066376	020124	047514	043507
16975	066404	042105	044440	020116
16976	066412	046106	043501	051040
16977	066420	043505	000	
16978	066423	114	043517	043040
16979	066430	051111	052123	046440
16980	066436	042117	020105	044504
16981	066444	020104	047516	020124
16982	066452	047111	044510	044502
16983	066460	020124	051105	047522
16984	066466	020122	047514	020107
16985	066474	043101	042524	020122
16986	066502	044506	051522	020124
16987	066510	051105	047522	000122
16988	066516	051105	047522	020122
16989	066524	047514	020107	040527
16990	066532	020123	047516	020124
16991	066540	042522	047105	041101
16992	066546	042514	026104	047440
16993	066554	042104	040440	051104
16994	066562	041040	052111	041440
16995	066570	051114	044440	020116
16996	066576	050103	042525	051122
16997	066604	000		
16998	066605	116	020117	040503
16999	066612	044103	020105	040520
17000	066620	044522	054524	052040
17001	066626	040522	000120	
17002	066632	047514	023040	044040
17003	066640	020111	054502	042524
17004	066646	023040	052040	043501
17005	066654	050040	051101	052111
17006	066662	020131	044502	051524
17007	066670	047040	052117	051440
17008	066676	052105	044440	020116
17009	066704	047514	020107	042523
17010	066712	053122	041511	000105
17011	066720	047514	023040	044040
17012	066726	020111	054502	042524
17013	066734	023040	052040	043501
17014	066742	050040	051101	052111
17015	066750	020131	044502	051524

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/

17016	066756	047040	052117	051440	
17017	066764	052105	044440	020116	
17018	066772	042515	020115	051105	
17019	067000	020122	042522	000107	
17020	067006	040503	044103	020105	EM45: .ASCIZ /CACHE TAG LOGGED WRONG/
17021	067014	040524	020107	047514	
17022	067022	043507	042105	053440	
17023	067030	047522	043516	000	
17024	067035	103	041501	042510	EM16: .ASCIZ /CACHE DATA LOGGED WRONG/
17025	067042	042040	052101	020101	
17026	067050	047514	043507	042105	
17027	067056	053440	047522	043516	
17028	067064	000			
17029	067065	105	051511	051440	EMEIS1: .ASCIZ 'EIS SET COND CODES WRONG'
17030	067072	052105	041440	047117	
17031	067100	020104	047503	042504	
17032	067106	020123	051127	047117	
17033	067114	000107			
17034	067116	044505	020123	040507	EMEIS2: .ASCIZ 'EIS GAVE WRONG RESULT'
17035	067124	042526	053440	047522	
17036	067132	043516	051040	051505	
17037	067140	046125	000124		
17038	067144	052501	047524	044455	EM46: .ASCIZ 'AUTO-INCREMENT (DECREMT) DID NOT OCCUR IN EIS'
17039	067152	041516	042522	042515	
17040	067160	052116	024040	042504	
17041	067166	051103	046505	024524	
17042	067174	042040	042111	047040	
17043	067202	052117	047440	041503	
17044	067210	051125	044440	020116	
17045	067216	044505	000123		
17046	067222	050040	053523	051011	DHEIS1: .ASCII ' PSW'<HT>'REG-WAS-REG+1'<HT>'REG-S/B-REG+1'<HT>
17047	067230	043505	053455	051501	
17048	067236	051055	043505	030453	
17049	067244	051011	043505	051455	
17050	067252	041057	051055	043505	
17051	067260	030453	011		
17052	067263	040	050040	004503	DH46: .ASCIZ ' PC'<HT>' (IR)'<HT>' TEST'
17053	067270	024040	051111	004451	
17054	067276	052040	051505	000124	
17055	067304	020040	041520	046411	DH15: .ASCIZ / PC/<HT>/MEDCODE MICROBK REG./
17056	067312	042105	047503	042504	
17057	067320	046440	041511	047522	
17058	067326	045502	051040	043505	
17059	067334	000056			
17060	067336	020040	041520	046411	DH17: .ASCIZ / PC/<HT>/MEDCODE EXPECTD RECEIVD/
17061	067344	042105	047503	042504	
17062	067352	042440	050130	041505	
17063	067360	042124	051040	041505	
17064	067366	044505	042126	000	
17065	067373	040	050040	000103	DH23: .ASCIZ / PC/
17066	067400	020040	041520	041411	DH24: .ASCIZ / PC/<HT>/CPUERR/<HT>/LOGJAM/
17067	067406	052520	051105	004522	
17068	067414	047514	045107	046501	
17069	067422	000			
17070	067423	040	050040	004503	DH25: .ASCIZ / PC/<HT>/FLGREG/
17071	067430	046106	051107	043505	

17072	067436	000							
17073	067437	040	050040	004503	DH26:	.ASCIZ	' PC'<HT>'<17:16>-S/B PA-<15:0> <17:16>-WAS PA-<15:0>'		
17074	067444	030474	035067	033061					
17075	067452	026476	027523	020102					
17076	067460	040520	036055	032461					
17077	067466	030072	020076	036040					
17078	067474	033461	030472	037066					
17079	067502	053455	051501	050040					
17080	067510	026501	030474	035065					
17081	067516	037060	000						
17082	067521	040	050040	004503	DH27:	.ASCIZ	/ PC/<HT>/LOGPBA/<HT>/LOGDATA/<HT>/LOGTAG/		
17083	067526	047514	050107	040502					
17084	067534	046011	043517	040504					
17085	067542	040524	046011	043517					
17086	067550	040524	000107						
17087	067554	020040	041520	046411	DH32:	.ASCIZ	/ PC/<HT>/MEMERR/		
17088	067562	046505	051105	000122					
17089	067570	020040	041520	046011	DH42:	.ASCIZ	/ PC/<HT>/LOGSERVICE/		
17090	067576	043517	042523	053122					
17091	067604	042503	000						
17092	067607	040	050040	004503	DH44:	.ASCIZ	/ PC/<HT>/EXPCT/<HT>/RECVD/		
17093	067614	054105	041520	004524					
17094	067622	042522	053103	000104					
17095						.EVEN			
17096	067630	001016	001076	001100	DT15:	.WORD	\$ERRPC,\$TMP0,\$TMP1,0		
17097	067636	000000							
17098	067640	001016	001100	001102	DT21:	.WORD	\$ERRPC,\$TMP1,\$TMP2,\$REG0,0		
17099	067646	001062	000000						
17100	067652	001016	001100	001102	DT22:	.WORD	\$ERRPC,\$TMP1,\$TMP2,\$TMP3,0		
17101	067660	001104	000000						
17102	067664	001016	000000		DT23:	.WORD	\$ERRPC,0		
17103	067670	001016	001064	001062	DT24:	.WORD	\$ERRPC,\$REG1,\$REG0,0		
17104	067676	000000							
17105	067700	001016	001062	000000	DT25:	.WORD	\$ERRPC,\$REG0,0		
17106	067706	001016	001064	001066	DT26:	.WORD	\$ERRPC,\$REG1,\$REG2,\$REG0,\$REG3,0		
17107	067714	001062	001070	000000					
17108	067722	001016	001070	001064	DT27:	.WORD	\$ERRPC,\$REG3,\$REG1,\$REG2,0		
17109	067730	001066	000000						
17110	067734	001060	001066	001070	DTEIS1:	.WORD	\$REGAD,\$REG2,\$REG3,\$REG1,\$REG4		
17111	067742	001064	001072						
17112	067746	001016	001076	001062	DT46:	.WORD	\$ERRPC,\$TMP0,\$REG0,0		
17113	067754	000000							
17114									
17115	067756	000	000		DF15:	.BYTE	0,0		
17116	067760	000	000	000	DF17:	.BYTE	0,0,0		
17117		067764				.EVEN			
17118	067764				DT1:				
17119	067764				DT2:				
17120	067764				DT4:				
17121	067764	001072			DT10:	.WORD	\$REG4		
17122	067766	001070			DT7:	.WORD	\$REG3		
17123	067770	001066			DT6:	.WORD	\$REG2		
17124	067772	001064	001062	001016	DT5:	.WORD	\$REG1,\$REG0,\$ERRPC,\$REG5,\$REGAD,0		
17125	070000	001074	001060	000000					
17126	070006	001072	001070	001064	DT3:	.WORD	\$REG4,\$REG3,\$REG1,\$REG0,\$ERRPC,\$REGAD,0		
17127	070014	001062	001016	001060					

CQKDA-D KD11-K BASIC LOGIC TESTS
CQKDAD.P11 11-SEP-79 13:31

MACY11 30A(1052) 11-SEP-79^{H 9} 13:53 PAGE 319
TRAP TABLE

SEQ 0318

17128	070022	000000							
17129	070024	001016	001124	001062	DT11:	.WORD	\$ERRPC,\$TESTN,\$REGO,0		
17130	070032	000000							
17131		000001				.END			

EM4	064640	986	16815#						
EM40	066516	1165	16988#						
EM41	066045	1172	16936#						
EM42	066532	1179	17002#						
EM43	066720	1186	17011#						
EM45	067006	1049	17020#						
EM46	067144	1207	17038#						
EM5	064666	991	16820#						
EM6	064660	996	16819#						
EM7	064650	1001	16817#						
EOP1	065103	15699	16846#						
EOP2	065121	15702	16849#						
ERRA	062040	4420	16058#						
ERRFLG	063240	4422*	4427*	16058*	16376#				
ERRVEC=	000004	815#	14901*	14914*	15948	15949*	15951*	15954*	
EX002	001640	1284#							
E001	001632	1274#							
E003	001656	1294	1295	1296	1299#				
E004	001674	1309	1310	1311	1314#				
E005	001712	1324	1325	1326	1329#				
E006	001724	1343#							
E007	001740	1357#							
E010	001756	1370	1373#						
E011	001776	1386	1389#						
E012	002016	1404#							
E013	002042	1421#							
E014	002064	1436#							
E016	002136	1469	1470	1471	1474#				
E022	002330	1562	1563	1564	1567#				
E024	002414	1606#							
E025	002434	1617	1620#						
E026	002454	1630	1633#						
E027	002474	1647#							
E030	002514	1661#							
E031	002534	1674#							
E035	002714	1753#							
E036	002734	1767#							
E037	002754	1781#							
E040	003002	1798#							
E042	003060	1836#							
E043	003102	1853#							
E044	003130	1869#							
E045	003156	1884#							
E1015A	002102	1450#							
E2002	001644	1288#							
E2015	002112	1457#							
E2017	002206	1502#							
E2020	002252	1528#							
E2021	002304	1551#							
E2023	002370	1591#							
E2032	002574	1696#							
E2033	002634	1718#							
E2034	002674	1739#							
E2041	003040	1820#							
E2046	003242	1918#							
FAC0.0	064324	16669#							

PWRVEC=	000024	821#	15726*	15727*	15736*	15742*	15757*	15758*
RCSR =	177560	1257#	4489	4575				
RDBR =	177562	1258#						
RDFLAG=	000144	1233#	15038	15088	15115			
RDLUA=	000103	1241#	15102					
RDLDAT=	000106	1247#	15257	15527				
RDLFGI=	000104	1243#	15382	15541	15654	15669		
RDLJAM=	000100	1235#	15169	15335	15365	15413	15588	15625
RDLPBA=	000102	1239#	15184	15251	15351	15502		
RDLSER=	000101	1237#	15191	15481				
RDLTAG=	000107	1249#	15263	15513				
RDLWHA=	000105	1245#						
RDWHAM=	000022	1231#	15083	15134				
RES	064457	16748#						
RESTAR	061176	15837	15846	15848#	15851	15905	15912	
RESVEC=	000010	816#	14902*	14915*				
REV	064453	16744#						
RSBERT	061150	15840#	15908					
RSERR	061122	4453	13835	14915	15833#			
RSMSG	065233	15839	16863#					
RSVFLG	063246	4441*	4447*	15830*	16379#			
RSVTST	061114	4438	15830#					
RT1A	064372	16690#						
RT1B	064376	16692#						
RT2A	064374	16691#						
RT2B	064400	16693#						
RVECT	064322	16668#						
RZERO	064320	16667#						
ROA	064422	16721#						
ROB	064426	16723#						
R1A	064166	16620#						
R1B	064256	16650#						
R2A	064170	16621#						
R2B	064260	16651#						
R3A	064172	16622#						
R3B	064262	16652#						
R4A	064174	16623#						
R4B	064264	16653#						
R5A	064176	16624#						
R5B	064266	16654#						
R6A	064200	16625#						
R6B	064270	16655#						
R7A	064424	15006	16722#					
R7B	064430	15018	16724#					
SCOFLG	063244	4314*	4321*	15992*	16378#			
SCOPEA	061612	4315	15992#					
SELTST	063242	15938*	15939*	15940	16377#			
SERV	064445	16738#						
SOBERR	043012	11972	11981#					
SOB1	042764	11967#	12050					
SOB2	043000	11962	11965	11975#				
SOB3	043156	11967	12048#					
SOB4	043160	12046	12050#					
SOB5	042772	11971#	11975					
SREG	064454	16745#						
STACK =	001000	720#	1428	1915	1931	14900	15771	

TST102	007366	3299	3308#
TST103	007446	3326	3333#
TST104	007526	3351	3358#
TST105	007606	3376	3383#
TST106	007666	3401	3409#
TST107	007744	3427	3435#
TST11	003564	2063	2071#
TST110	010022	3453	3461#
TST111	010102	3479	3487#
TST112	010162	3505	3513#
TST113	010224	3528	3535#
TST114	010266	3550	3558#
TST115	010356	3594#	
TST116	010446	3630#	
TST117	010510	3641	3648#
TST12	003622	2085	2093#
TST120	010552	3659	3666#
TST121	010614	3677	3684#
TST122	010670	3699	3707#
TST123	010714	3715	3723#
TST124	010752	3743	3751#
TST125	010770	3757	3765#
TST126	011004	3771	3779#
TST127	011020	3785	3793#
TST13	003656	2107	2115#
TST130	011040	3800	3808#
TST131	011060	3815	3823#
TST132	011076	3830	3838#
TST133	011114	3844	3852#
TST134	011132	3859	3867#
TST135	011150	3874	3882#
TST136	011170	3889	3897#
TST137	011210	3904	3912#
TST14	003710	2125	2133#
TST140	011230	3919	3927#
TST141	011250	3934	3942#
TST142	011264	3948	3956#
TST143	011304	3963	3971#
TST144	011324	3978	3986#
TST145	011344	3993	4001#
TST146	011362	4008	4016#
TST147	011376	4022	4030#
TST15	003744	2143	2151#
TST150	011416	4037	4045#
TST151	011436	4052	4060#
TST152	011456	4067	4075#
TST153	011524	4093	4101#
TST154	011566	4111	4119#
TST155	011622	4128	4136#
TST156	011710	4164	4172#
TST157	011744	4185	4193#
TST16	004010	2162	2170#
TST160	012012	4209	4217#
TST161	012100	4240	4248#
TST162	012152	4261	4270#
TST163	012274	4301	4310#

TST164	012342	4331#	
TST165	012464	4362	4371#
TST166	012540	4394#	
TST167	012620	4408	4416#
TST17	004054	2181	2189#
TST170	012664	4428	4435#
TST171	012746	4459#	
TST172	013016	4472	4480#
TST173	013116	4508#	
TST174	013156	4518	4529#
TST175	013210	4538	4547#
TST176	013244	4556	4573#
TST177	013556	4645#	
TST2	003350	1958	1966#
TST20	004110	2199	2206#
TST200	013600	4652	4659#
TST201	013620	4666	4673#
TST202	013660	4686	4693#
TST203	013722	4708	4715#
TST204	013760	4728	4735#
TST205	014020	4750	4757#
TST206	014042	4764	4771#
TST207	014064	4779	4786#
TST21	004154	2221	2229#
TST210	014106	4794	4801#
TST211	014130	4809	4816#
TST212	014154	4824	4831#
TST213	014176	4838	4845#
TST214	014220	4853	4860#
TST215	014242	4868	4875#
TST216	014264	4883	4890#
TST217	014332	4909	4916#
TST22	004214	2244	2252#
TST220	014374	4930	4937#
TST221	014442	4956	4963#
TST222	014474	4973	4980#
TST223	014622	5025	5032#
TST224	014656	5042	5049#
TST225	014726	5068	5075#
TST226	014764	5086	5093#
TST227	015034	5112	5119#
TST23	004250	2261	2269#
TST230	015104	5138	5145#
TST231	015234	5190	5197#
TST232	015310	5217	5224#
TST233	015354	5243	5250#
TST234	015424	5269	5276#
TST235	015472	5295	5302#
TST236	015542	5322	5329#
TST237	015616	5349	5356#
TST24	004310	2283	2291#
TST240	015672	5376	5383#
TST241	015744	5403	5410#
TST242	016014	5429	5436#
TST243	016062	5455	5462#
TST244	016132	5481	5488#

TST245	016200	5507	5514#
TST246	016250	5533	5540#
TST247	016320	5559	5566#
TST25	004360	2309	2317#
TST250	016376	5586	5593#
TST251	016452	5613	5621#
TST252	016530	5641	5649#
TST253	016604	5669	5677#
TST254	016662	5697	5705#
TST255	016740	5725	5733#
TST256	017020	5755	5763#
TST257	017072	5781	5789#
TST26	004434	2338	2347#
TST260	017170	5816	5824#
TST261	017244	5842	5850#
TST262	017330	5873	5881#
TST263	017404	5899	5907#
TST264	017464	5929	5937#
TST265	017534	5954	5962#
TST266	017600	5981	5988#
TST267	017646	6008	6015#
TST27	004470	2358	2366#
TST270	017714	6034	6041#
TST271	017760	6059	6066#
TST272	020030	6085	6092#
TST273	020076	6111	6118#
TST274	020144	6137	6144#
TST275	020214	6163	6170#
TST276	020262	6189	6196#
TST277	020332	6215	6222#
TST3	003364	1971	1978#
TST30	004526	2376	2384#
TST300	020376	6240	6247#
TST301	020444	6266	6273#
TST302	020514	6292	6299#
TST303	020560	6318	6325#
TST304	020630	6344	6351#
TST305	020700	6369	6376#
TST306	020744	6395	6402#
TST307	021014	6421	6428#
TST31	004552	2392	2400#
TST310	021062	6447	6454#
TST311	021132	6473	6480#
TST312	021200	6499	6506#
TST313	021250	6525	6532#
TST314	021320	6551	6558#
TST315	021366	6577	6584#
TST316	021440	6604	6612#
TST317	021516	6633	6641#
TST32	004606	2410	2418#
TST320	021604	6665	6673#
TST321	021666	6697	6705#
TST322	021744	6724	6732#
TST323	022020	6751	6759#
TST324	022074	6778	6786#
TST325	022152	6805	6813#

TST326	022226	6832	6840#
TST327	022304	6859	6867#
TST33	004646	2429	2437#
TST330	022356	6885	6893#
TST331	022432	6912	6920#
TST332	022510	6939	6947#
TST333	022562	6966	6974#
TST334	022640	6993	7001#
TST335	022716	7020	7028#
TST336	022770	7047	7055#
TST337	023046	7074	7082#
TST34	004702	2450	2458#
TST340	023122	7101	7109#
TST341	023200	7128	7136#
TST342	023254	7155	7163#
TST343	023332	7182	7190#
TST344	023410	7209	7217#
TST345	023464	7236	7244#
TST346	023534	7263	7270#
TST347	023604	7289	7296#
TST35	004736	2471	2478#
TST350	023654	7315	7322#
TST351	023724	7341	7348#
TST352	023774	7367	7374#
TST353	024054	7393	7401#
TST354	024124	7415	7423#
TST355	024200	7441	7449#
TST356	024244	7462	7470#
TST357	024324	7489	7497#
TST36	004772	2491	2499#
TST360	024406	7515	7523#
TST361	024462	7541	7549#
TST362	024526	7562	7570#
TST363	024600	7590	7597#
TST364	024654	7617	7624#
TST365	024722	7641	7648#
TST366	024770	7665	7672#
TST367	025044	7690	7697#
TST37	005026	2512	2520#
TST370	025112	7714	7721#
TST371	025166	7739	7746#
TST372	025226	7758	7765#
TST373	025272	7778	7785#
TST374	025336	7798	7806#
TST375	025402	7819	7827#
TST376	025470	7849	7856#
TST377	025556	7878	7885#
TST4	003402	1984	1992#
TST40	005066	2534	2542#
TST400	025646	7907	7914#
TST401	025736	7936	7943#
TST402	026024	7965	7972#
TST403	026112	7994	8001#
TST404	026202	8023	8030#
TST405	026272	8052	8059#
TST406	026344	8073	8081#

TST407	026416	8095	8103#
TST41	005132	2557	2566#
TST410	026466	8117	8125#
TST411	026536	8139	8147#
TST412	026602	8166	8173#
TST413	026652	8193	8200#
TST414	026726	8220	8227#
TST415	027002	8247	8254#
TST416	027056	8275	8283#
TST417	027136	8304	8312#
TST42	005174	2580	2589#
TST420	027232	8337	8345#
TST421	027314	8366	8374#
TST422	027364	8394	8401#
TST423	027434	8421	8428#
TST424	027510	8448	8455#
TST425	027564	8475	8482#
TST426	027640	8502	8509#
TST427	027720	8530	8538#
TST43	005236	2603	2612#
TST430	030000	8559	8567#
TST431	030060	8587	8595#
TST432	030132	8615	8622#
TST433	030202	8642	8649#
TST434	030260	8669	8676#
TST435	030344	8698	8706#
TST436	030432	8729	8737#
TST437	030506	8756	8763#
TST44	005274	2633#	
TST440	030564	8782	8789#
TST441	030640	8808	8815#
TST442	030716	8834	8841#
TST443	030766	8854	8862#
TST444	031036	8875	8883#
TST445	031122	8906	8915#
TST446	031206	8938	8947#
TST447	031272	8970	8979#
TST45	005322	2641	2649#
TST450	031356	9002	9011#
TST451	031452	9039	9048#
TST452	031546	9076	9085#
TST453	031644	9113	9122#
TST454	031742	9150	9159#
TST455	032036	9187	9196#
TST456	032132	9224	9233#
TST457	032242	9265	9274#
TST46	005364	2660	2668#
TST460	032340	9302	9311#
TST461	032430	9334	9343#
TST462	032520	9366	9375#
TST463	032610	9398	9407#
TST464	032700	9430	9439#
TST465	032740	9452	9460#
TST466	033000	9473	9481#
TST467	033042	9494	9502#
TST47	005420	2678	2686#

TST470	033104	9515	9523#
TST471	033146	9536	9545#
TST472	033212	9558	9566#
TST473	033264	9586	9593#
TST474	033336	9614	9621#
TST475	033374	9634	9641#
TST476	033434	9653	9660#
TST477	033502	9677	9684#
TST5	003424	1999	2007#
TST50	005462	2701	2709#
TST500	033550	9701	9708#
TST501	033614	9721	9729#
TST502	033672	9749	9756#
TST503	033752	9775	9783#
TST504	034030	9802	9810#
TST505	034110	9829	9837#
TST506	034162	9851	9859#
TST507	034234	9873	9881#
TST51	005520	2718	2726#
TST510	034302	9895	9903#
TST511	034350	9917	9925#
TST512	034420	9939	9947#
TST513	034470	9961	9969#
TST514	034542	9983	9991#
TST515	034616	10011	10018#
TST516	034676	10042	10049#
TST517	034752	10069	10076#
TST52	005544	2741#	
TST520	035020	10096	10103#
TST521	035076	10123	10131#
TST522	035150	10151	10158#
TST523	035222	10178	10185#
TST524	035274	10205	10212#
TST525	035346	10231	10238#
TST526	035430	10259	10267#
TST527	035504	10288	10296#
TST53	005600	2751	2759#
TST530	035566	10317	10325#
TST531	035642	10346	10354#
TST532	035724	10375	10383#
TST533	036006	10404	10411#
TST534	036070	10432	10440#
TST535	036150	10461	10469#
TST536	036230	10489	10497#
TST537	036304	10517	10524#
TST54	005632	2768	2776#
TST540	036354	10544	10551#
TST541	036434	10572	10579#
TST542	036504	10599	10606#
TST543	036556	10626	10633#
TST544	036630	10653	10659#
TST545	036702	10679	10686#
TST546	036754	10706	10713#
TST547	037032	10733	10740#
TST55	005700	2793	2801#
TST550	037114	10761	10769#

TST551	037172	10790	10798#		
TST552	037260	10820	10828#		
TST553	037340	10850	10858#		
TST554	037426	10880	10888#		
TST555	037526	10914	10922#		
TST556	037612	10944	10952#		
TST557	037676	10974	10982#		
TST56	005750	2818	2826#		
TST560	037762	11003	11011#		
TST561	040020	11023	11030#		
TST562	040064	11043	11051#		
TST563	040132	11065	11073#		
TST564	040202	11087	11095#		
TST565	040252	11109	11117#		
TST566	040322	11131	11139#		
TST567	040374	11153	11161#		
TST57	006010	2837	2845#		
TST570	040446	11175	11183#		
TST571	040510	11196	11204#		
TST572	040560	11218	11226#		
TST573	040622	11239	11247#		
TST574	040672	11261	11269#		
TST575	040732	11279	11284	11291#	
TST576	040772	11301	11306	11313#	
TST577	041042	11323	11333	11340#	
TST6	003444	2013	2021#		
TST60	006062	2862	2869#		
TST600	041102	11350	11355	11362#	
TST601	041160	11372	11382	11385	11394#
TST602	041226	11404	11409	11412	11420#
TST603	041304	11430	11434	11444	11451#
TST604	041344	11461	11466	11473#	
TST605	041422	11483	11493	11496	11503#
TST606	041470	11513	11518	11521	11529#
TST607	041536	11539	11544	11547	11555#
TST61	006124	2884	2892#		
TST610	041604	11565	11570	11573	11581#
TST611	041660	11591	11594	11599	11602
TST612	041734	11621	11624	11629	11632
TST613	042012	11657	11669#		11611#
TST614	042100	11700#			11641#
TST615	042162	11729#			
TST616	042232	11754#			
TST617	042310	11770	11783#		
TST62	006156	2901	2908#		
TST620	042374	11799	11816#		
TST621	042456	11835	11848#		
TST622	042542	11864	11881#		
TST623	042622	11897	11909#		
TST624	042710	11926	11943#		
TST625	042740	11951	11959#		
TST626	043014	11977	11980	11986#	
TST627	043050	11998	12005#		
TST63	006216	2918	2926#		
TST630	043104	12017	12024#		
TST631	043140	12036	12043#		

TST632	043174	12055	12062#			
TST633	043276	12092	12103#			
TST634	043402	12129	12140#			
TST635	043510	12166	12177#			
TST636	043666	12193	12201	12213	12222	12237#
TST637	044044	12253	12261	12273	12282	12297#
TST64	006274	2944	2952#			
TST640	044126	12321#				
TST641	044162	12331	12339#			
TST642	044220	12349	12357#			
TST643	044332	12387#				
TST644	044450	12419#				
TST645	044566	12452#				
TST646	044702	12485#				
TST647	045106	12542#				
TST65	006352	2972	2980#			
TST650	045310	12599#				
TST651	045426	12630#				
TST652	045544	12661#				
TST653	045662	12692#				
TST654	046000	12723#				
TST655	046116	12754#				
TST656	046250	12790#				
TST657	046370	12825#				
TST66	006406	2990	2998#			
TST660	046530	12870#				
TST661	047030	12953#				
TST662	047076	12973#				
TST663	047232	13013#				
TST664	047346	13051#				
TST665	047430	13076#				
TST666	047512	13101#				
TST667	047574	13126#				
TST67	006444	3008	3016#			
TST670	047660	13150#				
TST671	047766	13181	13189#			
TST672	050074	13219	13227#			
TST673	050234	13258	13279#			
TST674	050416	13327	13339#			
TST675	050600	13387	13399#			
TST676	050674	13429#				
TST677	051000	13461#				
TST7	003476	2031	2038#			
TST70	006504	3026	3034#			
TST700	051104	13493#				
TST701	051164	13516#				
TST702	051244	13539#				
TST703	051324	13561#				
TST704	051424	13594#				
TST705	051530	13612	13629#			
TST706	051636	13647	13664#			
TST707	051744	13682	13699#			
TST71	006534	3043	3051#			
TST710	052032	13725#				
TST711	052120	13752#				
TST712	052206	13779#				

	916#	917#	918#	919#	920#	921#														
SCM4 = 000005	916#	917#	918#	919#	920#	921#														
SCPUOP 001146	945#																			
SCRFL 001115	924#	15705	15843	16034	16057	16069	16088	16093	16097	16161	16196									
SDEVCT 001130	936#																			
SDOAGN 060654	15695	15707	15713#																	
SENDAD 060644	867	15709#																		
SENDCT 060600	15697#																			
SENULL 060660	15716#																			
SENV 001140	941#	16036	16140	16287	16311															
SENVN 001141	942#	4626	16142	16147	16289															
SEOP 060550	15688#																			
SEOPCT 060572	15694#	15698																		
SERFLG 001003	881#	15921	15957	15959	15965*	15991	16024*	16057												
SERMAX 001015	887#	15959	15982*	15991																
SERROR 061620	4620	15648	16011#																	
SERRPC 001016	888#	16028*	16029*	16030	16057	16075	17096	17098	17100	17102	17103	17105	17106							
	17108	17112	17124	17126	17129															
SERRTB 001150	968#	16083																		
SERTY 062046	16033	16068#																		
SERTTL 001012	885#	1925*	15703	15844*	16027*	16057														
SESCAP 001112	922#	15932*	15934*	15981*	16049	16051	16057													
SETABL 001140	940#																			
SETEND 001150	860	952#																		
SFATAL 001122	933#	16315*																		
SFFLG 063164	16278*	16281*	16309	16318*	16326#															
SFILLC 001056	906#	16165	16196																	
SFILLS 001055	905#	16196																		
SGDADR 001020	889#																			
SGDDAT 001024	891#																			
SGET42 060634	15706#																			
SGTSWR= ***** U	16369																			
SHD = 000000	705																			
SHIBTS 000700	855#																			
SICNT 001004	882#	15972*	15973	15975*	15990															
SILLUP 061036	15726	15742	15764#																	
SINTAG 001035	896#																			
SITEMB 001014	886#	16030*	16038	16057	16072															
SLF 001116	925#	16057	16196																	
SLFLG 063163	16319*	16325#																		
SLPADR 001006	883#	4637*	15963*	15979*	15984	15990														
SLPERR 001010	884#	11646*	11674*	11709*	11734*	11759*	11788*	11821*	11853*	11886*	11914*	12067*	12109*							
	12146*	12303*	12364*	12393*	12490*	12547*	12604*	12635*	12666*	12697*	12728*	12763*	12795*							
	12831*	12875*	12958*	13131*	13498*	13521*	13544*	13566*	13706*	13721*	13732*	13747*	13759*							
	13774*	13786*	13804*	13816*	13834*	13858*	13874*	13903*	13957*	14005*	14027*	14065*	14087*							
	14125*	14147*	14185*	14207*	14233*	14252*	14277*	14296*	15109*	15140*	15963	15980*	15990							
	16048																			
SMAIL 001120	856	860	931#	15978	16036	16140														
SMBADR 000702	856#																			
SMFLG 063162	16279*	16285	16320*	16324#																
SMSGAD 001134	938#	16295*	16298																	
SMSGLG 001136	939#	16300*																		
SMSGTY 001120	932#	16293	16301*	16313	16317*															
SMXCNT 061610	15976	15990#																		
SNULL 001054	904#	16167	16196																	
SNWTST= 000001	1936#	1949#	1963#	1975#	1989#	2004#	2018#	2035#	2051#	2068#	2090#	2112#	2130#							
	2148#	2167#	2186#	2203#	2226#	2249#	2266#	2288#	2314#	2344#	2363#	2381#	2397#							

2415#	2434#	2455#	2475#	2496#	2517#	2539#	2563#	2586#	2609#	2630#	2646#	2665#		
2683#	2706#	2723#	2738#	2756#	2773#	2798#	2823#	2842#	2866#	2889#	2905#	2923#		
2949#	2977#	2995#	3013#	3031#	3048#	3072#	3088#	3106#	3125#	3157#	3194#	3231#		
3268#	3305#	3330#	3355#	3380#	3406#	3432#	3458#	3484#	3510#	3532#	3555#	3591#		
3627#	3645#	3663#	3681#	3704#	3720#	3748#	3762#	3776#	3790#	3805#	3820#	3835#		
3849#	3864#	3879#	3894#	3909#	3924#	3939#	3953#	3968#	3983#	3998#	4013#	4027#		
4042#	4057#	4072#	4098#	4116#	4133#	4169#	4190#	4214#	4245#	4267#	4307#	4328#		
4368#	4391#	4413#	4432#	4456#	4477#	4505#	4526#	4544#	4562#	4564	4642#	4656#		
4670#	4690#	4712#	4732#	4754#	4768#	4783#	4798#	4813#	4828#	4842#	4857#	4872#		
4887#	4913#	4934#	4960#	4977#	5029#	5046#	5072#	5090#	5116#	5142#	5194#	5221#		
5247#	5273#	5299#	5326#	5353#	5380#	5407#	5433#	5459#	5485#	5511#	5537#	5563#		
5590#	5618#	5646#	5674#	5702#	5730#	5760#	5786#	5821#	5847#	5878#	5904#	5934#		
5959#	5985#	6012#	6038#	6063#	6089#	6115#	6141#	6167#	6193#	6219#	6244#	6270#		
6296#	6322#	6348#	6373#	6399#	6425#	6451#	6477#	6503#	6529#	6555#	6581#	6609#		
6638#	6670#	6702#	6729#	6756#	6783#	6810#	6837#	6864#	6890#	6917#	6944#	6971#		
6998#	7025#	7052#	7079#	7106#	7133#	7160#	7187#	7214#	7241#	7267#	7293#	7319#		
7345#	7371#	7398#	7420#	7446#	7467#	7494#	7520#	7546#	7567#	7594#	7621#	7645#		
7669#	7694#	7718#	7743#	7762#	7782#	7803#	7824#	7853#	7882#	7911#	7940#	7969#		
7998#	8027#	8056#	8078#	8100#	8122#	8144#	8170#	8197#	8224#	8251#	8280#	8309#		
8342#	8371#	8398#	8425#	8452#	8479#	8506#	8535#	8564#	8592#	8619#	8646#	8673#		
8703#	8734#	8760#	8786#	8812#	8838#	8859#	8880#	8912#	8944#	8976#	9008#	9045#		
9082#	9119#	9156#	9193#	9230#	9271#	9308#	9340#	9372#	9404#	9436#	9457#	9478#		
9499#	9520#	9542#	9563#	9590#	9618#	9638#	9657#	9681#	9705#	9726#	9753#	9780#		
9807#	9834#	9856#	9878#	9900#	9922#	9944#	9966#	9988#	10015#	10046#	10073#	10100#		
10128#	10155#	10182#	10209#	10235#	10264#	10293#	10322#	10351#	10380#	10408#	10437#	10466#		
10494#	10521#	10548#	10576#	10603#	10630#	10656#	10683#	10710#	10737#	10766#	10795#	10825#		
10855#	10885#	10919#	10949#	10979#	11008#	11027#	11048#	11070#	11092#	11114#	11136#	11158#		
11180#	11201#	11223#	11244#	11266#	11288#	11310#	11337#	11359#	11391#	11417#	11448#	11470#		
11500#	11526#	11552#	11578#	11608#	11638#	11666#	11697#	11726#	11751#	11780#	11813#	11845#		
11878#	11906#	11940#	11956#	11983#	12002#	12021#	12040#	12059#	12100#	12137#	12174#	12234#		
12294#	12318#	12336#	12354#	12384#	12416#	12449#	12482#	12539#	12596#	12627#	12658#	12689#		
12720#	12751#	12787#	12819#	12821	12863#	12865	12950#	12970#	13010#	13048#	13073#	13098#		
13123#	13147#	13186#	13224#	13276#	13336#	13396#	13426#	13458#	13490#	13513#	13536#	13558#		
13589#	13591	13626#	13661#	13696#	13722#	13749#	13776#	13806#	13846#	13848	13876#	13891#		
13916#	13932#	13949#	13969#	13971	14029#	14031	14089#	14091	14149#	14151	14209#	14211		
14254#	14256	14298#	14328#	14359#	14395#	14431#	14467#	14503#	14534#	14558#	14587#	14616#		
14645#	14667#	14669	14713#	14737#	14762#	14786#	14814#	14843#	14888#	14890	14926#	14928		
14977#	14979	15023#	15025	15058#	15060	15145#	15147	15206#	15208	15276#	15278	15313#		
15315	15390#	15392	15429#	15431	15551#	15553	15638#							
16229*	16258*	16271#												
16224*	16228*	16233	16236*	16247*	16273#									
15941	15944	15964	15974	15983#										
SPASS	001126	935#	1926*	15691*	15692*	15700	15716	15970	15991					
SPASTM	000706	858#												
SPOWER	061044	15760	15767#											
SPWRAD	061032	15762#												
SPWRDN	060664	4624	15726#	15757										
SPWRMG	061026	15760#												
SPWRUP	060736	15736	15742#											
SQUES	001114	923#	16057	16196										
SRDCHR=	***** U	16370												
SRDDEC=	***** U	16370												
SRDLIN=	***** U	16370												
SRDOCT=	***** U	16370												
SREGAD	001060	908#	16013	17110	17124	17126								
SREGO	001062	910#	15446*	15457*	15474	15544*	17098	17103	17105	17106	17112	17124	17126	17129

SOCNT 062714
SOMODE 062716
SOVER 061560
SPASS 001126
SPASTM 000706
SPOWER 061044
SPWRAD 061032
SPWRDN 060664
SPWRMG 061026
SPWRUP 060736
SQUES 001114
SRDCHR= ***** U
SRDDEC= ***** U
SRDLIN= ***** U
SRDOCT= ***** U
SREGAD 001060
SREGO 001062

\$REG1	001064	911#	17103	17106	17108	17110	17124	17126						
\$REG2	001066	912#	17106	17108	17110	17123								
\$REG3	001070	913#	17106	17108	17110	17122	17126							
\$REG4	001072	914#	17110	17121	17126									
\$REG5	001074	915#	1933*	12189*	12197*	12209*	12227*	12249*	12257*	12269*	12287*	12988*	12996*	13003*
		13027*	13035*	13042*	13066*	13091*	13116*	13292*	13302*	13312*	13322*	13352*	13362*	13372*
		13382*	13410*	13416*	13420*	13442*	13451*	13474*	13483*	16053*	17124			
\$RTNAD	060656	15715#												
\$R2A =	***** U	16370												
\$SAVRE =	***** U	16370												
\$SAVR6	061042	15735*	15743	15744*	15745*	15766#								
\$SCOPE	061260	4618	15663	15929#										
\$SETUP =	000000	15690	15930	16012	16046	16053								
\$SVLAD	061524	15952	15977#											
\$SVPC =	000714	865#	870											
\$SWR =	165000	695#	705	709	710	711	712	713	714	715	716	921	922	923
		1941	1954	1968	1980	1994	2009	2023	2040	2056	2073	2095	2117	2135
		2153	2172	2191	2208	2231	2254	2271	2293	2319	2349	2368	2386	2402
		2420	2439	2460	2480	2501	2522	2544	2568	2591	2614	2635	2651	2670
		2688	2711	2728	2743	2761	2778	2803	2828	2847	2871	2894	2910	2928
		2954	2982	3000	3018	3036	3053	3077	3093	3111	3130	3162	3199	3236
		3273	3310	3335	3360	3385	3411	3437	3463	3489	3515	3537	3560	3596
		3632	3650	3668	3686	3709	3725	3753	3767	3781	3795	3810	3825	3840
		3854	3869	3884	3899	3914	3929	3944	3958	3973	3988	4003	4018	4032
		4047	4062	4077	4103	4121	4138	4174	4195	4219	4250	4272	4312	4333
		4373	4396	4418	4437	4461	4482	4510	4531	4549	4575	4649	4663	4677
		4697	4719	4739	4761	4775	4790	4805	4820	4835	4849	4864	4879	4894
		4920	4941	4967	4984	5036	5053	5079	5097	5123	5149	5201	5228	5254
		5280	5306	5333	5360	5387	5414	5440	5466	5492	5518	5544	5570	5597
		5625	5653	5681	5709	5737	5767	5793	5828	5854	5885	5911	5941	5966
		5992	6019	6045	6070	6096	6122	6148	6174	6200	6226	6251	6277	6303
		6329	6355	6380	6406	6432	6458	6484	6510	6536	6562	6588	6616	6645
		6677	6709	6736	6763	6790	6817	6844	6871	6897	6924	6951	6978	7005
		7032	7059	7086	7113	7140	7167	7194	7221	7248	7274	7300	7326	7352
		7378	7405	7427	7453	7474	7501	7527	7553	7574	7601	7628	7652	7676
		7701	7725	7750	7769	7789	7810	7831	7860	7889	7918	7947	7976	8005
		8034	8063	8085	8107	8129	8151	8177	8204	8231	8258	8287	8316	8349
		8378	8405	8432	8459	8486	8513	8542	8571	8599	8626	8653	8680	8710
		8741	8767	8793	8819	8845	8866	8887	8919	8951	8983	9015	9052	9089
		9126	9163	9200	9237	9278	9315	9347	9379	9411	9443	9464	9485	9506
		9527	9549	9570	9597	9625	9645	9664	9688	9712	9733	9760	9787	9814
		9841	9863	9885	9907	9929	9951	9973	9995	10022	10053	10080	10107	10135
		10162	10189	10216	10242	10271	10300	10329	10358	10387	10415	10444	10473	10501
		10528	10555	10583	10610	10637	10663	10690	10717	10744	10773	10802	10832	10862
		10892	10926	10956	10986	11015	11034	11055	11077	11099	11121	11143	11165	11187
		11208	11230	11251	11273	11295	11317	11344	11366	11398	11424	11455	11477	11507
		11533	11559	11585	11615	11645	11673	11704	11733	11758	11787	11820	11852	11885
		11913	11947	11962	11990	12009	12028	12046	12066	12107	12144	12181	12241	12301
		12325	12343	12361	12391	12423	12456	12489	12546	12603	12634	12665	12696	12727
		12758	12794	12829	12874	12957	12977	13017	13055	13080	13105	13130	13154	13193
		13231	13283	13343	13403	13433	13465	13497	13520	13543	13565	13598	13633	13668
		13702	13728	13755	13782	13812	13854	13883	13898	13923	13939	13956	14004	14064
		14124	14184	14230	14274	14304	14334	14365	14401	14437	14473	14509	14540	14564
		14593	14622	14651	14678	14720	14744	14769	14792	14820	14849	14898	14941	14992
		15036	15078	15156	15219	15286	15327	15403	15444	15569	15644	15685	15690	15708
		15714	15716	15763	15922	15923	15924	15925	15926	15943	15955	15957	15958	15959

	15966	15967	15968	15980	15983	15990	16004	16005	16006	16007	16027	16031	16043
SSWREG 001142	16046	16057											
SSWRMK= 000000	943#	4628											
STESTN 001124	15926												
STIMES 001110	934#	4639*	15930	15978*	15988*	17129							
STKB 001046	921#	4638*	12423*	12456*	15690*	15966*	15973	15976*	15990				
STKS 001044	901#												
STMP0 001076	900#												
	916#	14304*	14334*	14365*	14401*	14437*	14473*	14509*	14540*	14564*	14593*	14622*	14651*
STMP1 001100	14792*	14820*	14853*	14951*	14960	15095*	15125*	17096	17112				
STMP2 001102	917#	14952*	14965*	14971	15012*	15052*	15096*	15126*	17096	17098	17100		
STMP3 001104	918#	14943*	14953	14963	14970*	14972	15013*	15053*	17098	17100			
STMP4 001106	919#	14959*	14963	15014*	17100								
STN = 000770	920#	14948*	14966	14969*									
	695#	705	1936	1940	1941#	1944	1949	1953	1954#	1958	1963	1967	1968#
	1971	1975	1979	1980#	1984	1989	1993	1994#	1999	2004	2008	2009#	2013
	2018	2022	2023#	2031	2035	2039	2040#	2047	2051	2055	2056#	2063	2068
	2072	2073#	2085	2090	2094	2095#	2107	2112	2116	2117#	2125	2130	2134
	2135#	2143	2148	2152	2153#	2162	2167	2171	2172#	2181	2186	2190	2191#
	2199	2203	2207	2208#	2221	2226	2230	2231#	2244	2249	2253	2254#	2261
	2266	2270	2271#	2283	2288	2292	2293#	2309	2314	2318	2319#	2338	2344
	2348	2349#	2358	2363	2367	2368#	2376	2381	2385	2386#	2392	2397	2401
	2402#	2410	2415	2419	2420#	2429	2434	2438	2439#	2450	2455	2459	2460#
	2471	2475	2479	2480#	2491	2496	2500	2501#	2512	2517	2521	2522#	2534
	2539	2543	2544#	2557	2563	2567	2568#	2580	2586	2590	2591#	2603	2609
	2613	2614#	2630	2634	2635#	2641	2646	2650	2651#	2660	2665	2669	2670#
	2678	2683	2687	2688#	2701	2706	2710	2711#	2718	2723	2727	2728#	2738
	2742	2743#	2751	2756	2760	2761#	2768	2773	2777	2778#	2793	2798	2802
	2803#	2818	2823	2827	2828#	2837	2842	2846	2847#	2862	2866	2870	2871#
	2884	2889	2893	2894#	2901	2905	2909	2910#	2918	2923	2927	2928#	2944
	2949	2953	2954#	2972	2977	2981	2982#	2990	2995	2999	3000#	3008	3013
	3017	3018#	3026	3031	3035	3036#	3043	3048	3052	3053#	3072	3076	3077#
	3083	3088	3092	3093#	3101	3106	3110	3111#	3120	3125	3129	3130#	3152
	3157	3161	3162#	3188	3194	3198	3199#	3225	3231	3235	3236#	3262	3268
	3272	3273#	3299	3305	3309	3310#	3326	3330	3334	3335#	3351	3355	3359
	3360#	3376	3380	3384	3385#	3401	3406	3410	3411#	3427	3432	3436	3437#
	3453	3458	3462	3463#	3479	3484	3488	3489#	3505	3510	3514	3515#	3528
	3532	3536	3537#	3550	3555	3559	3560#	3591	3595	3596#	3627	3631	3632#
	3641	3645	3649	3650#	3659	3663	3667	3668#	3677	3681	3685	3686#	3699
	3704	3708	3709#	3715	3720	3724	3725#	3743	3748	3752	3753#	3757	3762
	3766	3767#	3771	3776	3780	3781#	3785	3790	3794	3795#	3800	3805	3809
	3810#	3815	3820	3824	3825#	3830	3835	3839	3840#	3844	3849	3853	3854#
	3859	3864	3868	3869#	3874	3879	3883	3884#	3889	3894	3898	3899#	3904
	3909	3913	3914#	3919	3924	3928	3929#	3934	3939	3943	3944#	3948	3953
	3957	3958#	3963	3968	3972	3973#	3978	3983	3987	3988#	3993	3998	4002
	4003#	4008	4013	4017	4018#	4022	4027	4031	4032#	4037	4042	4046	4047#
	4052	4057	4061	4062#	4067	4072	4076	4077#	4093	4098	4102	4103#	4111
	4116	4120	4121#	4128	4133	4137	4138#	4164	4169	4173	4174#	4185	4190
	4194	4195#	4209	4214	4218	4219#	4240	4245	4249	4250#	4261	4267	4271
	4272#	4301	4307	4311	4312#	4328	4332	4333#	4362	4368	4372	4373#	4391
	4395	4396#	4408	4413	4417	4418#	4428	4432	4436	4437#	4456	4460	4461#
	4472	4477	4481	4482#	4505	4509	4510#	4518	4526	4530	4531#	4538	4544
	4548	4549#	4556	4562	4574	4575#	4642	4647	4649#	4652	4656	4661	4663#
	4666	4670	4675	4677#	4686	4690	4695	4697#	4708	4712	4717	4719#	4728
	4732	4737	4739#	4750	4754	4759	4761#	4764	4768	4773	4775#	4779	4783
	4788	4790#	4794	4798	4803	4805#	4809	4813	4818	4820#	4824	4828	4833

4835#	4838	4842	4847	4849#	4853	4857	4862	4864#	4868	4872	4877	4879#
4883	4887	4892	4894#	4909	4913	4918	4920#	4930	4934	4939	4941#	4956
4960	4965	4967#	4973	4977	4982	4984#	5025	5029	5034	5036#	5042	5046
5051	5053#	5068	5072	5077	5079#	5086	5090	5095	5097#	5112	5116	5121
5123#	5138	5142	5147	5149#	5190	5194	5199	5201#	5217	5221	5226	5228#
5243	5247	5252	5254#	5269	5273	5278	5280#	5295	5299	5304	5306#	5322
5326	5331	5333#	5349	5353	5358	5360#	5376	5380	5385	5387#	5403	5407
5412	5414#	5429	5433	5438	5440#	5455	5459	5464	5466#	5481	5485	5490
5492#	5507	5511	5516	5518#	5533	5537	5542	5544#	5559	5563	5568	5570#
5586	5590	5595	5597#	5613	5618	5623	5625#	5641	5646	5651	5653#	5669
5674	5679	5681#	5697	5702	5707	5709#	5725	5730	5735	5737#	5755	5760
5765	5767#	5781	5786	5791	5793#	5816	5821	5826	5828#	5842	5847	5852
5854#	5873	5878	5883	5885#	5899	5904	5909	5911#	5929	5934	5939	5941#
5954	5959	5964	5966#	5981	5985	5990	5992#	6008	6012	6017	6019#	6034
6038	6043	6045#	6059	6063	6068	6070#	6085	6089	6094	6096#	6111	6115
6120	6122#	6137	6141	6146	6148#	6163	6167	6172	6174#	6189	6193	6198
6200#	6215	6219	6224	6226#	6240	6244	6249	6251#	6266	6270	6275	6277#
6292	6296	6301	6303#	6318	6322	6327	6329#	6344	6348	6353	6355#	6369
6373	6378	6380#	6395	6399	6404	6406#	6421	6425	6430	6432#	6447	6451
6456	6458#	6473	6477	6482	6484#	6499	6503	6508	6510#	6525	6529	6534
6536#	6551	6555	6560	6562#	6577	6581	6586	6588#	6604	6609	6614	6616#
6633	6638	6643	6645#	6665	6670	6675	6677#	6697	6702	6707	6709#	6724
6729	6734	6736#	6751	6756	6761	6763#	6778	6783	6788	6790#	6805	6810
6815	6817#	6832	6837	6842	6844#	6859	6864	6869	6871#	6885	6890	6895
6897#	6912	6917	6922	6924#	6939	6944	6949	6951#	6966	6971	6976	6978#
6993	6998	7003	7005#	7020	7025	7030	7032#	7047	7052	7057	7059#	7074
7079	7084	7086#	7101	7106	7111	7113#	7128	7133	7138	7140#	7155	7160
7165	7167#	7182	7187	7192	7194#	7209	7214	7219	7221#	7236	7241	7246
7248#	7263	7267	7272	7274#	7289	7293	7298	7300#	7315	7319	7324	7326#
7341	7345	7350	7352#	7367	7371	7376	7378#	7393	7398	7403	7405#	7415
7420	7425	7427#	7441	7446	7451	7453#	7462	7467	7472	7474#	7489	7494
7499	7501#	7515	7520	7525	7527#	7541	7546	7551	7553#	7562	7567	7572
7574#	7590	7594	7599	7601#	7617	7621	7626	7628#	7641	7645	7650	7652#
7665	7669	7674	7676#	7690	7694	7699	7701#	7714	7718	7723	7725#	7739
7743	7748	7750#	7758	7762	7767	7769#	7778	7782	7787	7789#	7798	7803
7808	7810#	7819	7824	7829	7831#	7849	7853	7858	7860#	7878	7882	7887
7889#	7907	7911	7916	7918#	7936	7940	7945	7947#	7965	7969	7974	7976#
7994	7998	8003	8005#	8023	8027	8032	8034#	8052	8056	8061	8063#	8073
8078	8083	8085#	8095	8100	8105	8107#	8117	8122	8127	8129#	8139	8144
8149	8151#	8166	8170	8175	8177#	8193	8197	8202	8204#	8220	8224	8229
8231#	8247	8251	8256	8258#	8275	8280	8285	8287#	8304	8309	8314	8316#
8337	8342	8347	8349#	8366	8371	8376	8378#	8394	8398	8403	8405#	8421
8425	8430	8432#	8448	8452	8457	8459#	8475	8479	8484	8486#	8502	8506
8511	8513#	8530	8535	8540	8542#	8559	8564	8569	8571#	8587	8592	8597
8599#	8615	8619	8624	8626#	8642	8646	8651	8653#	8669	8673	8678	8680#
8698	8703	8708	8710#	8729	8734	8739	8741#	8756	8760	8765	8767#	8782
8786	8791	8793#	8808	8812	8817	8819#	8834	8838	8843	8845#	8854	8859
8864	8866#	8875	8880	8885	8887#	8906	8912	8917	8919#	8938	8944	8949
8951#	8970	8976	8981	8983#	9002	9008	9013	9015#	9039	9045	9050	9052#
9076	9082	9087	9089#	9113	9119	9124	9126#	9150	9156	9161	9163#	9187
9193	9198	9200#	9224	9230	9235	9237#	9265	9271	9276	9278#	9302	9308
9313	9315#	9334	9340	9345	9347#	9366	9372	9377	9379#	9398	9404	9409
9411#	9430	9436	9441	9443#	9452	9457	9462	9464#	9473	9478	9483	9485#
9494	9499	9504	9506#	9515	9520	9525	9527#	9536	9542	9547	9549#	9558
9563	9568	9570#	9586	9590	9595	9597#	9614	9618	9623	9625#	9634	9638
9643	9645#	9653	9657	9662	9664#	9677	9681	9686	9688#	9701	9705	9710

9712#	9721	9726	9731	9733#	9749	9753	9758	9760#	9775	9780	9785	9787#
9802	9807	9812	9814#	9829	9834	9839	9841#	9851	9856	9861	9863#	9873
9878	9883	9885#	9895	9900	9905	9907#	9917	9922	9927	9929#	9939	9944
9949	9951#	9961	9966	9971	9973#	9983	9988	9993	9995#	10011	10015	10020
10022#	10042	10046	10051	10053#	10069	10073	10078	10080#	10096	10100	10105	10107#
10123	10128	10133	10135#	10151	10155	10160	10162#	10178	10182	10187	10189#	10205
10209	10214	10216#	10231	10235	10240	10242#	10259	10264	10269	10271#	10288	10293
10298	10300#	10317	10322	10327	10329#	10346	10351	10356	10358#	10375	10380	10385
10387#	10404	10408	10413	10415#	10432	10437	10442	10444#	10461	10466	10471	10473#
10489	10494	10499	10501#	10517	10521	10526	10528#	10544	10548	10553	10555#	10572
10576	10581	10583#	10599	10603	10608	10610#	10626	10630	10635	10637#	10653	10656
10661	10663#	10679	10683	10688	10690#	10706	10710	10715	10717#	10733	10737	10742
10744#	10761	10766	10771	10773#	10790	10795	10800	10802#	10820	10825	10830	10832#
10850	10855	10860	10862#	10880	10885	10890	10892#	10914	10919	10924	10926#	10944
10949	10954	10956#	10974	10979	10984	10986#	11003	11008	11013	11015#	11023	11027
11032	11034#	11043	11048	11053	11055#	11065	11070	11075	11077#	11087	11092	11097
11099#	11109	11114	11119	11121#	11131	11136	11141	11143#	11153	11158	11163	11165#
11175	11180	11185	11187#	11196	11201	11206	11208#	11218	11223	11228	11230#	11239
11244	11249	11251#	11261	11266	11271	11273#	11279	11284	11288	11293	11295#	11301
11306	11310	11315	11317#	11323	11333	11337	11342	11344#	11350	11355	11359	11364
11366#	11372	11382	11385	11391	11396	11398#	11404	11409	11412	11417	11422	11424#
11430	11434	11444	11448	11453	11455#	11461	11466	11470	11475	11477#	11483	11493
11496	11500	11505	11507#	11513	11518	11521	11526	11531	11533#	11539	11544	11547
11552	11557	11559#	11565	11570	11573	11578	11583	11585#	11591	11594	11599	11602
11608	11613	11615#	11621	11624	11629	11632	11638	11643	11645#	11657	11666	11671
11673#	11697	11702	11704#	11726	11731	11733#	11751	11756	11758#	11770	11780	11785
11787#	11799	11813	11818	11820#	11835	11845	11850	11852#	11864	11878	11883	11885#
11897	11906	11911	11913#	11926	11940	11945	11947#	11951	11956	11961	11962#	11977
11980	11983	11988	11990#	11998	12002	12007	12009#	12017	12021	12026	12028#	12036
12040	12045	12046#	12055	12059	12064	12066#	12092	12100	12105	12107#	12129	12137
12142	12144#	12166	12174	12179	12181#	12193	12201	12213	12222	12234	12239	12241#
12253	12261	12273	12282	12294	12299	12301#	12318	12323	12325#	12331	12336	12341
12343#	12349	12354	12359	12361#	12384	12389	12391#	12416	12421	12423#	12449	12454
12456#	12482	12487	12489#	12539	12544	12546#	12596	12601	12603#	12627	12632	12634#
12658	12663	12665#	12689	12694	12696#	12720	12725	12727#	12751	12756	12758#	12787
12792	12794#	12819	12827	12829#	12863	12872	12874#	12950	12955	12957#	12970	12975
12977#	13010	13015	13017#	13048	13053	13055#	13073	13078	13080#	13098	13103	13105#
13123	13128	13130#	13147	13152	13154#	13181	13186	13191	13193#	13219	13224	13229
13231#	13258	13276	13281	13283#	13327	13336	13341	13343#	13387	13396	13401	13403#
13426	13431	13433#	13458	13463	13465#	13490	13495	13497#	13513	13518	13520#	13536
13541	13543#	13558	13563	13565#	13589	13596	13598#	13612	13626	13631	13633#	13647
13661	13666	13668#	13682	13696	13701	13702#	13722	13727	13728#	13749	13754	13755#
13776	13781	13782#	13806	13811	13812#	13846	13853	13854#	13876	13881	13883#	13891
13896	13898#	13916	13921	13923#	13932	13937	13939#	13949	13954	13956#	13969	14003
14004#	14029	14063	14064#	14089	14123	14124#	14149	14183	14184#	14209	14229	14230#
14245	14254	14273	14274#	14289	14298	14303	14304#	14324	14328	14333	14334#	14355
14359	14364	14365#	14391	14395	14400	14401#	14427	14431	14436	14437#	14463	14467
14472	14473#	14499	14503	14508	14509#	14530	14534	14539	14540#	14554	14558	14563
14564#	14583	14587	14592	14593#	14612	14616	14621	14622#	14641	14645	14650	14651#
14663	14667	14676	14678#	14706	14713	14718	14720#	14734	14737	14742	14744#	14759
14762	14767	14769#	14783	14786	14791	14792#	14811	14814	14819	14820#	14840	14843
14848	14849#	14872	14888	14896	14898#	14926	14939	14941#	14977	14990	14992#	15023
15034	15036#	15058	15076	15078#	15145	15154	15156#	15206	15217	15219#	15276	15284
15286#	15313	15325	15327#	15390	15401	15403#	15429	15442	15444#	15551	15567	15569#
15638	15642	15644#	15675									
903#	16185*	16196										

GETPRI	1#	827#													
GETSWR	1#	827#													
MSG	4562#	4564	12819#	12821	12863#	12865	13589#	13591	13846#	13848	13969#	13971	14029#	14031	14089#
	14091	14149#	14151	14209#	14211	14254#	14256	14667#	14669	15144#	15147	15205#	15208	15275#	15278
	15312#	15315	15389#	15392	15428#	15431	15550#	15553							
MSGJ	14888#	14890													
MSGM1	14925#	14928													
MSGM10	15022#	15025													
MSGM11	15057#	15060													
MSGM3	14977#	14979													
MULT	1#	827#													
NEWTST	1#	827#	1936	1949	1963	1975	1989	2004	2018	2035	2051	2068	2090	2112	2130
	2148	2167	2186	2203	2226	2249	2266	2288	2314	2344	2363	2381	2397	2415	2434
	2455	2475	2496	2517	2539	2563	2586	2609	2630	2646	2665	2683	2706	2723	2738
	2756	2773	2798	2823	2842	2866	2889	2905	2923	2949	2977	2995	3013	3031	3048
	3072	3088	3106	3125	3157	3194	3231	3268	3305	3330	3355	3380	3406	3432	3458
	3484	3510	3532	3555	3591	3627	3645	3663	3681	3704	3720	3748	3762	3776	3790
	3805	3820	3835	3849	3864	3879	3894	3909	3924	3939	3953	3968	3983	3998	4013
	4027	4042	4057	4072	4098	4116	4133	4169	4190	4214	4245	4267	4307	4328	4368
	4391	4413	4432	4456	4477	4505	4526	4544	4562	4642	4656	4670	4690	4712	4732
	4754	4768	4783	4798	4813	4828	4842	4857	4872	4887	4913	4934	4960	4977	5029
	5046	5072	5090	5116	5142	5194	5221	5247	5273	5299	5326	5353	5380	5407	5433
	5459	5485	5511	5537	5563	5590	5618	5646	5674	5702	5730	5760	5786	5821	5847
	5878	5904	5934	5959	5985	6012	6038	6063	6089	6115	6141	6167	6193	6219	6244
	6270	6296	6322	6348	6373	6399	6425	6451	6477	6503	6529	6555	6581	6609	6638
	6670	6702	6729	6756	6783	6810	6837	6864	6890	6917	6944	6971	6998	7025	7052
	7079	7106	7133	7160	7187	7214	7241	7267	7293	7319	7345	7371	7398	7420	7446
	7467	7494	7520	7546	7567	7594	7621	7645	7669	7694	7718	7743	7762	7782	7803
	7824	7853	7882	7911	7940	7969	7998	8027	8056	8078	8100	8122	8144	8170	8197
	8224	8251	8280	8309	8342	8371	8398	8425	8452	8479	8506	8535	8564	8592	8619
	8646	8673	8703	8734	8760	8786	8812	8838	8859	8880	8912	8944	8976	9008	9045
	9082	9119	9156	9193	9230	9271	9308	9340	9372	9404	9436	9457	9478	9499	9520
	9542	9563	9590	9618	9638	9657	9681	9705	9726	9753	9780	9807	9834	9856	9878
	9900	9922	9944	9966	9988	10015	10046	10073	10100	10128	10155	10182	10209	10235	10264
	10293	10322	10351	10380	10408	10437	10466	10494	10521	10548	10576	10603	10630	10656	10683
	10710	10737	10766	10795	10825	10855	10885	10919	10949	10979	11008	11027	11048	11070	11092
	11114	11136	11158	11180	11201	11223	11244	11266	11288	11310	11337	11359	11391	11417	11448
	11470	11500	11526	11552	11578	11608	11638	11666	11697	11726	11751	11780	11813	11845	11878
	11906	11940	11956	11983	12002	12021	12040	12059	12100	12137	12174	12234	12294	12318	12336
	12354	12384	12416	12449	12482	12539	12596	12627	12658	12689	12720	12751	12787	12819	12863
	12950	12970	13010	13048	13073	13098	13123	13147	13186	13224	13276	13336	13396	13426	13458
	13490	13513	13536	13558	13589	13626	13661	13696	13722	13749	13776	13806	13846	13876	13891
	13916	13932	13949	13969	14029	14089	14149	14209	14254	14298	14328	14359	14395	14431	14467
	14503	14534	14558	14587	14616	14645	14667	14713	14737	14762	14786	14814	14843	14888	14926
	14977	15023	15058	15145	15206	15276	15313	15390	15429	15551	15638				
NOINST	1263#	11959	12043	13699	13725	13752	13779	13809	14001	14061	14121	14181	14227	14271	14301
	14331	14362	14398	14434	14470	14506	14537	14561	14590	14619	14648	14789	14817	14846	
NOSCOF	1263#	1939	1952	1966	1978	1992	2007	2021	2038	2054	2071	2093	2115	2133	2151
	2170	2189	2206	2229	2252	2269	2291	2317	2347	2366	2384	2400	2418	2437	2458
	2478	2499	2520	2542	2566	2589	2612	2633	2649	2668	2686	2709	2726	2741	2759
	2776	2801	2826	2845	2869	2892	2908	2926	2952	2980	2998	3016	3034	3051	3075
	3091	3109	3128	3160	3197	3234	3271	3308	3333	3358	3383	3409	3435	3461	3487
	3513	3535	3558	3594	3630	3648	3666	3684	3707	3723	3751	3765	3779	3793	3808
	3823	3838	3852	3867	3882	3897	3912	3927	3942	3956	3971	3986	4001	4016	4030
	4045	4060	4075	4101	4119	4136	4172	4193	4217	4248	4270	4310	4331	4371	4394
	4416	4435	4459	4480	4508	4529	4547	4573	13852						

POP	1#	827#	15750	15751	16321	16322									
PREERR	15996#	16012													
PRENEW	1263#	14895	14938	14989	15033	15075	15153	15216	15283	15324	15400	15441	15566	15641	
PRESKO	15914#	15930													
PUSH	1#	827#	15728	15734	16282	16284	16305								
REPORT	1#	827#													
SCOPE	722#	4319	4646	4660	4674	4694	4716	4736	4758	4772	4787	4802	4817	4832	4846
	4861	4876	4891	4917	4938	4964	4981	5033	5050	5076	5094	5120	5146	5198	5225
	5251	5277	5303	5330	5357	5384	5411	5437	5463	5489	5515	5541	5567	5594	5622
	5650	5678	5706	5734	5764	5790	5825	5851	5882	5908	5938	5963	5989	6016	6042
	6067	6093	6119	6145	6171	6197	6223	6248	6274	6300	6326	6352	6377	6403	6429
	6455	6481	6507	6533	6559	6585	6613	6642	6674	6706	6733	6760	6787	6814	6841
	6868	6894	6921	6948	6975	7002	7029	7056	7083	7110	7137	7164	7191	7218	7245
	7271	7297	7323	7349	7375	7402	7424	7450	7471	7498	7524	7550	7571	7598	7625
	7649	7673	7698	7722	7747	7766	7786	7807	7828	7857	7886	7915	7944	7973	8002
	8031	8060	8082	8104	8126	8148	8174	8201	8228	8255	8284	8313	8346	8375	8402
	8429	8456	8483	8510	8539	8568	8596	8623	8650	8677	8707	8738	8764	8790	8816
	8842	8863	8884	8916	8948	8980	9012	9049	9086	9123	9160	9197	9234	9275	9312
	9344	9376	9408	9440	9461	9482	9503	9524	9546	9567	9594	9622	9642	9661	9685
	9709	9730	9757	9784	9811	9838	9860	9882	9904	9926	9948	9970	9992	10019	10050
	10077	10104	10132	10159	10186	10213	10239	10268	10297	10326	10355	10384	10412	10441	10470
	10498	10525	10552	10580	10607	10634	10660	10687	10714	10741	10770	10799	10829	10859	10889
	10923	10953	10983	11012	11031	11052	11074	11096	11118	11140	11162	11184	11205	11227	11248
	11270	11292	11314	11341	11363	11395	11421	11452	11474	11504	11530	11556	11582	11612	11642
	11670	11701	11730	11755	11784	11817	11849	11882	11910	11944	11960	11987	12006	12025	12044
	12063	12104	12141	12178	12238	12298	12322	12340	12358	12388	12420	12453	12486	12543	12600
	12631	12662	12693	12724	12755	12791	12826	12871	12954	12974	13014	13052	13077	13102	13127
	13151	13190	13228	13280	13340	13400	13430	13462	13494	13517	13540	13562	13595	13630	13665
	13700	13726	13753	13780	13810	13837	13880	13895	13920	13936	13953	14002	14062	14122	14182
	14228	14272	14302	14332	14363	14399	14435	14471	14507	14538	14562	14591	14620	14649	14675
	14717	14741	14766	14790	14818	14847	14897	14940	14991	15035	15077	15155	15218	15285	15326
	15402	15443	15568	15643	15689										
SETPRI	1#	827#													
SETTRA	16356#	16365	16366	16367											
SETUP	1#	827#													
SKIP	1#	827#	1943	1958	1970	1984	1999	2012	2031	2047	2063	2085	2107	2125	2143
	2162	2181	2199	2221	2244	2261	2283	2309	2338	2358	2376	2392	2410	2429	2450
	2471	2491	2512	2534	2557	2580	2603	2641	2660	2678	2701	2718	2751	2768	2793
	2818	2837	2862	2884	2901	2918	2944	2972	2990	3008	3026	3043	3083	3101	3120
	3152	3188	3225	3262	3299	3326	3351	3376	3401	3427	3453	3479	3505	3528	3550
	3641	3659	3677	3699	3715	3742	3757	3770	3784	3800	3815	3829	3844	3858	3873
	3889	3904	3919	3934	3947	3963	3978	3993	4007	4021	4037	4052	4067	4093	4111
	4128	4164	4185	4209	4240	4261	4301	4362	4408	4428	4472	4518	4538	4556	4652
	4665	4686	4708	4728	4750	4764	4778	4793	4808	4824	4838	4852	4867	4882	4909
	4930	4956	4973	5025	5042	5068	5086	5112	5138	5190	5217	5243	5269	5295	5322
	5349	5376	5403	5429	5455	5481	5507	5533	5559	5586	5613	5641	5669	5697	5725
	5755	5781	5816	5842	5873	5899	5929	5954	5981	6008	6034	6059	6085	6111	6137
	6163	6189	6215	6240	6266	6292	6318	6344	6369	6395	6421	6447	6473	6499	6525
	6551	6577	6604	6633	6665	6697	6724	6751	6778	6805	6832	6859	6885	6912	6939
	6966	6993	7020	7047	7074	7101	7128	7155	7182	7209	7236	7263	7289	7315	7341
	7367	7393	7415	7441	7462	7489	7515	7541	7562	7590	7617	7641	7665	7690	7714
	7739	7758	7778	7798	7819	7849	7878	7907	7936	7965	7994	8023	8052	8073	8095
	8117	8139	8166	8193	8220	8247	8275	8304	8337	8366	8394	8421	8448	8475	8502
	8530	8559	8587	8615	8642	8669	8698	8729	8756	8782	8808	8834	8854	8875	8906
	8938	8970	9002	9039	9076	9113	9150	9187	9224	9265	9302	9334	9366	9398	9430
	9452	9473	9494	9515	9536	9558	9586	9614	9634	9653	9677	9701	9721	9749	9775

SLASH
SPACE
STARS

9802	9829	9851	9873	9895	9917	9939	9961	9983	10011	10042	10069	10096	10123	10151
10178	10205	10231	10259	10288	10317	10346	10375	10404	10432	10461	10489	10517	10544	10572
10599	10626	10653	10679	10706	10733	10761	10790	10820	10850	10880	10914	10944	10974	11003
11023	11043	11065	11087	11109	11131	11153	11175	11196	11218	11239	11261	11279	11284	11301
11306	11323	11333	11350	11355	11372	11382	11385	11404	11409	11412	11430	11434	11444	11461
11466	11483	11493	11496	11513	11518	11521	11539	11544	11547	11565	11570	11573	11591	11594
11599	11602	11621	11624	11629	11632	11657	11770	11799	11835	11864	11897	11926	11951	11977
11980	11998	12017	12036	12055	12092	12129	12166	12193	12201	12213	12222	12253	12261	12273
12282	12331	12349	13181	13219	13258	13327	13387	13612	13647	13682	14245	14289	14324	14355
14391	14427	14463	14499	14530	14554	14583	14612	14641	14663	14706	14734	14759	14783	14811
14840	14872													
	1#													
	827#													
	1#													
	827#	841	843	850	863	873	926	929	1936	1938	1949	1951	1963	1965
1975	1977	1989	1991	2004	2006	2018	2020	2035	2037	2051	2053	2068	2070	2090
2092	2112	2114	2130	2132	2148	2150	2167	2169	2186	2188	2203	2205	2226	2228
2249	2251	2266	2268	2288	2290	2314	2316	2344	2346	2363	2365	2381	2383	2397
2399	2415	2417	2434	2436	2455	2457	2475	2477	2496	2498	2517	2519	2539	2541
2563	2565	2586	2588	2609	2611	2630	2632	2646	2648	2665	2667	2683	2685	2706
2708	2723	2725	2738	2740	2756	2758	2773	2775	2798	2800	2823	2825	2842	2844
2866	2868	2889	2891	2905	2907	2923	2925	2949	2951	2977	2979	2995	2997	3013
3015	3031	3033	3048	3050	3072	3074	3088	3090	3106	3108	3125	3127	3157	3159
3194	3196	3231	3233	3268	3270	3305	3307	3330	3332	3355	3357	3380	3382	3406
3408	3432	3434	3458	3460	3484	3486	3510	3512	3532	3534	3555	3557	3591	3593
3627	3629	3645	3647	3663	3665	3681	3683	3704	3706	3720	3722	3748	3750	3762
3764	3776	3778	3790	3792	3805	3807	3820	3822	3835	3837	3849	3851	3864	3866
3879	3881	3894	3896	3909	3911	3924	3926	3939	3941	3953	3955	3968	3970	3983
3985	3998	4000	4013	4015	4027	4029	4042	4044	4057	4059	4072	4074	4098	4100
4116	4118	4133	4135	4169	4171	4190	4192	4214	4216	4245	4247	4267	4269	4307
4309	4328	4330	4368	4370	4391	4393	4413	4415	4432	4434	4456	4458	4477	4479
4505	4507	4526	4528	4544	4546	4562	4572	4642	4644	4656	4658	4670	4672	4690
4692	4712	4714	4732	4734	4754	4756	4768	4770	4783	4785	4798	4800	4813	4815
4828	4830	4842	4844	4857	4859	4872	4874	4887	4889	4913	4915	4934	4936	4960
4962	4977	4979	5029	5031	5046	5048	5072	5074	5090	5092	5116	5118	5142	5144
5194	5196	5221	5223	5247	5249	5273	5275	5299	5301	5326	5328	5353	5355	5380
5382	5407	5409	5433	5435	5459	5461	5485	5487	5511	5513	5537	5539	5563	5565
5590	5592	5618	5620	5646	5648	5674	5676	5702	5704	5730	5732	5760	5762	5786
5788	5821	5823	5847	5849	5878	5880	5904	5906	5934	5936	5959	5961	5985	5987
6012	6014	6038	6040	6063	6065	6089	6091	6115	6117	6141	6143	6167	6169	6193
6195	6219	6221	6244	6246	6270	6272	6296	6298	6322	6324	6348	6350	6373	6375
6399	6401	6425	6427	6451	6453	6477	6479	6503	6505	6529	6531	6555	6557	6581
6583	6609	6611	6638	6640	6670	6672	6702	6704	6729	6731	6756	6758	6783	6785
6810	6812	6837	6839	6864	6866	6890	6892	6917	6919	6944	6946	6971	6973	6998
7000	7025	7027	7052	7054	7079	7081	7106	7108	7133	7135	7160	7162	7187	7189
7214	7216	7241	7243	7267	7269	7293	7295	7319	7321	7345	7347	7371	7373	7398
7400	7420	7422	7446	7448	7467	7469	7494	7496	7520	7522	7546	7548	7567	7569
7594	7596	7621	7623	7645	7647	7669	7671	7694	7696	7718	7720	7743	7745	7762
7764	7782	7784	7803	7805	7824	7826	7853	7855	7882	7884	7911	7913	7940	7942
7969	7971	7998	8000	8027	8029	8056	8058	8078	8080	8100	8102	8122	8124	8144
8146	8170	8172	8197	8199	8224	8226	8251	8253	8280	8282	8309	8311	8342	8344
8371	8373	8398	8400	8425	8427	8452	8454	8479	8481	8506	8508	8535	8537	8564
8566	8592	8594	8619	8621	8646	8648	8673	8675	8703	8705	8734	8736	8760	8762
8786	8788	8812	8814	8838	8840	8859	8861	8880	8882	8912	8914	8944	8946	8976
8978	9008	9010	9045	9047	9082	9084	9119	9121	9156	9158	9193	9195	9230	9232
9271	9273	9308	9310	9340	9342	9372	9374	9404	9406	9436	9438	9457	9459	9478
9480	9499	9501	9520	9522	9542	9544	9563	9565	9590	9592	9618	9620	9638	9640

9657	9659	9681	9683	9705	9707	9726	9728	9753	9755	9780	9782	9807	9809	9834
9836	9856	9858	9878	9880	9900	9902	9922	9924	9944	9946	9966	9968	9988	9990
10015	10017	10046	10048	10073	10075	10100	10102	10128	10130	10155	10157	10182	10184	10209
10211	10235	10237	10264	10266	10293	10295	10322	10324	10351	10353	10380	10382	10408	10410
10437	10439	10466	10468	10494	10496	10521	10523	10548	10550	10576	10578	10603	10605	10630
10632	10656	10658	10683	10685	10710	10712	10737	10739	10766	10768	10795	10797	10825	10827
10855	10857	10885	10887	10919	10921	10949	10951	10979	10981	11008	11010	11027	11029	11048
11050	11070	11072	11092	11094	11114	11116	11136	11138	11158	11160	11180	11182	11201	11203
11223	11225	11244	11246	11266	11268	11288	11290	11310	11312	11337	11339	11359	11361	11391
11393	11417	11419	11448	11450	11470	11472	11500	11502	11526	11528	11552	11554	11578	11580
11608	11610	11638	11640	11666	11668	11697	11699	11726	11728	11751	11753	11780	11782	11813
11815	11845	11847	11878	11880	11906	11908	11940	11942	11956	11958	11983	11985	12002	12004
12021	12023	12040	12042	12059	12061	12100	12102	12137	12139	12174	12176	12234	12236	12294
12296	12318	12320	12336	12338	12354	12356	12384	12386	12416	12418	12449	12451	12482	12484
12539	12541	12596	12598	12627	12629	12658	12660	12689	12691	12720	12722	12751	12753	12787
12789	12819	12824	12863	12869	12950	12952	12970	12972	13010	13012	13048	13050	13073	13075
13098	13100	13123	13125	13147	13149	13186	13188	13224	13226	13276	13278	13336	13338	13396
13398	13426	13428	13458	13460	13490	13492	13513	13515	13536	13538	13558	13560	13589	13593
13626	13628	13661	13663	13696	13698	13722	13724	13749	13751	13776	13778	13806	13808	13846
13851	13876	13878	13891	13893	13916	13918	13932	13934	13949	13951	13969	14000	14029	14060
14089	14120	14149	14180	14209	14226	14254	14270	14298	14300	14328	14330	14359	14361	14395
14397	14431	14433	14467	14469	14503	14505	14534	14536	14558	14560	14587	14589	14616	14618
14645	14647	14667	14673	14713	14715	14737	14739	14762	14764	14786	14788	14814	14816	14843
14845	14888	14894	14926	14937	14977	14988	15023	15032	15058	15074	15145	15152	15206	15215
15276	15282	15313	15323	15390	15399	15429	15440	15551	15565	15638	15640	15683	15724	15740
15784	15918	16000	16063	16119	16199	16277	16335							

SWRSU	1#	827#													
TESTNO	1263#	1940	1953	1967	1979	1993	2008	2022	2039	2055	2072	2094	2116	2134	2152
	2171	2190	2207	2230	2253	2270	2292	2318	2348	2367	2385	2401	2419	2438	2459
	2479	2500	2521	2543	2567	2590	2613	2634	2650	2669	2687	2710	2727	2742	2760
	2777	2802	2827	2846	2870	2893	2909	2927	2953	2981	2999	3017	3035	3052	3076
	3092	3110	3129	3161	3198	3235	3272	3309	3334	3359	3384	3410	3436	3462	3488
	3514	3536	3559	3595	3631	3649	3667	3685	3708	3724	3752	3766	3780	3794	3809
	3824	3839	3853	3868	3883	3898	3913	3928	3943	3957	3972	3987	4002	4017	4031
	4046	4061	4076	4102	4120	4137	4173	4194	4218	4249	4271	4311	4332	4372	4395
	4417	4436	4460	4481	4509	4530	4548	4574	4647	4661	4675	4695	4717	4737	4759
	4773	4788	4803	4818	4833	4847	4862	4877	4892	4918	4939	4965	4982	5034	5051
	5077	5095	5121	5147	5199	5226	5252	5278	5304	5331	5358	5385	5412	5438	5464
	5490	5516	5542	5568	5595	5623	5651	5679	5707	5735	5765	5791	5826	5852	5883
	5909	5939	5964	5990	6017	6043	6068	6094	6120	6146	6172	6198	6224	6249	6275
	6301	6327	6353	6378	6404	6430	6456	6482	6508	6534	6560	6586	6614	6643	6675
	6707	6734	6761	6788	6815	6842	6869	6895	6922	6949	6976	7003	7030	7057	7084
	7111	7138	7165	7192	7219	7246	7272	7298	7324	7350	7376	7403	7425	7451	7472
	7499	7525	7551	7572	7599	7626	7650	7674	7699	7723	7748	7767	7787	7808	7829
	7858	7887	7916	7945	7974	8003	8032	8061	8083	8105	8127	8149	8175	8202	8229
	8256	8285	8314	8347	8376	8403	8430	8457	8484	8511	8540	8569	8597	8624	8651
	8678	8708	8739	8765	8791	8817	8843	8864	8885	8917	8949	8981	9013	9050	9087
	9124	9161	9198	9235	9276	9313	9345	9377	9409	9441	9462	9483	9504	9525	9547
	9568	9595	9623	9643	9662	9686	9710	9731	9758	9785	9812	9839	9861	9883	9905
	9927	9949	9971	9993	10020	10051	10078	10105	10133	10160	10187	10214	10240	10269	10298
	10327	10356	10385	10413	10442	10471	10499	10526	10553	10581	10608	10635	10661	10688	10715
	10742	10771	10800	10830	10860	10890	10924	10954	10984	11013	11032	11053	11075	11097	11119
	11141	11163	11185	11206	11228	11249	11271	11293	11315	11342	11364	11396	11422	11453	11475
	11505	11531	11557	11583	11613	11643	11671	11702	11731	11756	11785	11818	11850	11883	11911
	11945	11961	11988	12007	12026	12045	12064	12105	12142	12179	12239	12299	12323	12341	12359
	12389	12421	12454	12487	12544	12601	12632	12663	12694	12725	12756	12792	12827	12872	12955

	12975	13015	13053	13078	13103	13128	13152	13191	13229	13281	13341	13401	13431	13463	13495
	13518	13541	13563	13596	13631	13666	13701	13727	13754	13781	13811	13853	13881	13896	13921
	13937	13954	14003	14063	14123	14183	14229	14273	14303	14333	14364	14400	14436	14472	14508
	14539	14563	14592	14621	14650	14676	14718	14742	14767	14791	14819	14848	14896	14939	14990
	15034	15076	15154	15217	15284	15325	15401	15442	15567	15642					
TRMTRP	16356#														
TYPBIN	1#	827#													
TYPDEC	1#	827#													
TYPNAM	1#	827#													
TYPNUM	1#	827#													
TYPOCS	1#	827#													
TYPOCT	1#	827#	16075	16099											
TYPTXT	1#	827#													
UPCODE	15722#	15747													
YESCOP	1263#	4645	4659	4673	4693	4715	4735	4757	4771	4786	4801	4816	4831	4845	4860
	4875	4890	4916	4937	4963	4980	5032	5049	5075	5093	5119	5145	5197	5224	5250
	5276	5302	5329	5356	5383	5410	5436	5462	5488	5514	5540	5566	5593	5621	5649
	5677	5705	5733	5763	5789	5824	5850	5881	5907	5937	5962	5988	6015	6041	6066
	6092	6118	6144	6170	6196	6222	6247	6273	6299	6325	6351	6376	6402	6428	6454
	6480	6506	6532	6558	6584	6612	6641	6673	6705	6732	6759	6786	6813	6840	6867
	6893	6920	6947	6974	7001	7028	7055	7082	7109	7136	7163	7190	7217	7244	7270
	7296	7322	7348	7374	7401	7423	7449	7470	7497	7523	7549	7570	7597	7624	7648
	7672	7697	7721	7746	7765	7785	7806	7827	7856	7885	7914	7943	7972	8001	8030
	8059	8081	8103	8125	8147	8173	8200	8227	8254	8283	8312	8345	8374	8401	8428
	8455	8482	8509	8538	8567	8595	8622	8649	8676	8706	8737	8763	8789	8815	8841
	8862	8883	8915	8947	8979	9011	9048	9085	9122	9159	9196	9233	9274	9311	9343
	9375	9407	9439	9460	9481	9502	9523	9545	9566	9593	9621	9641	9660	9684	9708
	9729	9756	9783	9810	9837	9859	9881	9903	9925	9947	9969	9991	10018	10049	10076
	10103	10131	10158	10185	10212	10238	10267	10296	10325	10354	10383	10411	10440	10469	10497
	10524	10551	10579	10606	10633	10659	10686	10713	10740	10769	10798	10828	10858	10888	10922
	10952	10982	11011	11030	11051	11073	11095	11117	11139	11161	11183	11204	11226	11247	11269
	11291	11313	11340	11362	11394	11420	11451	11473	11503	11529	11555	11581	11611	11641	11669
	11700	11729	11754	11783	11816	11848	11881	11909	11943	11986	12005	12024	12062	12103	12140
	12177	12237	12297	12321	12339	12357	12387	12419	12452	12485	12542	12599	12630	12661	12692
	12723	12754	12790	12825	12870	12953	12973	13013	13051	13076	13101	13126	13150	13189	13227
	13279	13339	13399	13429	13461	13493	13516	13539	13561	13594	13629	13664	13879	13894	13919
	13935	13952	14674	14716	14740	14765									
\$\$CMRE	871#	910	911	912	913	914	915								
\$\$CMTM	871#	916	917	918	919	920									
\$\$ESCA	1#	827#													
\$\$NEWT	1#	827#	1936	1949	1963	1975	1989	2004	2018	2035	2051	2068	2090	2112	2130
	2148	2167	2186	2203	2226	2249	2266	2288	2314	2344	2363	2381	2397	2415	2434
	2455	2475	2496	2517	2539	2563	2586	2609	2630	2646	2665	2683	2706	2723	2738
	2756	2773	2798	2823	2842	2866	2889	2905	2923	2949	2977	2995	3013	3031	3048
	3072	3088	3106	3125	3157	3194	3231	3268	3305	3330	3355	3380	3406	3432	3458
	3484	3510	3532	3555	3591	3627	3645	3663	3681	3704	3720	3748	3762	3776	3790
	3805	3820	3835	3849	3864	3879	3894	3909	3924	3939	3953	3968	3983	3998	4013
	4027	4042	4057	4072	4098	4116	4133	4169	4190	4214	4245	4267	4307	4328	4368
	4391	4413	4432	4456	4477	4505	4526	4544	4562	4642	4656	4670	4690	4712	4732
	4754	4768	4783	4798	4813	4828	4842	4857	4872	4887	4913	4934	4960	4977	5029
	5046	5072	5090	5116	5142	5194	5221	5247	5273	5299	5326	5353	5380	5407	5433
	5459	5485	5511	5537	5563	5590	5618	5646	5674	5702	5730	5760	5786	5821	5847
	5878	5904	5934	5959	5985	6012	6038	6063	6089	6115	6141	6167	6193	6219	6244
	6270	6296	6322	6348	6373	6399	6425	6451	6477	6503	6529	6555	6581	6609	6638
	6670	6702	6729	6756	6783	6810	6837	6864	6890	6917	6944	6971	6998	7025	7052
	7079	7106	7133	7160	7187	7214	7241	7267	7293	7319	7345	7371	7398	7420	7446

7467	7494	7520	7546	7567	7594	7621	7645	7669	7694	7718	7743	7762	7782	7803
7824	7853	7882	7911	7940	7969	7998	8027	8056	8078	8100	8122	8144	8170	8197
8224	8251	8280	8309	8342	8371	8398	8425	8452	8479	8506	8535	8564	8592	8619
8646	8673	8703	8734	8760	8786	8812	8838	8859	8880	8912	8944	8976	9008	9045
9082	9119	9156	9193	9230	9271	9308	9340	9372	9404	9436	9457	9478	9499	9520
9542	9563	9590	9618	9638	9657	9681	9705	9726	9753	9780	9807	9834	9856	9878
9900	9922	9944	9966	9988	10015	10046	10073	10100	10128	10155	10182	10209	10235	10264
10293	10322	10351	10380	10408	10437	10466	10494	10521	10548	10576	10603	10630	10656	10683
10710	10737	10766	10795	10825	10855	10885	10919	10949	10979	11008	11027	11048	11070	11092
11114	11136	11158	11180	11201	11223	11244	11266	11288	11310	11337	11359	11391	11417	11448
11470	11500	11526	11552	11578	11608	11638	11666	11697	11726	11751	11780	11813	11845	11878
11906	11940	11956	11983	12002	12021	12040	12059	12100	12137	12174	12234	12294	12318	12336
12354	12384	12416	12449	12482	12539	12596	12627	12658	12689	12720	12751	12787	12819	12863
12950	12970	13010	13048	13073	13098	13123	13147	13186	13224	13276	13336	13396	13426	13458
13490	13513	13536	13558	13589	13626	13661	13696	13722	13749	13776	13806	13846	13876	13891
13916	13932	13949	13969	14029	14089	14149	14209	14254	14298	14328	14359	14395	14431	14467
14503	14534	14558	14587	14616	14645	14667	14713	14737	14762	14786	14814	14843	14888	14926
14977	15023	15058	15145	15206	15276	15313	15390	15429	15551	15638				
16356#	16365	16366	16367											
1#	827#	1944	1958	1971	1984	1999	2013	2031	2047	2063	2085	2107	2125	2143
2162	2181	2199	2221	2244	2261	2283	2309	2338	2358	2376	2392	2410	2429	2450
2471	2491	2512	2534	2557	2580	2603	2641	2660	2678	2701	2718	2751	2768	2793
2818	2837	2862	2884	2901	2918	2944	2972	2990	3008	3026	3043	3083	3101	3120
3152	3188	3225	3262	3299	3326	3351	3376	3401	3427	3453	3479	3505	3528	3550
3641	3659	3677	3699	3715	3743	3757	3771	3785	3800	3815	3830	3844	3859	3874
3889	3904	3919	3934	3948	3963	3978	3993	4008	4022	4037	4052	4067	4093	4111
4128	4164	4185	4209	4240	4261	4301	4362	4408	4428	4472	4518	4538	4556	4652
4666	4686	4708	4728	4750	4764	4779	4794	4809	4824	4838	4853	4868	4883	4909
4930	4956	4973	5025	5042	5068	5086	5112	5138	5190	5217	5243	5269	5295	5322
5349	5376	5403	5429	5455	5481	5507	5533	5559	5586	5613	5641	5669	5697	5725
5755	5781	5816	5842	5873	5899	5929	5954	5981	6008	6034	6059	6085	6111	6137
6163	6189	6215	6240	6266	6292	6318	6344	6369	6395	6421	6447	6473	6499	6525
6551	6577	6604	6633	6665	6697	6724	6751	6778	6805	6832	6859	6885	6912	6939
6966	6993	7020	7047	7074	7101	7128	7155	7182	7209	7236	7263	7289	7315	7341
7367	7393	7415	7441	7462	7489	7515	7541	7562	7590	7617	7641	7665	7690	7714
7739	7758	7778	7798	7819	7849	7878	7907	7936	7965	7994	8023	8052	8073	8095
8117	8139	8166	8193	8220	8247	8275	8304	8337	8366	8394	8421	8448	8475	8502
8530	8559	8587	8615	8642	8669	8698	8729	8756	8782	8808	8834	8854	8875	8906
8938	8970	9002	9039	9076	9113	9150	9187	9224	9265	9302	9334	9366	9398	9430
9452	9473	9494	9515	9536	9558	9586	9614	9634	9653	9677	9701	9721	9749	9775
9802	9829	9851	9873	9895	9917	9939	9961	9983	10011	10042	10069	10096	10123	10151
10178	10205	10231	10259	10288	10317	10346	10375	10404	10432	10461	10489	10517	10544	10572
10599	10626	10653	10679	10706	10733	10761	10790	10820	10850	10880	10914	10944	10974	11003
11023	11043	11065	11087	11109	11131	11153	11175	11196	11218	11239	11261	11279	11284	11301
11306	11323	11333	11350	11355	11372	11382	11385	11404	11409	11412	11430	11434	11444	11461
11466	11483	11493	11496	11513	11518	11521	11539	11544	11547	11565	11570	11573	11591	11594
11599	11602	11621	11624	11629	11632	11657	11770	11799	11835	11864	11897	11926	11951	11977
11980	11998	12017	12036	12055	12092	12129	12166	12193	12201	12213	12222	12253	12261	12273
12282	12331	12349	13181	13219	13258	13327	13387	13612	13647	13682	14245	14289	14324	14355
14391	14427	14463	14499	14530	14554	14583	14612	14641	14663	14706	14734	14759	14783	14811
14840	14872													

.EQUAT 1# 695# 717
.HEADE 1# 695# 696
.KT11 1#
.SETUP 1#
.SWRMI 1# 695# 705

.SWPLO	716#		
.SACT1	1#	695#	861
.SAPT8	1#	695#	927#
.SAPTH	1#	695#	839
.SAPTY	1#	695#	16275
.SASTA	1#		
.SCATC	1#	695#	827
.SCMTA	1#	695#	871
.SDB2D	1#		
.SDB2O	1#		
.SDIV	1#		
.SEOP	1#	695#	15681
.SERRO	1#	695#	15998
.SERRT	1#	695#	16061
.SMULT	1#		
.SPOWE	1#	695#	15722
.SRAND	1#		
.SRDDE	1#		
.SRDOC	1#		
.SREAD	1#		
.SR2AZ	1#		
.SSAVE	1#		
.SSB2D	1#		
.SSB2O	1#		
.SSCOP	1#	695#	15916
.SSIZE	1#		
.SSUPR	1#		
.STRAP	1#	695#	16333
.STYPB	1#		
.STYPD	1#	695#	
.STYPE	1#	695#	16117
.STYPO	1#	695#	16197
.S4OCA	1#		
.1170	1#		

. ABS. 070034 000

ERRORS DETECTED: 0

DSKZ:CQKDA.D.BIN,CQKDA.D.LST/CRF/SOL/NL:TOC=CQKDA.D.SML,CQKDA.D.P11
RUN-TIME: 179 227 18 SECONDS
RUN-TIME RATIO: 762/425=1.7
CORE USED: 39K (77 PAGES)