

KD11-A

KD11-A CPU DIAG
CBQEACO

AH 7690C-MC
FICHE 1 OF 4

SEP 1980
COPYRIGHT © 1980
MADE IN USA



The main body of the document is a dense grid of technical diagrams and data tables. Each cell in the grid contains a small, detailed schematic or data set, likely representing a specific component or state of the CPU. The diagrams are arranged in a regular, repeating pattern across the page. The text within these diagrams is small and difficult to read, but they appear to be organized into columns and rows, possibly representing different stages of a diagnostic process or different parts of the hardware. The overall layout is highly structured and technical in nature.

KD11-A

KD11-A CPU DIAG
CBQEACO

AH 7690C MC
FICHE 2 OF 4

SEP 980
COPYR - T
MADE IN USA



A grid of microfiche frames containing technical data, likely CPU diagnostic information, arranged in approximately 15 columns and 15 rows. Each frame contains small, dense text and diagrams, which are too small to read clearly in this image. The frames are separated by thin white lines.

KD11-A

KD11-A CPU DIAG
CBQEACO

AH-7690C-MC
FICHE 3 OF 4

SEP 1980
COPYRIGHT © 76-80
MADE IN USA



A large grid of data, likely a CPU diagnostic table, consisting of many small columns and rows of text. The text is extremely faint and illegible due to the low resolution and high contrast of the scan. The grid appears to be organized into several vertical sections or columns of data.

KD11-A

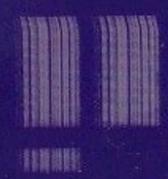
KD11-A CPU DIAG
CBQEACO

AH-7690C-MC
FICHE 4 OF 4

SEP 1980
COPYRIGHT © 76 80
MADE IN USA



The main body of the document is a large, dense grid of data, likely a CPU diagnostic report. It consists of approximately 15 columns and 25 rows of text. The text is extremely faint and difficult to read, but it appears to be organized into a structured format, possibly a table or a series of related data blocks. The content is mirrored across the page, suggesting it might be a scan of a document with a repeating pattern or a very low-contrast print.



.REM a

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

PRODUCT CODE: AC-7688C-MC
PRODUCT NAME: CBQEAC0 KD11-A CPU DIAG
PRODUCT DATE: 15 FEB 1980
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 MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILTY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1976,1980 BY DIGITAL EQUIPMENT CORPORATION

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
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105

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
- 4.0 PERFORMANCE AND PROGRESS REPORTS
 - 4.1 PERFORMANCE REPORTS
 - 4.2 PROGRESS REPORTS
- 5.0 DEVICE INFORMATION TABLES
 - 5.1 KD11-A MICROPROGRAMMING INFORMATION
 - 5.2 MICROWORD CONTROL SIGNAL TEST POINTS
 - 5.3 KD11-A LOGIC PRINT SUMMARY
- 6.0 MAINTENANCE PROCEDURES
 - 6.1 INTRODUCTION
 - 6.2 MICROPROGRAMMING / LOGIC INFORMATION
 - 6.3 KM11 MAINTENANCE MODULE
 - 6.4 UPP MATCH MAINTENANCE FEATURE
- 7.0 FLOW CHARTS
 - 7.1 FUNCTIONAL FLOW
 - 7.2 SUB-FUNCTIONAL FLOW
 - 7.3 FUNCTIONAL TEST FLOWS
 - 7.4 CORE MEMORY MAP
- 8.0 SUB-TITLE INDEX OF TESTS
- 9.0 HISTORY FILE FOR KD11-A
 - 9.1 PURPOSE
 - 9.2 ENTRIES
- 10.0 PROGRAM LISTING

107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161

1.0 GENERAL PROGRAM INFORMATION

1.1 PROGRAM PURPOSE

"CBQEAC" IS A DIAGNOSTIC PROGRAM DESIGNED TO DETECT, REPORT, AND IDENTIFY LOGIC FAULTS IN THE KD11-A CENTRAL PROCESSING UNIT OF THE PDP11/40 SYSTEM. IT CONSISTS OF 525(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 THREE MAJOR SECTIONS AS DESCRIBED BELOW:

A. 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 AND DATA DISPLAYS INDICATING 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

B. 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 MICROINSTRUCTION SEQUENCES AND DATA PATH DATA MANIPULATIONS IMPLICIT IN THE DESIGN SPECIFICATION OF THE KD11-A MICROPROGRAM. ALL LOGIC SEQUENCES THAT CAN BE ACTIVATED IN THE PROGRAM 'RUN' MODE ARE TESTED WITH THE EXCEPTION OF THE SMALL AMOUNT OF LOGIC THAT REQUIRES AN ACTIVE 'DMA' DEVICE. THIS EXCLUDES THOSE SEQUENCES AND LOGIC FUNCTIONS THAT SUPPORT THE CONSOLE FUNCTIONS (LOAD ADDRESS, DEPOSIT, ETC.). EACH TEST IN THIS SECTION CALLS A 'SCOPE LOOP' UTILITY THAT FACILITATES USER CONTROL OF TEST SELECTION AND EXECUTION VIA THE CONSOLE SWITCH REGISTER.

162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216

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.

C. 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. LIKE THE PREVIOUS SECTION, IT CALLS THE 'ERROR SERVICE' AND 'SCOPE LOOP' UTILITIES TO REPORT ERRORS AND ALLOW USER CONTROL OF TEST EXECUTION.

WHERE AT ALL POSSIBLE THE PROGRAM ANNOTATION ATTEMPTS TO CALL OUT THE MOST PROBABLE FAILURE TO THE FUNCTIONAL LOGIC AREA. EACH TEST OR GROUP OF GENERIC TESTS INCLUDES MICROPROGRAMMING AND LOGIC INFORMATION TO FACILITATE FURTHER ISOLATION OF THE FAULTY COMPONENT THRU THE USE OF ADDITIONAL LOWER LEVEL CHECKS BY THE MAINTENANCE TECHNICIAN USING THE KM-11 MAINTENANCE MODULE OR OSCILLOSCOPE. DETAILED PROCEDURES FOR USING THIS MAINTENANCE INFORMATION IS INCLUDED IN PARA. 6.0.

1.2 SYSTEM REQUIREMENTS

A. HARDWARE REQUIREMENTS

1. PDP11/40 CPU WITH OPERATOR'S CONSOLE
2. 16K OF CORE STORAGE - MF11/U OR EQUIVALENT
3. DL11 ASYNCHRONOUS LINE INTERFACE WITH TERMINAL
4. KW11-L LINE CLOCK (OPTIONAL)

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

- A. PDP11/40 PROCESSOR HANDBOOK
- B. PDP11 PERIPHERALS HANDBOOK
- C. KD11-A PROCESSOR MAINTENANCE MANUAL
- D. PDP11/40 SYSTEM ENGINEERING DRAWINGS
- E. DIAGNOSTIC ENGINEERING STANDARDS AND CONVENTIONS PROGRAMMING PRACTICES - DOC NO. 175-003-009-00

217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268

1.4 DIAGNOSTIC HIERARCHY PREREQUISITES

'CBQEAC' ASSUMES THAT THE HARDWARE VERIFIED BY THE 'BCPT' (RESIDENT BASIC CPU TEST FOR 1080 SYSTEMS) IS OPERATIONAL. A COPY OF THE 'BCPT' TEST IS INCLUDED AS PART OF THE LOAD MODULE AND IS RESIDENT IN CORE STARTING AT LOCATION 1000(8) WHENEVER 'CBQEAC' IS LOADED. AUTOMATIC LINKAGE TO THE START OF 'CBQEAC' OCCURS IF 'BCPT' RUNS ERROR FREE. THIS PROVIDES THE USER WITH THE OPTION OF RUNNING 'BCPT' PRIOR TO EXECUTING 'CBQEAC' IF HE SUSPECTS A FAULT IN THE 'HARD CORE'.

1.5 FAILURE ASSUMPTIONS

'CBQEAC' ASSUMES THAT THE STORAGE MEDIUM USED TO STORE THE PROGRAM IS INTACT AND THAT IT CAN BE LOADED INTO CORE. IT ALSO ASSUMES THAT THE BASIC TESTS DESCRIBED IN PARA. 1.4 RUN ERROR FREE AND ANY ASSUMPTIONS MADE BY THESE TESTS IS VALID. (REFER TO MAINDEC-10-DFQMAA FOR A DESCRIPTION OF THE 1080 RESIDENT TESTS)

2.0 OPERATING INSTRUCTIONS

2.1 LOADING AND STARTING PROCEDURES

A. LOADING PROCEDURES

1) PAPER TAPE SYSTEMS

USE THE STANDARD PDP11 ABSOLUTE LOADER PROCEDURES.

2) XXDP SYSTEMS

USE THE STANDARD XXDP MONITOR LOADING PROCEDURES.

B. STARTING PROCEDURES

1. TO RUN 'CBQEAC' ONLY

- A) SET SR = 000200
- B) DEPRESS LOAD ADDRESS
- C) SET SR = 000000 (NO SWITCH OPTIONS)
- D) SET HALT/ENABLE IN THE ENABLE POSITION
- E) DEPRESS START
- F) REFER TO PARA. 3.0 AND 4.0 FOR NORMAL PROGRAM RESPONSES AND ERROR REPORTS.

269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324

2. TO RUN 'BCPT' PRIOR TO 'CBQEAC'

- A) SET THE SR = 1000
- B) DEPRESS LOAD ADDRESS
- C) SET SR = 000000 (NO SWITCH OPTIONS)
- D) SET HALT/ENABLE IN THE ENABLE POSITION
- E) DEPRESS START
- F) REFER TO PARA. 3.0 AND 4.0 FOR NORMAL PROGRAM RESPONSES AND ERROR REPORTS OR TO THE 'BCPT' DOCUMENT IF AN ERROR HALT OCCURS IN THE BCPT SECTION.

C. RESTART PROCEDURES

1. TO INITIALIZE 'PASCNT', 'ERRCNT', AND 'PFCNT' TO ZERO
SAME AS IN 2.1(B1) ABOVE

2. TO PRESERVE 'PASCNT', 'ERRCNT', AND 'PFCNT'

- A) SET SR=003034
- B) DEPRESS LOAD ADDRESS
- C) SET SR=000000 (NO OPTIONS)
- D) SET HALT/ENABLE TO ENABLE POSITION
- E) DEPRESS START SWITCH

2.2 SPECIAL ENVIRONMENTS

A. 16K PDP11/40 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 'CBQEAC' WILL NOT PERMIT SIMULTANEOUS RESIDENCY OF THE UPDATE PROGRAMS. SUFFICIENT FREE CORE IS AVAILABLE FOR THE 'XXDP' MONITOR SO THAT 'CBQEAC' 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.

325

SW14 CONTINUOUSLY LOOP ON THE CURRENT TEST

326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381

- SW13 INHIBIT NORMAL ERROR PRINTOUTS - THIS DOES NOT INCLUDE POWER FAIL, BUS ERROR, RSVD INSTR TRAPS, OR MISSED TEST PRINTOUTS.
- SW12 INHIBIT ALL PRINTOUTS NOT COVERED UNDER SW13. THIS INCLUDES I.D., OPTIONS FOUND, ENDPAS, ETC.
- 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<09:00>. ONLY USE THIS OPTION FOR TESTS T0145 THRU T1015 SINCE THE 'SCOPE' UTILITY IS NOT ACTIVE UNTIL TEST T0145.
- SW09 IF SW10=0 , SW09=1 WILL ACTIVATE THE 'LOCK ON HARD ERROR' FEATURE. IF SW10=1, SW09 BECOMES THE HIGH ORDER BIT IN THE TEST NUMBER TO BE SELECTED.
- SW<9:0> USED TO SELECT A PARTICULAR TEST FOR LOOPING IF SW10=1 THERE IS NO TEST 000 DEFINED.

B. MEMORY LOCATIONS

1. FILLS: THERE IS A LOCATION TAGGED 'FILLS' THAT IS USED TO SPECIFY THE FILL COUNT AND FILL CHARACTER FOR THOSE TERMINAL DEVICES REQUIRING THE USE OF FILLERS. THE HIGH BYTE CONTAINS THE FILL COUNT AND THE LOW BYTE CONTAINS THE FILLER CHARACTER. IT IS PROGRAM LOADED AS A 002400(8) TO SPECIFY FIVE NULL CHARACTERS.
2. ITCOUNT: THERE IS A LOCATION TAGGED 'ITCOUNT' THAT CONTAINS THE ITERATION COUNT TO BE USED ON PASSES SUBSEQUENT TO PASS 0 TO SPECIFY THE NO. OF SUB-TEST ITERATIONS IT IS PROGRAM LOADED TO SPECIFY 32.(10) ITERATIONS.
3. OPTION: THERE IS A LOCATION TAGGED 'OPTION' THAT IS SET UP AUTOMATICALLY BY THE PROGRAM TO INDICATE THE PDP11/40 INTERNAL OPTIONS FOUND:

BIT15=1 KW11-L INSTALLED
BIT07=1 KT11-D INSTALLED
BIT02=1 KJ11-A INSTALLED
BIT01=1 KE11-F INSTALLED
BIT00=1 KE11-E INSTALLED

THE PROGRAM USES THE BITS IN 'OPTION' TO SKIP THOSE TESTS THAT ARE OPTION DEPENDENT TO PREVENT REDUNDANT ERROR REPORTS.

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

382
383
384

THE FAULT CAUSES A RUNAWAY OR HUNG PROGRAM. REFER TO
PARA. 4.2 FOR A DETAILED DESCRIPTION OF THE USE OF
THIS FEATURE.

385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440

2.4 EXECUTION TIMES

ONE COMPLETE ERROR FREE PASS OF DBQEAA WITH NO TEST ITERATIONS SHOULD TAKE LESS THAN 5 SECONDS. A SUCCESSFUL PASS WILL BE INDICATED BY THE FOLLOWING PRINTOUT ON THE CONSOLE DEVICE:

PASCNT = 000001 ERRCNT = 000000

WITH ITERATIONS ENABLED A COMPLETE ERROR FREE PASS SHOULD TAKE LESS THAN 2 MINUTES.

3.0 ERROR INFORMATION

3.1 ERROR REPORTING PROCEDURES

A. ERROR MESSAGE FORMATS

1. STANDARD ERROR MESSAGE HEADER

THE FOLLOWING HEADER IS PRINTED ON DETECTION OF THE FIRST ERROR DETECTED AFTER THE BASIC INSTRUCTION TEST SECTION. IT IS ONLY PRINTED ONCE PER PROGRAM PASS.

(PC)	(PS)	(SP)	(TEST)	(IR)	DEST	WAS	S/B
(R7)	(PSW)	(R6)	(R0)	(R1)	(R2)	(R3)	(R4)

WHERE:

- (PC) INDICATES THE CONTENTS OF THE PROGRAM COUNTER AT THE TIME OF THE ERROR CALL. THIS IS NORMALLY AN ADDRESS THAT IS USED TO LOCATE THE ERROR CALL STATEMENT IN THE FAILING TEST.
- (PSW) INDICATES THE CONTENTS OF THE PROCESSOR STATUS WORD AT THE TIME OF THE ERROR CALL
- (SP) INDICATES THE CONTENTS OF THE STACK POINTER (R6) AT THE TIME OF THE ERROR. (NOTE THAT THE ERROR CALL WILL PUSH THE STACK TWICE)
- (R0) INDICATES THE TEST NO. (IN OCTAL) THAT FAILED
- (R1) CONTAINS A COPY OF THE TEST INSTRUCTION THIS WILL BE THE FIRST WORD IN THE CASE OF TWO OR THREE WORD INSTRUCTIONS.
- (R2) FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS R2 NORMALLY CONTAINS THE DESTINATION ADDRESS
- (R3) FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS R3 CONTAINS WHAT THE RESULT (DEST. OPERAND) ACTUALLY

441

WAS AFTER THE TEST.

442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495

(R4) FOR SINGLE AND DOUBLE OPERAND INSTRUCTIONS R4
CONTAINS WHAT THE RESULT (DEST. OPERAND)
SHOULD HAVE BEEN (S/B).

IN SOME CASES THE ERROR INFORMATION MAY DEVIATE FROM THAT
DESCRIBED ABOVE BUT THE PROGRAM ANNOTATION FOR THOSE TESTS
WILL DESCRIBE THE MEANING OF THOSE ENTRIES THAT HAVE BEEN
RE-DEFINED.

THE ERROR CALL STATEMENT MAY BE ENCODED TO PRINT ONLY THE
INFORMATION RELATIVE TO THE PARTICULAR FUNCTION BEING TESTED.
INTERPRETATION OF THE ERROR CALLS IS AS FOLLOWS:

ERROR PRINTS ALL 8 COLUMNS
ERROR1 PRINT ONLY COLUMN 1
ERROR2 PRINT COLUMNS 1,2
ERROR3 PRINT COLUMNS 1,2,3
ERROR4 PRINT COLUMNS 1,2,3,4
ERROR5 PRINT COLUMNS 1,2,3,4,5
ERROR6 PRINT COLUMNS 1,2,3,4,5,6
ERROR7 PRINT COLUMNS 1,2,3,4,5,6,7

2. STANDARD ERROR PRINTOUT

A LINE OF FROM ONE TO EIGHT SIX DIGIT OCTAL NUMBERS THAT
LINE UP UNDER THE APPROPRIATE HEADER ENTRY AND HAVE THE
MEANINGS DESCRIBED IN THE PREVIOUS SECTION.

3. 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 CONTENTS OF THE PROGRAM COUNTER
AT THE TIME 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. 3.1,B1 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. 3.1,B2 BELOW.

496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547

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

WHERE: XXXXXX IS THE CONTENTS OF THE PC AT THE TIME 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. 3.1,B4 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. 3.1,B2 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.

EACH POWER FAIL ENCOUNTERED DURING ANY PASS IS COUNTED IN A LOCATION TAGGED 'PFCNT'. DURING END OF PASS PRINTOUT THIS LOCATION IS QUERIED AND IF FOUND NON-ZERO THE NORMAL ENDPAS MESSAGE IS MODIFIED AS SHOWN BELOW:

PASCNT=XXXXXX ERRCNT=YYYYYY PFCNT=ZZZZZZ

WHERE 'ZZZZZZ' GIVES THE TOTAL COUNT OF THE POWER FAILS (IN OCTAL) OCCURRING DURING ANY PROGRAM RUN.

IF THE POWER FAIL IS DETECTED BEFORE COMPLETION OF THE BASIC INSTRUCTION TESTS, THE PROGRAM WILL HALT VIA A TRAPCATCHER IN THE VECTOR. A DESCRIPTION OF THIS HALT IS GIVEN IN PARA. 3.1,B2 BELOW.

548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602

6. MISSED TEST MESSAGE

THERE IS A 512 BYTE TABLE TAGGED 'STAB1' THAT CONTAINS A BYTE ENTRY THAT CORRESPONDS TO EACH SEQUENTIAL TEST NO.. AFTER THE 'MOVB #N,X(R)' INSTRUCTION HAS BEEN VERIFIED IN THE 'BIT' SECTION, EACH TEST WILL USE THIS INSTRUCTION TO SET THE BYTE THAT CORRESPONDS TO THAT TEST NO. TO ALL ONES (377) THROUGHOUT THE REMAINDER OF THE 'BASIC INSTRUCTION TESTS' SECTION. DURING THE 'CIT' AND 'IEX' SECTIONS THE TABLE ENTRIES ARE UPDATED BY THE 'SCOPE' LOOP UTILITY. DURING END OF PASS SERVICE, A CHECK ROUTINE IS CALLED TO SCAN THE TABLE AND INSURE THAT EACH BYTE IS SET TO ALL ONES (377). ANY ENTRY THAT STILL CONTAINS ZEROES (TABLE IS INITIALIZED TO ZEROES BEFORE BEING USED) MEANS THAT ONE OR MORE TESTS WERE SKIPPED FOR SOME REASON. IF ANY TESTS ARE FOUND 'MISSED' THE PRINTOUT SHOWN BELOW IS DISPLAYED:

MISSED TEST
XXXXXX
YYYYYY

WHERE: XXXXXX AND YYYYYY ARE THE TEST NUMBERS MISSED.

B. ERROR HALTS

1. BASIC INSTRUCTION TESTS (BIT)

ANY ERROR DETECTED IN THE BASIC TESTS CAUSES THE PROGRAM TO HALT WITH THE FOLLOWING INFORMATION DISPLAYED IN THE CONSOLE INDICATORS:

ADDRESS DISPLAY: ADDRESS + 2 OF THE LOCATION CONTAINING THE HALT. THIS IS USED TO LOCATE THE SPECIFIC ERROR CALL IN THE FAILING TEST

DATA DISPLAY: NUMBER OF THE FAILING TEST (IN OCTAL). USING THE SBTTL INDEX THIS MAY BE USED TO LOCATE THE FAILING TEST IN THE LISTING.

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.

603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658

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. 6.0.

2. TRAPCATCHER HALTS

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 INSTRUCTION TESTS THE KW11-L 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 FOLLOWING INFORMATION DISPLAYED IN THE CONSOLE:

ADDRESS DISPLAY: V+4 OF THE VECTOR. THIS IS USED TO IDENTIFY THE CAUSE OF THE UNEXPECTED TRAP OR INTERRUPT.

DATA DISPLAY: TEST NO. OF THE LAST TEST BEING EXECUTED WHEN THE TRAP OR INTERRUPT OCCURRED.

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.

4. CATASTROPHIC ERROR HALTS

THERE ARE TWO HALTS, ONE IN THE BUS ERROR SERVICE ROUTINE 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

659
660

OR RSVD INSTRUCTION TRAP. AFTER THE HALT THE CONSOLE
DISPLAYS THE FOLLOWING INFORMATION:

661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716

ADDRESS DISPLAY: PC+2 OF THE ERROR HALT. THIS IS USED TO IDENTIFY WHICH OF THE TWO TYPES OF ERRORS - RSVD OR BUS ERROR.

DATA DISPLAY: LAST TEST NUMBER BEING EXECUTED WHEN THE TRAPS OCCURRED.

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:

[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 WHICH IS FROZEN WHEN IT REACHES 177777(8).

THIS PRINTOUT IS MODIFIED TO PRINT THE POWER FAIL COUNT AS DESCRIBED IN PARA. 3.1,A5 ABOVE IF POWER FAILED DURING A PASS.

4.2 PROGRESS REPORTS

THERE ARE THREE 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:

CBQEAC PDP11/40 CPU DIAGNOSTIC VERSION XXX

717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762

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).

C. PDP11/40 OPTIONS FOUND PRINTOUT

AFTER PRINTING THE PROGRAM I.D. MESSAGE, A SUBROUTINE IS CALLED TO 'LOOK FOR' THE 11/40 INTERNAL OPTIONS (KT11-D, KW11-L, KE11-E, KE11-F, AND KJ11-A). IF ANY OF THESE OPTIONS ARE FOUND THE FOLLOWING PRINTOUT OCCURS:

PDP11/40 INTERNAL OPTIONS FOUND

XXXX-X

YYYY-Y

WHERE 'X' AND 'Y' ETC. ARE THE NAMES OF THE OPTIONS.

IF NO OPTIONS ARE FOUND THE FOLLOWING PRINTOUT OCCURS:

PDP11/40 INTERNAL OPTIONS FOUND

NONE FOUND

THE OPTIONS FOUND MESSAGE IS ONLY PRINTED ON THE FIRST PASS THROUGH THE PROGRAM AND MAY ALSO BE INHIBITED BY SETTING SW12=1.

E. 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 AND DATA 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 32 TESTS). AT EACH CHECKPOINT THE PROGRAM EXAMINES A PARTICULAR BIT IN 'BPTLOC' AND HALTS IF THE BIT IS SET TO A '1' OTHERWISE IT CONTINUES IN NORMAL SEQUENCE. AFTER THE HALT DEPRESSING 'CONTINUE' WILL CAUSE RESUMPTION OF NORMAL PROGRAM EXECUTION. SETTING LOCATION 'BPTLOC' TO ALL 1'S (177777) WILL RESULT IN THE FOLLOWING SIXTEEN HALTS WITH THE INFORMATION SHOWN DISPLAYED IN THE CONSOLE:

	[BPTLOC]	DATA DISPLAY TEST NO.	ADDRESS DISPLAY HALT PC+2
763			
764			
765			
766			
767	BIT00=1	000040	005076
768	BIT01=1	000077	007460
769	BIT02=1	000134	012574
770	BIT03=1	000174	014700
771	BIT04=1	000234	016434
772	BIT05=1	000274	021552
773	BIT06=1	000334	024532
774	BIT07=1	000374	027532
775	BIT08=1	000444	033506
776	BIT09=1	000504	037340
777	BIT10=1	000544	042222
778	BIT11=1	000604	045446
779	BIT12=1	000644	050226
780	BIT13=1	000704	054400
781	BIT14=1	000745	061222
782	BIT15=1	001005	063566
783			
784			
785			
786			
787			
788			
789			
790			
791			
792			
793			
794			
795			
796			
797			
798			
799			
800			
801			

NOTE: IF THE USER DEPOSITED A 000400(8) IN LOCATION 'BPTLOC'
ONLY ONE HALT WOULD OCCUR AND AT THAT TIME THE DATA
DISPLAY SHOULD CONTAIN 000444 AND THE ADDRESS DISPLAY
SHOULD CONTAIN 033506.

THIS FEATURE IS USEFUL FOR TRACKING DOWN THE TEST THAT CAUSES
A 'RUNAWAY' OR 'HUNG' PROGRAM.

LOCATION 'BPTLOC' IS PROGRAM LOADED AS 000000 TO INHIBIT ANY HALTS.

5.0 DEVICE INFORMATION TABLES

5.1 KD11-A MICROPROGRAMMING INFORMATION

802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855

A. INTRODUCTION

THE KD11-A PROCESSOR EMPLOYS A MICROPROGRAMMED CONTROL SECTION THAT GENERATES THE PROPER SEQUENCES OF CONTROL SIGNALS REQUIRED TO EXECUTE THE PDP11 INSTRUCTION SET. THE HEART OF THE CONTROL IS A CONTROL STORE THAT CONSISTS OF 256 WORDS OF 56 BITS EACH, STORED IN A READ ONLY MEMORY (ROM). EACH 56 BIT WORD IS CALLED A MICROINSTRUCTION AND IS MICROPROGRAMMED TO PERFORM A UNIQUE ELEMENTARY OPERATION WITHIN THE PROCESSOR. EACH ONE OF THE 56 BITS IS ASSIGNED A PARTICULAR CONTROL FUNCTION THAT MAY BE TURNED 'ON' OR 'OFF' DEPENDENT UPON HOW EACH MICROWORD IS PROGRAMMED. TO ACTIVATE A SPECIFIC SET OF CONTROL SIGNALS SIMPLY MEANS READING OUT A SPECIFIC ROM ADDRESS. TO GENERATE A SPECIFIC SEQUENCE OF CONTROL SIGNAL SETS SIMPLY MEANS GENERATING A SPECIFIC ROM ADDRESS SEQUENCE. THE PURPOSE OF THIS SECTION IS TO DESCRIBE THE FUNCTION OF EACH BIT OR GROUP OF BITS (CALLED A FIELD) AND PROVIDE A LIST OF BACKPLANE TEST POINTS WHERE THE STATE OF ANY PARTICULAR BIT CAN BE OBSERVED.

B. MICROINSTRUCTION FORMAT

THIS SECTION LISTS THE FUNCTIONAL DESCRIPTION OF EACH BIT. FOR A MORE DETAILED DESCRIPTION OF THE SPECIFIC ENCODING REFER TO THE PDP11/40 SYSTEM ENGINEERING DRAWINGS, KD11A PROCESSOR (UWORD AND TABLES) SHEET.

<u>BIT NO.</u>	<u>NAME</u>	<u>FUNCTIONAL DESCRIPTION</u>
U<56:55>	CLKL<1:0>	THIS TWO BIT CONTROL FIELD IS USED TO SPECIFY THE CLOCK LENGTH FOR EACH MICROINSTRUCTION. THERE ARE THREE CLOCK LENGTHS POSSIBLE AS DESCRIBED BELOW: (PRT - PULSE REPETITION TIME)
		U<56:55>=00 -CLOCK LENGTH 1 - GENERATES A 'P1' PULSE WITH PRT=140NSEC OR 01
		U<56:55>=10 -CLOCK LENGTH 2 - GENERATES A 'P2' PULSE WITH PRT=200NS
		U<56:55>=11 -CLOCK LENGTH 3 - GENERATES A 'P2' PULSE FOLLOWED 100 NSEC LATER BY A 'P3' PULSE. TOTAL PRT=300 NSEC.

856				
857		U<54>	CLKOFF	THIS BIT ALLOWS TURNING OFF THE PRO-
858				CESSOR CLOCK TO STALL PROCESSOR OPERA-
859				TIONS WHILE WAITING FOR SOME INTERNAL
860				OR EXTERNAL RESPONSE.
861				
862		U<53>	CLKIR	THIS BIT ENABLES CLOCKING NEW DATA
863				INTO THE INSTRUCTION REGISTER. IT IS
864				ONLY ACTIVE WHEN THE PROCESSOR IS
865				FETCHING A NEW INSTRUCTION.
866				
867		U<52:51>	WRH,WRL	THESE TWO BITS CONTROL THE WRITING
868				INTO THE PROCESSOR'S GENERAL REGIS-
869				TERS. THEY PROVIDE THE MECHANISM FOR
870				CONTROLLING WHETHER WE READ, WRITE A
871				16 BIT WORD, OR WRITE AN 8 BIT BYTE AS
872				DESCRIBED BELOW:
873				
874				U<52:51>=00 READ ONLY - INHIBIT
875				WRITING
876				U<52:51>=01 WRITE ONLY THE LOW
877				BYTE BITS <7:0>
878				U<52:51>=11 WRITE A FULL 16 BIT
879				WORD BITS <15:00>
880				
881		U<50>	CLKB	THIS BIT ALLOWS CLOCKING NEW DATA
882				INTO THE 'B' REGISTER
883				
884				U<50>=0 HOLD B REGISTER
885				U<50>=1 LOAD B REGISTER
886				
887		U<49>	CLKD	THIS BIT ALLOWS CLOCKING THE OUTPUT
888				OF THE 'ALU' INTO THE 'D' REGISTER.
889				
890				U<49>=0 HOLD D REGISTER
891				U<49>=1 LOAD 'D' REGISTER
892				
893		U<48>	CLKBA	THIS BIT ALLOWS CLOCKING THE BUS
894				ADDRESS REGISTER 'BA'
895				
896				U<48>=0 HOLD BA REGISTER
897				U<48>=1 LOAD BA REGISTER
898				
899		U<47:46>	C1BUS,COBUS	THIS TWO BIT FIELD SPECIFIES THE
900				TYPE OF DATA TRANSFER BUS CYCLE WHEN
901				U<45> BGBUS IS ACTIVE
902				
903				U<46:47>=00 DATI
904				U<46:47>=01 DATIP
905				U<46:47>=10 DATO
906				U<46:47>=11 DATOB
907				
908				WHEN U<45> IS INACTIVE U<47:46> ARE
909				USED TO CONTROL INTERNAL PROCESSOR
910				LOGIC RELATED TO PRIORITY TRANSFER
911				OF UNIBUS CONTROL

912				
913				
914				
915				
916				
917				
918				
919				
920				
921				
922				
923				
924				
925				
926				
927				
928				
929				
930				
931				
932				
933				
934				
935				
936				
937				
938				
939				
940				
941				
942				
943				
944				
945				
946				
947				
948				
949				
950				
951				
952				
953				
954				
955				
956				
957				
958				
959				
960				
961				
962				
963				
964				
965				
966				
967				

U<45> BGBUS

THIS BIT IS USED TO TRIGGER INITIATION OF A UNIBUS DATA TRANSFER BUS CYCLE WHICH RESULTS IN ACTIVATING BUS MSYN:

U<44:41> DAD<3:0>

THIS FOUR BIT FIELD (DISCRETE ALTERATION OF DATA) PROVIDES THE MECHANISM FOR ALTERING THE OPERATION OF THE BASIC CONTROL FUNCTIONS BASED ON CONDITIONS EXISTING WITHIN THE PROCESSOR. IE: CHECK FOR STACK OVFLW IF PUSHING ON STACK, ALLOW ODD ADDRESS IF BYTE INSTRUCTION, MODIFY ALU OPERATION AS A FUNCTION OF THE IR DECODE ETC.

U<40:38> SPS<2:0>

THIS THREE BIT FIELD CONTROLS LOADING, CLOCKING, AND GATING OF THE PROCESSOR STATUS WORD.

U<37:33> ALUM,S<3:0>

THESE FIVE BITS CONTROL WHAT OPERATION IS TO BE PERFORMED BY THE PROCESSOR'S ALU. THE FINAL ALU CONTROL IS SUBJECT TO MODIFICATION BY THE 'DAD' CODE AS DESCRIBED ABOVE.

U<32:29> SBC<3:0>

THIS FOUR BIT FIELD IS ENCODED TO SELECT SPECIFIC CONSTANTS THAT MAY BE FED INTO THE ALU'S 'B' INPUT VIA THE B MUX. IE: SWR ADDRESS=177570

U<28:27> SBMH<1:0>

THIS TWO BIT FIELD IS USED TO CONTROL THE HIGH SECTION OF THE B MUX, BITS <15:08> AND DETERMINES WHAT APPEARS AT THE B INPUT TO THE ALU, BIN<15:08>

U<28:27>=00 BIN<15:08>=BREG<15:08>
U<28:27>=01 BIN<15:08>=BREG07(SEX)
U<28:27>=10 BIN<15:08>=BREG<07:00>(BYTE)
U<28:27>=11 BIN<15:08>=BCON<15:08>(CONSTANT)

U<26:25> SBML<1:0>

THIS TWO BIT FIELD IS USED TO CONTROL THE LOW SECTION OF THE B MUX, BITS <07:00>. AND DETERMINES WHAT APPEARS AT THE B INPUT TO THE ALU. BIN<07:00>

U<26:25>=00 BIN<07:00>=BREG<07:00>
U<26:25>=01 BIN<07:00>=BREG<07:00>
U<26:25>=10 BIN<07:00>=BREG<15:08>(BYTE)
U<26:25>=11 BIN<07:00>=BCON<07:00>(CONSTANT)

U<24:23> SDM<1:0>

THIS TWO BIT FIELD IS USED TO SELECT ONE OF FOUR INPUTS THAT APPEAR AT THE INPUT TO THE 'D' MUX.

U<24:23>=00 DMUX<15:00>=BUS RD<15:00>(REG)

968
969
970

U<24:23>=01 DMUX<15:00>=BUS D<15-00>(UNIBUS)
U<24:23>=10 DMUX<15:00>=D<15:00>
U<24:23>=11 DMUX<15:00>=D<C>,D<15:01>(SHF RIGHT

971			
972			
973			
974			
975			
976			
977			
978			
979			
980			
981			
982			
983			
984			
985			
986			
987			
988			
989			
990			
991			
992			
993			
994			
995			
996			
997			
998			
999			
1000			
1001			
1002			
1003			
1004			
1005			
1006			
1007			
1008			
1009			
1010			
1011			
1012			
1013			
1014			
1015			
1016			
1017			
1018			
1019			
1020			
1021			
1022			
1023			
1024			
1025			
1026			

	U<22>	SBAM	THIS BIT IS USED TO SELECT THE DATA LOADED INTO THE BA REGISTER
			U<22>=1 BA<15:00>=BUS RD<15:00>(REG) U<22>=0 BA<15:00>=ALU<15:00>
			BIT 48 (CLKBA) MUST BE ACTIVE TO ALLOW CLOCKING INTO THE BA REGISTER.
	U<21:17>	UBF<4:0>	THIS FOUR BIT FIELD (BUT BITS) IS USED TO SPECIFY ONE OF 16 POSSIBLE MICROBRANCH TESTS WHICH PROVIDES THE MECHANISM FOR MODIFYING THE ROM ADDRESS SEQUENCE GENERATED BASED ON THE PRESENCE OR ABSENCE OF SPECIFIC CONDITIONS.
	U<16:13>	SR<S,D,BA,I>	THESE FOUR BITS ARE USED TO SELECT THE SPECIFIC BITS USED TO ADDRESS A GENERAL REGISTER.
			SRS =1 USE IR<8:6> SOURCE REG BITS SRD =1 USE IR<2:0> DEST REG BITS SRBA=1 USE BA<3:0> CONSOLE (EXAM + DEP) SRI =1 USE U<12:9> MICROWORD
	U<12:09>	RIF<3:0>	THESE FOUR BITS SPECIFY WHICH GENERAL REGISTER TO ACCESS IF 'SRI' (U<13>) IS ACTIVE
			U<12:09>=0000 R0 U<12:09>=0001 R1 U<12:09>=0001 U<12:09>=0001 U<12:09>=1111 R17
	U<08>	UPF8	THIS BIT IS NOT STORED IN THE ROM BUT IS A SEPARATE CONTROL SIGNAL THAT PROVIDES THE MECHANISM FOR IMPLEMENTING AN EXPANSION ROM WHEN THE EIS/FIS OPTIONS ARE INSTALLED - NOT IN THE BASIC MACHINE.
	U<07:00>	UPF<7:0>	THIS EIGHT BIT FIELD (NEXT FIELD) PROVIDES A MEANS OF EACH MICROINSTRUCTION TO SPECIFY THE ADDRESS OF THE NEXT MICROWORD TO BE ACCESSED. THE CONTENTS OF THIS FIELD MAY BE MODIFIED AS A RESULT OF A MICROBRANCH TEST BEFORE BEING LOADED INTO THE ROM ADDRESS REGISTER. (UPP<7:0>.
			IT IS THE COMBINATION OF THIS FIELD AND THE BUT BITS U<21:17> THAT MAKE IT POSSIBLE TO GENERATE VARIABLE ROM ADDRESS SEQUENCES.

THESE 8 BITS ARE UNIQUE IN THAT THEY ARE ACTIVE (1) WHEN 'LOW' IN CONTRAST TO ALL OTHER BITS THAT ARE ACTIVE (1) WHEN HIGH.

IE: UPF<7:0>=01010010 = ADDRESS 255

5.2 MICROWORD CONTROL SIGNAL TEST POINTS

KD11-A MICROWORD CONTROL SIGNALS (SH1 OF 2)

1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066
1067
1068
1069
1070
1071
1072
1073
1074

U<N>	PRINT	SIGNAL NAME	ROM	UREG
56	K2-8	CLKL1(1) H	E38-09	D03N2
56	::	CLKL1(0) H	::	D03M2
55	::	CLKLO(1) H	E38-10	D03L2
55	::	CLKLO(1) H	::	D03R1
54	::	CLKOFF(1) H	E38-11	D03U2
53	::	CLKIR(1) H	E38-12	D03T2
52	K2-7	WRH(1) H	E34-09	D03J2
51	::	WRL(1) H	E34-10	D03H2
50	::	CLKB(1) H	E34-11	D03J1
49	::	CLKD(1) H	E34-12	D03K2
48	::	CLKBA(1) H	E35-09	D03L1
47	::	C1BUS(1) H	E35-10	D03K1
46	::	COBUS(1) H	E35-11	D03F2
45	::	BGBUS(1) H	E35-12	D03E2
44	::	DAD3(1) H	E37-09	D03M1
43	::	DAD2(1) H	E37-10	D03N1
42	::	DAD1(1) H	E37-11	D03P1
41	::	DAD0(1) H	E37-12	D03D2
40	K2-6	SPS2(1) H	E31-09	C03M2
39	::	SPS1(1) H	E31-10	D03H1
38	::	SPS0(1) H	E31-11	D03F1
37	::	SALUM(1) H	E31-12	D03E1
36	::	SALU3(1) H	E30-09	C03N2
35	::	SALU2(1) H	E30-10	C03P2
34	::	SALU1(1) H	E30-11	C03T2
33	::	SALU0(1) H	E30-12	C03U2
32	::	SBC3(1) H	E27-09	C03S2
31	::	SBC2(1) H	E27-10	C03U2
30	::	SBC1(1) H	E27-11	C03R2
29	::	SBC0(1) H	E27-12	D03D1

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

KD11-A MICROWORD CONTROL SIGNALS (SH2 OF 2)

U<N>	PRINT	SIGNAL NAME	ROM	UREG
28	K2-5	SBMH1(1) H	E22-09	C03N1
27	..	SBMH0(1) H	E22-10	C03M1
26	..	SBML1(1) H	E22-11	C03L1
25	..	SBML0(1) H	E22-12	C03J1
24	..	SDM1(1) H	E23-09	C03L2
23	..	SDM0(1) H	E23-10	C03K2
22	..	SBAM(1) H	E23-11	C03J2
21	..	UBF4(1) H	E23-12	C03H2
20	..	UBF3(1) H	E26-09	C03S1
19	..	UBF2(1) H	E26-10	C03R1
18	..	UBF1(1) H	E26-11	C03P1
17	..	UBF0(1) H	E26-12	C03F2
16	K2-4	SRS(1) H	E21-09	C03E1
15	..	SRD(1) H	E21-10	C03D2
14	..	SRBA(1) H	E21-11	C03E2
13	..	SR1(1) H	E21-12	C03B1
12	..	RIF3(1) H	E20-09	B03S1
11	..	RIF2(1) H	E20-10	B03U1
10	..	RIF1(1) H	E20-11	B03U2
09	..	RIF0(1) H	E20-12	B03T2
08	K2-3	BUPP8(1) H	N/A	B03E2
07	..	BUPP7(1) H	E14-09	B03J2
06	..	BUPP6(1) H	E14-10	B03M2
05	..	BUPP5(1) H	E14-11	B03P2
04	..	BUPP4(1) H	E14-12	B03R2
03	K2-2	BUPP3(1) H	E15-09	A03L2
02	..	BUPP2(1) H	E15-10	A03P2
01	..	BUPP1(1) H	E15-11	A03S2
00	..	BUPP0(1) H	E15-12	A03U2

NOTES:

1. THE ROM TEST POINTS CORRESPOND TO THE WORD BEING READ OUT AS DEFINED BY THE CONTENTS OF THE 'UPP'
2. THE UREG TEST POINTS CORRESPOND TO THE WORD BEING EXECUTED AS DEFINED BY THE CONTENTS OF THE 'PUPP'
3. AN 'H' (HIGH LEVEL) IS OBSERVED IF THE BIT IS PROGRAMMED AS A '1' AND A 'L' (LOW LEVEL) IS OBSERVED IF THE BIT IS PROGRAMMED AS A '0' WITH EXCEPTION OF THE 'UPF' FIELD BITS <7:0> WHEN OBSERVED AT THE OUTPUT OF THE ROM DIRECTLY. IN THIS CASE THE BITS ARE STORED IN 1'S COMPLEMENT FORM.

1120
1121
1122
1123
1124
1125
1126
1127
1128
1129
1130
1131
1132
1133
1134
1135
1136
1137
1138
1139
1140
1141

5.3 KD11-A LOGIC PRINT SUMMARY

THE LOGIC CIRCUIT DESIGN USED TO IMPLEMENT THE KD11-A CPU IS WELL ORGANIZED AND LOGICALLY PARTITIONED INTO FUNCTIONAL AREAS ON THE FIVE MODULES(4 HEX AND 1 QUAD). EACH MODULE IS IDENTIFIED FOR DOCUMENTATION PURPOSES BY A 'KN' NUMBER AS LISTED BELOW AND EACH PRINT APPROPRIATELY IDENTIFIED WITHIN THE PRINT SET:

M7231	DATA PATHS	PRINTS K1-1 THRU K1-9
M7232	UWORD (ROM)	PRINTS K2-1 THRU K2-8
M7233	IR DECODE	PRINTS K3-1 THRU K3-9
M7234	TIMING	PRINTS K4-1 THRU K4-6
M7235	STATUS	PRINTS K5-1 THRU K5-8

THE FIRST PRINT IN EACH SET (KN-1) SHOWS THE PHYSICAL LAYOUT OF THE MODULE AND INCLUDES A COMPONENT PARTS LIST. THE LOGIC PARTITIONING INTO FUNCTIONAL AREAS IS SUMMARIZED IN THE TABLE BELOW:

1142
1143
1144
1145
1146
1147
1148
1149
1150
1151
1152
1153
1154
1155
1156
1157
1158
1159
1160
1161
1162
1163
1164
1165
1166
1167
1168
1169
1170
1171
1172
1173
1174
1175
1176
1177
1178
1179
1180
1181
1182
1183
1184
1185
1186
1187

***** M7231 *****

- K1-2 DESCRIBES THE LOGIC AREAS THAT SUPPORT THE PRIMARY DATA PATH FACILITIES FOR BITS<03:00> THAT INCLUDES:
1. UNIBUS DATA RECEIVERS AND DRIVERS
 2. REGISTER DATA BUS RECEIVERS
 3. 'D' MULTIPLEXOR
 4. 'B' MULTIPLEXOR
 5. 'D' REGISTER
 6. 'B' REGISTER
 7. ARITHMETIC LOGIC UNIT
 8. 'BA' MULTIPLEXOR
- K1-3 SAME AS THE K1-2 EXCEPT IT SUPPORTS BITS<07:04>
- K1-4 SAME AS THE K1-2 EXCEPT IT SUPPORTS BITS<11:08>
- K1-5 SAME AS THE K1-2 EXCEPT IT SUPPORTS BITS<15:12> AND ALSO INCLUDES*
1. 'COUT' MULTIPLEXOR
 2. 'D<C>' CONTROL FLOP
- K1-6 DESCRIBES THE LOGIC THAT SUPPORTS:
1. BUS ADDRESS REGISTER (BA)
 2. UNIBUS ADDRESS LINE DRIVERS
- K1-7 DESCRIBES THE IMPLEMENTATION OF:
1. INTERNAL ADDRESS DECODERS
 2. 'D' REGISTER DECODER
- K1-8 DESCRIBES THE:
1. GENERAL REGESTERS (R00 THRU R17)
 2. GENERAL REGISTER ADDRESS SELECTION SWITCHES
- K1-9 DESCRIBES THE:
1. KY11-D DATA DISPLAY AND SWITCH REGISTER INTERFACES
 2. CABLE CONNECTOR

1188
1189
1190
1191
1192
1193
1194
1195
1196
1197
1198
1199
1200
1201
1202
1203
1204
1205
1206
1207
1208
1209
1210
1211
1212
1213
1214
1215
1216
1217
1218
1219
1220
1221
1222
1223
1224
1225
1226
1227
1228

***** M7232 *****

- K2-2 DESCRIBES THE:
1. UPP REGISTER BITS<03:00>
2. PUPP REGISTER BITS<03:00>
3. MICROBRANCH CONTROL 'OR' GATES BITS<03:00>
- K2-3 DESCRIBES THE:
1. UPP REGISTER BITS<08:04>
2. PUPP REGISTER BITS<08:04>
3. MICROBRANCH CONTROL 'OR' GATES BITS<07:04>
- K2-4 DESCRIBES THE:
1. ROM U<16:09>
2. UREG <16:09>
- K2-5 DESCRIBES THE:
1. ROM U <28:17>
2. UREG <28:17>
- K2-6 DESCRIBES THE:
1. ROM U <40:29>
2. UREG <40:29>
- K2-7 DESCRIBES THE:
1. ROM U <52:41>
2. UREG <52:41>
- K2-8 DESCRIBES THE:
1. ROM U <56:53>
2. UREG <56:53>
3. EXPANSION ROM CONNECTORS

1229
1230
1231
1232
1233
1234
1235
1236
1237
1238
1239
1240
1241
1242
1243
1244
1245
1246
1247
1248
1249
1250
1251
1252
1253
1254
1255
1256
1257
1258
1259
1260
1261
1262
1263
1264
1265
1266
1267

***** M7233 *****

- K3-2 DESCRIBES THE:
 - 1. BUT MULTIPLEXOR
- K3-3 DESCRIBES THE:
 - 1. INSTRUCTION REGISTER
 - 2. IR DECODERS
- K3-4 DESCRIBES THE:
 - 1. IR DECODERS
 - 2. "OVLAP" DECODER
- K3-5 DESCRIBES THE:
 - 1. MICRO BRANCH CONTROL LOGIC
- K3-6 DESCRIBES THE*
 - 1. IR DECODERS (DISCRETE)
- K3-7 DESCRIBES THE:
 - 1. MICROBRANCH CONTROL LOGIC
- K3-8 DESCRIBES THE:
 - 1. "COUT" MUX CONTROL LOGIC
 - 2. AUX ALU CONTROL MULTIPLEXORS
 - 3. "CIN" GENERATION LOGIC
- K3-9 DESCRIBES THE:
 - 1. "C" AND "V" BIT CONTROL LOGIC

1268
1269
1270
1271
1272
1273
1274
1275
1276
1277
1278
1279
1280
1281
1282
1283
1284
1285
1286
1287
1288
1289
1290
1291
1292
1293
1294
1295
1296
1297
1298

***** M7234 *****

K4-2 DESCRIBES THE:

1. CPU CLOCK WITH ASYNCHRONOUS CONTROL LOGIC
2. PRIMARY REGISTER TIMING LOGIC FOR:

- A. UPP AND PUPP REGISTERS
- E. U REGISTER
- C. INSTRUCTION REGISTER
- D. B, D, AND BA REGISTERS
- E. GENERAL REGISTERS - WRITE TIMING

K4-3 DESCRIBES THE:

1. JAMUPP CONTROL LOGIC AND TIMING

K4-4 DESCRIBES THE:

1. UNIBUS DATA TRANSFER CONTROL LOGIC

K4-5 DESCRIBES THE:

1. BUS PRIORITY TRANSFER CONTROL LOGIC

K4-6 DESCRIBES THE:

1. PRIORITY ARBITRATION LOGIC FOR 'BR'S'
2. BUS TIMEOUT AND NO SACK TIMEOUT CONTROL

1299
1300
1301
1302
1303
1304
1305
1306
1307
1308
1309
1310
1311
1312
1313
1314
1315
1316
1317
1318
1319
1320
1321
1322
1323
1324
1325
1326
1327
1328
1329
1330
1331
1332
1333
1334

***** M7235 *****

- K5-2 DESCRIBES THE:
1. PROCESSOR STATUS REGISTER
2. PSW CONTROL
3. PSW BUS DRIVERS (UNIBUS AND RD BUS)
- K5-3 DESCRIBES THE:
1. BRANCH INSTRUCTION DECODERS
2. 'BUT' DECODER (WORKING BUTS)
- K5-4 DESCRIBES THE:
1. MISCELLANEOUS HISTORY AND CONTROL FLOPS
- K5-5 DESCRIBES THE:
1. 'B' CONSTANTS GENERATOR
2. TRAP VECTOR ADDRESS GENERATOR
- K5-6 DESCRIBES THE:
1. CONSOLE CONTROL SWITCH INTERFACE
- K5-7 DESCRIBES THE:
1. CONSOLE INTERFACE CABLE CONNECTIONS
- K5-8 DESCRIBES THE:
1. POWER FAIL/AUTO RESTART CONTROL LOGIC

1391
1392
1393
1394
1395
1396
1397
1398
1399
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409
1410
1411
1412
1413
1414
1415
1416
1417
1418
1419
1420
1421
1422
1423
1424
1425
1426
1427
1428
1429
1430
1431
1432
1433
1434
1435
1436
1437
1438
1439
1440
1441
1442
1443
1444
1445

6.2 MICROPROGRAMMING / LOGIC INFORMATION

ALL OF THE TESTS DESIGNED TO TEST AND VERIFY A SPECIFIC INSTRUCTION OR LOGIC OPERATION CONTAIN A MAINTENANCE HEADER IN THE LISTING AT THE BEGINNING OF THE TEST. THIS HEADER PROVIDES THE TECHNICIAN WITH DETAILED MICROPROGRAMMING AND LOGIC INFORMATION RELATIVE TO THE INSTRUCTION OR LOGIC OPERATION BEING TESTED. THIS SECTION OF THE DOCUMENT DESCRIBES THE FORMAT AND CONTENT OF THIS INFORMATION AND SUGGESTS WAYS THAT THE TECHNICIAN MAY USE IT TO ISOLATE FAULTS TO THE FAILING MODULE OR IC.

INFORMATION FORMAT:

THE MAINTENANCE HEADER SHOWN BELOW IS FOR THE TEST THAT VERIFIES THE OPERATION OF THE INSTRUCTION:

'DECB 1(SP)'

WHERE: 1)THE INITIAL CONTENTS OF THE STACK WORD IS 000000
2)THE DEC (DECREMENT) MODIFIES THIS TO 177400 (DEC ODD BYTE)

HEADER:

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [166,261,262,267,237,270,222,253,075,374,375,
016] FC 1,3,9,8

;ACT BUTS: 37[004]100,166 / 17[166]262,262 / 33[266]220,237
34[237]220,222 / 16[374]016,016

;EXEC: [222]ALUC=LHHH :[375] D=177400

;CODES: [253]SPS=1, [275]SPS=3 / N:C = 1000

;SYNC: B05J2 (-) T = 4.0 USEC

;KEY SIG: K3-3 DM=6 L / K3-4 DEC L / K3-6 BYTE INSTR H
K3-7 ODD BYTE L

1) ROM SEQ ENTRIES:

THIS LISTS THE ROM ADDRESS SEQUENCE THAT MUST BE GENERATED TO PROPERLY EXECUTE THE INSTRUCTION. IT BEGINS WITH THE FIRST ROM WORD AFTER FETCH AND INCLUDES ALL MICROWORDS UP TO THE BEGINNING OF THE NEXT FETCH. THIS IS THE SEQUENCE THE TECHNICIAN SHOULD OBSERVE WHEN CLOCKING THE ROM USING THE KM-11 MAINTENANCE MODULE. IT ALSO INCLUDES A LIST OF THE FLOW CHART NUMBERS TO REFER TO TO OBTAIN THE DETAILS OF EACH MICROWORD.

2) ACT BUTS (ACTIVE MICROBRANCH TESTS) ENTRIES:

1446
1447
1448
1449
1450
1451
1452
1453
1454
1455
1456
1457
1458
1459
1460
1461
1462
1463
1464
1465
1466
1467
1468
1469
1470
1471
1472
1473
1474
1475
1476
1477
1478
1479
1480
1481
1482
1483
1484
1485
1486
1487
1488
1489
1490
1491
1492
1493
1494
1495
1496
1497
1498
1499
1500
1501

THIS IS A LIST OF ONE OR MORE ENTRIES THAT INDICATE ALL THE MICROBRANCHES THAT OCCUR DURING THE SEQUENCE. EACH ENTRY CONSISTS OF FOUR OCTAL NUMBERS AND EACH SEPARATE ENTRY IS DELIMITED BY A '/'. THE NO.S ARE INTERPRETED AS FOLLOWS:

- A) THE FIRST NO. IS THE OCTAL CODE OF THE BUT (UBF <4:0> FIELD)
- B) THE SECOND NO. (IN BRACKETS) INDICATES THE ADDRESS OF THE WORD CONTAINING THE BUT
- C) THE THIRD NO. INDICATES THE ENCODING OF THE NEXT FIELD (UPF <7:0>). THIS IS THE BASE ADDRESS THAT MAY OR MAY NOT BE MODIFIED BY THE MICROBRANCH TEST.
- D) THE FOURTH NO. INDICATES THE RESULT OF THE MICROBRANCH MODIFICATION. BY COMPARING THIS NO. WITH THE NEXT FIELD IT IS POSSIBLE TO DETERMINE WHICH MICROBRANCH CONTROL SIGNALS MUST BE ASSERTED TO OBTAIN THE PROPER ROM SEQUENCE.

THE KEY TO GENERATING THE PROPER MICROINSTRUCTION SEQUENCE IS WHAT THE 'BUT' DOES TO GENERATE THE MICROBRANCH CONTROL SIGNALS (K3-2 BUBC<5:0>). FROM THE 'ACT BUTS' INFORMATION THE TECHNICIAN CAN QUICKLY DETERMINE WHICH OF THESE SIGNALS MUST BE ACTIVE TO GENERATE THE PROPER ROM SEQUENCE.

FOR EXAMPLE THE ENTRY: 37[004]100,166

INDICATES THAT THE BUT37 IN LOC 004 MUST MODIFY THE ROM ADDRESS IN SUCH A WAY THAT THE 100 (NEXT FIELD) GETS CHANGED TO A 166. THIS MEANS THAT WE MUST SOMEHOW 'OR' IN A 066 WITH THE BASE ADDRESS OF 100. THIS IS DONE BY GENERATING THE FOLLOWING CONTROL SIGNALS:

K3-2 BUBC5, BUBC4, BUBC2, AND BUBC1

ALL OF THESE SIGNALS CAN BE OBSERVED AT THE OUTPUT OF THE 'BUT MUX' ON THE K3-2 PRINT. IT IS IMPORTANT THAT THEY BE OBSERVED WHEN THE PUPP=004 WHICH IS THE WORD CONTAINING THE BUT37.

SINCE OVER 50% OF THE LOGIC FAULTS THAT CAN OCCUR IN THE KD11-A WILL MANIFEST THEMSELVES BY CAUSING AN INCORRECT ROM SEQUENCE TO BE GENERATED, THIS INFORMATION PROVIDES THE TECHNICIAN WITH A CONVENIENT STARTING POINT TO PROCEED TO ISOLATE THE FAULTY COMPONENT.

3) EXEC ENTRIES:

THESE ENTRIES DESCRIBE TWO IMPORTANT MICROWORDS.

- A) THE WORD THAT EXECUTES THE INSTRUCTION

1502
1503

B) THE WORD WHERE THE RESULT MAY BE OBSERVED
IN THE DATA DISPLAY ON THE CONSOLE

1504
1505
1506
1507
1508
1509
1510
1511
1512
1513
1514
1515
1516
1517
1518
1519
1520
1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559

THE FIRST ENTRY INCLUDES A ROM ADDRESS THAT CONTAINS THE MICROINSTRUCTION THAT EXECUTES THE INSTRUCTION AND INCLUDES THE REQUIRED STATE OF THE ALU CONTROL SIGNALS TO PERFORM THE CORRECT OPERATION ON THE DATA AS DESCRIBED BELOW:

ALUC = ALU M,S<3:0> = LHHH

WHICH MEANS THE LOGIC LEVELS SHOWN BELOW MUST EXIST:

K3-8 ALUM L
K3-8 ALUS3 H
K3-8 ALUS2 H
K3-8 ALUS1 H
K3-8 ALUS0 H

THE SECOND ENTRY SPECIFIES THE ROM ADDRESS WHERE THE RESULT IS DISPLAYED IN THE CONSOLE DATA DISPLAY. IN THE EXAMPLE SHOWN WITH THE ROM AT PUPP=375 THE DATA DISPLAY ON THE CONSOLE SHOULD CONTAIN A 177400.

4) CODES ENTRIES:

THIS ENTRY SPECIFIES THE SPS CODES USED TO ALTER THE FLAGS AND THE ROM WORDS THAT CONTAIN THESE CODES. WHERE APPLICABLE IT ALSO SPECIFIES HOW THE CODES SHOULD APPEAR AFTER THE INSTRUCTION. IN THE EXAMPLE:

N:C = 1000 MEANS THE 'N' BIT IS SET AND THE 'Z', 'V', AND 'C' BITS ARE CLEAR.

5) SYNC ENTRIES:

THIS INFORMATION CONTAINS A BACKBOARD PIN NO. THAT MAY BE USED TO SYNC DURING A SCOPE LOOP, THE SCOPE TRIGGER SLOPE (-) TRAILING EDGE TO USE, AND THE APPROXIMATE SETTING OF THE HORIZONTAL SWEEP LENGTH. THIS INSURES THAT WHEN LOOPING ON A TEST, ONLY THE AREA OF INTEREST (THE TEST INSTRUCTION) IS DISPLAYED. THIS MINIMIZES THE POSSIBILITY OF LOOKING AT SIGNALS AT THE 'WRONG TIME'.

6) KEY SIGNALS:

THIS ENTRY CONTAINS A LIST OF UNIQUE SIGNALS THAT MUST BE ACTIVATED TO PROPERLY EXECUTE THE OPERATION UNDER TEST. THIS ALSO HELPS TO POINT THE TECHNICIAN TO THE KEY FUNCTIONAL AREAS OF LOGIC IN THE PRINT SETS.

6.3 KM11 MAINTENANCE MODULE

A. PURPOSE

THE KM11 MAINTENANCE MODULE OPTION PROVIDES THE TECH-

1560
1561

NICIAN WITH A MECHANISM FOR SINGLE CLOCKING THE KD11-A MICRO-PROGRAM. IT ALLOWS HIM TO VERIFY THAT THE PROPER ROM SEQUENCES

1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576
1577
1578
1579
1580
1581
1582
1583
1584
1585
1586
1587
1588
1589
1590
1591
1592
1593
1594
1595
1596
1597
1598
1599
1600
1601
1602
1603
1604
1605
1606
1607
1608
1609
1610
1611
1612
1613
1614
1615
1616
1617

ARE BEING GENERATED AND ALSO TO 'FREEZE' THE ROM AT ANY ADDRESS AND CHECK THE CONTROL AND DATA PATH SIGNALS THAT SHOULD BE ACTIVE FOR THE PARTICULAR MICROWORD BEING INVESTIGATED.

B. BASIC COMPONENTS

1. W130 SINGLE HEIGHT MODULE THAT CONTAINS THE INDICATOR DRIVERS FOR THE W131 AND ALSO DOUBLES AS A MODULE EXTENDER FOR THE W131. WHEN INSTALLED IT PLUGS INTO SLOT 'F1' IN THE KD11-A BACK-PLANE WHICH IS PREWIRED TO ACCEPT THE KM11.
2. W131 SINGLE HEIGHT MODULE THAT CONTAINS THE INDICATOR LAMPS AND CONTROL SWITCHES WITH FILTERS. IT PLUGS INTO THE W130 WHEN INSTALLED.
3. KD11-A OVERLAY (5509081-0-12)
A PLASTIC ETCHED OVERLAY THAT DEFINES THE INDICATORS AND SWITCHES ON THE W131.

C. INDICATORS

PUPP<08:00> NINE INDICATOR LAMPS THAT DISPLAY THE CONTENTS OF THE PUPP (PAST MICROPROGRAM POINTER). THEY INDICATE THE ADDRESS OF THE MICROWORD THAT IS CURRENTLY STORED IN THE 'UREG' (THIS WORD IS EXECUTED ON THE NEXT CLOCK PULSE)

BUPP<08:00> NINE INDICATOR LAMPS THAT DISPLAY THE CONTENTS OF THE 'UPP' (MICROPROGRAM POINTER). THEY INDICATE THE ADDRESS OF THE MICROWORD THAT IS CURRENTLY BEING READ OUT OF THE ROM. THIS MICROWORD IS LOADED INTO THE UREG ON THE NEXT CLOCK AND THE CONTENTS OF THE UPP GETS TRANSFERRED INTO THE PUPP.

<T,N:C> FIVE INDICATORS THAT DISPLAY THE STATE OF THE FOLLOWING PSW FLAGS:

- 'T' BIT
- 'N' BIT
- 'Z' BIT
- 'V' BIT
- 'C' BIT

MSYN A SINGLE INDICATOR THAT DISPLAYS THE STATE OF BUS MASTER SYNC.

SSYN A SINGLE INDICATOR THAT DISPLAYS THE STATE OF BUS SLAVE SYNC.

NOTE:

THE STATE OF ALL INDICATOR LAMPS IS DEFINED AS FOLLOWS:

1618
1619
1620

'ON' (1), ASSERTED, ACTIVE
'OFF' (0), NEGATED, INACTIVE

1621
1622
1623
1624
1625
1626
1627
1628
1629
1630
1631
1632
1633
1634
1635
1636
1637
1638
1639
1640
1641
1642
1643
1644
1645
1646
1647
1648
1649
1650
1651
1652
1653
1654
1655
1656
1657
1658
1659
1660
1661
1662
1663
1664
1665
1666
1667
1668
1669
1670
1671
1672
1673
1674
1675
1676

D. SWITCHES

MCLK ENAB WHEN ACTIVE THIS SWITCH DISABLES THE INTERNAL
PROCESSOR CLOCK AND ALLOWS TOGGING A MAINT
ENTANCE CLOCK USING THE 'MCLK' SWITCH.

MCLK WHEN ACTIVATED (ON-OFF SEQUENCE) THIS SWITCH
GENERATES ONE CLOCK PULSE TO THE KD11-A.

MSTOP WHEN ACTIVE THIS SWITCH ENABLES TURNING OFF THE
PROCESSOR CLOCK WHEN THE CONTENTS OF BUFP<8:0>
MATCHES THE ADDRESS CONTAINED IN SR<8:0> IN
THE CONSOLE SWITCH REGISTER. WHEN THE CLOCK
STOPS, THE CONTENTS OF THE ROM ADDRESS SELECT-
ED BY SR<8:0> IS STORED IN THE 'UREG' AND THE
ADDRESS ITSELF CONTAINED IN THE 'PUPP'.

NOTE:

ALL SWITCHES ARE INACTIVE WHEN POSITIONED
TOWARD THE INDICATOR DISPLAY ON THE W131.
AN 'ARROW' ETCHED ON THE OVERLAY SIGNIFIES THE
DIRECTION OF THE ACTIVE POSITION.

E. TYPICAL OPERATING PROCEDURES

1. INSTALLATION

- A) TURN OFF ALL POWER TO THE KD11-A
- B) SLIDE OUT THE KD11-A FROM THE CABINET
- C) OPEN THE HINGED COVER ON THE LEFT SIDE
- D) PLUG THE W130 INTO SLOT F1
- E) PLUG THE W131 INTO THE W130 (WITH OVERLAY ATTACHED)

*****CAUTION*****

INSURE THAT THE HINGED COVER IS SECURED PROPERLY TO PREVENT
IT FROM SWINGING FORWARD AND SHORTING THE PRINTED CIRCUIT
ETCH ON THE BACK OF THE W131.

- D) PLACE ALL THREE SWITCHES ON THE W131 TO 'OFF'
- E) TURN ON THE KD11-A POWER AND LOAD AND START
THE 'DBKDA' DIAGNOSTIC.

2. OPERATION

- A) ESTABLISH A SCOPE LOOP ON THE FAILING TEST
- B) CONSULT THE LISTING FOR THAT TEST TO DETERMINE
IF THE SCOPE SYNC INSTRUCTION IS A SET OR CLEAR
CONDITION CODES AND SET UP SR<8:0> ON THE
CONSOLE AS FOLLOWS:

SET CODES	SR<8:0> = 352
CLR CODES	SR<8:0> = 351

1677
1678
1679
1680

C) PLACE THE 'MSTOP' SWITCH IN THE ACTIVE POSITION.
THE CLOCK SHOULD STOP AND FREEZE THE ROM WITH THE
ADDRESS OF THE FIRST MICROWORD USED TO FETCH THE
TARGET INSTRUCTION CONTAINED IN THE 'UPP' REG.

1681
1682
1683
1684
1685
1686
1687
1688
1689
1690
1691
1692
1693
1694
1695
1696
1697
1698
1699
1700
1701
1702
1703
1704
1705
1706
1707
1708
1709
1710
1711
1712
1713
1714
1715
1716
1717
1718
1719
1720
1721
1722
1723
1724
1725
1726
1727
1728
1729
1730
1731
1732
1733
1734
1735

- D) PLACE THE 'MCLK ENAB' SWITCH ON AND THE 'MSTOP' SWITCH OFF.
- E) NOW YOU ARE READY TO 'TOGGLE' THROUGH THE ROM SEQUENCE FOR THE TEST INSTRUCTION USING THE 'MCLK' SWITCH TO LOCATE AND ISOLATE THE FAILING MICROWORD. EACH TIME THE 'MCLK' SWITCH IS TOGGLED THE CONTENTS OF THE 'BUPP' AND 'PUPP' INDICATORS SHOULD CHANGE TO INDICATE THE ROM SEQUENCE BEING GENERATED. COMPARE THIS SEQUENCE WITH THE CORRECT SEQUENCE INDICATED BY THE MAINTENANCE HEADER INCLUDED IN THE PROGRAM LISTING.

6.4 UPP MATCH MAINTENANCE FEATURE

THERE IS A VALUABLE HARDWARE MAINTENANCE AID BUILT INTO THE KD11-A LOGIC THAT PROVIDES THE TECHNICIAN WITH A METHOD OF ANALYZING THE ROM SEQUENCES BEING GENERATED. IT IS CONTAINED ON THE K1 MODULE AND CONSISTS OF A COMPARATOR CIRCUIT (K1-9 PRINT) THAT ALLOWS TWO UNIQUE LOGIC SIGNALS TO BE GENERATED AS A FUNCTION OF THE ROM ADDRESS CONTAINED IN THE 'UPP' AND THE SETTING OF THE CONSOLE SWITCH REGISTER SR<8:0>.

```
*****  
SR<8:> ---->* MATCH      *----> UPP MATCH H  
              * CIRCUIT  *  
BUPP<8:0>-->*          *----> P MATCH L  
*****
```

UPP MATCH H THIS SIGNAL IS ASSERTED WHENEVER A MATCH OCCURS BETWEEN THE CONTENTS OF THE 'UPP' AND THE CONTENTS OF SR<8:0>. IT CAN BE OBSERVED ON BACK-PLANE PIN C04L2 AND IS USEFUL FOR DETERMINING IF AND WHEN A SPECIFIC ROM WORD IS ACCESSED DURING A PARTICULAR ROM SEQUENCE. IT MAY BE USED AS A SCOPE SYNC TRIGGER OR AS A REFERENCE SIGNAL FOR INVESTIGATING ADDITIONAL CONTROL SIGNALS THAT SHOULD OCCUR DURING A SPECIFIC MICROWORD.

P MATCH L THIS SIGNAL WORKS IN CONJUNCTION WITH THE 'MSTOP' SWITCH ON THE KM11 MAINTENANCE MODULE TO STOP THE CLOCK AND 'FREEZE' THE ROM AT A SPECIFIC ROM ADDRESS. AS DESCRIBED IN PARA. 6.3 (E2), IT IS USEFUL FOR INITIALLY STOPPING THE ROM AT THE CORRECT POINT PRIOR TO SINGLE CLOCKING ROM SEQUENCES.

1778
1779
1780
1781
1782
1783
1784
1785
1786
1787
1788
1789
1790
1791
1792
1793
1794
1795
1796
1797
1798
1799
1800
1801
1802
1803
1804
1805
1806
1807
1808
1809
1810
1811
1812
1813
1814
1815
1816
1817
1818
1819
1820
1821
1822
1823
1824
1825
1826
1827
1828
1829
1830
1831
1832
1833

7.2 SUB-FUNCTIONAL FLOWS

A. BASIC INSTRUCTION TESTS SECTION

* 'BIT' STARTUP *
* SET UP STACK *
* POINTER AND *
* INIT FLAGS AND *
* COUNTERS. *

I
I

T0001-T0006

* SIX BASIC TESTS TO VERIFY THE *
* BNE, BEQ, AND BPL INSTRUCTIONS *
* FOR BOTH THE '1' AND '0' STATE *
* OF THE 'Z' AND 'N' FLAGS. *

I
I

T0007-T0017

* NINE BASIC TESTS OF THE MOV, *
* CMP, AND MOVB INSTRUCTIONS AS *
* THEY ARE USED TO CLEAR THE *
* MISSED TEST TABLE. *

I
I

* ROUTINE TO CLEAR THE MISSED *
* TEST TABLE - BEGIN FLAGGING *
* EACH TEST ENTERED STARTING *
* WITH T0020 *

I
I

T0020-T0035

* FOURTEEN BASIC TESTS OF THE *
* SINGLE OPERAND INSTRUCTIONS IN *
* THE FORMATS USED BY THE *
* UTILITIES (TST, COM, INC, DEC, CLR *
* AND ASL/ROL) *

I
I

T0036-T0041

* FOUR BASIC TESTS OF THE TSTB *
* INSTRUCTION FOR BOTH EVEN AND *
* ODD ADDRESSES. *

I
I

T0042

* BASIC TEST OF THE DECB INSTR- *

1834
1835
1836
1837

* UCTION IN ADDRESS MODE 6 *
* USING THE STACK POINTER. *

I

1838
1839
1840
1841
1842
1843
1844
1845
1846
1847
1848
1849
1850
1851
1852
1853
1854
1855
1856
1857
1858
1859
1860
1861
1862
1863
1864
1865
1866
1867
1868
1869
1870
1871
1872
1873
1874
1875
1876
1877
1878
1879
1880
1881
1882
1883
1884
1885
1886
1887
1888
1889
1890
1891
1892
1893

T0043-T0062 I
I

* SIXTEEN BASIC TESTS TO VERIFY *
* THE MOVE INSTRUCTION IN THOSE *
* FORMATS USED BY THE UTILITIES *

T0063-T0066 I
I

* FOUR BASIC TESTS OF THE COMP- *
* ARE INSTRUCTION IN THOSE FORM- *
* ATS USED BY THE UTILITIES *

T0067-T0074 I
I

* SIX BASIC TESTS OF THE LOGIC *
* INSTRUCTIONS (BIS,BIC,BIT) AS *
* USED BY THE UTILITIES. *

T0075-T0076 I
I

* TWO BASIC TESTS OF THE ADD *
* INSTRUCTIONS AS IT IS USED BY *
* THE UTILITIES. *

T0077-T0103 I
I

* FOUR BASIC TESTS OF THE CMPB *
* INSTRUCTIONS USING BOTH EVEN *
* AND ODD ADDRESSES. *

T0104-T0123 I
I

* SIXTEEN BASIC TESTS OF THE *
* MOVB INSTRUCTION USING BOTH *
* EVEN AND ODD ADDRESSES AND IN *
* ALL ADDRESS MODES USED BY THE *
* THE UTILITIES. *

T0124-T0125 I
I

* TWO TESTS TO VERIFY THE BASIC *
* RTS/JSR INSTRUCTIONS AS USED *
* IN THE UTILITIES. *

T0126-T0127 I
I

* TWO BASIC TESTS OF THE RTI *
* INSTRUCTION. *

1894
1895
1896

I

1897
1898
1899
1900
1901
1902
1903
1904
1905
1906
1907
1908
1909
1910
1911
1912
1913
1914
1915
1916
1917
1918
1919
1920
1921
1922
1923

T0130-T0137 I
I

* EIGHT BASIC TESTS OF THE VAR- *
* IOUS TRAP TYPE INSTRUCTIONS *
* TO VERIFY THAT THE UTILITIES *
* CAN BE CALLED VIA THE TRAP *
* MECHANISM. (IOT, TRAP, EMT, AND *
* RSDV INSTR AND BUS TIMEOUT *
* TRAPS) *

T0140-T0144 I
I

* FIVE BASIC TESTS TO VERIFY THE *
* DL11 INTERFACE USED TO REPORT *
* ERRORS. THE MAINTENANCE MODE *
* FEATURE IS USED TO TURNAROUND *
* AND CHECK AN ALL 1'S ALL 0'S *
* SEQUENCE. *

I

* ENTER THE 'CIT' *
* SECTION *

1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977

B. COMPREHENSIVE INSTRUCTION TESTS SECTION

* FROM 'BIT' SEC*

I
I

* START-UP ROUTINE FOR THE 'CIT'*
* (COMPREHENSIVE INSTRUCTION *
* TESTS SECTION). THE FOLLOWING:*
* FUNCTIONS ARE PERFORMED: *
* 1) SET UP REQUIRED VECTORS *
* 2) CALL SUBROUTINE TO SET *
* BITS IN 'OPTION' TO IND- *
* ICATE INTERNAL OPTIONS *
* FOUND. *
* 3) PRINT PROGRAM NAME *
* 4) PRINT OPTIONS FOUND *
* 5) CLEAR THE PSW AND INIT- *
* THE 'SCOPE LOOP' RETURN *

I
I

T0145-T0146

* TWO QUICK VERIFY TESTS OF THE *
* BASIC CONDITIONAL BRANCHES *
* (BMI,BEQ,BVS,BCS) FOR BOTH *
* THE '1' AND '0' STATE OF THE *
* PERTINENT PSW FLAG. *

I
I

T0147-T0223

* FORTY-FIVE LOGICALLY SEQUENCED*
* TESTS TO VERIFY ALL THE BRANCH*
* INSTRUCTIONS FOR ALL PERTIN- *
* ENT COMBINATIONS OF THE PSW *
* FLAGS. THESE TESTS FOCUS ON *
* THE BRANCH MICROROUTINES ON *
* FLOW CHART 7 AND THE BRANCH *
* INSTR DECISION LOGIC ON THE *
* K5-3 PRINT. *

I
I

T0224-T0227

* THESE FOUR TESTS VERIFY THE *
* SXT INSTR. IN MODE 0. THEY *
* FOCUS ON THE (SXT*DM0) MICRO- *
* ROUTINE ON FLOW CHART 8 *

I

1978
1979
1980
1981
1982
1983
1984
1985
1986
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001
2002
2003
2004
2005
2006
2007
2008
2009
2010
2011
2012
2013
2014
2015
2016
2017
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031

T0230-T0233 I

* THESE FOUR TESTS VERIFY THE *
* SXT INSTR. IN MODE 1 AND 2. *
* THEY FOCUS ON THE (SXT*-DMO) *
* MICROROUTINE ON FLOW CHART 8 *
* AND THE BUT33 IN ROM LOCATIONS*
* 260 AND 266 ON FLOW CHART 3 *

I
T0234-T0235 I

* TWO TESTS TO VERIFY THE SWAB *
* INSTR. IN MODE 0. THEY FOCUS *
* ON THE (SWAB*DMO) MICROROUTINE*
* ON FLOW CHART 7. *

I
T0236-T0237 I

* TWO TESTS TO VERIFY THE SWAB *
* INSTR. IN MODE 1. THEY FOCUS *
* ON THE (SWAB) MICROROUTINE ON *
* FLOW CHART 9 *

I
T0240-T0243 I

* FOUR TESTS TO VERIFY THE NEG *
* INSTR. IN MODE 0 THAT FOCUS ON*
* THE (SOPMORE*DMO*NEG) MICRO-*
* ROUTINE ON FLOW CHART 7. *

I
T0244-T0247 I

* FOUR TESTS TO VERIFY THE NEG *
* INSTRUCTION IN MODE 1 THAT *
* FOCUS ON THE [(SWAB+SOPMORE) *
* *-DMO*NEG] MICROROUTINE ON *
* FLOW CHART 9. *

I
T0250-T0273 I

* TWENTY TESTS THAT VERIFY THE *
* ROR/ASR INSTRUCTIONS THAT *
* FOCUS ON THE (ROTSHF) MICRO- *
* ROUTINES ON FLOW CHART 9. BOTH*
* WORD AND BYTE OPERATIONS ARE *
* TESTED FOR BOTH EVEN AND ODD *
* ADDRESSES. *

I

2032
2033
2034
2035
2036
2037
2038
2039
2040
2041
2042
2043
2044
2045
2046
2047
2048
2049
2050
2051
2052
2053
2054
2055
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070
2071
2072
2073
2074
2075
2076
2077
2078
2079
2080
2081
2082
2083
2084
2085
2086

T0274-T0323 I

* TWENTY FOUR TESTS THAT VERIFY *
* ALL SINGLE OPR. INSTR. OTHER *
* THAN SXT, SWAB, NEG, ASR, ROR, OR *
* JMP. THEY FOCUS ON THE MODE 0 *
* CASE AND THE MICROROUTINE *
* (SOPMORE*DMO*-NEG) ON FLOW *
* CHART 7. *

T0324-T0353 I

* TWENTY FOUR TESTS SIMILAR TO *
* THE PREVIOUS GROUP EXCEPT *
* THEY HANDLE THE NON MODE 0 *
* CASE FOCUSING ON THE (SOPMORE *
* +SWAB)*-NEG*-DMO MICROROUTINE *
* ON FLOW CHART 9 *

T0354-T0370 I

* THIRTEEN TESTS USING THE CLR *
* AND NEG INSTRUCTIONS TO VERIFY *
* THE BYTE MICROROUTINES THAT *
* SUPPORT SINGLE OPERAND INSTR- *
* UCTIONS FOR BOTH EVEN AND ODD *
* ADDRESS CASES. *

T0371-T0417 I

* TWENTY THREE TESTS THAT USE *
* THE ADD INSTRUCTION TO VERIFY *
* THE SOURCE AND DESTINATION *
* MICROBRANCHES ON FLOW CHARTS *
* 2 AND 3. *

T0420-T0427 I

* EIGHT TESTS THAT VERIFY THE *
* XOR INSTRUCTION FOR BOTH THE *
* MODE 0 AND 1 CASES. *

T0430-T0444 I

* THIRTEEN TESTS THAT VERIFY THE *
* MICROWORDS UNIQUE TO THE SUB- *
* TRACT INSTR. FOCUSING ON ROM *
* LOCATIONS 363, 370, AND 365. *

I

2087
2088
2089
2090
2091
2092
2093
2094
2095
2096
2097
2098
2099
2100
2101
2102
2103
2104
2105
2106
2107
2108
2109
2110
2111
2112
2113
2114
2115
2116
2117
2118
2119
2120
2121
2122
2123
2124
2125
2126
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140
2141
2142

T0445-T0452 I

* SIX TESTS USING THE NEG INSTR-*
* TO VERIFY FETCH TO DESTINATION*
* MICROBRANCHES FOR SINGLE OPER-*
* ERAND INSTRUCTIONS USING DM2 *
* THRU DM7. *

I
T0453-T0536 I

* FIFTY TWO MOVE INSTRUCTION *
* TESTS THAT FOCUS ON ALL THE *
* POSSIBLE MOV MICROINSTRUCTION *
* SEQUENCES ON FLOW CHART 4 AND *
* ALL MICRBRANCHES ENTERING AND *
* EXITING THIS SHEET. *

I
T0537-T0602 I

* THIRTY SIX TESTS THAT VERIFY *
* THE BIS,BIC,BIT, AND CMP INSTR*
* UCTIONS IN ALL SOURCE AND DEST*
* MODE COMBINATIONS FOR WORD OPS*

I
T0603-T0616 I

* TWELVE TESTS THAT USE THE BIS *
* INSTRUCTION TO VERIFY THE *
* BYTE MICROROUTINES THAT SUPP- *
* DOUBLE OPERAND INSTRUCTIONS *
* FOR BOTH THE EVEN AND ODD CASE*

I
T0617-T0634 I

* FOURTEEN TESTS THAT VERIFY THE*
* JMP MICRROUTINES ON FLOW CHART *
* 5 AND ALL MICROBRANCHES WITH- *
* WITHIN THESE ROUTINES. *

I
T0635-T0647 I

* TEN TESTS TO VERIFY THE JSR *
* MICROUTINE ON FLOW CHART 5 AND*
* ALL MICROBRANCHES INTO THIS *
* ROUTINE. *

I
T0647-T0654 I

* SIX TESTS TO VERIFY THE SOB *
* INSTRUCTION FOCUSING ON THE *

2143
2144
2145
2146

* MICROROUTINES ON FLOW CHART 7 *
* BOTH THE BRANCH AND NO BRANCH *
* CASES ARE TESTED. *

2147
2148
2149
2150
2151
2152
2153
2154
2155
2156
2157
2158
2159
2160
2161
2162
2163
2164
2165
2166
2167
2168
2169
2170
2171
2172
2173
2174
2175
2176
2177
2178
2179
2180
2181
2182
2183
2184
2185
2186
2187
2188
2189
2190
2191
2192
2193
2194
2195
2196
2197
2198
2199
2200

```
T0655-T0656      I
*****
* TWO TESTS TO VERIFY THE RTS *
* MICROROUTINE ON FLOW CHART 6 *
*****

T0657-T0660      I
*****
* TWO TESTS TO VERIFY THE RTT *
* MICROROUTINE ON FLOW CHART 6 *
*****

T0661-T0662      I
*****
* TWO TESTS TO VERIFY THE MARK *
* MICROROUTINE ON FLOW CHART 5 *
*****

T0663-T0667      I
*****
* FIVE TESTS TO VERIFY THE INT- *
* EGRITY OF THE KW11-L LINE *
* CLOCK OPTION. THESE TESTS ARE *
* SKIPPED IF THE KW11-L IS NOT *
* INSTALLED. *
*****

T0670-T0673      I
*****
* FOUR TESTS THAT VERIFY THE *
* RESET AND WAIT INSTRUCTIONS *
* THAT FOCUS ON THE SERVICE MIC- *
* ROROUTINE ON FLOW CHART 10 AND *
* THE RESET MICROROUTINE ON FLOW *
* CHART 6. *
*****

T0674-T0715      I
*****
* EIGHTEEN TESTS THAT VERIFY THE *
* PRIORITY ARBITRATION LOGIC *
* FOR BR REQUESTS. THEY FOCUS *
* ON THE SERVICE AND TRAP MICRO- *
* ROUTINES ON FLOW CHARTS 10 AND *
* 6. SEVERAL OF THESE TESTS MAY *
* BE SKIPPED IF THE KW11-L IS *
* NOT INSTALLED. *
*****

I
I
*****
* GO TO 'IEX' *
* SECTION *
*****
```

2201
2202
2203
2204
2205
2206
2207
2208
2209
2210
2211
2212
2213
2214
2215
2216
2217
2218
2219
2220
2221
2222
2223
2224
2225
2226
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239
2240
2241
2242
2243
2244
2245
2246
2247
2248
2249
2250
2251
2252

C. COMBINED INSTRUCTION EXERCISER TEST SECTION

* 'IEX' ENTRY *

I
T0716 I

* TEST TO VERIFY THAT THE 'BPT' *
* INSTRUCTION CAUSES A TRAP TO *
* THE VECTOR AT LOC. 14 *

I
T0717-T0734 I

* FOURTEEN TESTS TO VERIFY THAT *
* THE STACK OVERFLOW (BOTH RED *
* AND YELLOW ZONE) MECHANISM AND *
* THE ODD ADDRESS TRAP MECHANISM *
* FUNCTION PROPERLY FOR ALL CASES *
* OF ODD ADDR. ERRORS AND STACK *
* OVERFLOW. *

I
T0735-T0736 I

* TWO TESTS TO VERIFY THAT THE *
* JMP AND JSR CAUSE AN ILLEGAL *
* INSTRUCTION TRAP TO THE VECTOR *
* AT LOCATION 4 WHEN ENCODED IN *
* ADDRESS MODE 0. *

I
T0737-T0740 I

* TWO TESTS TO VERIFY THE BUS *
* TIMEOUT AND 'T' BIT TRAP *
* MECHANISM. *

I
T0741-T0743 I

* THREE TESTS TO VERIFY THAT A *
* 'RED' ZONE TRAP IS SPRUNG IF *
* AN ATTEMPT IS MADE TO PUSH *
* INTO THE PSW,SR, OR SLR USING *
* R6. T0743 IS CONDITIONAL ON *
* WETHER THE KJ11-A OPTION IS *
* INSTALLED OR NOT. *

I

2253
2254
2255
2256
2257
2258
2259
2260
2261
2262
2263
2264
2265
2266
2267
2268
2269
2270
2271
2272
2273
2274
2275
2276
2277
2278
2279
2280
2281
2282
2283
2284
2285
2286
2287
2288
2289

T0744-T0754 I

* NINE TESTS TO VERIFY THAT A *
* RSVI INSTRUCTION TRAP IS SPRU- *
* NG FOR ALL CASES OF RESERVED *
* OPERATION CODES. *

I
T0755-T1007 I

* TWENTY SEVEN TESTS THAT USE *
* THE 'T' BIT TRAP TO VERIFY *
* THAT ALL MICROWORDS ENCODED *
* WITH A 'BUT SERVICE' CAUSE *
* A MICROBRANCH TO THE SERVICE *
* MICROUTINE. *

I
T1010-T1015 I

* SIX ALU/DATA PATH TESTS THAT *
* VERIFY THE ALU OPERATION FOR *
* ALL POSSIBLE BIT INPUT COMBIN- *
* ATIONS FOR THE ADD, SUB, AND, OR, *
* AND LOGICAL FUNCTIONS USING *
* THE ADD, SUB, BIS, BIC, INC, AND *
* DEC INSTRUCTIONS IN VARIOUS *
* COMBINATIONS. *

I
I

* GO TO END OF *
* PASS SERVICE *

2346
2347
2348
2349
2350
2351
2352
2353
2354
2355
2356
2357
2358

ADDRESS DISPLAY:

ADDRESS + 2 OF THE
LOCATION CONTAINING
THE 'HALT'

DATA DISPLAY:

TEST NUMBER

8) TO ESTABLISH A CONTINUOUS SCOPE LOOP THE HALT INSTRUCTION
MUST BE REPLACED WITH A '000400' (BR .+2) BEFORE
DEPRESSING 'CONTINUE'.

2432
2433
2434
2435
2436
2437
2438
2439
2440
2441
2442
2443
2444
2445
2446
2447
2448
2449
2450
2451
2452
2453
2454
2455
2456
2457
2458
2459
2460
2461
2462
2463
2464
2465
2466
2467
2468
2469
2470
2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487

7.4 CORE MEMORY MAP

```
000000 *****  
*          VECTOR AREA          *  
* (ALL UNUSED VECTORS LOADED   *  
* WITH STANDARD PDP11 TRAP-   *  
*   CATCHER)                   *  
*                               *  
000200 *****  
* MOV #3000,PC                  *  
*                               *  
*          PROCESSOR STACK AREA *  
*                               *  
001000 *****  
*          'BCPT'                *  
* (BASIC CENTRAL PROCESSOR     *  
*   TESTS)                      *  
*                               *  
003000 *****  
* 'BIT' START-UP CODE          *  
*                               *  
T0001: *                               *  
*          'BIT'                *  
*                               *  
*          BASIC INSTRUCTION TESTS *  
*          (100[10] TESTS)        *  
*                               *  
CITST: *                               *  
*          'CIT' INITIALIZATION  *  
*                               *  
*                               *  
T0145: *                               *  
*          'CIT'                *  
*                               *  
*          COMPREHENSIVE INSTRUCTION *  
*          TESTS                  *  
*          (360[10] TESTS)        *  
*                               *  
*                               *  
T0715: *                               *  
*          'IEX'                 *  
*                               *  
*          COMBINED INSTRUCTION   *  
*          EXERCISER TESTS        *  
*          (65[10] TESTS)         *  
*                               *  
*                               *  
ENDPS: *                               *  
*          END OF PASS SERVICE    *  
*                               *  
*                               *  
*          UTILITIES AND MISCELLANEOUS *  
*                               *
```

2488
2489
2490

* SUBROUTINES *
* *

2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507

```

*****
OPTION: *
*   CONSTANTS, FLAGS, AND   *
*   VARIABLES                *
*                             *
*****
BELL:  *
*   ASCII MESSAGES          *
*                             *
*****
OBUF:  *
*   COMMON DATA STRUCTURES *
*                             *
*****

```

2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526
2527
2528
2529
2530
2531
2532
2533
2534
2535
2536
2537
2538
2539
2540
2541
2542
2543
2544
2545
2546
2547
2548
2549
2550
2551
2552
2553
2554
2555
2556
2557
2558
2559
2560
2561
2562
2563

8.0 SUB-TITLE INDEX OF TESTS

REFER TO THE TABLE OF CONTENTS IN THE LISTING FOR A DETAILED SUB-TITLE INDEX OF TESTS. THIS TABLE LISTS ALL TESTS SEQUENTIALLY BY TEST NO. WITH A BRIEF DESCRIPTION OF THE FUNCTION OF EACH TEST. THE LEFT HAND COLUMN CONTAINS LINE NUMBERS THAT FACILITATE RAPIDLY LOCATING ANY PARTICULAR TEST WITHIN THE LISTING.

HISTORY FILE FOR KD11-A CPU DIAG

9.0 PURPOSE

9.1 THE PURPOSE OF THIS FILE IS TO MAINTAIN A CONTINUOUS HISTORY OF WHAT HAPPENS TO THE PROGRAM AND ITS SUPPORTING DOCUMENTS THROUGHOUT THE LIFETIME OF THE PROGRAM. ANYONE MAKING ANY CHANGES IN THE PROGRAM OR ITS DOCUMENTS WILL UPDATE THIS FILE TO ACCURATELY DESCRIBE WHAT WAS CHANGED, WHY IT WAS CHANGED, THE DATE OF THE CHANGE, AND THE NAME OF THE PERSON MAKING THE CHANGE. OTHER USEFUL DATA SUCH AS THE NAME OF THE USER REPORTING ANY PROBLEMS AND A DESCRIPTION OF THE PROBLEM SHOULD ALSO BE INCLUDED.

9.2 ENTRIES

(1) DATE: 08-APR-75
NAME: ED CROWLEY, DIAGNOSTIC ENGINEERING, MARLBORO
ENTRY: REVISION A (1ST RELEASE) RELEASED ON THIS DATE.

(2) DATE: 05-AUG-75
NAME: ED CROWLEY, DIAGNOSTIC ENGINEERING, MARLBORO
ENTRY: REVISION B WAS CREATED TO CORRECT A PROGRAM PROBLEM THAT CAUSED AN UNWANTED ERROR HALT AT TEST 0144 IF THE PROGRAM WAS AUTO-STARTED BY THE 'XXDP' MONITOR.

THE CODE SHOWN BELOW WAS ADDED TO TEST 0140 TO STALL EXECUTION OF THE DL11 BASIC TTY TESTS UNTIL THE DL11 INTERFACE HAD SETTLED DOWN AFTER USE BY THE 'XXDP' MONITOR.

```
1$: CLR MBUFO ;INIT COUNTER  
DEC MBUFO ;COUNT THE TIMER  
BNE 1$ ;BR IF NO TIMEOUT
```

THE UPDATED VERSION IS SCHEDULED TO BE RELEASED ON 21-AUGUST-1975 AS MD-11-DBQEAB.

2564
2565
2566



2567
2568
2569
2570
2571
2572
2573
2574
2575
2576
2577
2578
2579
2580
2581
2582
2583
2584
2585
2586
2587
2588
2589
2590
2591
2592
2593
2594
2595
2596
2597
2598
2599
2600
2601
2602
2603
2604
2605
2606
2607
2608
2609
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622

000000

000000 000002
000002 000000

000004 000006
000006 000000

000010 000012

(3) DATE: 15 FEB 80
NAME: LEN LORANGER, DIAGNOSTIC ENGINEERING, MAYNARD
ENTRY: REVISION C WAS CREATED TO MAKE THE ASSEMBLY PROCEDURE
COMPATIBLE WITH ALL THE OTHER DIAGNOSTICS. THE FILES
WERE NOT CHANGED OTHERWISE AND SHOULD NOT AFFECT THE
OPERATION OF THE DIAGNOSTIC.

@
.ENABLE ABS

.=0

; *****
; .SBTTL STANDARD PDP11 'TRAP CATCHER'
; *****

; WHEN THE PROGRAM IS LOADED, LOCATIONS 000000-000776 (VECTOR AREA)
; GET LOADED WITH THE STANDARD PDP11 'TRAP CATCHER'. THE FIRST WORD
; IN EACH VECTOR (NEW PC) IS SET UP TO POINT TO THE SECOND WORD WHICH
; CONTAINS A 'HALT' INSTRUCTION. ANY UNEXPECTED TRAP OR INTERRUPT
; DIRECTED TO A VECTOR THAT HAS NOT BEEN INITIALIZED BY THE PROGRAM TO
; POINT TO AN APPROPRIATE SERVICE ROUTINE WILL CAUSE THE PROGRAM TO HALT.
; AFTER THE HALT THE FOLLOWING ERROR INFORMATION IS AVAILABLE FOR
; DETAILED ERROR ANALYSIS BY THE SERVICE TECHNICIAN:

; ADDRESS DISPLAY- V+4 - WHERE 'V' IS THE VECTOR THE
; TRAP OR INTR. TRAPPED TO.

; DATA DISPLAY- NUMBER OF THE TEST BEING EXECUTED WHEN
; THE TRAP WAS SPRUNG.

; CONTENTS OF THE SP- MEMORY ADDRESS CONTAINING THE CONTENTS
; OF THE PC WHEN THE TRAP WAS SPRUNG

; 'TRAP CATCHER' HALTS ARE CONSIDERED TO BE CATASTROPHIC ERRORS THAT
; NORMALLY PRECLUDE CONTINUING ON IN THE PROGRAM FROM THE POINT OF THE
; ERROR. THE PROGRAM MUST BE RESTARTED OR PROPER MODIFICATIONS MADE
; BASED ON THE ANALYSIS OF THE ERROR INFORMATION.

.+2
HALT

; AFTER EXECUTION OF THE BASIC INSTRUCTION TESTS AND BEFORE THE COM-
; PREHENSIVE INSTRUCTION TESTS, THE VECTOR BELOW IS SET UP TO POINT
; TO THE BUS ERROR SERVICE ROUTINE AT 'BERR:' WITH A PRIORITY OF 7

.+2
HALT

; AFTER EXECUTION OF THE BASIC INSTRUCTION TESTS AND BEFORE THE COM-
; PREHENSIVE INSTRUCTION TESTS, THE VECTOR BELOW IS SET UP TO POINT
; TO THE RSVD INSTR. TRAP SERVICE ROUTINE AT 'RSERR:' WITH A PRIORITY OF 7

.+2

2623	000012	000000
2624	000014	000016
2625	000016	000000
2626		
2627		
2628		
2629		
2630		
2631	000020	000022
2632	000022	000000
2633		
2634		
2635		
2636		
2637		
2638	000024	000026
2639	000026	000000
2640		
2641		
2642		
2643		
2644		
2645	000030	000032
2646	000032	000000
2647		
2648		
2649		
2650		
2651		
2652	000034	000036
2653	000036	000000
2654	000040	000042
2655	000042	000000
2656	000044	000046
2657	000046	000000
2658	000050	000052
2659	000052	000000
2660	000054	000056
2661	000056	000000
2662	000060	000062
2663	000062	000000
2664	000064	000066
2665	000066	000000
2666	000070	000072
2667	000072	000000
2668	000074	000076
2669	000076	000000
2670	000100	000102
2671	000102	000000
2672	000104	000106
2673	000106	000000
2674	000110	000112
2675	000112	000000
2676	000114	000116
2677	000116	000000
2678	000120	000122

HALT
.+2
HALT

:AFTER EXECUTION OF THE BASIC INSTRUCTION TESTS AND BEFORE THE COM-
:PREHENSIVE INSTRUCTION TESTS, THE VECTOR BELOW IS SET UP TO POINT
:TO THE SCOPE SERVICE ROUTINE AT 'SCOPEB:'' WITH A PRIORITY OF 0

.+2
HALT

:AFTER EXECUTION OF THE BASIC INSTRUCTION TESTS AND BEFORE THE COM-
:PREHENSIVE INSTRUCTION TESTS, THE VECTOR BELOW IS SET UP TO POINT
:TO THE POWER FAIL SERVICE ROUTINE AT 'PDWN:'' WITH A PRIORITY OF 7

.+2
HALT

:AFTER EXECUTION OF THE BASIC INSTRUCTION TESTS AND BEFORE THE COM-
:PREHENSIVE INSTRUCTION TESTS, THE VECTOR BELOW IS SET UP TO POINT
:TO THE ERROR SERVICE ROUTINE AT 'ERRB:'' WITH A PRIORITY OF 0

.+2
HALT

:AFTER EXECUTION OF THE BASIC INSTRUCTION TESTS AND BEFORE THE COM-
:PREHENSIVE INSTRUCTION TESTS, THE VECTOR BELOW IS SET UP TO POINT
:TO THE PRINT SERVICE ROUTINE AT 'PRINT:'' WITH A PRIORITY OF 0

.+2
HALT

.+2

2679	000122	000000		HALT		
2680	000124	000126		.+2		
2681	000126	000000		HALT		
2682	000130	000132		.+2		
2683	000132	000000		HALT		
2684	000134	000136		.+2		
2685	000136	000000		HALT		
2686	000140	000142		.+2		
2687	000142	000000		HALT		
2688	000144	000146		.+2		
2689	000146	000000		HALT		
2690	000150	000152		.+2		
2691	000152	000000		HALT		
2692	000154	000156		.+2		
2693	000156	000000		HALT		
2694	000160	000162		.+2		
2695	000162	000000		HALT		
2696	000164	000166		.+2		
2697	000166	000000		HALT		
2698	000170	000172		.+2		
2699	000172	000000		HALT		
2700	000174	000176		.+2		
2701	000176	000000		HALT		
2702	000200	012707	003000	MOV	#3000,PC	;GO START UP AT LOC. 3000
2703	000204	000206		.+2		
2704	000206	000000		HALT		
2705	000210	000212		.+2		
2706	000212	000000		HALT		
2707	000214	000216		.+2		
2708	000216	000000		HALT		
2709	000220	000222		.+2		
2710	000222	000000		HALT		
2711	000224	000226		.+2		
2712	000226	000000		HALT		
2713	000230	000232		.+2		
2714	000232	000000		HALT		
2715	000234	000236		.+2		
2716	000236	000000		HALT		
2717	000240	000242		.+2		
2718	000242	000000		HALT		
2719	000244	000246		.+2		
2720	000246	000000		HALT		
2721	000250	000252		.+2		
2722	000252	000000		HALT		
2723	000254	000256		.+2		
2724	000256	000000		HALT		
2725	000260	000262		.+2		
2726	000262	000000		HALT		
2727	000264	000266		.+2		
2728	000266	000000		HALT		
2729	000270	000272		.+2		
2730	000272	000000		HALT		
2731	000274	000276		.+2		
2732	000276	000000		HALT		
2733	000300	000302		.+2		
2734	000302	000000		HALT		

2735	000304	000306	.+2
2736	000306	000000	HALT
2737	000310	000312	.+2
2738	000312	000000	HALT
2739	000314	000316	.+2
2740	000316	000000	HALT
2741	000320	000322	.+2
2742	000322	000000	HALT
2743	000324	000326	.+2
2744	000326	000000	HALT
2745	000330	000332	.+2
2746	000332	000000	HALT
2747	000334	000336	.+2
2748	000336	000000	HALT
2749	000340	000342	.+2
2750	000342	000000	HALT
2751	000344	000346	.+2
2752	000346	000000	HALT
2753	000350	000352	.+2
2754	000352	000000	HALT
2755	000354	000356	.+2
2756	000356	000000	HALT
2757	000360	000362	.+2
2758	000362	000000	HALT
2759	000364	000366	.+2
2760	000366	000000	HALT
2761	000370	000372	.+2
2762	000372	000000	HALT
2763	000374	000376	.+2
2764	000376	000000	HALT
2765	000400	000402	.+2
2766	000402	000000	HALT
2767	000404	000406	.+2
2768	000406	000000	HALT
2769	000410	000412	.+2
2770	000412	000000	HALT
2771	000414	000416	.+2
2772	000416	000000	HALT
2773	000420	000422	.+2
2774	000422	000000	HALT
2775	000424	000426	.+2
2776	000426	000000	HALT
2777	000430	000432	.+2
2778	000432	000000	HALT
2779	000434	000436	.+2
2780	000436	000000	HALT
2781	000440	000442	.+2
2782	000442	000000	HALT
2783	000444	000446	.+2
2784	000446	000000	HALT
2785	000450	000452	.+2
2786	000452	000000	HALT
2787	000454	000456	.+2
2788	000456	000000	HALT
2789	000460	000462	.+2
2790	000462	000000	HALT

2791	000464	000466	.+2
2792	000466	000000	HALT
2793	000470	000472	.+2
2794	000472	000000	HALT
2795	000474	000476	.+2
2796	000476	000000	HALT
2797	000500	000502	.+2
2798	000502	000000	HALT
2799	000504	000506	.+2
2800	000506	000000	HALT
2801	000510	000512	.+2
2802	000512	000000	HALT
2803	000514	000516	.+2
2804	000516	000000	HALT
2805	000520	000522	.+2
2806	000522	000000	HALT
2807	000524	000526	.+2
2808	000526	000000	HALT
2809	000530	000532	.+2
2810	000532	000000	HALT
2811	000534	000536	.+2
2812	000536	000000	HALT
2813	000540	000542	.+2
2814	000542	000000	HALT
2815	000544	000546	.+2
2816	000546	000000	HALT
2817	000550	000552	.+2
2818	000552	000000	HALT
2819	000554	000556	.+2
2820	000556	000000	HALT
2821	000560	000562	.+2
2822	000562	000000	HALT
2823	000564	000566	.+2
2824	000566	000000	HALT
2825	000570	000572	.+2
2826	000572	000000	HALT
2827	000574	000576	.+2
2828	000576	000000	HALT
2829	000600	000602	.+2
2830	000602	000000	HALT
2831	000604	000606	.+2
2832	000606	000000	HALT
2833	000610	000612	.+2
2834	000612	000000	HALT
2835	000614	000616	.+2
2836	000616	000000	HALT
2837	000620	000622	.+2
2838	000622	000000	HALT
2839	000624	000626	.+2
2840	000626	000000	HALT
2841	000630	000632	.+2
2842	000632	000000	HALT
2843	000634	000636	.+2
2844	000636	000000	HALT
2845	000640	000642	.+2
2846	000642	000000	HALT

2847	000644	000646	.+2
2848	000646	000000	HALT
2849	000650	000652	.+2
2850	000652	000000	HALT
2851	000654	000656	.+2
2852	000656	000000	HALT
2853	000660	000662	.+2
2854	000662	000000	HALT
2855	000664	000666	.+2
2856	000666	000000	HALT
2857	000670	000672	.+2
2858	000672	000000	HALT
2859	000674	000676	.+2
2860	000676	000000	HALT
2861	000700	000702	.+2
2862	000702	000000	HALT
2863	000704	000706	.+2
2864	000706	000000	HALT
2865	000710	000712	.+2
2866	000712	000000	HALT
2867	000714	000716	.+2
2868	000716	000000	HALT
2869	000720	000722	.+2
2870	000722	000000	HALT
2871	000724	000726	.+2
2872	000726	000000	HALT
2873	000730	000732	.+2
2874	000732	000000	HALT
2875	000734	000736	.+2
2876	000736	000000	HALT
2877	000740	000742	.+2
2878	000742	000000	HALT
2879	000744	000746	.+2
2880	000746	000000	HALT
2881	000750	000752	.+2
2882	000752	000000	HALT
2883	000754	000756	.+2
2884	000756	000000	HALT
2885	000760	000762	.+2
2886	000762	000000	HALT
2887	000764	000766	.+2
2888	000766	000000	HALT
2889	000770	000772	.+2
2890	000772	000000	HALT
2891	000774	000776	.+2
2892	000776	000000	HALT

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

000000
000001
000002
000003
000004
000005
000006
000007

177570
177776

100000
040000
020000
010000
004000
002000
001000

000004

104000
104001
104002
104003
104004
104005
104006
104007

104400
177546

177560
177562

```
; *****  
; PROGRAM DEFINITIONS  
; *****  
  
; GENERAL REGISTER DEFINITIONS  
R0 = %0  
R1 = %1  
R2 = %2  
R3 = %3  
R4 = %4  
R5 = %5  
SP = %6  
PC = %7  
  
; DEFINITIONS FOR KD11-A PROCESSOR STATUS WORD AND CONSOLE SWITCH REGISTER  
SR = 177570 ; CONSOLE SWITCH REG. ADDR  
PSW = 177776 ; PROCESSOR STATUS REG. ADDR  
  
; DEFINITIONS FOR CONSOLE SWITCH REG. - BIT POSITIONS  
SW15 = 100000  
SW14 = 040000  
SW13 = 020000  
SW12 = 010000  
SW11 = 004000  
SW10 = 002000  
SW09 = 001000  
  
; IOT USED TO CALL 'SCOPE' LOOP UTILITY  
SCOPE = IOT  
  
; EMT USED TO CALL 'ERROR' SERVICE ROUTINE  
ERROR=EMT ; PRINT 8 COLUMNS  
ERROR1 = EMT+1 ; PRINT COLUMN 1 ONLY  
ERROR2=EMT+2 ; PRINT COLUMNS 1 AND 2  
ERROR3 = EMT+3 ; PRINT COLUMNS 1,2,3  
ERROR4 = EMT+4 ; PRINT COLUMNS 1,2,3,4  
ERROR5 = EMT+5 ; PRINT COLUMNS 1,2,3,4,5  
ERROR6 = EMT+6 ; PRINT COLUMNS 1,2,3,4,5,6  
ERROR7 = EMT+7 ; PRINT COLUMNS 1,2,3,4,5,6,7  
  
; TRAP USED TO CALL THE PRINT UTILITY  
TYPE = TRAP  
  
LKCSR= 177546 ; KW11-L LINE CLOCK ADDRESS  
  
; ADDRESS ASSIGNMENTS FOR DL11 CONSOLE TERMINAL INTERFACE  
RCSR=177560 ; RCVR. CONTROL / STATUS REG. ADDRESS  
RDBR = 177562 ; RECEIVER DATA BUFFER REG. ADDR.
```

2949	177564	XCSR = 177564	:TRANSMITTER CONTROL / STATUS REG. ADDR
2950	177566	XDBR = 177566	:TRANSMIT DATA BUFFER REG. ADDR.
2951	001000	. =1000	
2952			
2953	001000	STACKL=001000	:TOP OF STACK FOR LOWER TESTS

2980
2981
2982
2983
2984
2985
2986
2987
2988
2989
2990
2991
2992
2993
2994
2995
2996
2997
2998 001004 000402
2999
3000 001006 000403
3001
3002 001010 000000
3003
3004 001012 000775
3005
3006 001014 000000
3007

```
; *****  
; .SBTTL BT002 'BR' TEST - NEGATIVE OFFSET  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [111,340,341,016] FC 1,7  
;ACT BUTS: 37[004]100,111 / 16[340]016,016  
;EXEC: [341]ALUC=LHLLH :[016]D=001006  
;CODES: N/A / N:C=0000  
;SYNC: N/A T=1.76 USEC  
;KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR L / K3-3 SM=0 L / K3-3 IR(14:12)=0 L  
  
BT002: BR I002 ;GO TO TEST INSTRUCTION  
A002: BR BT003 ;GO TO NEXT TEST  
EX002: HALT ;JUST IN CASE  
I002: BR A002 ;TEST THE BR - NEG. OFFSET  
E2002: HALT ;BR FAILED WITH NEG. OFFSET
```

```
3008 ; *****  
3009 ; .SBTTL BT003 'BASIC COND. BR' TEST - FLAGS CLEARED  
3010 ; *****  
3011 ;  
3012 ;MICROPROGRAMMING / LOGIC INFORMATION (BMI,BEQ,BVS)  
3013 ;  
3014 ;ROM SEQ: [110,347,016] FC 1,7  
3015 ;  
3016 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
3017 ;  
3018 ;EXEC: NO BRANCH  
3019 ;  
3020 ;CODES: N/A / N:C=0000  
3021 ;  
3022 ;SYNC: N/A T=1.4 USEC  
3023 ;  
3024 ;KEY SIG: K5-3 BR INSTR L / (BMI)K3-3 SM=0 L / (BEQ)K3-3 SM=1 L / (BVS)K3-3 SM=2 L  
3025 ; (BMI,BVS)K3-4 IR15 L  
3026 ;  
3027 001016 100403 BT003: BMI E003 ;BR IF 'N' SET  
3028 001020 001402 BEQ E003 ;BR IF 'Z' SET  
3029 001022 102401 BVS E003 ;BR IF 'V' SET  
3030 001024 103002 BCC BT004 ;BR IF 'C' CLEAR  
3031 ;  
3032 001026 000000 E003: HALT ;ERROR - ONE OF THE ABOVE BR'S FAILED  
3033 ;OR THE FLAGS FAILED TO CLEAR ON 'START'  
3034 001030 000772 BR BT003 ;LOCK ON HARD ERROR  
3035
```

3036
3037
3038
3039
3040
3041
3042
3043
3044
3045
3046
3047
3048
3049
3050
3051
3052
3053
3054
3055
3056
3057
3058
3059
3060
3061
3062
3063
3064

001032 000277
001034 100003
001036 001002
001040 102001
001042 103402
001044 000000
001046 000771

```
; *****  
; .SBTTL BT004 'SCC AND COND. BR'S' TEST - FLAGS SET  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION (SCC)  
;ROM SEQ:      [117,352,016] FC 1,7  
;ACT BUTS:     37[004]100,117 / 16[117]016,016  
;EXEC:         [117]ALUC=HHLHH,[352]ALUC=HHHL :[016]D=000017  
;CODES:        [352]SPS=3      /      N:C=1111  
;SYNC:         B05J2 (+)      T=1.72 USEC  
;KEY SIG:      K3-6 CC INSTR H / K3-6 I1K0(CINSTR) L / K3-3 IR04(1) H  
BT004:  SCC                      ;MAKE N:C=1111  
I004:   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  
BR      BT004  
;ERROR - ONE OF THE ABOVE BR'S FAILED  
;OR THA SCC FAILED TO SET ALL THE FLAGS  
;LOCK ON HARD ERROR
```

```
3065 ; *****  
3066 ; .SBTTL BT005 'CCC AND COND. BR'S' TEST - FLAGS CLEARED  
3067 ; *****  
3068  
3069 ;MICROPROGRAMMING / LOGIC INFORMATION (CCC)  
3070  
3071 ;ROM SEQ: [116,350,351,016] FC 1,7  
3072  
3073 ;ACT BUTS: 37[004]100,116 / 16[350]016,016  
3074  
3075 ;EXEC: [116,351]ALUC=HHLHM,[350]ALUC=HLHLM :[016]D=000000  
3076  
3077 ;CODES: [351]SPS=3 / N:C=0000  
3078  
3079 ;SYNC: B05J2 (+) T=2.02 USEC  
3080  
3081 ;KEY SIG: K3-6 CC INSTR H  
3082  
3083 001050 000257 BT005: CCC ;MAKE N:C=0000  
3084  
3085 001052 100403 I005: BMI E005 ;BR IF 'N' STILL SET  
3086 001054 001402 BEQ E005 ;BR IF 'Z' STILL SET  
3087 001056 102401 BVS E005 ;BR IF 'V' STILL SET  
3088 001060 103002 BCC BT006 ;BR IF 'C' GOT CLEARED  
3089  
3090 001062 000000 E005: HALT ;ERROR - ONE OF THE ABOVE BR'S FAILED  
3091 ;OR THE CCC FAILED TO CLEAR ALL FLAGS  
3092 001064 000771 BR BT005 ;LOCK ON HARD ERROR  
3093
```

3094
3095
3096
3097
3098
3099
3100
3101
3102
3103
3104
3105
3106
3107
3108
3109
3110
3111
3112
3113
3114
3115
3116
3117
3118
3119
3120

001066 000257
001070 005000
001072 001402
001074 000000
001076 000773

```
; *****  
; .SBTTL BT006 'CLR %R' TEST - SETS THE 'Z' BIT  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [104,373,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
;EXEC: [104]ALUC=HLLMH :[373]D=000000  
;CODES: [373]SPS=1,[360]SPS=3 / N:C=0100  
;SYNC: B05J2 (-) T=1 USEC  
;KEY SIG: K3-4 CLR L / K3-3 DM=0 L / K3-4 OVLAP INSTR H  
BT006: CCC ;MAKE N:C=0000  
I006: CLR R0 ;TEST THE CLR - IT SHOULD SET 'Z'  
;BR IF CLR SET 'Z'  
E006: HALT ;ERROR - CLR FAILED TO SET 'Z'  
BR BT006 ;LOCK ON HARD ERROR
```

BT006 'CLR %R' TEST - SETS THE 'Z' BIT

SEQ 0082

3121
3122
3123
3124
3125
3126
3127
3128
3129
3130
3131
3132
3133
3134
3135
3136
3137
3138
3139 001100 005000
3140 001102 000257
3141
3142 001104 005700
3143
3144 001106 001402
3145
3146 001110 000000
3147
3148 001112 000772
3149

```
; *****  
; .SBTTL BT007 'TST %R' TEST - USING THE CLR  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [104,373,362,001] FC 1,7,8  
;ACT BUTS: 37[004]100,104 / 31[104]360,362 / 27[373]000,001  
;EXEC: [104]ALUC=LLLLL :[373]D=000000  
;CODES: [373]SPS=1,[362]SPS=3 / N:C=0100  
;SYNC: B05J2 (-) T=1 USEC  
;KEY SIG: K3-3 DM=0 L / K3-4 TST L / K3-4 OVLAP INSTR H  
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  
;ALL ZEROES OR TST FAILED  
BR BT007 ;LOCK ON HARD ERROR
```

3150
3151
3152
3153
3154
3155
3156
3157
3158
3159
3160
3161
3162
3163
3164
3165
3166
3167
3168
3169
3170
3171
3172
3173
3174
3175
3176
3177
3178

001114 005000
001116 000257
001120 005100
001122 100001
001124 103402
001126 000000
001130 000771

```
; *****  
; .SBTTL BT010 'COM %R' TEST - SHOULD SET 'N' AND 'C'  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [104,373,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
;EXEC: [104]ALUC=HLLLL :[373]D=177777  
;CODES: [373]SPS=1,[360]SPS=3 / N:C=1001  
;SYNC: B05J2 (-) T=1 USEC  
;KEY SIG: K3-4 COM L / K3-3 DM=0 L / K3-4 OVLAP INSTR H  
BT010: CLR R0 :MAKE [R0] = 000000  
CCC :MAKE N:C=0000  
J010: 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
```

BT010 'COM %R' TEST - SHOULD SET 'N' AND 'C'

SEQ 0084

3179
3180
3181
3182
3183
3184
3185
3186
3187
3188
3189
3190
3191
3192
3193
3194
3195
3196
3197
3198
3199
3200
3201
3202
3203
3204
3205
3206
3207
3208
3209
3210
3211
3212

```
; *****  
; .SBTTL BT011 'COM %R AND ADC %R' TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION (COM %0)  
;ROM SEQ:      [104,373,360,001] FC 1,7,8  
;ACT BUTS:     37[004]100,104 / 31[104]360,360 / 27[373]000,001  
;EXEC:        [104]ALUC=HLLLL :[373]D=177777  
;              (ADC) :[373]D=000000  
  
;CODES:       [373]SPS=1,[360]SPS=3 /      N:C=1001  
;              (ADC)      N:C=0101  
  
;SYNC:        B05J2 (-)      T=1 USEC  
  
;KEY SIG:     K3-3 DM=0 L / K3-4 COM L / K3-4 OVLAP INSTR H  
; (ADC)      K3-3 DM=0 L / K3-4 ADC L / K3-4 OVLAP INSTR H / K3-8 CIN00 L  
  
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 = 0C0000  
  
      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
```

```
3213 ; *****  
3214 ; .SBTTL BT012 'MOV #N,R' TEST WITH N=177777,[R]=000000  
3215 ; *****  
3216  
3217 ;MICROPROGRAMMING / LOGIC INFORMATION  
3218  
3219 ;ROM SEQ: [172,257,200,125,375,016] FC 1,4,8  
3220  
3221 ;ACT BUTS: 37[004]100,172 / 22[172]200,200 / 16[125]016,016  
3222  
3223 ;EXEC: [200]ALUC=LLLLL :[125]D=177777  
3224  
3225 ;CODES: [125]SPS=3 / N:C=1000  
3226  
3227 ;SYNC: B05J2 (-) T= 2.3 USEC  
3228  
3229 ;KEY SIG: K3-3 MOV L / K3-3 SM=2 L / K3-3 DM=0 L  
3230  
3231 001152 005000 BT012: CLR R0 ;MAKE [R0] = 000000  
3232 001154 000257 CCC ;MAKE N:C=0000  
3233  
3234 001156 012700 177777 I012: MOV #-1,R0 ;TEST THE MOV - [R0] S/B = 177777  
3235  
3236 001162 005100 COM R0 ;MAKE [R0] = 000000  
3237 001164 001402 BEQ BT013 ;BR IF 'Z' SET  
3238  
3239 001166 000000 E012: HALT ;ERROR - MOV FAILED TO LOAD R0 WITH ALL 1'S  
3240 001170 000770 BR BT012 ;LOCK ON HARD ERROR  
3241
```

BT012 'MOV #N,R' TEST WITH N=177777,[R]=000000

SEQ 0086

3242
3243
3244
3245
3246
3247
3248
3249
3250
3251
3252
3253
3254
3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271
3272
3273

; *****
; .SBTTL BT013 'MOV #N,R' TEST WITH N=000000,[R]=177777
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,160,204,000] FC 1,4,8
;ACT BUTS: 37[004]100,142 / 35[240]120,160 / 20[160]000,000
;EXEC: [160]ALUC=LLLLL :[204]D=000000
;CODES: [204]SPS=3 / N:C=0100
;SYNC: B05J2 (-) T=2.3 USEC
;KEY SIG: K3-3 MOV L / K3-3 SM=2 L / K3-3 DM=0 L

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

BT013 'MOV #N,R' TEST WITH N=000000,[R]=177777

SEQ 0087

3274
3275
3276
3277
3278
3279
3280
3281
3282
3283
3284
3285
3286
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300
3301
3302

001216 012706 001000
001222 012700 177776
001226 000277

001230 005010

001232 001002

001234 000000
001236 000767

```
; *****  
; .SBTTL BT014 'CLR (R)' TEST - [R] = 177776  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=HLLHH :[211]D=000000  
;CODES: [211]SPS=1,[367]SPS=3 / N:C=0000  
;SYNC: B05J2 (-) T=2.6 USEC  
;KEY SIG: K3-4 CLR L / K3-3 DM=1 L  
BT014: MOV #STACKL,SP ;SET UP STACK POINTER  
MOV #PSW,RO ;RO POINTS TO PSW  
SCC ;MAKE [PSW] = 017  
  
I014: CLR (R0) ;TEST THE CLR - IT SHOULD CLEAR PSW  
BNE BT015 ;BR IF CLR MADE 'Z' = 0 - IT SHOULD  
  
E014: HALT ;ERROR- CLR FAILED TO CLEAR PSW  
BR BT014 ;LOCK ON HARD ERROR
```

3303
3304
3305
3306
3307
3308
3309
3310
3311
3312
3313
3314
3315
3316
3317
3318
3319
3320
3321
3322
3323
3324
3325
3326
3327
3328
3329
3330
3331
3332
3333
3334
3335
3336
3337
3338

```
; *****  
; .SBTTL BT015 'CLR (R)'+ TEST - [R] = 177776  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [162,260,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,162 / 33[260]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=HLLHH :[211]D=000000  
;CODES: [211]SPS=1,[367]SPS=3 / N:C=0000  
;SYNC: B05J2 (-) T=2.6 USEC  
;KEY SIG: K3-4 CLR L / K3-3 DM=2 L / K5-5 BCON (1+2) H  
BT015: MOV #PSW,R0 ;R0 POINTS TO PSW  
SCC ;MAKE [PSW] = 017  
I015: CLR (R0)+ ;TEST THE CLR - IT SHOULD CLEAR PSW  
BNE A015 ;BR IF CLR MADE 'Z' = 0 - IT SHOULD  
E1015A: HALT ;ERROR- CLR FAILED TO CLEAR PSW  
BR BT015 ;LOCK ON HARD ERROR  
A015: TST R0 ;AUTO INC SHOULD ZERO R0  
BEQ BT016 ;BR IF IT DID  
E2015: HALT ;ERROR - AUTOINC. FAILED  
BR BT015 ;LOCK ON HARD ERROR
```

BT015 'CLR (R)'+ TEST - [R] = 177776

SEQ 0089

3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356
3357
3358
3359
3360
3361
3362
3363
3364
3365
3366
3367
3368
3369

; *****
; .SBTTL BT016 'COM (R)'+ TEST - [R] = 177776
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016
;EXEC: [220]ALUC=HLLLL :[211]D=000357
;CODES: [211]SPS=1,[367]SPS=3 / N:C=1111
;SYNC: B05J2 (-) T=1.8 USEC
;KEY SIG: K3-4 COM L / K3-3 DM=1 L

001266 012700 177776
001272 000257
001274 005110
001276 100003
001300 001002
001302 102001
001304 103402
001306 000000
001310 000766

BT016: MOV #PSW,R0 ;R0 POINTS TO PSW
CCC ;MAKE [PSW] = 000
I016: COM (R0) ;TEST THE COM - [PSW] S/B = 357
BPL E016 ;N:C=1111 ?
BNE E016
BVC E016
BCS BT017
E016: HALT ;ERROR - COM FAILED TO MAKE [PSW] = 357
BR BT016 ;LOCK ON HARD ERROR

3370
3371
3372
3373
3374
3375
3376
3377
3378
3379
3380
3381
3382
3383
3384
3385
3386
3387
3388 001312 012700 177776
3389 001316 005010
3390 001320 000257
3391
3392 001322 005120
3393
3394 001324 100003
3395 001326 001002
3396 001330 102001
3397 001332 103402
3398
3399 001334 000000
3400 001336 000765
3401
3402 001340 005100
3403 001342 005500
3404 001344 001402
3405
3406 001346 000000
3407 001350 000760
3408

```
; *****  
; .SBTTL BT017 'COM (R0)'+ TEST - [R0] = 177776  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ:      [162,160,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS:     37[004]100,162 / 33[260]220,220 / 16[367]016,016  
;EXEC:         [220]ALUC=HLLLL :[211]D=000357  
;CODES:        [211]SPS=1,[367]SPS=3 /      N:C=1111  
;SYNC:         B05J2 (-)      T=2 USEC  
;KEY SIG:      K3-4 COM L / K3-3 DM=2 L / K5-5 BCON(1+2) H  
  
BT017:  MOV #PSW,R0      ;R0 POINTS TO PSW  
        CLR (R0)        ;MAKE [PSW] = 000  
        CCC              ;SCOPE SYNC  
  
I017:   COM (R0)+      ;TEST THE COM - [PSW] S/B = 357  
  
        BPL E1017      ;N:C = 1111 ?  
        BNE E1017  
        BVC E1017  
        BCS A017  
  
E1017:  HALT           ;COM FAILED TO SET ALL FLAGS  
        BR BT017      ;LOCK ON HARD ERROR  
  
A017:   COM R0         ;SHOULD MAKE [R0] = 177777  
        ADC R0         ;SHOULD MAKE [R0] = 000000  
        BEQ BT020  
  
E2017:  HALT           ;ERROR - COM FAILED TO AUTO INC. R0  
        BR BT017      ;LOCK ON HARD ERROR
```

3409
3410
3411
3412
3413
3414
3415
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429
3430
3431
3432
3433
3434
3435
3436
3437
3438
3439
3440
3441
3442
3443
3444

001352 005000
001354 005001
001356 005101
001360 000257

001362 010100

001364 100402

001366 000000
001370 000770

001372 005100
001374 001402

001376 000000
001400 000764

```
; *****  
; .SBTTL BT020 'MOV RA,RB' TEST - WITH [RA]=177777,[RB]=000000  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [170,204,001] FC 1,4,8  
;ACT BUTS: 37[004]100,170 / 20[170]000,001  
;EXEC: [170]ALUC=LLLLL :[204]D=177777  
;CODES: [204]SPS=3 / N:C=1000  
;SYNC: B05J2 (-) T=1 USEC  
;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=0 L / K3-4 OVLAP INSTR H  
BT020: CLR R0 :MAKE [R0]=000000  
 CLR R1 :MAKE [R1]=000000  
 COM R1 :MAKE [R1]=0207777  
 CCC :SCOPE SYNC  
I020: MOV R1,R0 :TEST THE MOV  
 BMI A020 :BR IF 'N' GOT SET  
E1020: HALT :ERROR-MOV FAILED TO SET 'N'  
 BR BT020 :LOCK ON HARD ERROR  
A020: COM R0 :[R0] SHOULD GO TO 000000  
 BEQ BT021 :BR IF IT DID  
E2020: HALT :ERROR-MOV FAILED TO LOAD R0 WITH 1'S  
 BR BT020 :LOCK ON HARD ERROR
```

BT020 'MOV RA,RB' TEST - WITH [RA]=177777,[RB]=000000

SEQ 0092

3445
3446
3447
3448
3449
3450
3451
3452
3453
3454
3455
3456
3457
3458
3459
3460
3461
3462
3463
3464
3465
3466
3467
3468
3469
3470
3471
3472
3473
3474
3475
3476
3477
3478
3479
3480
3481

```
; *****  
; .SBTTL BT021 'MOV RA,RB' TEST WITH [RA]=000000,[RB]=177777  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ:      [170,204,001] FC 1,4,8  
;ACT BUTS:     37[004]100,170 / 20[170]000,001  
;EXEC:         [170]ALUC=LLLLL :[204]D=000000  
;CODES:        [204]SPS=3      /      N:C=0100  
;SYNC:         B05J2 (-)      T=1 USEC  
;KEY SIG:      K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=0 L / K3-4 OVLAP INSTR H  
  
BT021.  CLR    R0      ;MAKE [R0]=000000  
        COM    R0      ;MAKE [R0]=177777  
        CLR    R1      ;MAKE [R1]=000000  
        CCC                    ;SCOPE SYNC  
  
I021:   MOV    R1,R0   ;TEST THE MOV  
        BEQ    A021    ;BR IF 'Z' GOT SET  
  
E1021:  HALT                    ;MOV FAILED TO SET 'Z'  
        BR     BT021    ;LOCK ON HARD ERROR  
  
A021:   COM    R0      ;SHOULD MAKE [R0]=177777 AND SET 'C'  
        ADC    R0      ;SHOULD MAKE [R0]=000000  
        BEQ    BT022    ;BR IF 'Z' SET  
  
E2021:  HALT                    ;MOV FAILED TO ZERO R0  
        BR     BT021    ;LOCK ON HARD ERROR
```

BT021 'MOV RA,RB' TEST WITH [RA]=000000,[RB]=177777

SEQ 0093

3482
3483
3484
3485
3486
3487
3488
3489
3490
3491
3492
3493
3494
3495
3496
3497
3498
3499
3500
3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513

; *****
; .SBTTL BT022 'MOV #N,@#A' TEST WITH N=17,A=177776
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,173,207,210,200,125,375,016] FC 1,2,4,8
;ACT BUTS: 37[004]100,142 / 35[240]120,173 / 22[207]200,200 / 16[125]016,016
;EXEC: [200]ALUC=LLLLL :[125]D=000017
;CODES: [125]SPS=3 / N:C=1111
;SYNC: B05J2 (-) T=4 USEC
;KEY SIG: K3-3 MOV L / K3-3 SM=2 L / K3-3 DM=3 L / K5-5 BC01 H
; K5-5 BCON(1+2) H

001434 000257 BT022: CCC ;MAKE [PSW]=000
001436 012737 000017 177776 I022: MOV #17,@#PSW ;TEST THE MOV
001444 100003 BPL E022 ;N:C=1111
001446 001002 BNE E022
001450 102001 BVC E022
001452 103402 BCS BT023
001454 000000 E022: HALT ;MOV FAILED TO LOAD PSW
001456 000766 BR BT022 ;LOCK ON HARD ERROR

```
3514 ; *****
3515 ; .SBTTL BT023 'MOV RA,(RB)+' TEST WITH [RA]=17,[RB]=177776
3516 ; *****
3517
3518 ;MICROPROGRAMMING / LOGIC INFORMATION
3519
3520 ;ROM SEQ: [172,257,201,125,375,016] FC 1,4,8
3521
3522 ;ACT BUTS: 37[004]100,172 / 22[172]200,201 / 16[125]016,016
3523
3524 ;EXEC: [201]ALUC=LLLLL :[125]D=000017
3525
3526 ;CODES: [125]SPS=3 / N:C=1111
3527
3528 ;SYNC: B05J2 (-) T=2.42 USEC
3529
3530 ;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=2 L / K5-5 BCON(1+2) H
3531
3532 001460 012700 177776 BT023: MOV #PSW,R0 ;R0 POINTS TO PSW
3533 001464 012701 000017 MOV #17,R1 ;[SOURCE]=017
3534 001470 005010 CLR (R0) ;MAKE [PSW]=000
3535 001472 000257 CCC ;SCOPE SYNC
3536
3537 001474 010120 I023: MOV R1,(R0)+ ;TEST THE MOV
3538
3539 001476 100003 BPL E1023 ;N:C = 1111 ?
3540 001500 001002 BNE E1023
3541 001502 102001 BVC E1023
3542 001504 103402 BCS A023
3543
3544 001506 000000 E1023: HALT ;MOV FAILED TO LOAD PSW
3545 001510 000763 BR BT023 ;LOCK ON HARD ERROR
3546
3547 001512 005100 A023: COM R0 ;SHOULD MAKE [R0]=177777
3548 001514 005500 ADC R0 ;SHOULD MAKE [R0]=000000
3549 001516 001402 BEQ BT024 ;BR IF IT DID
3550
3551 001520 000000 E0232: HALT ;MOV FAILED TO AUTO INC. R0
3552 001522 000756 BR BT023 ;LOCK ON HARD ERROR
3553
```

BT023 'MOV RA,(RB)+' TEST WITH [RA]=17,[RB]=177776

SEQ 0095

3554
3555
3556
3557
3558
3559
3560
3561
3562
3563
3564
3565
3566
3567
3568
3569
3570
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582
3583
3584

; *****
; .SBTTL BT024 'CMP #N,@#A' TEST WITH N=(A)
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,163,264,265,266,267,225,367,375,016] FC 1,2,3,8
;ACT BUTS: 37[004]100,142 / 35[240]120,163 / 33[266]220,225 / 16[367]016,016
;EXEC: [225]ALUC=LLHHL :[367]D=000000
;CODES: [367]SPS=3 / N:C=0100
;SYNC: B05J2 (-) T=5.2 USEC
;KEY SIG: K3-3 CMP L / K3-3 SM=2 L / K3-3 DM=3 L / K3-8 CIN00 L
; K4-4 ALLOW CLK L / K5-5 BC01 H

001524	012700	177776	BT024:	MOV	#PSW,R0	:R0 POINTS TO PSW
001530	005010			CLR	(R0)	:MAKE [PSW]=000
001532	000273			273		:MAKE N:C=1011
001534	022737	000013 177776	I024:	CMP	#13,@#PSW	:TEST THE CMP
001542	001402			BEQ	BT025	:BR IF 'Z' GOT SET
001544	000000		E024:	HALT		:CMP FAILED TO SET 'Z'
001546	000766			BR	BT024	:LOCK ON HARD ERROR

3585
3586
3587
3588
3589
3590
3591
3592
3593
3594
3595
3596
3597
3598
3599
3600
3601
3602
3603
3604
3605
3606
3607
3608
3609
3610
3611
3612

; *****
; .SBTTL BT025 'CMP #N,@#A' WITH N > (A)
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,163,264,265,266,267,225,367,375,016] FC 1,2,3,8
;ACT BUTS: 37[004]100,142 / 35[240]120,163 / 33[266]220,225 / 16[367]016,016
;EXEC: [225]ALUC=LLHHL :[367]D=177761
;CODES: [367]SPS=3 / N:C=1001
;SYNC: B05J2 (-) T=5.2 USEC
;KEY SIG: K3-3 CMP L / K3-3 SM=2 L / K3-3 DM=3 L / K3-8 CIN00 L
; K4-4 ALLOW CLK L / K5-5 BC01 H

001550 000257 BT025: CCC ;MAKE [PSW]=000
001552 022737 000017 177776 I025: CMP #17,@#PSW ;TEST THE CMP
001560 001401 BEQ E025 ;BR IF 'Z' GOT SET
001562 000402 BR BT026 ;GO TO NEXT TEST
001564 000000 E025: HALT ;CMP FAILED TO CLEAR 'Z'
001566 000770 BR BT025 ;LOCK ON HARD ERROR

3642
3643
3644
3645
3646
3647
3648
3649
3650
3651
3652
3653
3654
3655
3656
3657
3658
3659
3660
3661
3662
3663
3664
3665
3666
3667
3668
3669
3670

```
; *****  
; .SBTTL BT027 'CMP R,#N' TEST WITH [R]=N  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [162,260,267,224,367,375,016] FC 1,3,8  
;ACT BUTS: 37[004]100,162 / 33[260]220,224 / 16[367]016,016  
;EXEC: [224]ALUC=LLHHL :[367]D=000000  
;CODES: [367]SPS=3 / N:C=0100  
;SYNC: B05J2 (-) T=2.6 USEC  
;KEY SIG: K3-3 CMP L / K3-3 SM=0 L / K3-3 DM=2 L / K3-8 CIN00 L  
; K4-4 ALLOW CLK L  
BT027: MOV # -1,R0 ;MAKE [R0]=177777  
;N:C=0000  
I027: CMP R0,# -1 ;TEST THE CMP  
;BEQ BT030 ;BR IF CMP SET 'Z'  
E027: HALT ;CMP FAILED  
BR BT027 ;LOCK ON HARD ERROR
```

001610 012700 177777
001614 000257
001616 020027 177777
001622 001402
001624 000000
001626 000770

BT027 'CMP R,#N' TEST WITH [R]=N

SEQ 0099

3671
3672
3673
3674
3675
3676
3677
3678
3679
3680
3681
3682
3683
3684
3685
3686
3687
3688
3689
3690
3691
3692
3693
3694
3695
3696
3697
3698

; *****
; .SBTTL BT030 'CMP R,#N' TEST WITH [R] > N
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [162,260,267,224,367,375,016] FC 1,3,8
;ACT BUTS: 37[004]100,162 / 33[260]220,224 / 16[367]016,016
;EXEC: [224]ALUC=LLHHL :[367]D=000002
;CODES: [367]SPS=3 / N:C=0001
;SYNC: B05J2 (-) T=2.6 USEC
;KEY SIG: K3-3 CMP L / K3-3 SM=0 L / K3-3 DM=2 L / K3-8 CIN00 L
; K4-4 ALLOW CLK L

001630 012700 000001
001634 000264
001636 020027 177777
001642 001002
001644 000000
001646 000770

BT030: MOV #1,R0 ;MAKE [R0]=000001
SEZ ;SET THE 'Z' BIT
I030: CMP R0,#-1 ;TEST THE CMP
BNE BT031 ;BR IF CMP CLEARED 'Z'
E030: HALT ;CMP FAILED
BR BT030 ;LOCK ON HARD ERROR

BT030 'CMP R,#N' TEST WITH [R] > N

SEQ 0100

3699
3700
3701
3702
3703
3704
3705
3706
3707
3708
3709
3710
3711
3712
3713
3714
3715
3716
3717
3718
3719
3720
3721
3722
3723
3724
3725
3726
3727

; *****
; .SBTTL BT031 'CMP R,#N' TEST WITH [R] < N
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [162,260,267,224,367,375,016] FC 1,3,8
;ACT BUTS: 37[004]100.162 / 33[260]220,224 / 16[367]016,016
;EXEC: [224]JALUC=LLHHL :[367]D=177762
;CODES: [367]SPS=3 / N:C=1001
;SYNC: B05J2 (-) T=2.6 USEC
;KEY SIG: K3-3 CMP L / K3-3 SM=0 L / K3-3 DM=2 L / K3-8 CIN00 L
; K4-4 ALLOW CLK L

001650 012700 000001
001654 000264
001656 020027 000017
001662 001002
001664 000000
001666 000770

BT031: MOV #1,R0 ;MAKE [R0] = 000001
SEZ ;SET THE 'Z' BIT
I031: CMP R0,#17 ;TEST THE CMP
BNE BT032 ;BR IF CMP CLEARED 'Z'
E031: HALT ;CMP FAILED TO SET 'Z'
BR BT031 ;LOCK ON HARD ERROR

3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766

; *****
; .SBTTL BT032 'CMP (RA)+,RB' TEST WITH [SOURCE]=[RB]
; *****
; MICROPROGRAMMING / LOGIC INFORMATION
; ROM SEQ: [142,240,250,120,371,362,000] FC 1,2,8
; ACT BUTS: 37[004]100,142 / 35[240]120,120 / 31[120]360,362 / 27[371]016,016
; EXEC: [371]ALUC=LLHHL :[362]D=000000
; CODES: [362]SPS=3 / N:C=0100
; SYNC: B05J2 (-) T=2.5 USEC
; KEY SIG: K3-3 CMP L / K3-3 SM=2 L / K3-3 DM=0 L / K5-5 BCON(1+2) H
; K3-8 CIN00 L

001670	012700	177776		BT032: MOV	#PSW,R0	:R0 POINTS TO PSW
001674	012737	000340	177776	MOV	#340,@#PSW	:MAKE [PSW]=340
001702	012701	000340		MOV	#340,R1	:MAKE [DEST]=340
001706	000257			CCC		:N:C=0000
001710	022001			I032: CMP	(R0)+,R1	:TEST THE CMP
001712	001402			BEQ	A032	:BR IF 'Z' GOT SET
001714	000000			E1032: HALT		:CMP FAILED TO ACCESS PSW
001716	000764			BR	BT032	:LOCK ON HARD ERROR
001720	005100			A032: COM	R0	:MAKE [R0]=177777
001722	005500			ADC	R0	:MAKE [R0]=000000
001724	001402			BEQ	BT033	:BR IF 'Z' SET
001726	000000			E2032: HALT		:CMP FAILED TO AUTO INC. R0
001730	000757			BR	BT032	:LOCK ON HARD ERROR

BT032 'CMP (RA)+,RB' TEST WITH [SOURCE]=[RB]

SEQ 0102

3767
3768
3769
3770
3771
3772
3773
3774
3775
3776
3777
3778
3779
3780
3781
3782
3783
3784
3785
3786
3787
3788
3789
3790
3791
3792
3793
3794
3795
3796
3797
3798
3799
3800
3801
3802
3803

: *****
: .SBTTL BT033 'CMP (RA)+,RB' TEST WITH [SOURCE]>[RB]
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [142,240,250,120,371,362,000] FC 1,2,8
:ACT BUTS: 37[004]100,142 / 35[240]120,120 / 31[120]360,362 / 27[371]016,016
:EXEC: [371]ALUC=LLHHL :[362]D=000010
:CODES: [362]SPS=3 / N:C=0000
:SYNC: B05J2 (-) T=2.5 USEC
:KEY SIG: K3-3 CMP L / K3-3 SM=2 L / K3-3 DM=0 L / K5-5 BCON(1+2) H
: K3-8 CIN00 L

001732 012700 177776
001736 012737 000340 177776
001744 012701 000330
001750 000264
001752 022001
001754 001002
001756 000000
001760 000764
001762 005100
001764 005500
001766 001402
001770 000000
001772 000757

BT033: MOV #PSW,R0 :R0 POINTS TO PSW
MOV #340,@#PSW :MAKE [PSW]=340
MOV #330,R1 :MAKE [DEST]=330
SEZ :SET THE 'Z' BIT
I033: CMP (R0)+,R1 :TEST THE CMP
BNE A033 :BR IF 'Z' GOT CLEARED
E1033: HALT :CMP FAILED TO ACCESS PSW
BR BT033 :LOCK ON HARD ERROR
A033: COM R0 :MAKE [R0]=177777
ADC R0 :MAKE [R0]=000000
BEQ BT034 :BR IF 'Z' SET
E2033: HALT :CMP FAILED TO AUTO INC. R0
BR BT033 :LOCK ON HARD ERROR

3804 ; *****
3805 ; .SBTTL BT034 'CMP (RA)+,RB' TEST WITH [SOURCE]<[RB]
3806 ; *****

3807
3808 ;MICROPROGRAMMING / LOGIC INFORMATION

3809
3810 ;ROM SEQ: [142,240,250,120,371,362,000] FC 1,2,8
3811
3812 ;ACT BUTS: 37[004]100,142 / 35[240]120,120 / 31[120]360,362 / 27[371]016,016
3813
3814 ;EXEC: [371]ALUC=LLHHL :[362]D=177770
3815
3816 ;CODES: [362]SPS=3 / N:C=1001
3817
3818 ;SYNC: B05J2 (-) T=2.5 USEC
3819
3820 ;KEY SIG: K3-3 CMP L / K3-3 SM=2 L / K3-3 DM=0 L / K5-5 BCON(1+2) H
3821 ; K3-8 CIN00 L
3822

3823	001774	012700	177776		BT034:	MOV	#PSW,R0	:R0 POINTS TO PSW
3824	002000	012737	000330	177776		MOV	#330,@#PSW	:MAKE [PSW]=330
3825	002006	012701	000340			MOV	#340,R1	:MAKE [DEST]=340
3826	002012	000264				SEZ		:SET THE 'Z' BIT
3827								
3828	002014	022001			I034:	CMP	(R0)+,R1	:TEST THE CMP
3829								
3830	002016	001002				BNE	A034	:BR IF 'Z' GOT CLEARED
3831								
3832	002020	000000			E1034:	HALT		:CMP FAILED TO ACCESS PSW
3833	002022	000764				BR	BT034	:LOCK ON HARD ERROR
3834								
3835	002024	005100			A034:	COM	R0	:MAKE [R0]=177777
3836	002026	005500				ADC	R0	:MAKE [R0]=000000
3837	002030	001402				BEQ	BT035	:BR IF 'Z' SET
3838								
3839	002032	000000			E2034:	HALT		:CMP FAILED TO AUTO INC. R0
3840	002034	000757				BR	BT034	:LOCK ON HARD ERROR

BT034 'CMP (RA)+,RB' TEST WITH [SOURCE]<[RB]

SEQ 0104

3841
3842
3843
3844
3845
3846
3847
3848
3849
3850
3851
3852
3853
3854
3855
3856
3857
3858
3859
3860
3861
3862
3863
3864
3865
3866
3867
3868
3869

: *****
: .SBTTL BT035 'CMP RA,RB' TEST WITH [RA] = [RB]
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [102,364,362,001] FC 1,8
:ACT BUTS: 37[004]100,102 / 31[102]360,362 / 27[364]000,001
:EXEC: [364]ALUC=LLHHL :[362]D=000000
:CODES: [362]SPS=3 / N:C=0100
:SYNC: B05J2 (-) T=1 USEC
:KEY SIG: K3-3 CMP L / K3-3 SM=0 L / K3-3 DM=0 L / K3-4 OVLAP INSTR H
: K3-8 CIN00 L

002036 012700 125252
002042 010001
002044 000257
002046 020100
002050 001402
002052 000000
002054 000770

BT035: MOV #125252,R0 ;MAKE [R0] = 125252
(MOV R0,R1 ;MAKE [R1] = 125252
CCC ;SCOPE SYNC
I035: CMP R1,R0 ;TEST THE CMP
BEQ BT036 ;BR IF 'Z' GOT SET
E035: HALT ;ERROR - CMP FAILED TO SET 'Z'
BR BT035 ;LOCK ON HARD ERROR

BT035 'CMP RA,RB' TEST WITH [RA] = [RB]

; *****
; .SBTTL BT036 'CMP RA,RB' TEST WITH [RA] < [RB]
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [102,364,362,001] FC 1,8

;ACT BUTS: 37[004]100,102 / 31[102]360,362 / 27[364]000,001

;EXEC: [364]ALUC=LLHML :[362]D=152526

;CODES: [362]SPS=3 / N:C=1001

;SYNC: B05J2 (-) T=1 USEC

;KEY SIG: K3-3 CMP L / K3-3 SM=0 L / K3-3 DM=0 L / K3-4 OVLAP INSTR H
; K3-8 CIN00 L

3870
3871
3872
3873
3874
3875
3876
3877
3878
3879
3880
3881
3882
3883
3884
3885
3886
3887
3888
3889 002056 012700 025252
3890 002062 005001
3891 002064 000264
3892
3893 002066 020100
3894
3895 002070 001002
3896
3897 002072 000000
3898 002074 000770

BT036: MOV #25252,R0 ;MAKE [R0] = 25252
CLR R1 ;MAKE [R1] = 000000
SEZ ;SCOPE SYNC - SET 'Z'

I036: CMP R1,R0 ;TEST THE CMP

BNE BT037 ;BR IF 'Z' GOT CLEARED

E036: HALT ;ERROR - CMP FAILED TO SET 'Z'
BR BT036 ;LOCK ON HARD ERROR

BT036 'CMP RA,RB' TEST WITH [RA] < [RB]

SEQ 0106

3899
3900
3901
3902
3903
3904
3905
3906
3907
3908
3909
3910
3911
3912
3913
3914
3915
3916
3917
3918
3919
3920
3921
3922
3923
3924
3925
3926
3927
3928

```
; *****  
; .SBTTL BT037 'CMP RA,RB' TEST WITH [RA] > [RB]  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ:      [102,364,362,001] FC 1,8  
;ACT BUTS:     37[004]100,102 / 31[102]360,362 / 27[364]000,001  
;EXEC:        [364]ALUC=LLHHL :[362]D=000017  
;CODES:       [362]SPS=3      /      N:C=0000  
;SYNC:        B05J2  (-)      T=1 USEC  
;KEY SIG:     K3-3 CMP L / K3-3 SM=0 L / K3-3 DM=0 L / K3-4 OVLAP INSTR H  
;             ; K3-8 CIN00 L  
  
BT037:  CLR    R0           ;MAKE [R0] = 000000  
        MOV    #17,R1      ;MAKE [R1] = 000017  
        SEZ           ;SCOPE SYNC - SET 'Z'  
  
I037:   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
```

```
002076 005000  
002100 012701 000017  
002104 000264  
  
002106 020100  
  
002110 001002  
  
002112 000000  
002114 000770
```

BT037 'CMP RA,RB' TEST WITH [RA] > [RB]

SEQ 0107

3929
3930
3931
3932
3933
3934
3935
3936
3937
3938
3939
3940
3941
3942
3943
3944
3945
3946
3947
3948
3949
3950
3951
3952
3953
3954
3955
3956
3957
3958

; *****
; .SBTTL BT040 'MOV (RA),RB' TEST WITH [SOURCE]=[RB]=17
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,160,204,000] FC 1,2,4,8
;ACT BUTS: 37[004]100,141 / 35[247]120,160 / 20[160]000,000
;EXEC: [160]ALUC=LLLLL :[204]D=000017
;CODES: [204]SPS=3 / N:C=0001
;SYNC: B05J2 (-) T=2.3 USEC
;KEY SIG: K3-3 MOV L / K3-3 SM=1 L / K3-3 DM=0 L

002116 012700 177776
002122 005010
002124 005001
002126 000277
002130 011001
002132 020127 000017
002136 001402
002140 000000
002142 000765

BT040: MOV #PSW,R0 ;R0 POINTS TO PSW
CLR (R0) ;MAKE [PSW]=000
CLR R1 ;MAKE [R1]=000000
SCC ;MAKE N:C=1111
I040: 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

```
3959 ; *****  
3960 ; .SBTTL BT041 'MOV (RA)+,RB' TEST WITH [SOURCE]=[RB]=17  
3961 ; *****  
3962  
3963 ;MICROPROGRAMMING / LOGIC INFORMATION  
3964  
3965 ;ROM SEQ: [141,247,250,160,204,000] FC 1,2,4,8  
3966  
3967 ;ACT BUTS: 37[004]100,141 / 35[247]120,160 / 20[160]000,000  
3968  
3969 ;EXEC: [160]ALUC=L L L L L : [204]D=000017  
3970  
3971 ;CODES: [204]SPS=3 / N:C=0001  
3972  
3973 ;SYNC: B05J2 (-) T=2.3 USEC  
3974  
3975 ;KEY SIG: K3-3 MOV L / K3-3 SM=1 L / K3-3 DM=0 L  
3976  
3977 002144 012700 177776 BT041: MOV #PSW,R0 ;R0 POINTS TO PSW  
3978 002150 005010 CLR (R0) ;MAKE [PSW]=000  
3979 002152 005001 CLR R1 ;MAKE [R1]=000000  
3980 002154 000277 SCC ;MAKE N:C=1111  
3981  
3982 002156 012001 I041: MOV (R0)+,R1 ;TEST THE MOV  
3983  
3984 002160 020127 000017 CMP R1,#17 ;DID R1 GET LOADED WITH 000017 ?  
3985 002164 001402 BEQ A041 ;BR IF YES  
3986  
3987 002166 000000 E1041: HALT ;MOV FAILED TO LOAD R1  
3988 002170 000765 BR BT041 ;LOCK ON HARD ERROR  
3989  
3990 002172 005100 A041: COM R0 ;[R0] SHOULD GO TO 177777  
3991 002174 005500 ADC R0 ;[R0] SHOULD GO TO 000000  
3992 002176 001402 BEQ BT042 ;BR IF 'Z' GOT SET  
3993  
3994 002200 000000 E2041: HALT ;MOV FAILED TO AUTO INC. R0  
3995 002202 000760 BR BT041 ;LOCK ON HARD ERROR  
3996
```

BT041 'MOV (RA)+,RB' TEST WITH [SOURCE]=[RB]=17

SEQ 0109

3997
3998
3999
4000
4001
4002
4003
4004
4005
4006
4007
4008
4009
4010
4011
4012
4013
4014
4015
4016
4017
4018
4019
4020
4021
4022
4023
4024
4025
4026
4027

; *****
; .SBTTL BT042 'XOR RA,RB' TEST WITH [RA] = [RB] = 000000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [102,364,360,001] FC 1,7,8
;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001
;EXEC: [364]ALUC=HLHHL :[360]D=000000
;CODES: [360]SPS=3 / N:C=0100
;SYNC: B05J2 (-) T=1 USEC
;KEY SIG: K3-3 DM=0 L / K3-3 IR(08:06)=1 L / K3-5 XOR L / K3-4 OVLAP INSTR H

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

4028
4029
4030
4031
4032
4033
4034
4035
4036
4037
4038
4039
4040
4041
4042
4043
4044
4045
4046
4047
4048
4049
4050
4051
4052
4053
4054
4055
4056
4057
4058

002224 005000
002226 005100
002230 010001
002232 000257

002234 074100

002236 005700
002240 001402

002242 000000
002244 000767

```
; *****  
; .SBTTL BT043 'XOR RA,RB' TEST WITH [RA] = [RB] = 177777  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ:      [102,364,360,001] FC 1,7,8  
  
;ACT BUTS:     37[004]100,102 / 31[102]360,360 / 27[364]000,001  
  
;EXEC:        [364]ALUC=HLHHL :[360]D=000000  
  
;CODES:       [360]SPS=3      /      N:C=0100  
  
;SYNC:        B05J2  (-)      T=1 USEC  
  
;KEY SIG:     K3-3 DM=0 L / K3-3 IR(08:06)=1 L / K3-5 XOR L / K3-4 OVLAP INSTR H  
  
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  
        BR   BT043        ;XOR FAILED  
                          ;LOCK ON HARD ERROR
```

BT043 'XOR RA,RB' TEST WITH [RA] = [RB] = 177777

SEQ 0111

4059
4060
4061
4062
4063
4064
4065
4066
4067
4068
4069
4070
4071
4072
4073
4074
4075
4076
4077 002246 012701 125252
4078 002252 012700 052525
4079 002256 000257
4080
4081 002260 074100
4082
4083 002262 020027 177777
4084 002266 001402
4085
4086 002270 000000
4087 002272 000400

```
; *****  
; .SBTTL BT044 'XOR RA,RB' TEST WITH [RB]=052525,[RA]=125252  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [102,364,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001  
;EXEC: [364]ALUC=HLHHL :[360]D=177777  
;CODES: [360]SPS=3 / N:C=1000  
;SYNC: B05J2 (-) T=1 USEC  
;KEY SIG: K3-3 DM=0 L / K3-3 IR(08:06)=1 L / K3-5 XOR L / K3-4 OVLAP INSTR H  
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
```

4088
4089
4090
4091
4092
4093
4094
4095
4096
4097
4098
4099
4100
4101
4102
4103
4104
4105
4106 002274 012700 125252
4107 002300 012701 052525
4108 002304 000257
4109
4110 002306 074100
4111
4112 002310 020027 177777
4113 002314 001402
4114
4115 002316 000000
4116 002320 000765
4117

```
; *****  
; .SBTTL BT045 'XOR RA,RB' TEST WITH [RA]=052525,[RB]=125252  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [102,364,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001  
;EXEC: [364]ALUC=HLHHL :[360]D=177777  
;CODES: [360]SPS=3 / N:C=1000  
;SYNC: B05J2 (-) T=1 USEC  
;KEY SIG: K3-3 DM=0 L / K3-3 IR(08:06)=1 L / K3-5 XOR L / K3-4 OVLAP INSTR H  
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 ?  
BEQ BT046 ;BR IF YES  
E045: HALT ;XOR FAILED  
BR BT045 ;LOCK ON HARD ERROR
```

```
4118 ; *****  
4119 ; .SBTTL BT046 GPR ADDRESS INTERACTION TEST  
4120 ; *****  
4121  
4122 002322 012700 125252 BT046: MOV #125252,R0 ;[R0] = 125252  
4123 002326 010001 MOV R0,R1  
4124 002330 005101 COM R1 ;[R1] = 052525  
4125 002332 010102 MOV R1,R2  
4126 002334 005102 COM R2 ;[R2] = 125252  
4127 002336 010203 MOV R2,R3  
4128 002340 005103 COM R3 ;[R3] = 052525  
4129 002342 010304 MOV R3,R4  
4130 002344 005104 COM R4 ;[R4] = 125252  
4131 002346 010405 MOV R4,R5  
4132 002350 005105 COM R5 ;[R5] = 052525  
4133  
4134 002352 074100 I046: XOR R1,R0 ;[R0] S/B = 177777  
4135 002354 074200 XOR R2,R0 ;[R0] S/B = 125252  
4136 002356 074300 XOR R3,R0 ;[R0] S/B = 177777  
4137 002360 074400 XOR R4,R0 ;[R0] S/B = 125252  
4138 002362 074500 XOR R5,R0 ;[R0] S/B = 177777  
4139 002364 005100 COM R0 ;[R0] S/B = 000000  
4140  
4141 002366 001402 BEQ A046 ;BR IF [R0] WAS 000000  
4142  
4143 002370 000000 E1046: HALT ;GPR ADDRESSING PROBLEM  
4144 002372 000753 BR BT046 ;LOCK ON HARD ERROR  
4145  
4146 002374 020627 001000 A046: CMP SP,#STACKL ;DID R6 GET DISTURBED  
4147 002400 001577 BEQ INIT ;BR IF NOT  
4148  
4149 002402 000000 E2046: HALT ;R6 ADDRESS PROBLEM  
4150 002404 000746 BR BT046 ;LOCK ON HARD ERROR  
4151  
4152 003000 .=3000
```

```
4153 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4154 ; / / / / / / BASIC INSTRUCTION TESTS / / / / / /
4155 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
4156
4157 003000 012706 001000 INIT: MOV #BT001,SP ;SET UP THE STACK POINTER
4158 003004 012737 000001 066662 MOV #1,@#ICOUNT ;NO ITERATIONS ON INITIAL PASS
4159 003012 012737 000001 066664 MOV #1,@#ITCNT
4160 003020 012701 066666 MOV #ERRCNT,R1 ;SET UP TO INIT. COUNTERS AND FLAGS
4161 003024 005021 1$: CLR (R1)+ ;CLEAR ONE WORD
4162 003026 020127 066724 CMP R1,#ONCE+2 ;CLEARED ALL FLAGS AND COUNTERS?
4163 003032 001374 BNE 1$ ;BR IF NOT
4164 003034 012706 001000 BEGIN: MOV #BT001,SP ;SET UP THE STACK POINTER
4165
4166 ; *****
4167 ; .SBTTL T0001 BASIC 'BNE' TEST WITH Z=0
4168 ; *****
4169
4170 ;MICROPROGRAMMING / LOGIC INFORMATION
4171
4172 ;ROM SEQ: [111,340,341,016] FC 1,7
4173
4174 ;ACT BUTS: 37[004]100,111 / 16[340]016,016
4175
4176 ;EXEC: [341]ALUC = LHLLH :[016] D = #T0002
4177
4178 ;CODES: N / A
4179
4180 ;SYNC: B05J2 (-) / T= 1.8 USEC
4181
4182 ;KEY SIG: K5-3 FALSE BR L / K5-3 BR INSTR L
4183
4184 003040 012700 000001 T0001: MOV #0001,R0 ;LOAD R0 WITH TEST NO.
4185 003044 000257 R0001: CCC ;MAKE Z=0
4186
4187 003046 001002 I0001: BNE T0002 ;TEST THE BNE - IT SHOULD BR
4188
4189 003050 000000 E0001: HALT ;BNE FAILED TO LOAD PC
4190 003052 000774 BR R0001 ;LOCK ON HARD ERROR
4191
```

```
4192 ; *****  
4193 ; .SBTTL T0002 BASIC 'BNE' TEST WITH Z=1  
4194 ; *****  
4195  
4196 ;MICROPROGRAMMING / LOGIC INFORMATION  
4197  
4198 ;ROM SEQ: [110,347,016]FC 1,7  
4199  
4200 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
4201  
4202 ;EXEC: NO-OP=[016] D= #I0002  
4203  
4204 ;CODES: N / A  
4205  
4206 ;SYNC: B05J2 (-) / T= 1.4 USEC  
4207  
4208 ;KEY SIG: K5-3 BR INSTR L  
4209  
4210 003054 012700 000002 T0002: MOV #0002,R0 ;LOAD R0 WITH TEST NO.  
4211 003060 000264 R0002: SEZ ;SET THE 'Z' BIT  
4212  
4213 003062 001001 I0002: BNE E0002 ;TEST THE BNE - IT SHOULD NOT BR  
4214  
4215 003064 000402 BR T0003 ;GO TO NEXT TEST  
4216  
4217 003066 000000 E0002: HALT ;BNE BRANCHED WITH Z=1  
4218 003070 000773 BR R0002 ;LOCK ON HARD ERROR  
4219
```

```

4220 ; *****
4221 ; .SBTTL T0003 BASIC 'BEQ' TEST WITH Z=1
4222 ; *****
4223
4224 ;MICROPROGRAMMING / LOGIC INFORMATION
4225
4226 ;ROM SEQ: [111,340,341,016] FC 1,7
4227
4228 ;ACT BUTS: 37[004]100,111 / 16[340]16,16
4229
4230 ;EXEC: [341]ALUC=LHLLH :[016]D=#T0004
4231
4232 ;CODES: N / A
4233
4234 ;SYNC: B05J2 (-) / T= 1.8 USEC
4235
4236 ;KEY SIG: K5-3 FALSE BRL / K5-3 BR INSTRL
4237
4238 003072 012700 000003 T0003: MOV #0003,R0 ;LOAD R0 WITH THE TEST NO.
4239 003076 000264 R0003: SEZ ;MAKE Z=1
4240
4241 003100 001402 I0003: BEQ T0004 ;TEST THE BEQ - IT SHOULD BR
4242
4243 003102 000000 E0003: HALT ;BEQ FAILED TO LOAD THE PC
4244 003104 000774 BR R0003 ;LOCK ON HARD ERROR

```

4245
4246
4247
4248
4249
4250
4251
4252
4253
4254
4255
4256
4257
4258
4259
4260
4261
4262
4263 003106 012700 000004
4264 003112 000257
4265
4266 003114 001401
4267
4268 003116 000402
4269
4270 003120 000000
4271 003122 000773
4272

```
; *****  
; .SBTTL T0004 BASIC 'BEQ' TEST WITH Z=0  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [110,347,016] FC 1,7  
  
;ACT BUTS: 37[004]100,110 / 16[110]016,016  
  
;EXEC: NO-OP=[016] D= #I0004  
  
;CODES: N / A  
  
;SYNC: B05J2 (-) / T= 1.4 USEC  
  
;KEY SIG: K5-3 BR INSTR L  
  
T0004: MOV #0004,R0 ;LOAD R0 WITH THE TEST NO.  
R0004: CCC ;MAKE Z=0  
  
I0004: BEQ E0004 ;TEST THE BEQ - IT SHOULD NOT BR  
BR T0005 ;GO TO NEXT TEST  
  
E0004: HALT ;BEQ BRANCHED WITH Z=0  
BR R0004 ;LOCK ON HARD ERROR
```

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

003124 012700 000005
003130 005037 1777.76
003134 000270
003136 100001
003140 000402
003142 000000
003144 000771

```
; *****  
; .SBTTL T0005 BASIC 'BPL' TEST WITH N=1  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [110,347,016] FC 1,7  
;ACT BUTS: 37[004]100,110 / 16[110]16,16  
;EXEC: NO-OP=[016] D=#I0005  
;CODES: N / A  
;SYNC: B05J2 (-) / T= 1.4 USEC  
;KEY SIG: K5-3 BR INSTR L  
T0005: MOV #0005,RO ;LOAD RO WITH TEST NO.  
R0005: CLR @#PSW ;CLEAR THE PSW  
;MAKE N=1  
SEN  
I0005: BPL E0005 ;TEST THE BPL - IT SHOULDN'T BR  
BR T0006 ;GO TO NEXT TEST  
E0005: HALT ;BPL BRANCHED WITH N=1  
BR R0005 ;LOCK ON HARD ERROR
```

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

; *****
; .SBTTL T0006 BASIC 'BPL' TEST WITH N=0
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [111,340,341,016] FC 1,7
;ACT BUTS: 37[004]100,111 / 16[340]16,16
;EXEC: [341]ALUC LHLLH :[016]D=#T0007
;CODES: N / A
;SYNC: B05J2 (-) / T= 1.8 USEC
;KEY SIG: K5-3 BR INSTR L / K5-3 FALSE BR L

003146 012700 000006
003152 005037 177776
003156 000257
003160 100002
003162 000000
003164 000772

T0006: MOV #0006,R0 ;LOAD R0 WITH TEST NO.
R0006: CLR @#PSW ;CLEAR THE PSW
CCC ;SCOPE SYNC
I0006: BPL T0007 ;TEST THE BPL - IT SHOULD BR
E0006: HALT ;BPL FAILED TO LOAD THE PC
BR R0006 ;LOCK ON HARD ERROR

4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341
4342
4343
4344
4345
4346
4347 003166 012700 000007
4348 003172 012702 177703
4349 003176 012705 177776
4350 003202 012704 000017
4351 003206 005015
4352 003210 005003
4353 003212 000277
4354
4355 003214 011503
4356
4357 003216 020403
4358 003220 001402
4359
4360 003222 000000
4361 003224 000770

```
; *****  
; .SBTTL T0007 BASIC 'MOV (RA),RB' TEST - (RA)=177776  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [141,247,250,160,204,000] FC 1,2,4,8  
  
;ACT BUTS: 37[004]100,141 / 35[247]120,160 / 20[160]000,000  
  
;EXEC: [160]ALUC=LLLLL :[204]D=000017  
  
;CODES: [204]SPS=3 / N:C=0000  
  
;SYNC: B05J2 (-) T=2.25 USEC  
  
;KEY SIG: K3-3 MOV L / K3-3 SM=1 L / K3-3 DM=0 L  
  
T0007: MOV #0007,R0 ;LOAD R0 WITH TEST NO.  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #PSW,R5 ;SOURCE ADDR = 177776  
MOV #17,R4 ;RESULT S / B=000017  
R0007: CLR (R5) ;MAKE [PSW]=000  
CLR R3 ;[DEST] = 000000  
SCC ;MAKE [PSW]=017  
  
I0007: MOV (R5),R3 ;TEST THE MOV  
  
CMP R4,R3 ;CORRECT RESULT ?  
BEQ T0010 ;BR IF YES  
  
E0007: HALT ;ERROR-MOV FAILED  
BR R0007 ;LOCK ON HARD ERROR
```

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

```
; *****  
; .SBTTL T0010 BASIC 'CMP RA,(RB)' TEST - [RA] = [DEST]  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016  
;EXEC: [224]ALUC=LLHHL :[367]D=000000  
;CODES: [367]SPS=3 / N:C=0100  
;SYNC: B05J2 (-) T=2.6 USEC  
;KEY SIG: K3-3 CMP L / K3-3 SM=0 L / K3-3 DM=1 L / K4-4 ALLOW CLK L  
; K3-8 CIN00 L
```

003226 012700 000010
003232 012702 067560
003236 012704 125252
003242 012737 125252 067560
003250 000257
003252 020412
003254 001403
003256 011203
003260 000000
003262 000767

```
T0010: MOV #0010,R0 ;LOAD R0 WITH TEST NO.  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #125252,R4 ;RESULT S / B = 125252  
R0010: MOV #125252,@MBUF0 ;MAKE [DEST] = 125252  
CCC ;MAKE N:C=0000  
I0010: CMP R4,(R2) ;TEST THE CMP  
BEQ T0011 ;BR IF 'Z' GOT SET  
E0010: MOV (R2),R3 ;GET THE WAS DATA  
HALT ;ERROR - CMP FAILED TO SET 'Z'  
BR R0010 ;LOCK OF HARD ERROR
```

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

```
; *****  
; .SBTTL T0011 BASIC 'CMP RA,(RB)' TEST - [RA] NOT EQUAL TO [DEST]  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8  
  
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016  
  
;EXEC: [224]ALUC=LLHML :[367]D=177777  
  
;CODES: [367]SPS=3 / N:C=1001  
  
;SYNC: B05J2 (-) T=2.6 USEC  
  
;KEY SIG: K3-3 CMP L / K3-3 SM=0 L / K3-3 DM=1 L / K4-4 ALLOW CLK L  
; K3-8 CIN00 L
```

```
003264 012700 000011 T0011: MOV #0011,R0 ;LOAD R0 WITH TEST NO.  
003270 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
003274 012704 000001 MOV #1,R4 ;RESULT S / B = 000001  
003300 012737 000000 067560 R0011: MOV #0,@MBUF0 ;MAKE [DEST] = 000000  
003306 000264 SEZ ;MAKE N:C=0100  
  
003310 020412 I0011: CMP R4,(R2) ;TEST THE CMP  
  
003312 001003 BNE T0012 ;BR IF 'Z' GOT CLEARED  
  
003314 011203 MOV (R2),R3 ;GET THE WAS DATA  
003316 000000 E0011: HALT ;ERROR - CMP FAILED TO CLR 'Z'  
003320 000767 BR R0011 ;LOCK ON HARD ERROR
```

4428
4429
4430
4431
4432
4433
4434
4435
4436
4437
4438
4439
4440
4441
4442
4443
4444
4445
4446 003322 012700 000012
4447 003326 012704 125252
4448 003332 012702 177703
4449 003336 010403
4450 003340 000257
4451
4452 003342 022703 125252
4453
4454 003346 001402
4455
4456 003350 000000
4457 003352 000771
4458
4459
4460 003354 020403
4461 003356 001402
4462
4463 003360 000000
4464 003362 000765
4465

; *****
; .SBTTL T0012 BASIC "CMP #N,R" TEST - N = [R]
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,120,371,362,000] FC 1,2,8
;ACT BUTS: 37[004]100,142 / 35[240]120,120 / 27[371]000,000
;EXEC: [371]ALUC=LLHHL :[362]D=125252
;CODES: [362] SPS=3 / N:C=
;SYNC: B05J2 (-) / T= 2.5 USEC
;KEY SIG: K3-3 CMPL / K3-3 DM=0L / K3-6 BYTE INSTR H

T0012: MOV #0012,R0 ;LOAD R0 WITH TEST NO.
MOV #125252,R4 ;RESULT S / B = 125252
MOV #177703,R2 ;DEST ADDR = R3
R0012: MOV R4,R3 ;[DEST] = 125252
CCC ;SCOPE SYNC
I0012: CMP #125252,R3 ;TEST THE CMP
BEQ A0012 ;BR IF N = [R]
E10012: HALT ;CMP FAILED
BR R0012 ;LOCK ON HARD ERROR
A0012: CMP R4,R3 ;DID CMP ALTER [DEST]?
BEQ T0013 ;BR IF NO
E20012: HALT ;CMP DELIVERED A RESULT
BR R0012 ;LOCK ON HARD ERROR

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

```
; *****  
; .SBTTL T0013 BASIC 'CMP #N,R' TEST - N NOT EQUAL TO [R]  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [142,240,250,120,371,362,000] FC 1,2,8  
;ACT BUTS: 37[004]100,142 / 35[240]120,120 / 27[371]000,000  
;EXEC: [371]ALUC=LLHHL :[362]D=177777  
;CODES: [362] SPS=3 / N:C=1001  
;SYNC: B05J2 (-) T=2.5 USEC  
;KEY SIG: K3-3 CMPL / K3-3 DM=0L / K3-6 BYTE INSTR H / K3-8 CIN00 L  
T0013: MOV #0013,R0 ;LOAD R0 WITH TEST NO.  
CLR R4 ;RESULT S / B = 000000  
MOV #177703,R2 ;DEST ADDR = R3  
R0013: MOV R4,R3 ;[DEST] = 125252  
SEZ ;SCOPE SYNC  
I0013: CMP #1,R3 ;TEST THE CMP  
BNE A0013 ;BR IF N NOT EQUAL TO [R]  
E10013: HALT ;CMP FAILED  
BR R0013 ;LOCK ON HARD ERROR  
A0013: CMP R4,R3 ;DID CMP ALTER [DEST]?  
BEQ T0014 ;BR IF NO  
E20013: HALT ;CMP DELIVERED A RESULT  
BR R0013 ;LOCK ON HARD ERROR
```

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

003424 012700 000014
003430 012702 067560
003434 012704 177777
003440 005012
003442 000257
003444 010412
003446 020412
003450 001403
003452 011203
003454 000000
003456 000770

```
; *****  
; .SBTTL T0014 BASIC 'MOV RA,(R2)' TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [171,257,201,125,375,016] FC 1,4,8  
;ACT BUTS: 37[004]100,171 / 22[171]200,201 / 16[125]016,016  
;EXEC: [201]ALUC=LLLLL :[125]D=177777  
;CODES: [125]SPS=3 / N:C=1000  
;SYNC: B05J2 (-) T=2.42 USEC  
;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=1 L  
T0014: MOV #0014,R0 ;LOAD R0 WITH TEST NO.  
MOV #MBUF0,R2 ;DEST ADDR=MBUF0  
MOV #-1,R4 ;RESULT S / B = 177777  
R0014: CLR (R2) ;MAKE [DEST] = 000000  
CCC ;SCOPE SYNC - N:C=0000  
I0014: MOV R4,(R2) ;TEST THE MOV  
CMP R4,(R2) ;RESULT CORRECT ?  
BEQ T0015 ;BR IF YES  
E0014: MOV (R2),R3 ;GET THE WAS DATA  
HALT ;ERROR - MOV FAILED  
BR R0014 ;LOCK ON HARD ERROR
```

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

; *****
; .SBTTL T0015 BASIC 'MOV #N,(R)'' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ [142,240,250,171,257,200,125,375,016] FC 1,2,4,8

;ACT BUTS: 37[004]100,142 / 35[240]120,171 / 22[171]200,200 / 16[125]016,016

;EXEC: [200]ALUC=LLLLL :[125]D=177777

;CODES: [125]SPS=3 / N:C=1000

;SYNC: B05J2 (-) T=3.26 USEC

;KEY SIG: K3-3 MOV L / K3-3 SM=2 L / K3-3 DM=1 L / K5-5 BCON(1+2) H

003460 012700 000015
003464 012702 067560
003470 012704 177777
003474 005012
003476 000257

003500 012712 177777

003504 020412
003506 001403

003510 011203
003512 000000
003514 000767

T0015: MOV #0015,R0 ;LOAD R0 WITH TEST NO.
MOV #M00F0,R2 ;DEST ADDR = M00F0
R0015: MOV #-1,R4 ;RESULT S / B = 177777
CLR (R2) ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC

I0015: MOV #-1,(R2) ;TEST THE MOV

CMP R4,(R2) ;RESULT OK ?
BEQ T0016 ;BR IF YES

E0015: MOV (R2),R3 ;GET THE WAS DATA
HALT ;ERROR - MOV FAILED
BR R0015 ;LOCK ON HARD ERROR

```
4571 ; *****  
4572 ; .SBTTL T0016 BASIC 'MOVB #N,X(R)'' TEST - DEST EVEN  
4573 ; *****  
4574  
4575 ;MICROPROGRAMMING / LOGIC INFORMATION  
4576  
4577 ;ROM SEQ: [142,240,250,177,206,212,202,205,125,375,016] FC 1,2,4  
4578  
4579 ;ACT BUTS: 37[004]100,142 / 35[240]120,177 / 17[177]212,212 / 21[206]200,202  
4580 ; / 16[125]016,016  
4581  
4582 ;EXEC: [205]ALUC=LLLLL :[125]D=001001  
4583  
4584 ;CODES: [125] SPS=3 / N:C=0000  
4585  
4586 ;SYNC: B05J2 (-) / T= 4.2 USEC  
4587  
4588 ;KEY SIG: K3-3 MOVL / K3-3 DM=6L / K3-5 DOPL / K3-6 BYTE INSTR H  
4589  
4590 003516 012700 000016 T0016: MOV #0016,R0 ;LOAD R0 WITH TEST NO.  
4591 003522 012704 177401 MOV #177401,R4 ;RESULT S / B = 177401  
4592 003526 012702 067564 MOV #MBUF1,R2 ;DEST ADDR = MBUF1  
4593 003532 012705 067560 MOV #MBUFO,R5 ;BASE DEST ADDR = MBUFO  
4594 003536 012712 177777 R0016: MOV #-1,(R2) ;[DEST] = 177777  
4595 003542 000257 CCC ;SCOPE SYNC  
4596  
4597 003544 112765 000001 000004 I0016: MOVB #1,4(R5) ;TEST THE MOVB  
4598  
4599 003552 020412 CMP R4,(R2) ;RESULT OK?  
4600 003554 001403 BEQ T0017 ;BR IF YES  
4601  
4602 003556 011203 E0016: MOV (R2),R3 ;GET WAS DATA  
4603 003560 000000 HALT ;MOVB DELIVERED WRONG RESULT  
4604 003562 000765 BR R0016 ;LOCK ON HARD ERROR  
4605
```

```

4606 ; *****
4607 ; .SBTTL T0017 BASIC 'MOVB #N,X(R)'' TEST - DEST ODD
4608 ; *****
4609 ;MICROPROGRAMMING / LOGIC INFORMATION
4610 ;ROM SEQ: [142,240,250,177,206,212,202,205,125,375,016] FC 1,2,4
4611 ;ACT BUTS: 37[004]100,142 / 35[240]120,177 / 17[177]212,212 / 21[206]200,202
4612 ; / 16[125]016,016
4613 ;EXEC: [205]ALUC=LLLLL :[125]D=001001
4614 ;CODES: [125] SPS=3 / N:C=0000
4615 ;SYNC: B05J2 (-) / T= 4.2 USEC
4616 ;KEY SIG: K3-3 MOVL / K3-3 DM=6L / K3-5 DOPL / K3-6 BYTE INSTR H
4617
4618 T0017: MOV #0017,R0 ;LOAD R0 WITH TEST NO.
4619 MOV #777,R4 ;RESULT S / B = 777
4620 MOV #MBUF1,R2 ;DEST ADDR = MBUF1
4621 MOV #MBUF0,R5 ;BASE DEST ADDR = MBUF0
4622 R0017: MOV #-1,(R2) ;[DEST] = 177777
4623 CCC ;SCOPE SYNC
4624
4625 003564 012700 000017 I0017: MOVB #1,5(R5) ;TEST THE MOVB
4626 003570 012704 000777 ;RESULT OK?
4627 003574 012702 067564 BEQ CLMT ;BR IF YES
4628 003600 012705 067560
4629 003604 012712 177777
4630 003610 000257
4631
4632 003612 112765 000001 000005 E0017: MOV (R2),R3 ;GET WAS DATA
4633 ;MOVB DELIVERED WRONG RESULT
4634 003620 020412 ;LOCK ON HARD ERROR
4635 003622 001403
4636
4637 003624 011203 ;THIS ROUTINE CLEARS THE 512 BYTE MISSED TEST STATUS TABLE
4638 003626 000000
4639 003630 000765
4640
4641
4642
4643 003632 012701 070140 CLMT: MOV #STAB1,R1 ;R1 POINTS TO BEGINNING OF TABLE
4644 003636 012702 071162 MOV #STAB2,R2 ;R2 POINTS TO END OF TABLE
4645 003642 012721 000000 MT: MOV #0,(R1)+ ;CLEAR ONE WORD
4646 003646 020102 CMP R1,R2 ;AT END OF TABLE ?
4647 003650 001374 BNE MT ;BR IF NOT AT END
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

; *****
; .SBTTL T0020 BASIC 'TST @WA' TEST WITH [A]>0
; *****

; MICROPROGRAMMING / LOGIC INFORMATION

; ROM SEQ: [163,264,265,266,267,220,211,367,376,016] FC 1,3,9,8

; ACT BUTS: 37[004]100,163 / 33[266]220,220 / 16[367]016,016

; EXEC: [220]ALUC LLLL :[211]D=377

; CODES: [367]SPS=3 / N:C=0000

; SYNC: B05J2 (-) / T= 2.8 USEC

; KEY SIG: K3-4 TSTL / K3-3 DM=3L

003652 012700 000020
003656 112760 000377
003664 012703 066704
003670 010013
003672 012702 067560
003676 012704 000377
003702 010412
003704 000257
003706 005737 067560
003712 001401
003714 100002
003716 000000
003720 000770

070140

T0020: MOV #0020,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #FIRST,R3 ;R3 POINTS TO LOCATION THAT STORES NO OF FIRST TEST
MOV R0,(R3) ;SAVE FIRST TEST NO. CHECKED
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377 (NO CHANGE)
R0020: MOV R4,(R2) ;[DEST] = 377
CCC ;SCOPE SYNC
I0020: TST @MBUF0 ;TEST THE TST
BEQ E0020 ;BR IF 'Z' SET - IT SHOULDN'T BE
BPL T0021 ;BR IF 'N' CLEAR - IT SHOULD BE
E0020: HALT ;TST FAILED TO ALTER CODES PROPERLY
BR R0020 ;LOCK ON HARD ERROR

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

; *****
; .SBTTL T0021 BASIC 'TST @WA' TEST WITH [A] < 0
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [163,264,265,266,267,220,211,367,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,163 / 33[266]220,220 / 16[367]016,016
;EXEC: [220]ALUC=LLLLL :[211]D=100000
;CODES: [367]SP=3 / N:C=1000
;SYNC: B05J2 (-) / T= 2.8 USEC
;KEY SIG: K3-4 TSTL / K3-3 DM=3L / K5-2 PS(N)(1)H

003722 012700 000021
003726 112760 000377 070140
003734 012702 067560
003740 012704 100000
003744 010412
003746 000257
003750 005737 067560
003754 001401
003756 100402
003760 000000
003762 000770
003764 020412
003766 001403
003770 011203
003772 000000
003774 000763

T0021: MOV #0021,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #100000,R4 ;MAKE S / B = 100000
R0021: MOV R4,(R2) ;MAKE [DEST] = 100000
CCC ;SCOPE SYNC
I0021: TST @MBUF0 ;TEST THE TST
BEQ E10021 ;BR IF 'Z' SET - IT SHULDN'T BE
BMI A0021 ;BR IF 'N' SET - IT SHOULD BE
E10021: HALT ;TST FAILED TO ALTER CODES PROPERLY
BR R0021 ;LOCK ON HARD ERROR
A0021: CMP R4,(R2) ;DID TST DISTURB [DEST] ?
BEQ T0022 ;BR IF NOT
E20021: MOV (R2),R3 ;GET THE WAS DATA
HALT ;TST DELIVERED A RESULT
BR R0021 ;LOCK ON HARD ERROR

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

; *****
; .SBTTL T0022 BASIC 'TST @WA' WITH [A] = 0
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [163,264,265,266,267,220,211,367,375,016] FC 1,3,9,8

;ACT BUTS: 37[004]100,163 / 33[266]220,220 / 16[367]016,016

;EXEC: [220]ALUC=LLLLL :[211]D=000000

;CODES: [367]SPS=3 / N:C=0100

;SYNC: B05J2 (-) / T=2.8 USEC

;KEY SIG: K3-4 TSTL / K3-3 DM=3L / K5-2 PS(Z)(1)H

003776 012700 000022
004002 112760 000377 070140
004010 012702 067560
004014 005004
004016 005012
004020 000257
004022 005737 067560
004026 001402
004030 000000
004032 000771
004034 020412
004036 001403
004040 011203
004042 000000
004044 000764

T0022: MOV #0022,R0 ;LOAD R0 WITH TEST NO.
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 0 (IT SHOULDN'T CHANGE
R0022: CLR (R2) ;[DEST] = 0
CCC ;SCOPE SYNC - Z=0
I0022: TST @MBUF0 ;TEST THE TST
BEQ A0022 ;BR IF TST SET 'Z'
E10022: HALT ;TST FAILED TO SET 'Z'
BR R0022 ;LOCK ON HARD ERROR
A0022: CMP R4,(R2) ;[DEST] STILL = 000000
BEQ T0023 ;BR IF YES
E20022: MOV (R2),R3 ;GET THE WAS DATA
HALT ;TST ALTERED THE [DEST]
BR R0022 ;LOCK ON HARD ERROR

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

; *****
; .SBTTL T0023 BASIC 'TST (R)+' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [162,260,267,220,211,367,375,016] FC 1,3,8,9
;ACT BUTS: 37[004]100,162 / 33[260]220,220 / 16[367]016,016
;EXEC: [220]ALUC=LLLLL :[211]D=000000
;CODES: [367] SPS=3 / N:C=0100
;SYNC: B05J2 (-) / T= 1.84 USEC
;KEY SIG: K3-4 TSTL / K3-3 DM=2L / K5-2 PS(Z)(1)H

004046 012700 000023
004052 112760 000377 070140
004060 012702 067560
004064 005004
004066 005012
004070 000257
004072 005722
004074 001402
004076 000000
004100 000772
004102 022702 067562
004106 001402
004110 000000
004112 000765

T0023: MOV #0023,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUFO,R2 ;INITIAL DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 0 (NO CHANGE)
R0023: CLR (R2) ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC
I0023: TST (R2)+ ;TEST THE TST
BEQ A0023 ;BR IF 'Z' SET - IT SHOULD BE
E10023: HALT ;TST FAILED TO SET 'Z'
BR R0023 ;LOCK ON HARD ERROR
A0023: CMP #MBUFO+2,R2 ;DID REG. GET AUTO-INCREMENTED ?
BEQ T0024 ;BR IF YES
E20023: HALT ;TST FAILED TO UPDATE REGISTER
BR R0023 ;LOCK ON HARD ERROR

```
4800 ; *****  
4801 ; .SBTTL T0024 BASIC 'TST -(R)'+ TEST  
4802 ; *****  
4803  
4804 ;MICROPROGRAMMING / LOGIC INFORMATION  
4805  
4806 ;ROM SEQ: [164,260,267,220,211,367,375,016]  
4807  
4808 ;ACT BUTS: 37[004]100,164 / 33[260]220,220 / 16[367]016,016  
4809  
4810 ;EXEC: [220]ALUC=LLLLL :[211]D=125252  
4811  
4812 ;CODES: [367] SPS=3 / N:C=0000  
4813  
4814 ;SYNC: B05J2 (-) / T= 1.84 USEC  
4815  
4816 ;KEY SIG: K3-4 TSTL / K3-3 DM=4  
4817  
4818 004114 012700 000024 T0024: MOV #0024,R0 ;LOAD R0 WITH TEST NO.  
4819 004120 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
4820 004126 012702 067576 MOV #DWTA+6,R2 ;DEST ADDR = DWTA+6  
4821 004132 012704 000377 MOV #377,R4 ;RESULT S / B = 377  
4822 004136 012705 067600 R0024: MOV #DWTA+10,R5 ;BASE DEST ADDR = DWTA+10  
4823 004142 000270 SEN ;SCOPE SYNC  
4824  
4825 004144 005745 I0024: TST -(R5) ;TEST THE TST  
4826  
4827 004146 100002 BPL A0024 ;BR IF 'N' CLEAR  
4828  
4829 004150 000000 E10024: HALT ;TST FAILED TO CLEAR 'N'  
4830 004152 000771 BR R0024 ;LOCK ON HARD ERROR  
4831  
4832 004154 020502 A0024: CMP R5,R2 ;DID DEST REG GET DECREMENTED?  
4833 004156 001402 BEQ B0024 ;BR IF YES  
4834  
4835 004160 000000 E20024: HALT ;ERROR - TST FAILED TO UPDATE DEST REG  
4836 004162 000765 BR R0024 ;LOCK ON HARD ERROR  
4837  
4838 004164 020412 B0024: CMP R4,(R2) ;DID TST ALTER [DEST]?  
4839 004166 001404 BEQ T0025 ;BR IF NOT  
4840  
4841 004170 011203 E30024: MOV (R2),R3 ;GET WAS DATA  
4842 004172 000000 HALT ;TST ALTERED [DEST]  
4843 004174 010412 MOV R4,(R2) ;RESTORE [DEST]  
4844 004176 000757 BR R0024 ;LOCK ON HARD ERROR  
4845
```

```
4846 ; *****  
4847 ; .SBTTL T0025 BASIC 'COM @WA' TEST  
4848 ; *****  
4849  
4850 ;MICROPROGRAMMING / LOGIC INFORMATION  
4851  
4852 ;ROM SEQ: [163,264,265,266,267,220,211,267,375,016] FC 1,3,9,8  
4853  
4854 ;ACT BUS: 37[004]100,163 / 33[266]220,220 / 16 [367]016,016  
4855  
4856 ;EXEC: [220]ALUC=HLLLL :[211]D=177777  
4857  
4858 ;CODES: [367] SPS=3 / N:C=1001  
4859  
4860 ;SYNC: B05J2 (-) / T= 2.8 USEC  
4861  
4862 ;KEY SIG: K3-4 COM L / K3-3 DM=3L / K5-2 PS(N)(1)H / K5-2 PS(C)(1)H  
4863  
4864 004200 012700 000025 T0025: MOV #0025,R0 ;LOAD R0 WITH TEST NO.  
4865 004204 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
4866 004212 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
4867 004216 005004 CLR R4 ;RESULT S / B = 177777  
4868 004220 005104 COM R4  
4869 004222 005012 R0025: CLR (R2) ;MAKE [DEST] = 000000  
4870 004224 000257 CCC ;SCOPE SYNC  
4871  
4872 004226 005137 067560 I0025: COM @MBUF0 ;TEST THE COM  
4873  
4874 004232 020412 CMP R4,(R2) ;RESULT = 177777 ??  
4875 004234 001403 BEQ T0026 ;BR IF YES  
4876  
4877 004236 011203 E0025: MOV (R2),R3 ;GET THE WAS DATA  
4878 004240 000000 HALT ;COM DELIVERED THE WRONG RESULT  
4879 004242 000767 BR R0025  
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

; *****
; .SBTTL T0026 BASIC 'INC @#A' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [163,264,265,266,267,220,211,367,375,016] FC 1,3,9,8

;ACT BUTS: 37[004]100,163 / 33[266]220,220 / 16[367]016,016

;EXEC: [220]ALUC=LLLLL :[211]D=100

;CODES: [367] SPS=3 / N:C=0000

;SYNC: B05J2 (-) / T= 2.8 USEC

;KEY SIG: K3-4 INC L / K3-3 DM=3L / K3-8 CIN 00 L

004244 012700 000026
004250 112760 000377
004256 012702 067560
004262 012704 000100
004266 012712 000077
004272 000257
004274 005237 067560
004300 020412
004302 001403
004304 011203
004306 000000
004310 000766

070140

T0026: MOV #0026,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #100,R4 ;RESULT S / B = 100
R0026: MOV #77,(R2) ;[DEST] = 77
CCC ;SCOPE SYNC
I0026: INC @MBUF0 ;TEST THE INC
CMP R4,(R2) ;DID RESULT = 100 ??
BEQ T0027 ;BR IF YES
E0026: MOV (R2),R3 ;GET THE WAS DATA
HALT ;INC DELIVERED WRONG RESULT
BR R0026 ;LOCK ON HARD ERROR

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

; *****
; .SBTTL T0027 BASIC 'DEC RN' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [104,373,360,001] FC 1,7,8
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001
;EXEC: [104]ALUC=LHHHH :[373]D=000000
;CODES: [367] SPS=3 / N:C=0100
;SYNC: B05J2 (-) / T= 1 USEC
;KEY SIG: K3-4 DEC L / K3-4 OVLAP INSTR H / K3-3 DM=0L / K5-2 PS(Z)(1)H

004312 012700 000027
004316 112760 000377 070140
004324 012702 177703
004330 005004
004332 012703 000001
004336 000257
004340 005303
004342 005703
004344 001402
004346 000000
004350 000770

T0027: MOV #0027,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #177703,R2 ;DEST ADDR = R3
CLR R4 ;RESULT S / B = 000000
R0027: MOV #1,R3 ;[DEST0030
CCC ;SCOPE SYNC
I0027: DEC R3 ;TEST THE DEC
TST R3 ;RESULT = 000000 ??
BEQ T0030 ;BR IF YES
E0027: HALT ;DEC DELIVERED THE WRONG RESULT
BR R0027 ;LOCK ON HARD ERROR

4982
4983
4984
4985
4986
4987
4988
4989
4990
4991
4992
4993
4994
4995
4996
4997
4998
4999
5000
5001
5002
5003
5004
5005
5006
5007
5008
5009
5010
5011
5012
5013
5014
5015
5016

004416 012700 000031
004422 112760 000377
004430 012702 067562
004434 005004
004436 012705 067560
004442 012712 177777
004446 000257
004450 005065 0000C2
004454 020412
004456 001403
004460 011203
004462 000000
004464 000764

070140

```
; *****  
; .SBTTL T0031 BASIC 'CLR X(R)' TESTS  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [167,261,262,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,167 / 17[167]262,262 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC HLLHH :[211]D=000000  
;CODES: [367] SPS=3 / N:C=0100  
;SYNC: B05J2 (-) / T= 2.5 USEC  
;KEY SIG: K3-3 DM=6L / K3-4 CLRL  
  
T0031: MOV #0031,R0 ;LOAD R0 WITH TEST NO.  
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #MBUF0+2,R2 ;DEST ADDR = MBUF0+2  
CLR R4 ;RESULT S / B = 000000  
R0031: MOV #MBUF0,R5 ;BASE DEST ADDR = MBUF0  
MOV #-1,(R2) ;[DEST] = 177777  
CCC ;SCOPE SYNC  
  
I0031: CLR 2(R5) ;TEST THE CLR  
  
CMP R4,(R2) ;RESULT = 0?  
BEQ T0032 ;BR IF YES  
  
E0031: MOV (R2),R3 ;GET WAS DATA  
HALT ;CLR FAILED TO ZERO [DEST]  
BR R0031 ;LOCK ON HARD ERROR.
```

5017
5018
5019
5020
5021
5022
5023
5024
5025
5026
5027
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

; *****
; .SBTTL T0032 BASIC 'ASL RN' TEST WITH [DEST]=125252 AND C(0)
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [104,373,360,000] FC 1,7,8
;ACT BUTS: 37[004]160,104 / 31[104]360,360 / 27[373]000,000
;EXEC: [104]ALUC=LHLL :[373]D=52524
;CODES: [360] SPS=3 / N:C=0001
;SYNC: B05J2 (-) / T= 1 USEC
;KEY SIG: K3-5 ROTSHF H / K3-3 SM=6L / K5-2 PS (C)(1)H

070140

T0032: MOV #0032,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #52524,R4 ;RESULT S / B = 52524
MOV #177703,R2 ;DEST ADDR = R3
R0032: MOV #125252,R3 ;MAKE [DEST] = 125252
CCC ;MAKE C=0
I0032: ASL R3 ;TEST THE ASL - IT SHOULD SET 'C'
BCS A0032 ;BR IF 'C' GOT SET
E10032: HALT ;ASL FAILED TO SET 'C' BIT
BR R0032 ;LOCK ON HRD ERROR
A0032: CMP R4,R3 ;WAS RESULT = 52524 ??
BEQ T0033 ;BR IF YES
E20032: HALT ;ASL DELIVERED THE WRONG RESULT
BR R0032 ;LOCK ON HARD ERROR

```
5055 ; *****
5056 ; .SBTTL T0033 BASIC 'ASL RN' TEST WITH [DEST]=052525 AND C(1)
5057 ; *****
5058
5059 ;MICROPROGRAMMING / LOGIC INFORMATION
5060
5061 ;ROM SEQ: [104,373,360,000] FC 1,7,8]
5062
5063 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,000
5064
5065 ;EXEC: [104]ALUC=LHLL :[373]D=125252
5066
5067 ;CODES: [360] SPS=3 / N:C=1000
5068
5069 ;SYNC: B05J2 (-) / T= 1 USEC
5070
5071 ;KEY SIG: K3-5 ROTSHF H / K3-3 SM=6L / K5-2 PS(N)(1)H
5072
5073 004536 012700 000033 T0033: MOV #0033,R0 ;LOAD R0 WITH TEST NO.
5074 004542 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5075 004550 012704 125252 MOV #125252,R4 ;RESULT S / B = 125252
5076 004554 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
5077 004560 012703 052525 R0033: MOV #052525,R3 ;MAKE [DEST] = 052525
5078 004564 000261 SEC ;MAKE C=1
5079
5080 004566 006303 I0033: ASL R3 ;TEST THE ASL - IT SHOULD CLR 'C'
5081
5082 004570 103002 BCC A0033 ;BR IF 'C' GOT CLEARED
5083
5084 004572 000000 E10033: HALT ;ASL FAILED TO CLEAR 'C'
5085 004574 000771 BR R0033 ;LOCK ON HARD ERROR
5086
5087 004576 020403 A0033: CMP R4,R3 ;RESULT = 125252 ??
5088 004600 001402 BEQ T0034 ;BR IF YES
5089
5090 004602 000000 E20033: HALT ;ASL DELIVERED WRONG REULT
5091 004604 000765 BR R0033 ;LOCK ON HARD ERROR
```

```
5092 ; *****
5093 ; .SBTTL T0034 BASIC 'ROL RN' TEST WITH [DEST]=125252 AND C(0)
5094 ; *****
5095
5096 ;MICROPROGRAMMING / LOGIC INFORMATION
5097
5098 ;ROM SEQ: [104,373,360,000] FC 1,7,8
5099
5100 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,000
5101
5102 ;EXEC: [104]ALUC=LHLL :[373]D=052524
5103
5104 ;CODES: [360] SPS=3 / N:C=0001
5105
5106 ;SYNC: B05J2 (-) / T= 1 USEC
5107
5108 ;KEY SIG: K3-5 ROTSHF H / K3-3 SM=6L / K5-2 PS(C)(1)H
5109
5110 004606 012700 000034 T0034: MOV #0034,R0 ;LOAD R0 WITH TEST NO.
5111 004612 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5112 004620 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
5113 004624 012704 052524 MOV #052524,R4 ;RESULT S / B = 052524
5114 004630 012703 125252 R0034: MOV #125252,R3 ;MAKE [DEST] = 125252
5115 004634 000257 CCC ;MAKE C=0
5116
5117 004636 006103 I0034: ROL R3 ;TEST THE ROL - IT SHOULD SET C
5118
5119 004640 103402 BCS A0034 ;BR IF 'C' GOT SET
5120
5121 004642 000000 E10034: HALT ;ROL FAILED TO SET 'C'
5122 004644 000771 BR R0034 ;LOCK ON HARD ERROR
5123
5124 004646 020403 A0034: CMP R4,R3 ;RESULT = 052524 ??
5125 004650 001402 BEQ T0035 ;BR IF YES
5126
5127 004652 000000 E20034: HALT ;ROL DELIVERED WRONG RESULT
5128 004654 000765 BR R0034 ;LOCK ON HARD ERROR
5129
```

```
5130 ; *****
5131 ; .SBTTL T0035 BASIC 'ROL RN' TEST WITH [DEST]=052524 AND C(1)
5132 ; *****
5133 ;MICROPROGRAMMING / LOGIC INFORMATION
5134 ;ROM SEQ: [104,373,360,000] FC 1,7,8
5135 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,000
5136 ;EXEC: [104]ALUC=LHLL :[373]D=125251
5137 ;CODES: [360] SPS=3 / N:C=1000
5138 ;SYNC: B05J2 (-) / T= 1 USEC
5139 ;KEY SIG: K3-5 ROTSHF H / K3-3 SM=6L / K5-2 PS(N)(1)H
5140
5141
5142
5143
5144
5145
5146
5147
5148 004656 012700 000035 T0035: MOV #0035,R0 ;LOAD R0 WITH TEST NO.
5149 004662 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5150 004670 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
5151 004674 012704 125251 MOV #125251,R4 ;RESULT S / B = 125251
5152 004700 012703 052524 R0035: MOV #052524,R3 ;MAKE [DEST] = 052524
5153 004704 000261 SEC ;MAKE C=1
5154
5155 004706 006103 I0035: ROL R3 ;TEST THE ROL - IT SHOULD CLEAR C
5156
5157 004710 103002 BCC A0035 ;BR IF 'C' IS CLEAR
5158
5159 004712 000000 E10035: HALT ;ROL FAILED TO CLEAR 'C'
5160 004714 000771 BR R0035 ;LOCK ON HARD ERROR
5161
5162 004716 020403 A0035: CMP R4,R3 ;RESULT = 125251 ??
5163 004720 001402 BEQ T0036 ;BR IF YES
5164
5165 004722 000000 E20035: HALT ;ROL DELIVERED WRONG RESULT
5166 004724 000765 BR R0035 ;LOCK ON HARD ERROR
5167
```

```
5168 ; *****  
5169 ; .SBTTL T0036 BASIS 'TSTB (R)'' TEST - EVEN ADDRESS  
5170 ; *****  
5171 ;  
5172 ;MICROPROGRAMMING / LOGIC INFORMATION  
5173 ;  
5174 ;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
5175 ;  
5176 ;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
5177 ;  
5178 ;EXEC: [220]ALUC=LLLLL :[211]D=000377  
5179 ;  
5180 ;CODES: [367] SPS=3 / N:C=1000  
5181 ;  
5182 ;SYNC: B05J2 (-) / T= 1.8 USEC  
5183 ;  
5184 ;KEY SIG: K3-4 TSTL / K3-3 DM=1L / K5-2 PS(N)(1)H / K3-6 BYTE INSTR H  
5185 ;  
5186 004726 012700 000036 T0036: MOV #0036,R0 ;LOAD R0 WITH TEST NO.  
5187 004732 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
5188 004740 012702 067576 MOV #DWTA+6,R2 ;DEST ADDR = DWTA+6  
5189 004744 012704 000377 MOV #377,R4 ;RESULT S / B = 377  
5190 004750 000257 R0036: CCC ;SCOPE SYNC  
5191 ;  
5192 004752 105712 I0036: TSTB (R2) ;TEST THE T3TB  
5193 ;  
5194 004754 100402 BMI A0036 ;BR IF 'N' SET - IT SHOULD BE  
5195 ;  
5196 004756 000000 E10036: HALT ;TSTB FAILED TO SET 'N'  
5197 004760 000773 BR R0036 ;LOCK ON HARD ERROR  
5198 ;  
5199 004762 020412 A0036: CMP R4,(R2) ;DID TSTB DISTURB [DEST]  
5200 004764 001404 BEQ T0037 ;BR IF NOT  
5201 ;  
5202 004766 011203 E20036: MOV (R2),R3 ;GET WAS DATA  
5203 004770 000000 HALT ;TSTB ALTERED [DEST]  
5204 004772 010412 MOV R4,(R2) ;RESTORE [DEST]  
5205 004774 000765 BR R0036 ;LOCK ON HARD EROR
```

5206
5207
5208
5209
5210
5211
5212
5213
5214
5215
5216
5217
5218
5219
5220
5221
5222
5223
5224
5225
5226
5227
5228
5229
5230
5231
5232
5233
5234
5235
5236
5237
5238
5239
5240
5241
5242
5243
5244
5245

; *****
; .SBTTL T0037 BASIS 'TSTB (R)'' TEST - ODD ADDRESS
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,237,270,222,253,075,374,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,161 / 33[266]220,237 / 34[237]220,222 / 16[374]016,016
;EXEC: [222]ALUC=LLLLL :[253]D=000377
;CODES: [075] SPS=3 / N:C=1000
;SYNC: B05J2 (-) / T= 1.9 USEC
;KEY SIG: K3-4 TSTL / K3-3 DM=1L / K5-2 PS(N)(1)H / K3-6 BYTE INSTR H

004776 012700 000037
005002 112760 000377 070140
005010 012702 070122
005014 012704 177401
005020 012703 070123
005024 000257
005026 105713
005030 100402
005032 000000
005034 000773
005036 020412
005040 001404
005042 011203
005044 000000
005046 010412
005050 000765

T0037: MOV #0037,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #DWTB+6,R2 ;DEST ADDR = DWTB+6
MOV #177401,R4 ;RESULT S / B = 177401
MOV #DWTB+7,R3 ;DEST ADDR USED = DWTB+7
R0037: CCC ;SCOPE SYNC
I0037: TSTB (R3) ;TEST THE TSTB
BMI A0037 ;BR IF 'N' SET - IT SHOULD BE
E10037: HALT ;TSTB FAILED TO SET 'N'
BR R0037 ;LOCK ON HARD ERROR
A0037: CMP R4,(R2) ;DID TSTB DISTURB [DEST]
BEQ T0040 ;BR IF NOT
E20037: MOV (R2),R3 ;GET WAS DATA
HALT ;TSTB ALTERED [DEST]
MOV R4,(R2) ;RESTORE [DEST]
BR R0037 ;LOCK ON HARD EROR

5246
5247
5248
5249
5250
5251
5252
5253
5254
5255
5256
5257
5258
5259
5260
5261
5262
5263
5264
5265
5266
5267
5268
5269
5270
5271
5272
5273
5274
5275
5276
5277
5278
5279
5280
5281
5282
5283
5284
5285
5286
5287

005052 012700 000040
005056 112760 000377 070140
005064 032737 000001 066642
005072 001401
005074 000000
005076 012702 067574
005102 012704 177400
005106 000257
005110 105737 067574
005114 001402
005116 000000
005120 000772
005122 020412
005124 001404
005126 011203
005130 000000
005132 010412
005134 000764

```
; *****  
; .SBTTL T0040 BASIC 'TSTB @#A'' TEST - EVEN ADDRESS  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [163,264,265,266,267,225,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,163 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=LLLLL :[211]D=177400  
;CODES: [075] SPS=3 / N:C=0100  
;SYNC: B05J2 (-) / T= 2.8 USEC  
;KEY SIG: K3-4 TSTL / K3-3 M=3L / K5-2 PS(Z)(1)H / K3-6 BYTE INSTR H  
T0040: MOV #0040,R0 ;LOAD R0 WITH TEST NO.  
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
BIT #1,@#BPTLOC ;BREAKPOINT HALT SET ??  
BEQ .+4 ;BR IF NOT  
HALT ;BREAK - DEPRESS CONTINUE TO RESTART  
MOV #DWTA+4,R2 ;DEST ADDR = DWTA+4  
MOV #177400,R4 ;RESULT S / B = 177400  
R0040: CCC ;SCOPE SYNC  
I0040: TSTB @#DWTA+4 ;TEST THE TSTB  
BEQ A0040 ;BR IF 'Z' SET - IT SHOULD BE  
E10040: HALT ;TSTB FAILED TO SET 'Z'  
BR R0040 ;LOCK ON HARD ERROR  
A0040: CMP R4,(R2) ;DID TSTB DISTURB [DEST]?  
BEQ T0041 ;BR IF NOT  
E20040: MOV (R2),R3 ;GET WAS DATA  
HALT ;TSTB ALTERED [DEST]  
MOV R4,(R2) ;RESTORE [DEST]  
BR R0040 ;LOCK ON HARD ERROR
```

```

5288 ; *****
5289 ; .SBTTL T0041 BASIC 'TSTB @#A' TEST - ODD ADDRESS
5290 ; *****
5291
5292 ;MICROPROGRAMMING / LOGIC INFORMATION
5293
5294 ;ROM SEQ: [163,264,265,266,267,237,270,222,253,075,374,375,016] FC 1,3,9,8
5295
5296 ;ACT BUTS: 37[004]100,163 / 33[266]220,237 / 34[237]220,222 / 16[374]016,016
5297
5298 ;EXEC: [222]ALUC=LLLLL :[253]D=000377
5299
5300 ;CODES: [075] SPS=3 / N:C=0100
5301
5302 ;SYNC: B05J2 (-) / T= 2.8 USEC
5303
5304 ;KEY SIG: K3-4 TSTL / K3-3 DM=3L / K5-2 PS(2)(1)H / K3-6 BYTE INSTR H
5305
5306 005136 012700 000041 T0041: MOV #0041,R0 ;LOAD R0 WITH TEST NO.
5307 005142 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5308 005150 012702 067576 MOV #DWTA+6,R2 ;DEST ADDR = DWTA+6
5309 005154 012704 000377 MOV #377,R4 ;RESULT S / B = 377
5310 005160 000257 R0041: CCC ;SCOPE SYNC
5311
5312 005162 105737 067577 I0041: TSTB @#DWTA+7 ;TEST THE TSTB
5313
5314 005166 001402 BEQ A0041 ;BR IF 'Z' SET - IT SHOULD BE
5315
5316 005170 000000 E10041: HALT ;TSTB FAILED TO SET 'Z'
5317 005172 000772 BR R0041 ;LOCK ON HARD ERROR
5318
5319 005174 020412 A0041: CMP R4,(R2) ;DID TSTB DISTURB [DEST]?
5320 005176 001404 BEQ T0042 ;BR IF NOT
5321
5322 005200 011203 E20041: MOV (R2),R3 ;GET WAS DATA
5323 005202 000000 HALT ;TSTB ALTERED [DEST]
5324 005204 010412 MOV R4,(R2) ;RESTORE [DEST]
5325 005206 000764 BR R0041 ;LOCK ON HARD ERROR
5326

```

5327
5328
5329
5330
5331
5332
5333
5334
5335
5336
5337
5338
5339
5340
5341
5342
5343
5344
5345
5346
5347
5348
5349
5350
5351
5352
5353
5354
5355
5356
5357
5358
5359
5360
5361
5362
5363
5364
5365

; *****
; .SBTTL T0042 BASIC 'DECB 1(SP)'
; *****
;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ: [167,261,262,266,267,237,270,222,253,075,374,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,167 / 33[266]220,237 / 34[237]220,222 / 16[374]016,016
;EXEC: [222]ALUC=LHHHH :[253]D=177400
;CODES: [075] SPS=3 / N:C=1000
;SYNC: B05J2 (-) / T= 2.8 USEC
;KEY SIG: K3-4 DEC L / K3-3 DM=6L / K5-2 PS(N)(1)H / K3-6 BYTE INSTR H

005210 012700 000042
005214 112760 000377 070140
005222 010605
005224 012704 177400
005230 010506
005232 012746 000000
005236 000257
005240 105366 000001
005244 020416
005246 001405
005250 011603
005252 010602
005254 005202
005256 000000
005260 000763
005262 010506

T0042: MOV #0042,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV SP,R5 ;SAVE SP
MOV #177400,R4 ;RESULT S / B = 177400
R0042: MOV R5,SP
MOV #0,-(SP) ;[DEST] = 000000
CCC ;SCOPE SYNC
I0042: DECB 1(SP) ;TEST THE DECB
CMP R4,(SP) ;RESULT = 177400?
BEQ A0042 ;BR IF YES
MOV (SP),R3 ;GET WAS DATA
MOV SP,R2 ;GENERATE DEST ADDR IN R2
INC R2
E0042: HALT ;ERROR - DECB FAILED
BR R0042 ;LOCK ON HARD ERROR
A0042: MOV R5,SP ;RESET THE SP

```

5366 ; *****
5367 ; .SBTTL T0043 BASIC 'MOV @#A,R' TEST
5368 ; *****
5369
5370 ;MICROPROGRAMMING / LOGIC INFORMATION
5371
5372 ;ROM SEQ: [143,245,246,247,250,160,204,000] FC 1,2,4
5373
5374 ;ACT BUTS: 37[004]100,143 / 35[247]120,160 / 20[160]000,000
5375
5376 ;EXEC: [160]ALUC=LLLLL :[204]D=#DWTA
5377
5378 ;CODES: [204] SPS=3 / N:C=0000
5379
5380 ;SYNC: B05J2 (-) / T= 3.2 USEC
5381
5382 ;KEY SIG: K3-3 MOV L / K3-3 DM=0L / K3-5 DOPL
5383
5384 005264 012700 000043 T0043: MOV #0043,R0 ;LOAD R0 WITH TEST NO.
5385 005270 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5386 005276 012702 177703 MOV #177703,R2 ;DEST ADDR = 3
5387 005302 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA
5388 005306 005003 R0043: CLR R3 ;[DEST] = 000000
5389 005310 000257 CCC ;SCOPE SYNC
5390
5391 005312 013703 067544 I0043: MOV @#ATA,R3 ;TEST THE MOV
5392
5393 005316 020403 CMP R4,R3 ;RESULT = DWTA?
5394 005320 001402 BEQ T0044 ;BR IF YES
5395
5396 005322 000000 E0043: HALT ;MOV FAILED TO DELIVER CORRECT RESULT
5397 005324 000770 BR R0043 ;LOCK ON HARD ERROR
5398

```

5399
5400
5401
5402
5403
5404
5405
5406
5407
5408
5409
5410
5411
5412
5413
5414
5415
5416
5417
5418
5419
5420
5421
5422
5423
5424
5425
5426
5427
5428
5429
5430
5431
5432
5433
5434

; *****
; .SBTTL T0044 BASIC 'MOV #N,X(R)' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,177,206,212,200,125,375,016] FC 1,2,4,8
;ACT BUTS: 37[004]100,142 / 35[240]120,177 / 17[177]212,212 / 16[125]016,016
;EXEC: [200]ALUC=LLLLL :[125]D=125252
;CODES: [125] SPS=3 / N:C=0000
;SYNC: B05J2 (-) / T= 4 USEC
;KEY SIG: K3-3 MOVL / K3-3 DM=0 / K3-5 DOPL

005326	012700	000044		T0044:	MOV	#0044,R0	:LOAD R0 WITH TEST NO.
005332	112760	000377	070140		MOVB	#377,STAB1(R0)	:SET FLAG FOR THIS TEST IN MISSED TABLE
005340	012702	067562			MOV	#MBUF0+2,R2	:DEST ADDR = MBUF0+2
005344	012704	125252			MOV	#125252,R4	:RESULT S / B = 125252
005350	012703	067560		R0044:	MOV	#MBUF0,R3	: [R3] = BASE DEST ADDR
005354	005012				CLR	(R2)	: [DEST] = 000000
005356	000257				CCC		:SCOPE SYNC
005360	012763	125252	000002	I0044:	MOV	#125252,2(R3)	:TEST THE MOV
005366	020412				CMP	R4,(R2)	:RESULT OK?
005370	001403				BEQ	T0045	:BR IF YES
005372	011203				MOV	(R2),R3	:GET WAS DATA
005374	000000			E0044:	HALT		:MOV DELIVERED WRONG RESULT
005376	000764				BR	R0044	:LOCK ON HARD ERROR

5435
5436
5437
5438
5439
5440
5441
5442
5443
5444
5445
5446
5447
5448
5449
5450
5451
5452
5453
5454
5455
5456
5457
5458
5459
5460
5461
5462
5463
5464
5465
5466
5467
5468
5469

005400 012700 000045
005404 112760 000377 070140
005412 012702 067560
005416 012704 125252
005422 010203
005424 005013
005426 000257

005430 012713 125252

005434 020412
005436 001403

005440 011203
005442 000000
005444 000766

```
; *****  
; .SBTTL T0045 BASIC 'MOV #N,(R)'' TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [142,240,250,171,257,200,125,375,016] FC 1,2,4  
;ACT BUTS: 37[004]100,142 / 35[240]120,171 / 22[171]200,200 / 16[125]016,016  
;EXEC: [200]ALUC=L L L L L :[125]D=125252  
;CODES: [125] SPS=3 / N:C=1000  
;SYNC: B05J2 (-) / T= 2.3 USEC  
;KEY SIG: K3-3 MOVL / K3-3 DM=1L / K3-5 DOPL / K5-2 PS (N)(1)H  
T0045: MOV #0045,R0 ;LOAD R0 WITH TEST NO.  
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #125252,R4 ;RESULT S / B = 125252  
R0045: MOV R2,R3 ;R3 GETS DEST ADDR  
CLR (R3) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
  
I0045: MOV #125252,(R3) ;TEST THE MOV  
  
CMP R4,(R2) ;RESULT OK?  
BEQ T0046 ;BR IF YES  
  
E0045: MOV (R2),R3 ;GET WAS DATA  
HALT ;MOV DELIVERED WRONG RESULT  
BR R0045 ;LOCK ON HARD ERROR
```

```

5470 ; *****
5471 ; .SBTTL T0046 BASIC 'MOV (RA)+,RB'' TEST
5472 ; *****
5473
5474 ;MICROPROGRAMMING / LOGIC INFORMATION
5475
5476 ;ROM SEQ: [142,240,250,160,204,000] FC 1,2,4
5477
5478 ;ACT BUTS: 37[004]100,142 / 35[240]120,160 / 20[160]000,000
5479
5480 ;EXEC: [160]ALUC=LLLLL :[204]D=#DWTA
5481
5482 ;CODES: [204] SPS=3 / N:C=0000
5483
5484 ;SYNC: B05J2 (-) / T= 2.3 USEC
5485
5486 ;KEY SIG: K3-3 MOVL / K3-3 DM=0L / K3-5 DOPL
5487
5488 005446 012700 000046 T0046: MOV #0046,R0 ;LOAD R0 WITH TEST NO.
5489 005452 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5490 005460 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA
5491 005464 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
5492 005470 012705 067544 R0046: MOV #ATA,R5 ;SRC ADDR = ATA
5493 005474 005003 CLR R3 ;[DEST] = 000000
5494 005476 000257 CCC ;SCOPE SYNC
5495
5496 005500 012503 I0046: MOV (R5)+,R3 ;TEST THE MOV
5497
5498 005502 020403 CMP R4,R3 ;RESULT OK?
5499 005504 000402 BR A0046 ;BR IF YES
5500
5501 005506 000000 E10046: HALT ;MOV DELIVERED WRONG RESULT
5502 005510 000767 BR R0046 ;LOCK ON HARD ERROR
5503
5504 005512 022705 067546 A0046: CMP #ATA+2,R5 ;DID SRC REG GET INCREMENTED?
5505 005516 001402 BEQ T0047 ;BR IF YES
5506
5507 005520 000000 E20046: HALT ;MOV FAILED TO UPDATE SRC. REG.
5508 005522 000762 BR R0046 ;LOCK ON HARD ERROR
5509

```

```

5510 ; *****
5511 ; .SBTTL T0047 BASIC 'MOV @A,@B'
5512 ; *****
5513
5514 ;MICROPROGRAMMING / LOGIC INFORMATION
5515
5516 ;ROM SEQ: [143,245,256,247,250,173,207,210,200,125,375,016] FC 1,2,4
5517
5518 ;ACT BUTS: 37[004]100,143 / 35[247]120,173 / 22[207]200,200 / 16[125]016,016
5519
5520 ;EXEC: [200]ALUC=LLLLL :[125]D=#DWTA
5521
5522 ;CODES: [125] SPS=3 / N:C=0000
5523
5524 ;SYNC: B05J2 (-) / T= 5 USEC
5525
5526 ;KEY SIG: K3-3 MOVL / K3-3 SM=3L / K3-3 DM=3L
5527
5528 005524 012700 000047 T0047: MOV #0047,R0 ;LOAD R0 WITH TEST NO.
5529 005530 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5530 005536 012702 067564 MOV #MBUF1,R2 ;DEST ADDR = MBUF1
5531 005542 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA
5532 005546 005012 R0047: CLR (R2) ;MAKE [DEST] = 000000
5533 005550 000257 CCC ;SCOPE SYNC
5534
5535 005552 013737 067544 067564 I0047: MOV @A,@MBUF1 ;TEST THE MOV
5536 005560 020412 CMP R4,(R2) ;DID RESULT = #DWTA ?
5537 005562 001403 BEQ T0050 ;BR IF YES
5538
5539 005564 011203 E0047: MOV (R2),R3 ;GET THE WAS DATA
5540 005566 000000 HALT ;MOV DELIVERED THE WRONG RESULT
5541 005570 000766 BR R0047 ;LOCK ON HARD ERROR
5542

```

```

5543 ; *****
5544 ; .SBTTL T0050 BASIC 'MOV X(R),PC' TEST
5545 ; *****
5546
5547 ;MICROPROGRAMMING / LOGIC INFORMATION
5548
5549 ;ROM SEQ: [146,241,242,247,250,160,204,000] FC 1,2,4
5550
5551 ;ACT BUTS: 37[004]100,146 / 35[247]120,160 / 20[160]000,000
5552
5553 ;EXEC: [160]ALUC=LLLLL :[204]D=#T077
5554
5555 ;CODES: [204] SPS=3 / N:C=0000
5556
5557 ;SYNC: B05J2 (-) / T= 4 USEC
5558
5559 ;KEY SIG: K3-3 MOVL / K3-3 SM=6L / K3-3 DM=0L / K3-4 IR (02:00)=7L
5560
5561 005572 012700 000050 T0050: MOV #0050,R0 ;LOAD R0 WITH TEST NO.
5562 005576 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5563 005604 012705 005612 R0050: MOV #I0050,R5 ;[R5] = I0050 (BASE ADDRESS)
5564 005610 000257 CCC ;SCOPE SYNC
5565
5566 005612 016507 000010 I0050: MOV A0050-I0050(R5),PC ;TEST THE MOV - GO TO A0050
5567
5568 005616 000000 E0050: HALT ;MOV FAILED TO LOAD THE PC
5569 005620 000771 BR R0050 ;LOCK ON HARD ERROR
5570
5571 005622 005624 A0050: T0051 ;POINTER TO NEXT TEST
5572

```

```

5573 ; *****
5574 ; .SBTTL T0051 BASIC 'MOV @WA,(R)'' TEST
5575 ; *****
5576 ;MICROPROGRAMMING / LOGIC INFORMATION
5577 ;ROM SEQ: [143,245,246,247,250,171,257,200,125,375,016] FC 1,2,4
5578 ;ACT BUTS: 37[004]100,143 / 35[247]120,171 / 22[171]200,200 / 16[125]016,016
5579 ;EXEC: [200]ALUC=LLLLL :[125]D=051300
5580 ;CODES: [125] SPS=3 / N:C=0000
5581 ;SYNC: B05J2 (-) / T= 4.2 USEC
5582 ;KEY SIG: K3-3 MOVL / K3-3 SM=3L / K3-3 DM=1L
5583
5584 T0051: MOV #0051,R0 ;LOAD R0 WITH TEST NO.
5585 MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5586 MOV #DWTA,R4 ;RESULT S / B = #DWTA
5587 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
5588 R0051: CLR (R2) ;MAKE [DEST]=000000
5589 CCC ;SCOPE SYNC - Z=0
5590
5591 005624 012700 000051 070140 I0051: MOV @WA,(R2) ;TEST THE MOV
5592 005630 112760 000377
5593 005636 012704 067570
5594 005642 012702 067560
5595 005646 005012
5596 005650 000257
5597
5598 005652 013712 067544
5599
5600 005656 020412
5601 005660 001403
5602
5603 005662 011203
5604 005664 000000 E0051: MOV (R2),R3 ;GET THE WAS DATA
5605 005666 000767 BR R0051 ;MOV DELIVERED WRONG RESULT
5606 ;LOCK ON HARD ERROR

```

5607
5608
5609
5610
5611
5612
5613
5614
5615
5616
5617
5618
5619
5620
5621
5622
5623
5624
5625
5626
5627
5628
5629
5630
5631
5632
5633
5634
5635
5636
5637
5638
5639
5640

; *****
; .SBTTL T0052 BASIC 'MOV X(RA),RB'' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ: [146,241,242,247,250,160,204,000] FC 1,2,4
;ACT BUTS: 37[004]100,146 / 35[247]120,160 / 20[160]000,000
;EXEC: [160]ALUC=LLLLL :[204]D=#DBTA
;CODES: [204] SPS=3 / N:C=0000
;SYNC: B05J2 (-) / T= 3 USEC
;KEY SIG: K3-3 MOVL / K3-3 SM=6L / K3-3 DM=0L

005670 012700 000052
005674 112760 000377 070140
005702 012704 070130
005706 012702 177703
005712 012705 067544
005716 005003
005720 000257

005722 016503 000004

005726 020403
005730 001402

005732 000000
005734 000770

T0052: MOV #0052,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #DBTA,R4 ;RESULT S / B = #DBTA
MOV #177703,R2 ;DEST ADDR = R3
MOV #ATA,R5 ;[R5] = BASE ADDR FOR SOURCE (ATA)
R0052: CLR R3 ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC

I0052: MOV 4(R5),R3 ;TEST THE MOV

CMP R4,R3 ;RESULT = #DBTA ??
BEQ T0053 ;BR IF YES

E0052: HALT ;MOV DELIVERED WRONG RESULT
BR R0052 ;LOCK ON HARD ERROR

```

5641 ; *****
5642 ; .SBTTL T0053 BASIC 'MOV RA,-(RB)'' TEST
5643 ; *****
5644
5645 ;MICROPROGRAMMING / LOGIC INFORMATION
5646
5647 ;ROM SEQ: [174,257,201,125,375,016] FC 1,4
5648
5649 ;ACT BUTS: 37[004]100,174 / 22[174]200,201 / 16[125]016,016
5650
5651 ;EXEC: [201]ALUC=LLLLL :[125]D=125252
5652
5653 ;CODES: [125] SPS=3 / N:C=1000
5654
5655 ;SYNC: B05J2 (-) / T= 1.8 USEC
5656
5657 ;KEY SIG: K-3 MOVL / K3-3 SM=0L / K3-3 DM=4L / K5-2 PS(N)(1)H
5658

```

```

5659 005736 012700 000053 T0053: MOV #0053,R0 ;LOAD R0 WITH TEST NO.
5660 005742 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5661 005750 012702 067560 MOV #MBUFO,R2 ;FINAL DEST ADDR = MBUFO
5662 005754 012704 125252 MOV #125252,R4 ;RESULT S / B = 125252
5663 005760 012705 067562 R0053: MOV #MBUFO+2,R5 ;INITIAL DEST ADDR = TEMP2 + 2
5664 005764 005012 CLR (R2) ;MAKE [DEST] = 000000
5665 005766 000257 CCC ;SCOPE SYNC
5666
5667 005770 010445 I0053: MOV R4,-(R5) ;TEST THE MOV
5668
5669 005772 020412 CMP R4,(R2) ;RESULT = 125252
5670 005774 001403 BEQ A0053 ;BR IF YES
5671
5672 005776 011203 MOV (R2),R3 ;GET THE S / B DATA
5673 006000 000000 E10053: HALT ;MOV DELIVERED THE WRONG RESULT
5674 006002 000766 BR R0053 ;LOCK ON HARD ERROR
5675
5676 006004 020205 A0053: CMP R2,R5 ;DID REGISTER GET DECREMENTED ?
5677 006006 001402 BEQ T0054 ;BR IF YES
5678
5679 006010 000000 E20053: HALT ;MOV FAILED TO UPDATE REGISTER
5680 006012 000762 BR R0053 ;LOCK ON HARD ERROR
5681
5682

```

5683
5684
5685
5686
5687
5688
5689
5690
5691
5692
5693
5694
5695
5696
5697
5698
5699
5700
5701
5702
5703
5704
5705
5706
5707
5708
5709
5710
5711
5712
5713
5714
5715
5716
5717
5718
5719
5720
5721
5722
5723

; *****
; .SBTTL T0054 BASIC 'MOV @#A,-(R)'' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [143,245,246,247,250,174,200,125,375,016] FC 1,2,4
;ACT BUTS: 37[004]100,143 / 35[247]120,174 / 22[174]200,200 / 16[125]016,016
;EXEC: [200]ALUC=LLLLL :[125]D=#DWTA
;CODES: [125] SPS=3 / N:C=0100
;SYNC: B05J2 (-) / T= 4.2 USEC
;KEY SIG: K3-3 MOVL / K3-3 SM=3L / K3-3 DM=4L / K5-2 PS(Z)(1)H

006014 012700 000054
006020 112760 000377 070140
006026 012704 067570
006032 012702 067560
006036 012705 067562
006042 005012
006044 000257
006046 013745 067544
006052 020412
006054 001403
006056 011203
006060 000000
006062 000765
006064 020502
006066 001402
006070 000000
006072 000761

T0054: MOV #0054,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #DWTA,R4 ;RESULT S / B = #DWTA
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
R0054: MOV #MBUF0+2,R5 ;INITIAL DEST ADDR = MBUF0+2
CLR (R2) ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC
I0054: MOV @#ATA,-(R5) ;TEST THE MOV
CMP R4,(R2) ;RESULT = 000000
BEQ A0054 ;BR IF YES
E10054: MOV (R2),R3 ;GET THE WAS DATA
HALT ;MOV DELIVERED THE WRONG RESULT
BR R0054 ;LOCK ON HARD ERROR
A0054: CMP R5,R2 ;DID DEST REG GET DECREMENTED ??
BEQ T0055 ;BR IF YES
E20054: HALT ;MOV FAILED TO UPDATE REGISTER
BR R0054 ;LOCK ON HARD ERROR

5724
5725
5726
5727
5728
5729
5730
5731
5732
5733
5734
5735
5736
5737
5738
5739
5740
5741
5742
5743
5744
5745
5746
5747
5748
5749
5750
5751
5752
5753
5754
5755
5756
5757
5758

```
; *****  
; .SBTTL T0055 BASIC 'MOV (R),@WA'' TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,173,210,200,125,375,016] FC 1,2,4  
;ACT BUTS: 37[004]100,141 / 35[247]120,173 / 22[207]200,200 / 16[125]016,016  
;EXEC: [200]ALUC=LLLLL :[125]D=#DWTA  
;CODES: [125] SPS=3 / N:C=0100  
;SYNC: B05J2 (-) / T= 4 USEC  
;KEY SIG: K3-3 MOVL / K3-3 SM=1L / K3-3 DM=3L / K5-2 PS(2)(1)H
```

```
006074 012700 000055  
006100 112760 000377 070140  
006106 012702 067560  
006112 012704 067570  
006116 012705 067544  
006122 005012  
006124 000257  
006126 011537 067560  
006132 020412  
006134 001403  
006136 011203  
006140 000000  
006142 000767
```

```
T0055: MOV #0055,R0 ;LOAD R0 WITH TEST NO.  
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #DWTA,R4 ;RESULT S / B = #DWTA  
MOV #ATA,R5 ;SOURCE ADDR = ATA  
R0055: CLR (R2) ;MAKE [DEST] = 000000  
CCC ;SCOPE SYNC  
I0055: MOV (R5),@MBUF0 ;TEST THE MOV  
CMP R4,(R2) ;RESULT = #DWTA ??  
BEQ T0056 ;BR IF YES  
E0055: MOV (R2),R3 ;GET THE WAS DATA  
HALT ;MOV DELIVERED THE WRONG RESULT  
BR R0055 ;LOCK ON HARD ERROR
```

5759
5760
5761
5762
5763
5764
5765
5766
5767
5768
5769
5770
5771
5772
5773
5774
5775
5776
5777
5778
5779
5780
5781
5782
5783
5784
5785
5786
5787
5788
5789
5790
5791
5792
5793
5794
5795
5796
5797
5798

```
; *****  
; .SBTTL T0056 BASIC 'MOV -(R),@#A' TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [144,240,250,173,207,210,200,125,375,016] FC 1,2,4  
;ACT BUTS: 37[004]100,144 / 35[240]120,173 / 22[207]200,200 / 16[125]016,016  
;EXEC: [200]ALUC=LLLLL :[125]D=#DWTA  
;CODES: [125] SPS=3 / N:C=0100  
;SYNC: B05J2 (-) / T= 4 USEC  
;KEY SIG: K3-3 MOVL / K3-3 SM=4L / K3-3 DM=3L / K5-2 PS(2)(1)H  
T0056: MOV #0056,R0 ;LOAD R0 WITH TEST NO.  
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #DWTA,R4 ;RESULT S / B = #DWTA  
R0056: MOV #ATA+2,R5 ;INITIAL SOURCE ADDR = ATA+2  
CLR (R2) ;MAKE [DEST] = 000000  
CCC ;SCOPE SYNC  
I0056: MOV -(R5),@#MBUF0 ;TEST THE MOV  
CMP R4,(R2) ;RESULT = #DWTA ?  
BEQ A0056 ;BR IF YES  
E10056: MOV (R2),R3 ;GET THE WAS DATA  
HALT ;MOV DELIVERED THE WRONG RESULT  
BR R0056 ;LOCK ON HARD ERROR  
A0056: CMP #ATA,R5 ;DID THE SRC REG GET DECREMENTED ?  
BEQ T0057 ;BR IF YES  
E20056: HALT ;MOV FAILED TO UPDATE SOURCE REG  
BR R0056 ;LOCK ON HARD ERROR
```

070140

067544

5799
5800
5801
5802
5803
5804
5805
5806
5807
5808
5809
5810
5811
5812
5813
5814
5815
5816
5817
5818
5819
5820
5821
5822
5823
5824
5825
5826
5827
5828
5829
5830
5831
5832
5833
5834
5835
5836
5837
5838

```

; *****
; .SBTTL T0057 BASIC 'MOV (RA),RB' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,160,204,000] FC 1,2,4

;ACT BUTS: 37[004]100,142 / 35[240]120,160 / 20[160]000,000

;EXEC: [160]ALUC=LLLLL :[204]D=#DWTA

;CODES: [204] SPS=3 / N:C=0100

;SYNC: B05J2 (-) / T= 1.8 USEC

;KEY SIG: K3-3 MOVL / K3-3 SM=2L / K3-3 DM=0L / K5-2 PS(Z)(1)H

T0057: MOV #0057,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #177703,R2 ;DEST ADDR = R3
MOV #DWTA,R4 ;RESULT S / B = #DWTA
R0057: MOV #ATA,R5 ;INITIAL SOURCE ADDR = ATA
CLR R3 ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC

I0057: MOV (R5)+,R3 ;TEST THE MOV

CMP R4,R3 ;RESULT = #DWTA ?
BEQ A0057 ;BR IF YES

E10057: HALT ;MOV DELIVERED WRONG RESULT
BR R0057 ;LOCK ON HARD ERROR

A0057: CMP #ATA+2,R5 ;DID SOURCE REG GET INCREMENTED
BEQ T0060 ;BR IF YES

E20057: HALT ;MOV FAILED TO UPDATE SOURCE REGISTER
BR R0057 ;LOCK ON HARD ERROR

```

070140

067546

006226	012700	000057	
006232	112760	000377	
006240	012702	177703	
006244	012704	067570	
006250	012705	067544	
006254	005003		
006256	000257		
006260	012503		
006262	020403		
006264	001402		
006266	000000		
006270	000767		
006272	022705	067546	
006276	001402		
006300	000000		
006302	000762		

```

5839 ; *****
5840 ; .SBTTL T0060 BASIC 'MOV X(RA),RB' TEST
5841 ; *****
5842 ; MICROPROGRAMMING / LOGIC INFORMATION
5843 ; ROM SEQ: [146,241,242,247,250,160,204,000] FC 1,2,4
5844 ; ACT BUTS: 37[004]100,146 / 35[247]120,160 / 20[160]000,000
5845 ; EXEC: [160]ALUC=LLLLL :[204]D=#DWTB
5846 ; CODES: [204] SPS=3 / N:C=0100
5847 ; SYNC: B05J2 (-) / T= 2.5 USEC
5848 ; KEY SIG: K3-3 MOVL / K3-3 SM=6L / K3-3 DM=0L / K5-2 PS(Z)(1)H
5849
5850
5851
5852
5853
5854
5855
5856
5857 006304 012700 000060 T0060: MOV #0060,R0 ;LOAD R0 WITH TEST NO.
5858 006310 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
5859 006316 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
5860 006322 012704 070114 MOV #DWTB,R4 ;RESULT S / B = #DWTB
5861 006326 012705 067544 MOV #ATA,R5 ;BASE SOURCE ADDR = ATA
5862 006332 005003 R0060: CLR R3 ;MAKE [DEST] = 000000
5863 006334 000257 CCC ;SCOPE SYNC
5864
5865 006336 016503 000002 I0060: MOV 2(R5),R3 ;TEST THE MOV
5866
5867 006342 020403 CMP R4,R3 ;RESULT = #DWTB ?
5868 006344 001402 BEQ T0061 ;BR IF YES
5869
5870 006346 000000 E0060: HALT ;MOV FAILED TO DELIVER CORRECT RESULT
5871 006350 000770 BR R0060 ;LOCK ON HARD ERROR
    
```

5872
5873
5874
5875
5876
5877
5878
5879
5880
5881
5882
5883
5884
5885
5886
5887
5888
5889
5890
5891
5892
5893
5894
5895
5896
5897
5898
5899
5900
5901
5902
5903
5904
5905
5906
5907

```
; *****  
; .SBTTL T0061 BASIC 'MOV @X(RA),RB' TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [147,243,244,245,246,247,250,160,204,000] FC 1,2,4  
;ACT BUTS: 37[004]100,147 / 35[247]120,160 / 20[160]000,000  
;EXEC: [160]ALUC=LLLLL :[204]D=177777  
;CODES: [204] SPS=3 / N:C=1000  
;SYNC: B05J2 (-) / T= 3.4 USEC  
;KEY SIG: K3-3 MOVL / K3-3 SM=7L / K3-3 DM=0L / K5-2 PS(N)(1)H  
T0061: MOV #0061,R0 ;LOAD R0 WITH TEST NO.  
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
CLR R4 ;RESULT S / B = 177777  
COM R4  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #DWTA+2,@#MBUFO+2 ;SET UP ADDRESS TABLE MBUFO  
MOV #MBUFO,R5 ;BASE ADDRESS IN R5  
R0061: CLR R3 ;MAKE [DEST] = 000000  
CCC ;SCOPE SYNC  
I0061: MOV @2(R5),R3 ;TEST THE MOV  
CMP R4,R3 ;RESULT = 177777  
BEQ T0062 ;BR IF YES  
E0061: HALT ;MOV DELIVERED THE WRONG RESULT  
BR R0061 ;LOCK ON HARD ERROR
```

5908
5909
5910
5911
5912
5913
5914
5915
5916
5917
5918
5919
5920
5921
5922
5923
5924
5925
5926
5927
5928
5929
5930
5931
5932
5933
5934
5935
5936
5937
5938
5939
5940
5941
5942
5943
5944
5945
5946
5947
5948
5949
5950

```
; *****  
; .SBTTL T0062 BASIC 'MOV (R)+,X(R)' TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [142,240,250,177,206,212,200,125,375,016] FC 1,2,4  
;ACT BUTS: 37[004]100,142 / 35[240]120,177 / 17[177]212,212 / 21[206]200,200  
; / 16[125]016,016  
;EXEC: [200]ALUC=LLLLL :[125]D=125252  
;CODES: [125] SPS=3 / N:C=1000  
;SYNC: B05J2 (-) / T= 4 USEC  
;KEY SIG: K3-3 MOVL / K3-3 SM=2L / K3-3 DM=6L / K5-2 PS(N)(1)H
```

070140

```
T0062: MOV #0062,R0 ;LOAD R0 WITH TEST NO.  
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #125252,R4 ;RESULT S / B = 125252  
MOV #MBUF1+2,R2 ;FINAL DEST ADDR = MBUF1+2  
MOV R4,@MBUF0 ;SOURCE OPERAND = 125252  
R0062: MOV #MBUF0,R5 ;[R5] = INITIAL SRC ADDR = MBUF0  
CLR (R2) ;MAKE [DEST] = 000000  
CCC ;SCOPE SYNC  
I0062: MOV (R5)+,4(R5) ;TEST THE MOV  
CMP R4,(R2) ;RESULT = 125252 ?  
BEQ A0062 ;BR IF YES  
E10062: MOV (R2),R3 ;GET THE WAS DATA  
HALT ;MOV DELIVERED WRONG RESULT  
BR R0062 ;LOCK ON HARD ERROR  
A0062: CMP #MBUF0+2,R5 ;DID REGISTER GET INCREMENTED ?  
BEQ T0063 ;BR IF YES  
E20062: HALT ;MOV FAILED TO UPDATE REGISTER  
BR R0062 ;LOCK ON HARD ERROR
```

```
5951 ; *****  
5952 ; .SBTTL T0063 BASIC 'CMP R,@VA'' TEST WITH [R] = [A]  
5953 ; *****  
5954 ;  
5955 ;MICROPROGRAMMING / LOGIC INFORMATION  
5956 ;  
5957 ;ROM SEQ: [163,264,265,266,267,224,367,375,016] FC 1,3,8  
5958 ;  
5959 ;ACT BUTS: 37[004]100,163 / 33[266]220,224 / 16[367]016,016  
5960 ;  
5961 ;EXEC: [224]ALUC=LLHHL :[367]D=125252  
5962 ;  
5963 ;CODES: [367] SPS=3 / N:C=1100  
5964 ;  
5965 ;SYNC: B05J2 (-) / T= 3.5 USEC  
5966 ;  
5967 ;KEY SIG: K3-3 CMP L / K3-3 SM=0L / K3-3 DM=3L / K5-2 PS(Z)(1)H  
5968 ; K3-8 BTI + CMP + TST H  
5969 ;  
5970 006514 012700 000063 T0063: MOV #0063,R0 ;LOAD R0 WITH TEST NO.  
5971 006520 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
5972 006526 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
5973 006532 012704 125252 MOV #125252,R4 ;RESULT S / B = 125252  
5974 006536 010405 R0063: MOV R4,R5 ;[R5] = SOURCE OP = 125252  
5975 006540 010412 MOV R4,(R2) ;MAKE [DEST] = 125252  
5976 006542 000257 CCC ;SCOPE SYNC  
5977 ;  
5978 006544 020537 067560 I0063: CMP R5,@MBUF0 ;TEST THE CMP  
5979 ;  
5980 006550 001402 BEQ A0063 ;BR IF 'Z' WAS SET - IT SHOULD BE  
5981 ;  
5982 006552 000000 E10063: HALT ;CMP FAILED TO SET 'Z'  
5983 006554 000770 BR R0063 ;LOCK ON HARD ERROR  
5984 ;  
5985 006556 020412 A0063: CMP R4,(R2) ;IS RESULT STILL = 125252 ?  
5986 006560 001403 BEQ T0064 ;BR IF YES  
5987 ;  
5988 006562 011203 E20063: MOV (R2),R3 ;GET THE WAS DATA  
5989 006564 000000 HALT ;CMP ALTERED [DEST]  
5990 006566 000763 BR R0063 ;LOCK ON HARD ERROR  
5991 ;
```

5992
5993
5994
5995
5996
5997
5998
5999
6000
6001
6002
6003
6004
6005
6006
6007
6008
6009
6010
6011
6012
6013
6014
6015
6016
6017
6018
6019
6020
6021
6022
6023
6024

006570 012700 000064
006574 112760 000377
006602 012702 067560
006606 012704 125252
006612 005005
006614 010412
006616 000277

006620 020537 067560

006624 001002

006626 000000
006630 000770

070140

```
; *****  
; .SBTTL T0064 BASIC 'CMP R,@#A' WITH [R] NOT EQUAL TO [A]  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [163,264,266,267,224,367,375,016] FC 1,3,8  
  
;ACT BUTS: 37[004]100,163 / 33[266]220,224 / 16[367]016,016  
  
;EXEC: [224]ALUC=LLHHL :[367]D=052526  
  
;CODES: [367] SPS=3 / N:C=1000  
  
;SYNC: B05J2 (-) / T= 3.5 USEC  
  
;KEY SIG: K3-3 MOVL / K3-3 SM=0L / K3-3 DM=3L / K3-8 BIT+CMP+TSTH  
  
T0064: MOV #0064,R0 ;LOAD R0 WITH TEST NO.  
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #125252,R4 ;MAKE RESULT S / B = 125252  
R0064: CLR R5 ;[R5] = SOURCE OP = 000000  
MOV R4,(R2) ;MAKE [DEST] = 125252  
SCC ;SCOPE SYNC - MAKE Z=1  
  
I0064: CMP R5,@#MBUF0 ;TEST THE CMP  
  
BNE T0065 ;BR IF Z=0 - IT SHOULD BE  
  
E0064: HALT ;CMP FAILED TO CLEAR 'Z'  
BR R0064 ;LOCK ON HARD ERROR
```

```

6025 ; *****
6026 ; .SBTTL T0065 BASIC 'CMP #N,@#A' TEST WITH [A] = N
6027 ; *****
6028 ;MICROPROGRAMMING / LOGIC INFORMATION
6029 ;ROM SEQ: [142,240,250,163,264,265,266,267,225,367,375,016] FC 1,2,3,8
6030 ;ACT BUTS: 37[004]100,142 / 35[240]120,163 / 33[266]220,225 / 16[367]016,016
6031 ;EXEC: [225]ALUC=LLHHL :[367]D=000000
6032 ;CODES: [367] SPS=3 / N:C=0100
6033 ;SYNC: B05J2 (-) / T= 4.3 USEC
6034 ;KEY SIG: K3-3 CMPL / K3-3 SM=2L / K3-3 DM=3L / K5-2 PS(Z)(1)H
6035 ; K3-8 BIT+CMPL+TSTH
6036
6037 T0065: MOV #0065,R0 ;LOAD R0 WITH TEST NO.
6038 MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
6039 MOV #125252,R4 ;RESULT S / B = 125252
6040 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
6041 R0065: MOV R4,(R2) ;MAKE [DEST] = 125252
6042 CCC ;SCOPE SYNC - Z=0
6043
6044 006632 012700 000065 T0065: CMP #125252,@#MBUF0 ;TEST THE CMP
6045 006636 112760 000377 070140 BEQ T0066 ;BR IF Z=1 - IT SHOULD BE
6046 006644 012704 125252
6047 006650 012702 067560
6048 006654 010412
6049 006656 000257
6050
6051 006660 022737 125252 067560 I0065: HALT ;CMP FAILED TO SET 'Z'
6052
6053 006666 001402 BR R0065 ;LOCK ON HARD ERROR
6054
6055 006670 000000
6056 006672 000770
6057

```

```

6058 ; *****
6059 ; .SBTTL T0066 BASIC 'CMP #N,@#A' TEST WITH [A] NOT EQUAL TO #N
6060 ; *****
6061 ;MICROPROGRAMMING / LOGIC INFORMATION
6062 ;ROM SEQ: [142,240,250,163,264,265,266,267,225,367,375,016] FC 1,2,3,8
6063 ;ACT BUTS: 37[004]100,142 / 35[240]120,163 / 33[266]220,225 / 16[367]016,016
6064 ;EXEC: [225]ALUC=LLHHL :[367]D=125252
6065 ;CODES: [367] SPS=3 / N:C=1000
6066 ;SYNC: B05J2 (-) / T= 4.3 USEC
6067 ;KEY SIG: K3-3 CMPL / K3-3 SM=2L / K3-3 DM=3L / K5-2 PS(N)(1)H
6068 ; K3-8 BIT+CMPT+TSTH
6069
6070 006674 012700 000066 T0066: MOV #0066,R0 ;LOAD R0 WITH TEST NO.
6071 006700 112760 000377 070140 T0066: MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
6072 006706 012704 125252 T0066: MOV #125252,R4 ;RESULT S / B=125252
6073 006712 012702 067560 T0066: MOV #MBUF0,R2 ;DEST ADDR = MBUF0
6074 006716 005012 R0066: CLR (R2) ;MAKE [DEST] = 000000
6075 006720 000277 R0066: SCC ;SCOPE SYNC - Z=1
6076
6077 006722 022737 125252 067560 I0066: CMP #125252,@#MBUF0 ;TEST THE CMP
6078 006730 001002 I0066: BNE T0067 ;BR IF Z=0 - IT SHOULD BE
6079
6080 006732 000000 E0066: HALT ;CMP FAILED TO CLEAR 'Z'
6081 006734 000770 E0066: BR R0066 ;LOCK ON HARD ERROR
6082
6083
6084
6085
6086
6087
6088
6089
6090

```

6091
6092
6093
6094
6095
6096
6097
6098
6099
6100
6101
6102
6103
6104
6105
6106
6107
6108
6109
6110
6111
6112
6113
6114
6115
6116
6117
6118
6119
6120
6121
6122
6123
6124

; *****
; .SBTTL T0067 BASIC 'BIS #N,@#A' TEST - N=177777,[A]=000000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,163,264,265,266,267,225,367,375,016] FC 1,3,8

;ACT BUTS: 37[004]100,142 / 35[240]120,163 / 33[266]220,225 / 16[367]016,016

;EXEC: [225]ALUC LLLH :[367]D=177777

;CODES: [367] SPS=3 / N:C=1000

;SYNC: B05J2 (-) / T= 4.3 USEC

;KEY SIG: K3-5 DOPL / K3-3 DM=3L / K3-3 BIS L

006736 012700 000067
006742 112760 000377 070140
006750 012702 067560
006754 012704 177777
006760 005012
006762 000257
006764 052737 177777 067560
006772 020412
006774 001403
006776 011203
007000 000000
007002 000766

T0067: MOV #0067,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #M0067,R2 ;DEST ADDR = M0067
MOV #-1,R4 ;RESULT S / B = 177777
R0067: CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
I0067: BIS #-1,@M0067 ;TEST THE BIS
CMP R4,(R2) ;RESULT OK?
BEQ T0070 ;BR IF YES
E0067: MOV (R2),R3 ;GET WAS DATA
HALT ;BIS FAILED TO SET ALL BITS IN BITFLG
BR R0067 ;LOCK ON HARD ERROR

6125
6126
6127
6128
6129
6130
6131
6132
6133
6134
6135
6136
6137
6138
6139
6140
6141
6142
6143
6144
6145
6146
6147
6148
6149
6150
6151
6152
6153
6154
6155
6156
6157
6158

: .SBTTL T0070 BASIC 'BIC #N,@#A' TEST

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [142,240,250,163,264,265,266,267,225,367,375,016] FC 1,2,3,8
:ACT BUTS: 37[004]100,142 / 35[240]120,163 / 33[266]220,225 / 16[367]016,016
:EXEC: [225]ALUC=HLLHL :[367]D=000077
:CODES: [367] SPS=3 / N:C=0000
:SYNC: B05J2 (-) / T= 4.3 USEC
:KEY SIG: K3-3 BICL / K3-3 SM=2L / K3-3 DM=3L

007004 012700 000070
007010 112760 000377 070140
007016 012702 067560
007022 012704 000077
007026 012712 177777
007032 000257
007034 042737 177700 067560
007042 020412
007044 001403
007046 011203
007050 000000
007052 000765

T0070: MOV #0070,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #M0070,R2 ;DEST ADDR = M0070
MOV #77,R4 ;RESULT S / B = 77
R0070: MOV #-1,(R2) ;MAKE [DEST] = 177777
CCC ;SCOPE SYNC
I0070: BIC #177700,@M0070 ;TEST THE BIC
CMP R4,(R2) ;DID RESULT = 77 ?
BEQ T0071 ;BR IF YES
E0070: MOV (R2),R3 ;GET THE WAS DATA
HALT ;BIC DELIVERED THE WRONG RESULT
BR R0070 ;LOCK ON HARD ERROR

6159
6160
6161
6162
6163
6164
6165
6166
6167
6168
6169
6170
6171
6172
6173
6174
6175
6176
6177
6178
6179
6180
6181
6182
6183
6184
6185
6186
6187
6188
6189
6190
6191
6192

; *****
; .SBTTL T0071 BASIC 'BIC #N,R' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,120,371,360,000] FC 1,2,8
;ACT BUTS: 35[240]120,120 / 31[120]360,360 / 27[371]000,000
;EXEC: [371]JALUC HLLHL :[360]D=377
;CODES: [360] SPS=3 / N:C=0000
;SYNC: B05J2 (-) / T= 4.3 USEC
;KEY SIG: K3-5 DOPL / K3-3 DM=0L / K3-3 BIC L

007054 012700 000071
007060 112760 000377 070140
007066 012704 000377
007072 012702 177703
007076 005003
007100 005103
007102 000257
007104 042703 177400
007110 020304
007112 001402
007114 000000
007116 000767

T0071: MOV #0071,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #377,R4 ;RESULT S / B = 377
MOV #177703,R2 ;DEST ADDR = R3
R0071: CLR R3 ;[DEST] = 177777
COM R3
CCC ;SCOPE SYNC
I0071: BIC #177400,R3 ;TEST THE BIC
CMP R3,R4 ;RESULT OK?
BEQ T0072 ;BR IF YES
E0071: HALT ;BIC FAILED TO CLEAR HI-BYTE
BR R0071 ;LOCK ON HARD ERROR

```
6193 ; *****
6194 ; .SBTTL T0072 BASIC 'BIC #N,2(SP)' TEST
6195 ; *****
6196
6197 ;MICROPROGRAMMING / LOGIC INFORMATION
6198
6199 ;ROM SEQ: [142,240,250,167,261,262,266,267,225,367,375,016] FC 1,2,3,8
6200
6201 ;ACT BUTS: 37[004]100,142 / 35[250]120,167 / 17[167]262,262 / 33[266]220,225
6202 ; / 16[367]016,016
6203
6204 ;EXEC: [225]ALUC HLLHL :[367]D=357
6205
6206 ;CODES: [367] SPS=3 / N:C=0000
6207
6208 ;SYNC: B05J2 (-) / T= 3.3 USEC
6209
6210 ;KEY SIG: K3-5 DOPL / K3-3 DM=6 / K3-3 BIC L
6211
6212 007120 012700 000072 T0072: MOV #0072,R0 ;LOAD R0 WITH TEST NO.
6213 007124 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
6214 007132 012704 000357 MOV #357,R4 ;RESULT S / B = 357
6215 007136 010605 SP,R5 ;SAVE SP
6216 007140 010506 R0072: MOV R5,SP ;RESET SP FOR ERROR LOOP
6217 007142 012746 000377 MOV #377,-(SP) ;[DEST] = 377 PUT ON STACK
6218 007146 005746 TST -(SP) ;DECREMENT SP
6219 007150 000257 CCC ;SCOPE SYNC
6220
6221 007152 042766 000020 000002 I0072: BIC #20,2(SP) ;TEST THE BIC - CLEAR BIT 4
6222
6223 007160 010602 MOV SP,R2 ;[R2] = DEST ADDR
6224 007162 005722 TST (R2)+
6225 007164 020412 CMP R4,(R2) ;RESULT = 357?
6226 007166 001403 BEQ A0072 ;BR IF YES
6227
6228 007170 011203 MOV (R2),R3 ;GET WAS DATA
6229 007172 000000 E0072: HALT ;BIC FAILED TO CLR BIT2 OF DEST
6230 007174 000761 BR R0072 ;LOCK ON HARD ERROR
6231
6232 007176 010506 A0072: MOV R5,SP
6233
```

6234
6235
6236
6237
6238
6239
6240
6241
6242
6243
6244
6245
6246
6247
6248
6249
6250
6251
6252
6253
6254
6255
6256
6257
6258
6259
6260
6261
6262
6263
6264
6265

```
; *****  
; .SBTTL T0073 BASIC 'BIT #N,@#A' WITH BIT SET IN 'A'  
; *****  
;MICRO PROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [142,240,250,163,264,265,266,267,225,367,375,016] FC 1,2,3,8  
;ACT BUTS: 37[004]100,142 / 35[240]120,163 / 33[266]220,225 / 16[367]016,016  
;EXEC: [225]ALUC=HHLHH :[367]D=040000  
;CODES: [367] SPS=3 / N:C=0000  
;SYNC: B05J2 (-) / T= 4.3 USEC  
;KEY SIG: K3-3 BITL / K3-3 SM=2L / K3-3 DM=3L / K3-8 BIT+CMP+TSTH  
T0073: MOV #0073,R0 ;LOAD R0 WITH TEST NO.  
;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #377,STAB1(R0)  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #40000,R4 ;RESULT S / B = 40000  
R0073: MOV R4,(R2) ;MAKE [DEST] = 40000  
;SCOPE SYNC - Z=1  
;SCC  
I0073: BIT #40000,@MBUF0 ;TEST THE BIT  
;BNE T0074 ;BR IF Z=0 - IT SHOULD BE  
E0073: HALT ;BIT FAILED TO CLEAR 'Z'  
BR R0073 ;LOCK ON HARD ERROR
```

T0073 BASIC 'BIT #N,@#A' WITH BIT SET IN 'A'

SEQ 0173

```
6266 ; *****
6267 ; .SBTTL T0074 BASIC 'BIT #N,@#A' WITH BIT CLEAR IN 'A'
6268 ; *****
6269
6270 ;MICROPROGRAMMING / LOGIC INFORMATION
6271
6272 ;ROM SEQ: [142,240,250,163,264,265,266,267,225,367,375,016] FC 1,2,3,8
6273
6274 ;ACT BUTS: 37[004]100,142 / 35[240]120,163 / 33[266]220,225 / 16[367]016,016
6275
6276 ;EXEC: [225]ALUC=HHLHH :[367]D=000000
6277
6278 ;CODES: [367] SPS=3 / N:C=0100
6279
6280 ;SYNC: B05J2 (-) / T= 4.3 USEC
6281
6282 ;KEY SIG: K3-3 BITL / K3-3 SM=2L / K3-3 DM=3L / K3-8 BIT+CMP+TSTH
6283 ; K5-2 PS(Z)(1)H
6284
6285 007242 012700 000074 T0074: MOV #0074,R0 ;LOAD R0 WITH TEST NO.
6286 007246 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
6287 007254 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
6288 007260 005004 CLR R4 ;RESULT S / B = 000000
6289 007262 005012 R0074: CLR (R2) ;MAKE [DEST] = 000000
6290 007264 000257 CCC ;SCOPE SYNC - Z=0
6291
6292 007266 032737 040000 067560 I0074: BIT #40000,@#MBUFO ;TEST THE BIT
6293
6294 007274 001402 BEQ A0074 ;BR IF Z=1 - IT SHOULD BE
6295
6296 007276 000000 E10074: HALT ;BIT FAILED TO SET 'Z'
6297 007300 000770 BR R0074 ;LOCK ON HARD ERROR
6298
6299 007302 020412 A0074: CMP R4,(R2) ;DID BIT DELIVER A RESULT
6300 007304 001403 BEQ T0075 ;BR IF NOT
6301
6302 007306 011203 E20074: MOV (R2),R3 ;GET THE WAS DATA
6303 007310 000000 HALT ;BIT DISTURBED THE [DEST]
6304 007312 000763 BR R0074 ;LOCK ON HARD ERROR
6305
```

```
6306 ; *****  
6307 ; .SBTTL T0075 BASIC 'ADD #N,(R) ' TEST  
6308 ; *****  
6309 ;MICROPROGRAMMING / LOGIC INFORMATION  
6310 ;ROM SEQ: [142,240,250,161,266,267,225,367,375,016] FC 1,2,3,8  
6311 ;ACT BUTS: 37[004]100,142 / 35[240]120,161 / 33[266]220,225 / 16[367]016,016  
6312 ;EXEC: [225]ALUC=LHLLH :[367]D=000004  
6313 ;CODES: [367] SPS=3 / N:C=0000  
6314 ;SYNC: B05J2 (-) / T=3.4 USEC  
6315 ;KEY SIG: K3-3 ADD+SUBL / K3-3 SM=2L / K3-3 DM=1L  
6316  
6317  
6318  
6319  
6320  
6321  
6322  
6323  
6324 007314 012700 000075 T0075: MOV #0075,R0 ;LOAD R0 WITH TEST NO.  
6325 007320 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
6326 007326 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
6327 007332 012704 000004 MOV #4,R4 ;RESULT S / B = 4  
6328 007336 012712 000002 R0075: MOV #2,(R2) ;MAKE [DEST] = 2  
6329 007342 000257 CCC ;SCOPE SYNC  
6330  
6331 007344 062712 000002 I0075: ADD #2,(R2) ;TEST THE ADD  
6332  
6333 007350 020412 CMP R4,(R2) ;RESULT = 4 ?  
6334 007352 001403 BEQ T0076 ;BR IF YES  
6335  
6336 007354 011203 E0075: MOV (R2),R3 ;GET THE WAS DATA  
6337 007356 000000 HALT ;ADD DELIVERED THE WRONG RESULT  
6338 007360 000766 BR R0075 ;LOCK ON HARD ERROR  
6339
```

6340
6341
6342
6343
6344
6345
6346
6347
6348
6349
6350
6351
6352
6353
6354
6355
6356
6357
6358
6359
6360
6361
6362
6363
6364
6365
6366
6367
6368
6369
6370
6371
6372
6373
6374
6375
6376

; *****
; .SBTTL T0076 BASIC 'ADD #N,X(R)'' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,167,261,262,266,267,225,367,375,016] FC 1,2,3,8

;ACT BUTS: 37[004]100,142 / 35[240]120,167 / 17[167]262,262 / 33[266]220,225
; / 16[367]016,016

;EXEC: [225]ALUC=LHLLH :[367]D=000004

;CODES: [367] SPS=3 / N:C=0000

;SYNC: B05J2 (-) / T= 3.4 USEC

;KEY SIG: K3-3 ADD+SUBL / K3-3 SM=2L / K3-3 DM=6L

007362 012700 000076
007366 112760 000377 070140
007374 012704 000002
007400 012702 067562
007404 012705 067560
007410 005012
007412 000257
007414 062765 000002 000002
007422 020412
007424 001403
007426 011203
007430 000000
007432 000764

T0076: MOV #0076,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #2,R4 ;RESULT S / B = 4
MOV #MBUFO+2,R2 ;DEST ADDR = MBUFO + 2
R0076: MOV #MBUFO,R5 ;BASE DEST ADDR = MBUFO
CLR (R2) ;MAKE [DEST] = 000000
CCC ;SCOPE SYNC
I0076: ADD #2,2(R5) ;TEST THE ADD
CMP R4,(R2) ;RESULT = 4 ?
BEQ T0077 ;BR IF YES
E0076: MOV (R2),R3 ;GET THE WAS DATA
HALT ;ADD DELIVERED THE WRONG RESULT
BR R0076 ;LOOP ON HARD ERROR

6377
6378
6379
6380
6381
6382
6383
6384
6385
6386
6387
6388
6389
6390
6391
6392
6393
6394
6395
6396
6397
6398
6399
6400
6401
6402
6403
6404
6405
6406
6407
6408
6409
6410
6411
6412
6413
6414
6415
6416
6417
6418
6419
6420
6421
6422
6423
6424
6425
6426
6427
6428
6429

007434 012700 000077
007440 112760 000377 070140
007446 032737 000002 066642
007454 001401
007456 000000
007460 012704 177400
007464 010605
007466 010602
007470 005742
007472 010506
007474 010446
007476 000257
007500 122726 000000
007504 001403
007506 011203
007510 000000
007512 000767
007514 020506
007516 001402
007520 000000
007522 000763
007524 020412
007526 001403
007530 011203
007532 000000
007534 000756

; *****
; .SBTTL T0077 BASIC 'CMPB #N,(SP)+' TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,162,260,267,225,367,375,016] FC 1,2,3,8
;ACT BUTS: 37[004]100,142 / 35[240]120,162 / 33[260]220,225 / 16[367]016,016
;EXEC: [225]ALUC=LLHHL :[367]D=177400
;CODES: [367] SPS=3 / N:C=0100
;SYNC: B05J2 (-) / T= 4 USEC
;KEY SIG: K3-3 CMPL / K3-3 DM=2L / K3-6 BYTE INSTR H

T0077: MOV #0077,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
BIT #2,@#BPTLOC ;BREAKPOINT HALT SET ??
BEQ .+4 ;BR IF NOT
HALT ;BREAK - DEPRESS CONTINUE TO RESTART
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
R0077: MOV R5,SP ;RESET SP FOR ERROR LOOP

MOV R4,-(SP) ;MAKE [DEST] = 177400
CCC ;SCOPE SYNC - 'Z' = 0

I0077: CMPB #0,(SP)+ ;TEST THE CMPB

BEQ A0077 ;BR IF 'Z' SET - IT SHOULD BE

MOV (R2),R3 ;GET WAS DATA
E10077: HALT ;CMPB FAILED TO SET 'Z'
BR R0077 ;LOCK ON HARD ERROR

A0077: CMP R5,SP ;DID SP GET UPDATED BY ??
BEQ B0077 ;BR IF YES

E20077: HALT ;CMPB FAILED TO UPDATE SP PROPERLY
BR R0077 ;LOCK ON HARD ERROR

B0077: CMP R4,(R2) ;[DEST] ALTERED?
BEQ T0100 ;BR IF NOT

MOV (R2),R3 ;GET WAS DATA
E30077: HALT ;CMPB MODIFIED [DEST]
BR R0077 ;LOCK ON HARD ERROR.

```
6430 ; *****  
6431 ; .SBTTL T0100 BASIC 'CMPB (RA)+,(RB)+' - SRC AND DEST EVEN  
6432 ; *****  
6433  
6434 ;MICROPROGRAMMING / LOGIC INFORMATION  
6435  
6436 ;ROM SEQ: [142,240,250,162,260,267,225,367,375,016] FC 1,2,3,8  
6437  
6438 ;ACT BUTS: 37[004]100,142 / 35[240]120,162 / 33[260]220,225 / 16[367]016,016  
6439  
6440 ;EXEC: [225]ALUC=LLHHL :[326]D=177400  
6441  
6442 ;CODES: [367] SPS=3 / N:C=0100  
6443  
6444 ;SYNC: B05J2 (-) / T= 4.5 USEC  
6445  
6446 ;KEY SIG: K3-3 CMPL / K3-3 DM=2L / K3-6 BYTE INSTR H  
6447  
6448 007536 012700 000100 T0100: MOV #0100,R0 ;LOAD R0 WITH TEST NO.  
6449 007542 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
6450 007550 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777  
6451 007554 012702 067572 MOV #DWTA+2,R2 ;DEST ADDR = DWTA+2  
6452 007560 012705 067576 R0100: MOV #DWTA+6,R5 ;SRC ADDR = DWTA+6  
6453 007564 010203 MOV R2,R3 ;R3 GETS DEST ADDR  
6454 007566 000257 CCC ;SCOPE SYNC  
6455  
6456 007570 122523 I0100: CMPB (R5)+,(R3)+ ;TEST THE CMPB  
6457  
6458 007572 001402 BEQ A0100 ;BR IF 'Z' = 1 - IT SHOULD BE  
6459  
6460 007574 000000 E10100: HALT ;CMPB FAILED TO SET 'Z'  
6461 007576 000770 BR R0100 ;LOCK ON HARD ERROR  
6462  
6463 007600 022703 067573 A0100: CMP #DWTA+3,R3 ;DID DEST REG GET UPDATED?  
6464 007604 001402 BEQ B0100 ;BR IF YES  
6465  
6466 007606 000000 E20100: HALT ;CMPB FAILED TO UPDATE DEST REG  
6467 007610 000763 BR R0100 ;LOCK ON HARD ERROR  
6468  
6469 007612 022705 067577 B0100: CMP #DWTA+7,R5 ;DID SRC REG GET UPDATED?  
6470 007616 001402 BEQ C0100 ;BR IF YES  
6471  
6472 007620 000000 E30100: HALT ;CMPB FAILED TO UPDATE SRC REG  
6473 007622 000756 BR R0100 ;LOCK ON HARD ERROR  
6474  
6475 007624 020412 C0100: CMP R4,(R2) ;DID [DEST] GET ALTERED?  
6476 007626 001404 BEQ T0101 ;BR IF NOT  
6477  
6478 007630 011203 E40100: MOV (R2),R3 ;GET WAS DATA  
6479 007632 000000 HALT ;CMPB DELIVERED A RESULT  
6480 007634 010412 MOV R4,(R2) ;RESTORE [DEST]  
6481 007636 000750 BR R0100 ;LOCK ON HARD ERROR  
6482
```

6483
6484
6485
6486
6487
6488
6489
6490
6491
6492
6493
6494
6495
6496
6497
6498
6499
6500
6501
6502
6503
6504
6505
6506
6507
6508
6509
6510
6511
6512
6513
6514
6515
6516
6517
6518
6519
6520
6521
6522
6523
6524
6525
6526
6527
6528
6529
6530
6531
6532
6533
6534
6535
6536

; *****
; .SBTTL T0101 BASIC 'CMPB (RA)+,(RB)+' - SRC AND DEST ODD
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,137,251,162,260,267,237,270,231,254,074,366,375,016] FC 1,2
;ACT BUTS: 37[004]100,142 / 35[240]120,137 / 36[137]120,162 / 33[260]220,237
; / 34[237]220,231 / 16[366]016,016
;EXEC: [231]ALUC=LLHHL :[254]D=000000
;CODES: [074] SPS=3 / N:C=0100
;SYNC: B05J2 (-) / T= 4.5 USEC
;KEY SIG: K3-3 CMPL / K3-3 DM=2L / K3-6 BYTE INSTR H

007640 012700 000101
007644 112760 000377
007652 012704 177777
007656 012702 067572
007662 012705 067575
007666 012703 067573
007672 000257

007674 122523
007676 001402
007700 000000
007702 000767
007704 022703 067574
007710 001402
007712 000000
007714 000762
007716 022705 067576
007722 001402
007724 000000
007726 000755
007730 020412
007732 001404
007734 011203
007736 000000
007740 010412
007742 000747

070140

T0101: MOV #0101,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #-1,R4 ;RESULT S / B = 177777
R0101: MOV #DWTA+2,R2 ;DEST ADDR = DWTA+2
MOV #DWTA+5,R5 ;SRC ADDR = DWTA+5
MOV #DWTA+3,R3 ;R3 GETS DEST ADDR+1
CCC ;SCOPE SYNC

I0101: CMPB (R5)+,(R3)+ ;TEST THE CMPB
BEQ A0101 ;BR IF 'Z' = 1 - IT SHOULD BE

E10101: HALT ;CMPB FAILED TO SET 'Z'
BR R0101 ;LOCK ON HARD ERROR

A0101: CMP #DWTA+4,R3 ;DID DEST REG GET UPDATED?
BEQ B0101 ;BR IF YES

E20101: HALT ;CMPB FAILED TO UPDATE DEST REG
BR R0101 ;LOCK ON HARD ERROR

B0101: CMP #DWTA+6,R5 ;DID SRC REG GET UPDATED?
BEQ C0101 ;BR IF YES

E30101: HALT ;CMPB FAILED TO UPDATE SRC REG
BR R0101 ;LOCK ON HARD ERROR

C0101: CMP R4,(R2) ;DID [DEST] GET ALTERED?
BEQ T0102 ;BR IF NOT

E40101: MOV (R2),R3 ;GET WAS DATA
HALT ;CMPB DELIVERED A RESULT
MOV R4,(R2) ;RESTORE [DEST]
BR R0101 ;LOCK ON HARD ERROR

6537
6538
6539
6540
6541
6542
6543
6544
6545
6546
6547
6548
6549
6550
6551
6552
6553
6554
6555
6556
6557
6558
6559
6560
6561
6562
6563
6564
6565
6566
6567
6568
6569
6570
6571
6572
6573
6574
6575
6576
6577
6578
6579
6580
6581
6582
6583
6584
6585
6586
6587
6588
6589
6590

: *****
: .SBTTL T0102 BASIC 'CMPB (RA)+,(RB)+' - SRC / EVEN,DEST / ODD
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [142,240,250,162,260,267,237,270,231,254,074,366,375,016] FC 1,2,3,8

:ACT BUTS: 37[004]100,142 / 35[240]120,162 / 33[260]220,237 / 34[237]220,231
: / 16[366]016,016

:EXEC: [231]ALUC=LLHHL :[367]D=000400

:CODES: [074] SPS=3 / N:C=0100

:SYNC: B05J2 (-) / T= 4.5 USEC

:KEY SIG: K3-3 CMPL / K3-3 DM=2 / K3-6 BYTE INSTR H

007744 012700 000102
007750 112760 000377
007756 012704 177400
007762 012702 067574
007766 012705 067576
007772 012703 067575
007776 000257
010000 122523
010002 001402
010004 000000
010006 000767
010010 022703 067576
010014 001402
010016 000000
010020 000762
010022 022705 067577
010026 001402
010030 000000
010032 000755
010034 020412
010036 001404
010040 011203
010042 000000
010044 010412
010046 000747

070140

T0102: MOV #0102,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #177400,R4 ;RESULT S / B = 177400
R0102: MOV #DWTA+4,R2 ;DEST ADDR = DWTA+4
MOV #DWTA+6,R5 ;SRC ADDR = DWTA+6
MOV #DWTA+5,R3 ;R3 GETS DEST ADDR
CCC ;SCOPE SYNC
I0102: CMPB (R5)+,(R3)+ ;TEST THE CMPB
BEQ A0102 ;BR IF 'Z' = 1 - IT SHOULD BE
E10102: HALT ;CMPB FAILED TO SET 'Z'
BR R0102 ;LOCK ON HARD ERROR
A0102: CMP #DWTA+6,R3 ;DID DEST REG GET UPDATED?
BEQ B0102 ;BR IF YES
E20102: HALT ;CMPB FAILED TO UPDATE DEST REG
BR R0102 ;LOCK ON HARD ERROR
B0102: CMP #DWTA+7,R5 ;DID SRC REG GET UPDATED?
BEQ C0102 ;BR IF YES
E30102: HALT ;CMPB FAILED TO UPDATE SRC REG
BR R0102 ;LOCK ON HARD ERROR
C0102: CMP R4,(R2) ;DID [DEST] GET ALTERED?
BEQ T0103 ;BR IF NOT
E40102: MOV (R2),R3 ;GET WAS DATA
HALT ;CMPB DELIVERED A RESULT
MOV R4,(R2) ;RESTORE [DEST]
BR R0102 ;LOCK ON HARD ERROR

```
6591 ; *****
6592 ; .SBTTL T0103 BASIC 'CMPB (RA)+,(RB)+' - SRC / ODD,DEST / EVEN
6593 ; *****
6594
6595 ;MICROPROGRAMMING / LOGIC INFORMATION
6596
6597 ;ROM SEQ: [142,240,250,137,251,162,260,267,225,367,375,016] FC 1,2,3,8
6598
6599 ;ACT BUTS: 37[004]100,142 / 35[240]120,137 / 36[137]120,162 / 33[260]220,225
6600 ; / 16[367]016,016
6601
6602 ;EXEC: [225]ALUC=LLHHL :[367]D=000000
6603
6604 ;CODES: [367] SPS=3 / N:C=0100
6605
6606 ;SYNC: B05J2 (-) / T= 4.5 USEC
6607
6608 ;KEY SIG: K3-3 CMPL / K3-3 DM=2 / K3-6 BYTE INSTR H
6609
6610 010050 012700 000103 T0103: MOV #0103,R0 ;LOAD R0 WITH TEST NO.
6611 010054 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
6612 010062 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777
6613 010066 012702 067572 MOV #DWTA+2,R2 ;DEST ADDR = DWTA+2
6614 010072 012705 067575 R0103: MOV #DWTA+5,R5 ;SRC ADDR = DWTA+5
6615 010076 010203 MOV R2,R3 ;R3 GETS DEST ADDR
6616 010100 000257 CCC ;SCOPE SYNC
6617
6618 010102 122523 I0103: CMPB (R5)+,(R3)+ ;TEST THE CMPB
6619
6620 010104 001402 BEQ A0103 ;BR IF 'Z' = 1 - IT SHOULD BE
6621
6622 010106 000000 E10103: HALT ;CMPB FAILED TO SET 'Z'
6623 010110 000770 BR R0103 ;LOCK ON HARD ERROR
6624
6625 010112 022703 067573 A0103: CMP #DWTA+3,R3 ;DID DEST REG GET UPDATED?
6626 010116 001402 BEQ B0103 ;BR IF YES
6627
6628 010120 000000 E20103: HALT ;CMPB FAILED TO UPDATE DEST REG
6629 010122 000763 BR R0103 ;LOCK ON HARD ERROR
6630
6631 010124 022705 067576 B0103: CMP #DWTA+6,R5 ;DID SRC REG GET UPDATED?
6632 010130 001402 BEQ C0103 ;BR IF YES
6633
6634 010132 000000 E30103: HALT ;CMPB FAILED TO UPDATE SRC REG
6635 010134 000756 BR R0103 ;LOCK ON HARD ERROR
6636
6637 010136 020412 C0103: CMP R4,(R2) ;DID [DEST] GET ALTERED?
6638 010140 001404 BEQ T0104 ;BR IF NOT
6639
6640 010142 011203 E40103: MOV (R2),R3 ;GET WAS DATA
6641 010144 000000 HALT ;CMPB DELIVERED A RESULT
6642 010146 010412 MOV R4,(R2) ;RESTORE [DEST]
6643 010150 000750 BR R0103 ;LOCK ON HARD ERROR
6644
```

```
6645 ; *****
6646 ; .SBTTL T0104 BASIC 'MOVB (RA)+,X(RB) - SRC EVEN / DEST EVEN
6647 ; *****
6648
6649 ;MICROPROGRAMMING / LOGIC INFORMATION
6650
6651 ;ROM SEQ: [142,240,250,177,206,212,202,205,125,375,016] FC 1,2,4
6652
6653 ;ACT BUTS: 37[004]100,142 / 35[240]120,177 / 17[177]212,212 / 21[206]200,202
6654 ; / 16[125]016,016
6655
6656 ;EXEC: [205]ALUC=LLLLL :[125]D=000000
6657
6658 ;CODES: [125] SPS=3 / N:C=0100
6659
6660 ;SYNC: B05J2 (-) / T=4.5 USEC
6661
6662 ;KEY SIG: K3-3 MOVL / K3-3 DM=6L / K3-5 DOPL / K5-? PS(Z)(1)H
6663 ; K3-6 BYTE INSTR H
6664
6665 010152 012700 000104 T0104: MOV #0104,R0 ;LOAD R0 WITH TEST NO.
6666 010156 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
6667 010164 012702 067564 MOV #MBUF1,R2 ;DEST ADDR = MBUF1
6668 010170 012703 067560 MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0
6669 010174 012704 177400 MOV #177400,R4 ;RESULT S / B = 177400
6670 010200 012705 070130 R0104: MOV #DBTA,R5 ;SRC ADDR = DBTA
6671 010204 012712 177777 MOV #-1,(R2) ;[DEST] = 177777
6672 010210 000257 CCC ;SCOPE SYNC
6673
6674 010212 112563 000004 I0104: MOVB (R5)+,4(R3) ;TEST THE MOVB
6675
6676 010216 020412 CMP R4,(R2) ;RESULT OK?
6677 010220 001403 BEQ A0104 ;BR IF YES
6678
6679 010222 011203 MOV (R2),R3 ;GET WAS DATA
6680 010224 000000 E10104: HALT ;MOV DELIVERED WRONG RESULT
6681 010226 000764 BR R0104 ;LOCK ON HARD ERROR
6682
6683 010230 022705 070131 A0104: CMP #DBTA+1,R5 ;DID SRC REG GET INCREMENTED BY +1
6684 010234 001402 BEQ T0105 ;BR IF YES
6685
6686 010236 000000 E20104: HALT ;MOVB FAILED TO UPDATE SRC REG
6687 010240 000757 BR R0104 ;LOCK ON HARD ERROR
```

6688
6689
6690
6691
6692
6693
6694
6695
6696
6697
6698
6699
6700
6701
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730

010242 012700 000105
010246 112760 000377
010254 012702 067564
010260 012703 067560
010264 012704 000777
010270 012705 070135
010274 012712 177777
010300 000257
010302 112563 000005
010306 020412
010310 001403
010312 011203
010314 000000
010316 000764
010320 022705 070136
010324 001402
010326 000000
010330 000757

070140

: *****
: .SBTTL T0105 BASIC 'MOVB (RA)+,X(RB) - SRC ODD / DEST ODD
: *****
:MICROPROGRAMMING / LOGIC INFORMATION
:ROM SEQ: [142,240,250,137,251,177,206,212,202,205,125,375,016] FC 1,2,4
:ACT BUTS: 37[004]100,142 / 35[240]120,137 / 36[137]120,177 / 17[177]212,212
: / 21[206]200,202 / 16[125]016,016
:EXEC: [205]ALUC=LLLLL :[125]D=177777
:CODES: [125] SPS=3 / N:C=1000
:SYNC: B05J2 (-) / T=4.5 USEC
:KEY SIG: K3-3 MOVL / K3-3 DM=6L / K3-5 DOPL / K5-2 PS (N)(1) H
: K3-6 BYTE INSTR H

T0105: MOV #0105,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUF1,R2 ;DEST ADDR = MBUF1
MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0
MOV #777,R4 ;RESULT S / B = 777
R0105: MOV #DBTB+1,R5 ;SRC ADDR = DBTB+1
MOV #-1,(R2) ;[DEST] = 177777
CCC ;SCOPE SYNC
I0105: MOVB (R5)+,5(R3) ;TEST THE MOVB
CMP R4,(R2) ;RESULT OK?
BEQ A0105 ;BR IF YES
E10105: MOV (R2),R3 ;GET WAS DATA
HALT ;MOV DELIVERED WRONG RESULT
BR R0105 ;LOCK ON HARD ERROR
A0105: CMP #DBTB+2,R5 ;DID SRC REG GET INCREMENTED BY +1
BEQ T0106 ;BR IF YES
E20105: HALT ;MOVB FAILED TO UPDATE SRC REG
BR R0105 ;LOCK ON HARD ERROR

6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747
6748
6749
6750
6751
6752
6753
6754
6755
6756
6757
6758
6759
6760
6761
6762
6763
6764
6765
6766
6767
6768
6769
6770
6771
6772
6773

010332 012700 000106
010336 112760 000377 070140
010344 012702 067564
010350 012703 067560
010354 012704 000377
010360 012705 070130
010364 012712 177777
010370 000257
010372 112563 000005
010376 020412
010400 001403
010402 011203
010404 000000
010406 000764
010410 022705 070131
010414 001402
010416 000000
010420 000757

: *****
.SBTTL T0106 BASIC 'MOVB (RA)+,X(RB) - SRC EVEN / DEST ODD
: *****
:MICROPROGRAMMING / LOGIC INFORMATION
:ROM SEQ: [142,240,250,177,206,212,202,205,125,375,016] FC 1,2,4
:ACT BUTS: 37[004]100,142 / 35[240]120,177 / 17[177]212,212 / 21[206]200,202
: / 16[125]016,016
:EXEC: [205]ALUC=LLLLL :[125]D=177777
:CODES: [125] SPS=3 / N:C=1000
:SYNC: B05J2 (-) / T=4.5 USEC
:KEY SIG: K3-3 MOVL / K3-3 DM=6L / K3-5 DOPL / K5-2 PS (N)(1) H
: K3-6 BYTE INSTR

T0106: MOV #0106,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUF1,R2 ;DEST ADDR = MBUF1
MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
R0106: MOV #DBTA,R5 ;SRC ADDR = DBTA
MOV #-1,(R2) ;[DEST] = 177777
CCC ;SCOPE SYNC
I0106: MOVB (R5)+,5(R3) ;TEST THE MOVB
CMP R4,(R2) ;RESULT OK?
BEQ A0106 ;BR IF YES
E10106: MOV (R2),R3 ;GET WAS DATA
HALT ;MOV DELIVERED WRONG RESULT
BR R0106 ;LOCK ON HARD ERROR
A0106: CMP #DBTA+1,R5 ;DID SRC REG GET INCREMENTED BY +1
BEQ T0107 ;BR IF YES
E20106: HALT ;MOVB FAILED TO UPDATE SRC REG
BR R0106 ;LOCK ON HARD ERROR

T0106 BASIC 'MOVB (RA)+,X(RB) - SRC EVEN / DEST ODD

SEQ 0184

6774
6775
6776
6777
6778
6779
6780
6781
6782
6783
6784
6785
6786
6787
6788
6789
6790
6791
6792
6793
6794
6795
6796
6797
6798
6799
6800
6801
6802
6803
6804
6805
6806
6807
6808
6809
6810
6811
6812
6813
6814
6815
6816

010422 012700 000107
010426 112760 000377
010434 012702 067564
010440 012703 067560
010444 012704 177401
010450 012705 070135
010454 012712 177777
010460 000257

010462 112563 000004

010466 020412
010470 001403

010472 011203
010474 000000
010476 000764

010500 022705 070136
010504 001402

010506 000000
010510 000757

070140

```
; *****  
; .SBTTL T0107 BASIC 'MOVB (RA)+,X(RB) - SRC ODD / DEST EVEN  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [142,240,250,137,251,177,206,212 202,205,125,375,016] FC 1,2,4  
;ACT BUTS: 37[004]100,142 / 35[240]120,137 / 36[137]120,177 / 17[177]212,212  
; / 21[206]200,202 / 16[125]016,016  
  
;EXEC: [205]ALUC=LLLLL :[125]D=001001  
;CODES: [125] SPS=3 / N:C=0000  
;SYNC: B05J2 (-) / T=4.5 USEC  
;KEY SIG: K3-3 MOVL / K3-3 DM=6L / K3-5 DOPL / K3-6 BYTE INSTR H  
  
T0107: MOV #0107,R0 ;LOAD R0 WITH TEST NO.  
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #MBUF1,R2 ;DEST ADDR = MBUF1  
MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0  
MOV #177401,R4 ;RESULT S / B = 177401  
R0107: MOV #DBTB+1,R5 ;SRC ADDR = DBTB+1  
MOV #-1,(R2) ;[DEST] = 177777  
CCC ;SCOPE SYNC  
  
I0107: MOVB (R5)+,4(R3) ;TEST THE MOVB  
  
CMP R4,(R2) ;RESULT OK?  
BEQ A0107 ;BR IF YES  
  
E10107: MOV (R2),R3 ;GET WAS DATA  
HALT ;MOV DELIVERED WRONG RESULT  
BR R0107 ;LOCK ON HARD ERROR  
  
A0107: CMP #DBTB+2,R5 ;DID SRC REG GET INCREMENTED BY +1  
BEQ T0110 ;BR IF YES  
  
E20107: HALT ;MOVB FAILED TO UPDATE SRC REG  
BR R0107 ;LOCK ON HARD ERROR
```

6817
6818
6819
6820
6821
6822
6823
6824
6825
6826
6827
6828
6829
6830
6831
6832
6833
6834
6835
6836
6837
6838
6839
6840
6841
6842
6843
6844
6845
6846
6847
6848
6849
6850
6851
6852
6853
6854
6855
6856
6857
6858

010512 012700 000110
010516 112760 000377 070140
010524 012702 067560
010530 012704 177401
010534 012705 070114
010540 010203
010542 012713 177777
010546 000257
010550 116523 000002
010554 020412
010556 001403
010560 011203
010562 000000
010564 000765
010566 022703 067561
010572 001402
010574 000000
010576 000760

: *****
: .SBTTL T0110 BASIC 'MOVB 2(RA),(RB)+' TEST - SRC EVEN / DEST EVEN
: *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [146,241,242,247,250,172,257,202,205,125,375,016] FC 1,2,4
;ACT BUTS: 37[004]100,146 / 35[247]120,172 / 22[172]200,202 / 16[125]016,016
;EXEC: [205]ALUC=LLLLL :[125]D=001001
;CODES: [125] SPS=3 / N:C=0000
;SYNC: B05J2 (-) / T=4.2 USEC
;KEY SIG: K3-3 MOVL / K3-3 DM=2L / K3-5 DOPL / K3-6 BYTE INSTR H

T0110: MOV #0110,R0 ;LOAD R0 WITH TEST NO.
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177401,R4 ;RESULT S / B = 177401
MOV #DWTB,R5 ;SRC ADDR = DWTB
R0110: MOV R2,R3 ;R3 GETS DEST ADDR
MOV #-1,(R3) ;[DEST] = 177400
CCC ;SCOPE SYNC
I0110: MOVB 2(R5),(R3)+ ;TEST THE MOVB
CMP R4,(R2) ;RESULT OK?
BEQ A0110 ;BR IF YES
MOV (R2),R3 ;GET WAS DATA
E10110: HALT ;MOVB DELIVERED WRONG RESULT
BR R0110 ;LOCK ON HARD ERROR
A0110: CMP #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED?
BEQ T0111 ;BR IF YES
E20110: HALT ;MOVB FAILED TO AUTO INCREMENT DEST REG
BR R0110 ;LOCK ON HARD ERROR

6859
6860
6861
6862
6863
6864
6865
6866
6867
6868
6869
6870
6871
6872
6873
6874
6875
6876
6877
6878
6879
6880
6881
6882
6883
6884
6885
6886
6887
6888
6889
6890
6891
6892
6893
6894
6895
6896
6897
6898
6899
6900
6901

010600 012700 000111
010604 112760 000377 070140
010612 012702 067560
010616 012704 177401
010622 012705 070134
010626 010203
010630 012713 177777
010634 000257
010636 116523 000001
010642 020412
010644 001403
010646 011203
010650 000000
010652 000765
010654 022703 067561
010660 001402
010662 000000
010664 000760

070140

; *****
; .SBTTL T0111 BASIC 'MOVB 2(RA),(RB)++' TEST - SRC ODD / DEST EVEN
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [146,241,242,247,250,137,251,172,257,202,205,125,375,016] FC 1,2,4
;ACT BUTS: 37[004]100,146 / 35[247]120,137 / 36[137]120,172 / 22[172]200,202
; / 16[125]016,016
;EXEC: [205]ALUC=LLLLL :[125]D=001001
;CODES: [125] SPS=3 / N:C=0000
;SYNC: B05J2 (-) / T=4.2 USEC
;KEY SIG: K3-3 MOVL / K3-3 DM=2L / K3-5 DOPL / K3-6 BYTE INSTR H

T0111: MOV #0111,R0 ;LOAD R0 WITH TEST NO.
MOVE #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #177401,R4 ;RESULT S / B = 177401
MOV #DBTB,R5 ;SRC ADDR = DBTB
R0111: MOV R2,R3 ;R3 GETS DEST ADDR
MOV #-1,(R3) ;[DEST] = 177777
CCC ;SCOPE SYNC
I0111: MOVB 1(R5),(R3)+ ;TEST THE MOVB
CMP R4,(R2) ;RESULT OK?
BEQ A0111 ;BR IF YES
MOV (R2),R3 ;GET WAS DATA
E10111: HALT ;MOVB DELIVERED WRONG RESULT
BR R0111 ;LOCK ON HARD ERROR
A0111: CMP #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED?
BEQ T0112 ;BR IF YES
E20111: HALT ;MOVB FAILED TO AUTO INCREMENT DEST REG
BR R0111 ;LOCK ON HARD ERROR

6902
6903
6904
6905
6906
6907
6908
6909
6910
6911
6912
6913
6914
6915
6916
6917
6918
6919
6920
6921
6922
6923
6924
6925
6926
6927
6928
6929
6930
6931
6932
6933
6934
6935
6936
6937
6938
6939
6940
6941
6942
6943

010666 012700 000112
010672 112760 000377
010700 012702 067560
010704 012704 000777
010710 012705 070114
010714 012703 067561
010720 012712 177777
010724 000257
010726 116523 000002
010732 020412
010734 001403
010736 011203
010740 000000
010742 000764
010744 022703 067562
010750 001402
010752 000000
010754 000757

070140

; *****
; .SBTTL T0112 BASIC 'MOVB 2(RA),(RB)++' TEST - SRC EVEN / DEST ODD
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [146,241,242,247,250,172,257,202,205,125,375,016] FC 1,2,4
;ACT BUTS: 37[004]100,146 / 35[247]120,172 / 22[172]200,202 / 16[125]016,016
;EXEC: [205]ALUC=LLLLL :[125]D=001001
;CODES: [125] SPS=3 / N:C=0000
;SYNC: B05J2 (-) / T=4.2 USEC
;KEY SIG: K3-3 MOVL / K3-3 DM=2L / K3-5 DOPL / K3-6 BYTE INSTR H

T0112: MOV #0112,R0 ;LOAD R0 WITH TEST NO.
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #777,R4 ;RESULT S / B = 777
MOV #DWTB,R5 ;SRC ADDR = DWTB
R0112: MOV #MBUF0+1,R3 ;R3 GETS DEST ADDR
MOV #-1,(R2) ;[DEST] = 177777
CCC ;SCOPE SYNC
I0112: MOVB 2(R5),(R3)+ ;TEST THE MOVB
CMP R4,(R2) ;RESULT OK?
BEQ A0112 ;BR IF YES
E10112: MOV (R2),R3 ;GET WAS DATA
HALT ;MOVB DELIVERED WRONG RESULT
BR R0112 ;LOCK ON HARD ERROR
A0112: CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED?
BEQ T0113 ;BR IF YES
E20112: HALT ;MOVB FAILED TO AUTO INCREMENT DEST REG
BR R0112 ;LOCK ON HARD ERROR

6944
6945
6946
6947
6948
6949
6950
6951
6952
6953
6954
6955
6956
6957
6958
6959
6960
6961
6962
6963
6964
6965
6966
6967
6968
6969
6970
6971
6972
6973
6974
6975
6976
6977
6978
6979
6980
6981
6982
6983
6984
6985
6986

010756 012700 000113
010762 112760 000377 070140
010770 012702 067560
010774 012704 000777
011000 012705 070134
011004 012703 067561
011010 012712 177777
011014 000257
011016 116523 000001
011022 020412
011024 001403
011026 011203
011030 000000
011032 000764
011034 022703 067562
011040 001402
011042 000000
011044 000757

: *****
: .SBTTL T0113 BASIC 'MOVB 2(RA),(RB)++' TEST - SRC ODD / DEST ODD
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [146,241,242,247,250,137,251,172,257,202,205,125,375,016] FC 1,2,4
:ACT BUTS: 37[004]100,146 / 35[247]120,137 / 36[137]120,172 / 22[172]200,202
: / 16[125]016,016
:EXEC: [205]ALUC=LLLLL :[125]D=001001
:CODES: [125] SPS=3 / N:C=0000
:SYNC: B05J2 (-) / T=4.2 USEC
:KEY SIG: K3-3 MOVL / K3-3 DM=2L / K3-5 DOPL / K3-6 BYTE INSTR H

T0113: MOV #0113,R0 ;LOAD R0 WITH TEST NO.
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #777,R4 ;RESULT S / B = 777
MOV #DBTB,R5 ;SRC ADDR = DBTB
R0113: MOV #MBUF0+1,R3 ;R3 GETS DEST ADDR = MBUF0+1
MOV #-1,(R2) ;[DEST] = 177777
CCC ;SCOPE SYNC
I0113: MOVB 1(R5),(R3)+ ;TEST THE MOVB
CMP R4,(R2) ;RESULT OK?
BEQ A0113 ;BR IF YES
E10113: MOV (R2),R3 ;GET WAS DATA
HALT ;MOVB DELIVERED WRONG RESULT
BR R0113 ;LOCK ON HARD ERROR
A0113: CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED?
BEQ T0114 ;BR IF YES
E20113: HALT ;MOVB FAILED TO AUTO INCREMENT DEST REG
BR R0113 ;LOCK ON HARD ERROR

```

6987 ; *****
6988 ; .SBTTL T0114 BASIC 'MOVB -(RA),RB' TEST - SRC EVEN ADDR
6989 ; *****
6990
6991 ;MICROPROGRAMMING / LOGIC INFORMATION
6992
6993 ;ROM SEQ: [144,240,250,160,204,003,204,000] FC 1,2,4
6994
6995 ;ACT BUTS: 37[004]100,144 / 35[240]120,160 / 20[160]000,003 / 27[003]000,000
6996
6997 ;EXEC: [003]ALUC=LLLLL :[204]D=177777
6998
6999 ;CODES: [204] SPS=3 / N:C=1000
7000
7001 ;SYNC: B05J2 (-) / T=2 USEC
7002
7003 ;KEY SIG: K3-3 MOVL / K3-3 DM=0L / K3-5 DOPL / K3-6 BYTE INSTR
7004 ; K5-2 PS (N)(1)H
7005
7006 011046 012700 000114 T0114: MOV #0114,R0 ;LOAD R0 WITH TEST NO.
7007 011052 112760 000377 070140 T0114: MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
7008 011060 012702 177703 T0114: MOV #177703,R2 ;DEST ADDR = R3
7009 011064 012704 177777 T0114: MOV #-1,R4 ;RESULT S / B = 177777
7010 011070 012705 067577 R0114: MOV #DWTA+7,R5 ;SRC ADDR = DWTA+7
7011 011074 005003 R0114: CLR R3 ;[DEST] = 000000
7012 011076 000257 R0114: CCC ;SCOPE SYNC
7013
7014 011100 114503 I0114: MOVB -(R5),R3 ;TEST THE MOVB
7015
7016 011102 020403 I0114: CMP R4,R3 ;RESULT OK?
7017 011104 001402 I0114: BEQ A0114 ;BR IF YES
7018
7019 011106 000000 E10114: HALT ;MOVB FAILED - WRONG RESULT
7020 011110 000767 E10114: BR R0114 ;LOCK ON HARD ERROR
7021
7022 011112 022705 067576 A0114: CMP #DWTA+6,R5 ;SRC REG GET DECREMENTED?
7023 011116 001402 A0114: BEQ T0115 ;BR IF YES
7024
7025 011120 000000 E20114: HALT ;MOVB FAILED TO UPDATE SRC REG
7026 011122 000762 E20114: BR R0114 ;LOCK ON HARD ERROR

```

7027
7028
7029
7030
7031
7032
7033
7034
7035
7036
7037
7038
7039
7040
7041
7042
7043
7044
7045
7046
7047
7048
7049
7050
7051
7052
7053
7054
7055
7056
7057
7058
7059
7060
7061
7062
7063
7064
7065
7066
7067
7068

: *****
: .SBTTL T0115 BASIC 'MOVB -(RA),RB' TEST - SRC ODD ADDR
: *****
:MICROPROGRAMMING / LOGIC INFORMATION
:ROM SEQ: [144,240,250,137,251,160,204,003,204,000] FC 1,2,4
:ACT BUTS: 37[004]100,144 / 35[240]120,137 / 36[137]120,160 / 20[160]000,003
: / 27[003]000,000
:EXEC: [003]ALUC=LLLLL :[204]D=177777
:CODES: [204] SPS=3 / N:C=1000
:SYNC: B05J2 (-) / T= 2 USEC
:KEY SIG: K3-3 MOVL / K3-3 DM=0L / K3-5 DOPL / K3-6 BYTE INSTR H
: K5-2 PS(N)(1)H

011124 012700 000115
011130 112760 000377 070140
011136 012702 177703
011142 012704 177777
011146 012705 067576
011152 005003
011154 000257

011156 114503

011160 020403
011162 001402

011164 000000
011166 000767

011170 022705 067575
011174 001402

011176 000000
011200 000762

T0115: MOV #0115,R0 ;LOAD R0 WITH TEST NO.
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #177703,R2 ;DEST ADDR = R3
MOV #-1,R4 ;RESULT S / B = 177777
R0115: MOV #DWTA+6,R5 ;SRC ADDR = DWTA+6
CLR R3 ;[DEST] = 000000
CCC ;SCOPE SYNC

I0115: MOVB -(R5),R3 ;TEST THE MOVB

CMP R4,R3 ;RESULT OK?
BEQ A0115 ;BR IF YES

E10115: HALT ;MOVB FAILED - WRONG RESULT
BR R0115 ;LOCK ON HARD ERROR

A0115: CMP #DWTA+5,R5 ;SRC REG GET DECREMENTED?
BEQ T0116 ;BR IF YES

E20115: HALT ;MOVB FAILED TO UPDATE SRC REG
BR R0115 ;LOCK ON HARD ERROR

7069
7070
7071
7072
7073
7074
7075
7076
7077
7078
7079
7080
7081
7082
7083
7084
7085
7086
7087
7088
7089
7090
7091
7092
7093
7094
7095
7096
7097
7098
7099
7100
7101
7102
7103
7104
7105
7106
7107
7108
7109
7110
7111
7112
7113
7114
7115
7116
7117
7118
7119
7120
7121
7122

011202 012700 000116
011206 112760 000377 070140
011214 010605
011216 012704 177400
011222 010506
011224 012703 070130
011230 012746 177777
011234 010602
011236 005726
011240 000257
011242 112346
011244 022703 070131
011250 001402
011252 000000
011254 000762
011256 020412
011260 001403
011262 011203
011264 000000
011266 000755
011270 020206
011272 001402
011274 000000
011276 000751
011300 010506

: *****
: .SBTTL T0116 BASIC 'MOVB (RA)+,-(SP)' TEST - SRC ADDR EVEN
: *****
:MICROPROGRAMMING / LOGIC INFORMATION
:ROM SEQ: [142,240,250,174,257,202,205,125,375,016] FC 1,2,4
:ACT BUTS: 37[004]100,142 / 35[240]120,174 / 22[174]200,202 / 16[125]016,016
:EXEC: [205]ALUC=LLLLL :[125]D=000000
:CODES: [125] SPS=3 / N:C=0100
:SYNC: B05J2 (-) / T= 3.3 USEC
:KEY SIG: K3-3 MOVL / K3-3 DM=4L / K3-5 DOPL / K3-6 BYTE INSTR H
: K5-2 PS(Z)(1)H

T0116: MOV #0116,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV SP,R5 ;SAVE SP
MOV #177400,R4 ;RESULT S / B = 177400
R0116: 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
I0116: MOVB (R3)+,-(SP) ;TEST THE MOVB
CMP #DBTA+1,R3 ;DID MOVB INCREMENT SRC REG?
BEQ A0116 ;BR IF YES
E0116: HALT ;MOVB FAILED TO UPDATE SRC REG
BR R0116 ;LOCK ON HARD ERROR
A0116: CMP R4,(R2) ;RESULT OK?
BEQ B0116 ;BR IF YES
E10116: MOV (R2),R3 ;GET WAS DATA
HALT ;MOVB FAILED TO DELIVER CORRECT RESULT
BR R0116 ;LOCK ON HARD ERROR
B0116: CMP R2,SP ;DID SP GET PUSHED BY 2 ?
BEQ C0116 ;BR IF YES
E20116: HALT ;MOVB FAILED TO PUSH SP PROPERLY
BR R0116 ;LOCK ON HARD ERROR
C0116: MOV R5,SP ;RESET SP IN CASE OF ERROR

7123
7124
7125
7126
7127
7128
7129
7130
7131
7132
7133
7134
7135
7136
7137
7138
7139
7140
7141
7142
7143
7144
7145
7146
7147
7148
7149
7150
7151
7152
7153
7154
7155
7156
7157
7158
7159
7160
7161
7162
7163
7164
7165
7166
7167
7168
7169
7170
7171
7172
7173
7174
7175
7176

011302 012700 000117
011306 112760 000377 070140
011314 010605
011316 012704 177400
011322 010506
011324 012703 070117
011330 012746 177777
011334 010602
011336 005726
011340 000257

011342 112346

011344 022703 070120
011350 001402

011352 000000
011354 000762

011356 020412
011360 001403

011362 011203
011364 000000
011366 000755

011370 020206
011372 001402

011374 000000
011376 000751

011400 010506

```
: *****  
: .SBTTL T0117 BASIC 'MOVB (RA)+,-(SP)'' TEST - SRC ADDR ODD  
: *****  
  
:MICROPROGRAMMING / LOGIC INFORMATION  
  
:ROM SEQ: [142,240,250,137,251,174,257,202,205,125,375,016] FC 1,2,4  
  
:ACT BUTS: 37[004]100,142 / 35[240]120,137 / 36[137]120,174 / 22[174]200,202  
: / 16[125]016,016  
  
:EXEC: [205]ALUC=LLLLL :[125]D=000000  
  
:CODES: [125] SPS=3 / N:C=0100  
  
:SYNC: B05J2 (-) / T= 3.3 USEC  
  
:KEY SIG: K3-3 MOVL / K3-3 DM=4L / K3-5 DOPL / K3-6 BYTE INSTR H  
: K5-2 PS(Z)(1)H  
  
T0117: MOV #0117,R0 ;LOAD R0 WITH TEST NO.  
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV SP,R5 ;SAVE SP  
MOV #177400,R4 ;RESULT S / B = 177400  
R0117: 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 (SP)+ ;RESET SP  
CCC ;SCOPE SYNC  
  
I0117: MOVB (R3)+,-(SP) ;TEST THE MOVB  
  
CMP #DWTB+4,R3 ;DID MOVB INCREMENT SRC REG?  
BEQ A0117 ;BR IF YES  
  
E0117: HALT ;MOVB FAILED TO UPDATE SRC REG  
BR R0117 ;LOCK ON HARD ERROR  
  
A0117: CMP R4,(R2) ;RESULT OK?  
BEQ B0117 ;BR IF YES  
  
E10117: MOV (R2),R3 ;GET WAS DATA  
HALT ;MOVB FAILED TO DELIVER CORRECT RESULT  
BR R0117 ;LOCK ON HARD ERROR  
  
B0117: CMP R2,SP ;DID SP GET PUSHED BY 2  
BEQ C0117 ;BR IF YES  
  
E20117: HALT ;MOVB FAILED TO PUSH SP  
BR R0117 ;LOCK ON HARD ERROR  
  
C0117: MOV R5,SP ;RESET SP IN CASE OF ERROR
```

7177
7178
7179
7180
7181
7182
7183
7184
7185
7186
7187
7188
7189
7190
7191
7192
7193
7194
7195
7196
7197
7198
7199
7200
7201
7202
7203
7204
7205
7206
7207
7208
7209
7210

011402 012700 000120
011406 112760 000377 070140
011414 012702 067560
011420 012704 000001
011424 012705 070114
011430 005012
011432 000257
011434 116537 000006 067560
011442 020412
011444 001403
011446 011203
011450 000000
011452 000766

```
: *****  
: .SBTTL T0120 BASIC 'MOVB X(R),@#A'' TEST - SRC EVEN / DEST EVEN  
: *****  
  
:MICROPROGRAMMING / LOGIC INFORMATION  
  
:ROM SEQ: [146,241,242,247,250,173,207,210,202,205,125,375,016] FC 1,2,4  
:ACT BUTS: 37[004]100,146 / 35[247]120,173 / 22[207]200,202 / 16[125]016,016  
:EXEC: [205]ALUC=LLLLL :[125]D=001001  
:CODES: [125] SPS=3 / N:C=0000  
:SYNC: B05J2 (-) / T= 4.7 USEC  
:KEY SIG: K3-3 MOVL / K3-3 DM=3L / K3-5 DOPL / K3-6 BYTE INSTR H  
  
T0120: MOV #0120,R0 ;LOAD R0 WITH TEST NO.  
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
MOV #1,R4 ;RESULT S / B = 1  
MOV #DWTB,R5 ;BASE SRC ADDR = DWTB  
R0120: CLR (R2) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
  
I0120: MOVB 6(R5),@MBUFO ;TEST THE MOVB  
  
CMP R4,(R2) ;RESULT OK?  
BEQ T0121 ;BR IF YES  
  
E0120: MOV (R2),R3 ;GET WAS DATA  
HALT ;MOVB DELIVERED WRONG RESULT  
BR R0120 ;LOCK ON HARD ERROR
```

7211
7212
7213
7214
7215
7216
7217
7218
7219
7220
7221
7222
7223
7224
7225
7226
7227
7228
7229
7230
7231
7232
7233
7234
7235
7236
7237
7238
7239
7240
7241
7242
7243
7244

011454 012700 000121
011460 112760 000377 070140
011466 012702 067560
011472 012704 000001
011476 012705 070134
011502 005012
011504 000257
011506 116537 000001 067560
011514 020412
011516 001403
011520 011203
011522 000000
011524 000766

```
: *****  
      .SBTTL T0121 BASIC 'MOVB X(R),@#A' TEST - SRC ODD / DEST EVEN  
: *****  
:MICROPROGRAMMING / LOGIC INFORMATION  
:ROM SEQ:      [146,241,242,247,250,137,251,173,207,210,202,205,125,375,016] FC 1,2,4  
:ACT BUTS:     37[004]100,146 / 35[247]120,137 / 22[207]200,202 / 16[125]016,016  
:EXEC:         [205]ALUC=LLLLL :[125]D=001001  
:CODES:        [125] SPS=3 / N:C=0000  
:SYNC:         B05J2 (-) / T= 4.7 USEC  
:KEY SIG:      K3-3 MOVL / K3-3 DM=3 / K3-5 DOPL / K3-6 BYTE INSTR H  
T0121: MOV      #0121,R0          ;LOAD R0 WITH TEST NO.  
        MOVVB   #377,STAB1(R0)   ;SET FLAG FOR THIS TEST IN MISSED TABLE  
        MOV      #MBUF0,R2       ;DEST ADDR = MBUF0  
        MOV      #1,R4           ;RESULT S / B = 1  
        MOV      #DBTB,R5        ;BASE SRC ADDR = DBTB  
R0121: CLR      (R2)             ;[DEST] = 000000  
        CCC                               ;SCOPE SYNC  
I0121: MOVVB   1(R5),@MBUF0     ;TEST THE MOVVB  
        CMP      R4,(R2)         ;RESULT OK?  
        BEQ      T0122           ;BR IF YES  
E0121: MOV      (R2),R3         ;GET WAS DATA  
        HALT                               ;MOVVB DELIVERED WRONG RESULT  
        BR      R0121           ;LOCK ON HARD ERROR
```

7245
7246
7247
7248
7249
7250
7251
7252
7253
7254
7255
7256
7257
7258
7259
7260
7261
7262
7263
7264
7265
7266
7267
7268
7269
7270
7271
7272
7273
7274
7275
7276
7277
7278

011526 012700 000122
011532 112760 000377 070140
011540 012702 067560
011544 012704 000400
011550 012705 070114
011554 005012
011556 000257

011560 116537 000006 067561

011566 020412
011570 001403

011572 011203
011574 000000
011576 000766

```
; *****  
; .SBTTL T0122 BASIC 'MOVB X(R),@#A' TEST - SRC EVEN / DEST ODD  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [146,241,242,247,250,173,207,210,202,205,125,375,016] FC 1,2,4  
;ACT BUTS: 37[004]100,146 / 35[247]120,177 / 21[206]200,202 / 16[125]016,016  
;EXEC: [205]ALUC=LLLLL :[125]D=000401  
;CODES: [125] SPS=3 / N:C=0000  
;SYNC: B05J2 (-) / T= 4.7 USEC  
;KEY SIG: K3-3 MOVL / K3-3 DM=3L / K3-5 DOPL / K3-6 BYTE INSTR H  
T0122: MOV #0122,R0 ;LOAD R0 WITH TEST NO.  
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #400,R4 ;RESULT S / B = 400  
MOV #DWTB,R5 ;BASE SRC ADDR = DWTB  
R0122: CLR (R2) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
I0122: MOVB 6(R5),@#MBUF0+1 ;TEST THE MOVB  
CMP R4,(R2) ;RESULT OK?  
BEQ T0123 ;BR IF YES  
E0122: MOV (R2),R3 ;GET WAS DATA  
HALT ;MOVB DELIVERED WRONG RESULT  
BR R0122 ;LOCK ON HARD ERROR
```

7279
7280
7281
7282
7283
7284
7285
7286
7287
7288
7289
7290
7291
7292
7293
7294
7295
7296
7297
7298
7299
7300
7301
7302
7303
7304
7305
7306
7307
7308
7309
7310
7311
7312
7313
7314

; *****
; .SBTTL T0123 BASIC 'MOVB X(R),@#A' TEST - SRC ODD / DEST ODD
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [146,241,242,247,250,137,251,173,207,210,202,205,125,375,016] FC 1,2,4

;ACT BUTS: 37[004]100,146 / 35[247]120,137 / 36[137]120,173 / 22[207]200,202
; / 16[125]016,016

;EXEC: [205]ALUC=LLLLL :[125]D=001001

;CODES: [125] SPS=3 / N:C=0000

;SYNC: B05J2 (-) / T= 4.7 USEC

;KEY SIG: K3-3 MOVL / K3-3 DM=3L / K3-5 DOPL / K3-6 BYTE INSTR H

011600 012700 000123
011604 112760 000377 070140
011612 012702 067560
011616 012704 000400
011622 012705 070134
011626 005012
011630 000257
011632 116537 000001 067561
011640 020412
011642 001403
011644 011203
011646 000000
011650 000766

T0123: MOV #0123,R0 ;LOAD R0 WITH TEST NO.
MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #400,R4 ;RESULT S / B = 400
MOV #DBTB,R5 ;BASE SRC ADDR = DBTB
R0123: CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
I0123: MOVB 1(R5),@#MBUF0+1 ;TEST THE MOVB
CMP R4,(R2) ;RESULT OK?
BEQ T0124 ;BR IF YES
E0123: MOV (R2),R3 ;GET WAS DATA
HALT ;MOVB DELIVERED WRONG RESULT
BR R0123 ;LOCK ON HARD ERROR

```

7315 ; *****
7316 ; .SBTTL T0124 BASIC 'RTS PC' TEST
7317 ; *****
7318
7319 ;MICROPROGRAMMING / LOGIC INFORMATION
7320
7321 ;ROM SEQ: [124,323,324,325,016] FC 1,6
7322
7323 ;ACT BUTS: 37[004]100,124 / 16[324]016,016
7324
7325 ;EXEC: N / A
7326
7327 ;CODES: N / A
7328
7329 ;SYNC: B05J2 (-) / T= 2.5 USEC
7330
7331 ;KEY SIG: K3-6 RTSL
7332
7333 011652 012700 000124 T0124: MOV #0124,R0 ;LOAD R0 WITH TEST NO.
7334 011656 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
7335 011664 010605 MOV SP,R5 ;SAVE THE ORIGINAL SP
7336 011666 010506 R0124: MOV R5,SP ;RESET SP FOR ERROR LOOP
7337 011670 012746 011704 MOV #A0124,-(SP) ;PUSH NEW PC ON STACK
7338 011674 000257 CCC ;SCOPE SYNC
7339
7340 011676 000207 I0124: RTS PC ;TEST THE RTS - GO TO A0124
7341
7342 011700 000000 E10124: HALT ;RTS FAILED TO LOAD PC
7343 011702 000771 BR R0124 ;LOCK ON HARD ERROR
7344
7345 011704 020605 A0124: CMP SP,R5 ;DID SP GET POPPED ?
7346 011706 001402 BEQ T0125 ;BR IF YES
7347
7348 011710 000000 E20124: HALT ;RTS FAILED TO UPDATE SP
7349 011712 000765 BR R0124 ;LOCK ON HARD ERROR
7350

```

```

7351 ; *****
7352 ; .SBTTL T0125 BASIC 'JSR PC,@#A' TEST
7353 ; *****
7354
7355

```

;MICROPROGRAMMING / LOGIC INFORMATION

```

7356 ;ROM SEQ: [153,303,307,310,311,312,306,313,016] FC 1,5
7357 ;ACT BUTS: 37[004]100,153 / 15[153]306,307 / 16[306]016,016
7358 ;EXEC: N / A
7359 ;CODES: N / A
7360 ;SYNC: B05J2 (-) / T= 3.5 USEC
7361 ;KEY SIG: K3-5 JMP+JSRH / K3-3 DM=3L
7362
7363
7364
7365
7366
7367
7368

```

```

7369 011714 012700 000125 T0125: MOV #0125,R0 ;LOAD R0 WITH TEST NO.
7370 011720 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
7371 011726 010605 MOV SP,R5 ;SAVE ORIGINAL SP
7372 011730 010506 R0125: MOV R5,SP ;RESET SP FOR ERROR LOOP
7373 011732 000257 CCC ;SCOPE SYNC
7374
7375 011734 004737 011744 I0125: JSR PC,@#A0125 ;TEST THE JSR - GO TO A0125
7376
7377 011740 000000 E10125: HALT ;JSR FAILED TO LOAD PC
7378 011742 000772 BR R0125 ;LOCK ON HARD ERROR
7379
7380 011744 022726 011740 A0125: CMP #E10125,(SP)+ ;DID JSR SAVE OLD PC ON STACK ?
7381 011750 001402 BEQ T0126 ;BR IF YES
7382
7383
7384 011752 000000 E20125: HALT ;JSR FAILED TO SAVE OLD PC
7385 011754 000765 BR R0125 ;LOCK ON HARD ERROR
7386

```

```

7387 ; *****
7388 ; .SBTTL T0126 BASIC 'RTI' TEST - N:C=0000
7389 ; *****
7390
7391 ;MICROPROGRAMMING / LOGIC INFORMATION
7392
7393 ;ROM SEQ: [101,320,321,322,017,015,013] FC 6,10
7394
7395 ;ACT BUTS: 37[004]100,101 / 26[017]010,013
7396
7397 ;EXEC: N / A
7398
7399 ;CODES: N / A
7400
7401 ;SYNC: B05J2 (-) / T= 3 USEC
7402
7403 ;KEY SIG: K3-6 RTI+RTTL
7404
7405 011756 012700 000126 T0126: MOV #0126,R0 ;LOAD R0 WITH TEST NO.
7406 011762 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
7407 011770 010605 MOV SP,R5 ;SAVE THE SP
7408 011772 010506 R0126: MOV R5,SP ;RESET THE SP FOR ERROR LOOP
7409 011774 012746 000357 MOV #357,-(SP) ;NEW PSW = 357
7410 012000 012746 012020 MOV #A0126,-(SP) ;NEW PC = A0126
7411 012004 005037 177776 CLR @#PSW ;MAKE [PSW] = 000
7412 012010 000257 CCC ;MAKE N:C=0000
7413
7414 012012 000002 I0126: RTI ;TEST THE RTI - GO TO A0126
7415
7416 012014 000000 E10126: HALT ;RTI FAILED TO LOAD PC
7417 012016 000765 BR R0126 ;LOOP ON HARD ERROR
7418
7419 012020 013702 177776 A0126: MOV @#PSW,R2 ;SAVE THE [PSW] IN R2
7420 012024 022702 000357 CMP #357,R2 ;WAS [PSW] = 357 ?
7421 012030 001404 BEQ B0126 ;BR IF YES
7422
7423 012032 010237 177776 E20126: MOV R2,@#PSW ;RESTORE THE ERROR PSW
7424 012036 000000 HALT ;RTI FAILED TO LOAD PSW
7425 012040 000754 BR R0126 ;LOCK ON HARD ERROR
7426
7427 012042 020605 B0126: CMP SP,R5 ;DID SP GET UPDATED OK ?
7428 012044 001402 BEQ T0127 ;BR IF YES
7429
7430 012046 000000 E30126: HALT ;RTI FAILED TO UPDATE THE SP
7431 012050 000750 BR R0126 ;LOCK ON HARD ERROR
7432

```

7433
7434
7435
7436
7437
7438
7439
7440
7441
7442
7443
7444
7445
7446
7447
7448
7449
7450
7451
7452
7453
7454
7455
7456
7457
7458
7459
7460
7461
7462
7463
7464
7465
7466
7467
7468
7469

012052 012700 000127
012056 112760 000377 070140
012064 010605
012066 010506
012070 005046
012072 012746 012110
012076 012737 000357 177776
012104 000240
012106 000002
012110 013702 177776
012114 022702 000000
012120 001404
012122 010237 177776
012126 000000
012130 000756

```

: *****
: .SBTTL T0127 BASIC 'RTI' TEST WITH N:C=1111
: *****
;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ:      [101,320,321,322,017,015,013] FC 6,10
;ACT BUTS:     37[004]100,101 / 26[017]010,013
;EXEC:         N / A
;CODES:        N / A
;SYNC:         B05J2 (-) / T= 3 USEC
;KEY SIG:      K3-6 RTI+RTTL
T0127:  MOV    #0127,R0          ;LOAD R0 WITH TEST NO.
        MOVB  #377,STAB1(R0)   ;SET FLAG FOR THIS TEST IN MISSED TABLE
        MOV   SP,R5           ;SAVE THE SP IN R5
R0127:  MOV   R5,SP           ;RESET SP FOR ERROR LOOP
        CLR   -(SP)          ;NEW PSW = 000000
        MOV  #A0127,-(SP)     ;NEW PC = A0127
        MOV  #357,@#PSW      ;MAKE OLD PSW = 357
        NOP                   ;SCOPE SYNC
I0127:  RTI                   ;TEST THE RTI - GO TO A0127
A0127:  MOV   @#PSW,R2        ;GET THE PSW
        CMP  #0,R2           ;WAS [PSW]=000
        BEQ  T0130          ;BR IF YES
E0127:  MOV   R2,@#PSW       ;RESTORE ERROR PSW
        HALT                  ;RTI FAILED TO CLEAR PSW
        BR   R0127           ;LOCK ON HARD ERROR

```

7470
7471
7472
7473
7474
7475
7476
7477
7478
7479
7480
7481
7482
7483
7484
7485
7486
7487
7488
7489
7490
7491
7492
7493
7494
7495
7496
7497
7498
7499
7500
7501
7502
7503
7504
7505
7506
7507
7508
7509
7510
7511
7512
7513
7514
7515
7516
7517
7518
7519
7520
7521
7522
7523
7524
7525

012132 012700 000130
012136 112760 000377 070140
012144 010605
012146 010506
012150 012737 012206 000020
012156 012737 000357 000022
012164 012766 177777 177776
012172 005037 177776
012176 000257
012200 000004
012202 000000
012204 000760
012206 013702 177776
012212 022702 000357
012216 001404
012220 010237 177776
012224 000000
012226 000747
012230 022726 012202
012234 001404
012236 010237 177776
012242 000000
012244 000740
012246 005726
012250 001404
012252 010237 177776
012256 000000
012260 000732

```
; *****  
; .SBTTL T0130 BASIC 'IOT' TEST -VERIFY LOADING PSW WITH 357  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEG: [126,007,115,326,327,113,330,331,77,140,332,333,123,015,013] FC 1,6,10  
;ACT BUTS: 37[004]100,126 / 04[140]REG EXAM / 01[332]122,123 / 03[333]REG DEP  
; / 26[123]010,013  
;EXEC: N / A  
;CODES: N / A  
;SYNC: B05J2 (-) / T= 6 USEC  
;KEY SIG: K3-6 TRAP INSTR L  
T0130: MOV #0130,R0 ;LOAD R0 WITH TEST NO.  
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
R0130: MOV SP,R5 ;SAVE THE SP  
MOV R5,SP ;RESET SP FOR ERROR LOOP  
MOV #A0130,@#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  
I0130: IOT ;TEST THE IOT  
E10130: HALT ;IOT FAILED TO LOAD PC  
BR R0130 ;LOCK ON HARD ERROR  
A0130: MOV @#PSW,R2 ;GET THE PSW  
CMP #357,R2 ;DID IOT LOAD A 357 ?  
BEQ B0130 ;BR IF YES  
E20130: MOV R2,@#PSW ;RESTORE ERROR PSW  
HALT ;IOT FAILED TO LOAD PSW  
BR R0130 ;LOCK ON HARD ERROR  
B0130: CMP #E10130,(SP)+ ;DID IOT SAVE OLD PC ?  
BEQ C0130 ;BR IF YES  
E30130: MOV R2,@#PSW ;RESTORE ERROR PSW  
HALT ;IOT FAILED TO SAVE OLD PC  
BR R0130 ;LOCK ON HARD ERROR  
C0130: TST (SP)+ ;DID IOT SAVE OLD PSW ?  
BEQ T0131 ;BR IF YES  
E40130: MOV R2,@#PSW ;RESTORE ERROR PSW  
HALT ;IOT FAILED TO SAVE OLD PSW  
BR R0130 ;LOCK ON HARD ERROR
```

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 202
CBQEAC.P11 03-JUL-80 08:05 T0130 BASIC ''IOT'' TEST -VERIFY LOADING PSW WITH 357

SEQ 0202

7526

7527
7528
7529
7530
7531
7532
7533
7534
7535
7536
7537
7538
7539
7540
7541
7542
7543
7544
7545
7546
7547
7548
7549
7550
7551
7552
7553
7554
7555
7556
7557
7558
7559
7560
7561
7562
7563

: *****
: .SBTTL T0131 BASIC 'IOT' TEST - VERIFY LINKAGE TO SCOPE SERVICE
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [126,007,115,326,327,113,330,331,77,140,332,333,123,015,013] FC 1,6,10

:ACT BUTS: 37[004]100,126 / 04[140]REG EXAM / 01[332]122,123 / 03[333]REG DEP
: / 26[123]010,013

:EXEC: N / A

:CODES: N / A

:SYNC B05J2 (-) / T= 6 USEC

:KEY SIG: K3-6 TRAP INSTR L

012262 012700 000131
012266 112760 000377 070140
012274 010605
012276 010506
012300 005037 066702
012304 012737 065260 000020
012312 005037 000022
012316 000257
012320 000004
012322 005137 066702
012326 001402
012330 000000
012332 000761
012334 010506

T0131: MOV #0131,R0 ;LOAD R0 WITH TEST NO.
MOVB #377,S*AB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
MOV SP,R5 ;SAVE SP
R0131: 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
I0131: SCOPE ;TEST THE IOT
COM @#SCOFLG ;SCOFLG SHOULD BECOME 000000
BEQ A0131 ;BR IF IT DID
E0131: HALT ;IOT FAILED TO LINK TO SCOPE SERVICE
BR R0131 ;LOCK ON HARD ERROR
A0131: MOV R5,SP ;RESET SP IN CASE OF ERROR

7564
7565
7566
7567
7568
7569
7570
7571
7572
7573
7574
7575
7576
7577
7578
7579
7580
7581
7582
7583
7584
7585
7586
7587
7588
7589
7590
7591
7592
7593
7594
7595
7596
7597
7598
7599
7600
7601
7602
7603
7604
7605
7606
7607
7608
7609
7610
7611
7612
7613
7614
7615
7616
7617
7618
7619

012336 012700 000132
012342 112760 000377 070140
012350 010605
012352 010506
012354 012737 012412 000020
012362 012737 000357 000022
012370 012766 177777 177776
012376 005037 177776
012402 000257

012404 000004

012406 000000
012410 000760

012412 013702 177776
012416 022702 000357
012422 001404

012424 010237 177776
012430 000000
012432 000747

012434 022726 012406
012440 001404

012442 010237 177776
012446 000000
012450 000740

012452 005726
012454 001404

012456 010237 177776
012462 000000
012464 000732

```
; *****  
; .SBTTL T0132 BASIC 'IOT' TEST -VERIFY LOADING PSW WITH 357  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [126,007,115,326,327,113,330,331,77,140,332,333,123,015,013] FC 1,6,10  
;ACT BUTS: 37[004]100,126 / 04[140]REG EXAM / 01[332]122,123 / 03[333]REG DEP  
; / 26[123]010,013  
;EXEC: N / A  
;CODES: N / A  
;SYNC: B05J2 (-) / T= 6 USEC  
;KEY SIG: K3-6 TRAP INSTR L  
T0132: MOV #0132,R0 ;LOAD R0 WITH TEST NO.  
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV SP,R5 ;SAVE THE SP  
R0132: MOV R5,SP ;RESET SP FOR ERROR LOOP  
MOV #A0132,@#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  
  
I0132: IOT ;TEST THE IOT  
  
E10132: HALT ;IOT FAILED TO LOAD PC  
BR R0132 ;LOCK ON HARD ERROR  
  
A0132: MOV @#PSW,R2 ;GET THE PSW  
CMP #357,R2 ;DIF IOT LOAD A 357 ?  
BEQ B0132 ;BR IF YES  
  
E20132: MOV R2,@#PSW ;RESTORE ERROR PSW  
HALT ;IOT FAILED TO LOAD PSW  
BR R0132 ;LOCK ON HARD ERROR  
  
B0132: CMP #E10132,(SP)+ ;DID IOT SAVE OLD PC ?  
BEQ C0132 ;BR IF YES  
  
E30132: MOV R2,@#PSW ;RESTORE ERROR PSW  
HALT ;IOT FAILED TO SAVE OLD PC  
BR R0132 ;LOCK ON HARD ERROR  
  
C0132: TST (SP)+ ;DID IOT SAVE OLD PSW ?  
BEQ T0133 ;BR IF YES  
  
E40132: MOV R2,@#PSW ;RESTORE ERROR PSW  
HALT ;IOT FAILED TO SAVE OLD PSW  
BR R0132 ;LOCK ON HARD ERROR
```

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 205
CBQEAC.P11 03-JUL-80 08:05 T0132 BASIC "IOT" TEST -VERIFY LOADING PSW WITH 357

SEQ 0205

7620

7621
7622
7623
7624
7625
7626
7627
7628
7629
7630
7631
7632
7633
7634
7635
7636
7637
7638
7639
7640
7641
7642
7643
7644
7645
7646
7647
7648
7649
7650
7651
7652
7653
7654
7655
7656
7657
7658
7659
7660

: *****
: .SBTTL T0133 BASIC IOT TEST - VERIFY LOADING PSW WITH 000
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [126,007,115,326,327,113,330,331,77,140,323,333,123,015,013] FC 1,6,10

:ACT BUTS: 37[004]100,126 / 04[140]REG EXAM / 01[332]122,123 / 03[333]REG DEP
: / 26[123]010,013

:EXEC: N / A

:CODES: N / A

:SYNC: B05J2 (-) / T= 6 USEC

:KEY SIG: K3-6 TRAP INSTR L

T0133:	MOV	#0133,R0	:LOAD R0 WITH TEST NO.
	MOVB	#377,STAB1(R0)	:SET FLAG FOR THIS TEST IN MISSED TABLE
	MOV	SP,R5	:SAVE THE SP
R0133:	MOV	R5,SP	:RESET SP FOR ERROR LOOP
	MOV	#A0133,@#20	:SET UP IOT VECTOR
	CLR	@#22	:
	MOV	#340,@#PSW	:MAKE [PSW] = 340
	SCC		:MAKE N:C=1111
I0133:	IOT		:TEST THE IOT
A0133:	MOV	@#PSW,R2	:GET THE [PSW]
	BEQ	B0133	:BR IF [PSW] = 000
E0133:	MOV	R2,@#PSW	:RESTORE THE ERROR PSW
	HALT		:IOT FAILED TO CLEAR THE PSW
	BR	R0133	:LOCK ON HARD ERROR
B0133:	MOV	R5,SP	:RESET THE SP BEFORE CONTINUING

7661
7662
7663
7664
7665
7666
7667
7668
7669
7670
7671
7672
7673
7674
7675
7676
7677
7678
7679
7680
7681
7682
7683
7684
7685
7686
7687
7688
7689
7690
7691
7692
7693
7694
7695
7696
7697
7698
7699

: *****
: .SBTTL T0134 BASIC 'TRAP' TEST - LINKAGE TO PRINT ROUTINE
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [126,007,115,326,327,113,330,331,77,140,332,333,123,015,013] FC 1,6,10

:ACT BUTS: 37[004]100,126 / 04[140]REG EXAM / 01[332]122,123 / 03[333]REG DEP
: / 26[123]010,013

:EXEC: N / A

:CODES: N / A

:SYNC: B05J2 (-) / T= 6 USEC

:KEY SIG: K3-6 TRAP INSTR L

012550 012700 000134
012554 112760 000377 070140
012562 032737 000004 066642
012570 001401
012572 000000
012574 010605
012576 010506
012600 012737 066042 000034
012606 005037 000036
012612 005037 066674
012616 000257
012620 104400
012622 005137 066674
012626 001402
012630 000000
012632 000761

T0134: MOV #0134,R0 ;LOAD R0 WITH TEST NO.
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
BIT #4,@#BPTLOC ;BREAKPOINT HALT SET ??
BEQ .+4 ;BR IF NOT
HALT ;BREAK - DEPRESS CONTINUE TO RESTART
R0134: MOV SP,R5 ;SAVE THE SP
MOV R5,SP ;RESET SP FOR ERROR LOOP
MOV #PRINA,@#34 ;SET UP THE 'TRAP' VECTOR
CLR @#36
CLR @#PRIFLG ;INITIALIZE TEST FLAG
CCC ;SCOPE SYNC
I0134: TYPE ;TEST THE TRAP
COM @#PRIFLG ;SHOULD MAKE [PRIFLG] = 000000
BEQ T0135 ;BR IF IT DID
E0134: HALT ;TRAP FAILED TO LINK TO PRINT SERV.
BR R0134 ;LOCK ON HARD ERROR

```
7700 ; *****  
7701 ; .SBTTL T0135 BASIC 'EMT' TEST - LINKAGE TO ERROR SERVICE  
7702 ; *****  
7703 ;MICROPROGRAMMING / LOGIC INFORMATION  
7704 ;ROM SEQ: [126,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6,10  
7705 ;ACT BUTS: 37[004]100,126 / 04[140]REG EXAM / 01[332]122,123 / 03[333]REG DEP  
7706 ; / 26[123]010,013  
7707 ;EXEC: N / A  
7708 ;CODES: N / A  
7709 ;SYNC: B05J2 (-) / T= 6 USEC  
7710 ;KEY SIG: K3-6 TRAP INSTR L  
7711  
7712  
7713  
7714  
7715  
7716  
7717  
7718  
7719 012634 012700 000135 T0135: MOV #0135,R0 ;LOAD R0 WITH TEST NO.  
7720 012640 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
7721 012646 010605 MOV SP,R5 ;SAVE THE SP  
7722 012650 010506 R0135: MOV R5,SP ;RESET SP FOR ERROR LOOP  
7723 012652 012737 065400 000030 MOV #ERRA,@#30 ;SET UP THE EMT VECTOR  
7724 012660 005037 000032 CLR @#32  
7725 012664 005037 066676 CLR @#ERRFLG ;EMT SERVICE WILL COM [ERRFLG]  
7726 012670 000257 CCC ;SCOPE SYNC  
7727  
7728 012672 104000 I0135: ERROR ;TEST THE EMT  
7729  
7730 012674 005137 066676 COM @#ERRFLG ;DID EMT SERV. COM ERRFLG?  
7731 012700 001402 BEQ T0136 ;BR IF YES  
7732  
7733 012702 000000 E0135: HALT ;EMT DID NOT LINK PROPERLY  
7734 012704 000761 BR R0135 ;LOCK ON HARD ERROR
```

```
7735 ; *****
7736 ; .SBTTL T0136 BASIC TEST OF RSVD INSTR. TRAP LINKAGE
7737 ; *****
7738 ;MICROPROGRAMMING / LOGIC INFORMATION
7739 ;ROM SEQ: [100,126,007,115,327,113,330,331,077,140,332,333,123,015,013] FC 1,6,10
7740 ;ACT BUTS: 37[004]100,100 / 04[140]REG EXAM / 01[332]122,123 / 03[333]REG DEP
7741 ; / 26[123]010,013
7742 ;EXEC: N / A
7743 ;CODES: N / A
7744 ;SYNC: B05J2 (-) / T= 6 USEC
7745 ;KEY SIG: K3-6 RSVD INSTR L
7746
7747
7748
7749
7750
7751
7752
7753
7754 012706 012700 000136 T0136: MOV #0136,R0 ;LOAD R0 WITH TEST NO.
7755 012712 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
7756 012720 013701 012752 MOV @#I0136,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
7757 012724 010605 MOV SP,R5 ;SAVE THE SP
7758 012726 012737 065042 000010 MOV #RSVTST,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
7759 012734 012737 000340 000012 MOV #340,@#12
7760 012742 010506 R0136: MOV R5,SP ;RESET SP FOR ERROR LOOP
7761 012744 005037 066712 CLR @#RSVFLG ;INITIALIZE TEST FLAG THAT WILL GET
7762 ;COMPLEMENTED BY TRAP SERVICE
7763 012750 000257 CCC ;SCOPE SYNC
7764
7765 012752 177777 I0136: 177777 ;FORCE RSVD INSTR. TRAP
7766
7767 012754 005137 066712 COM @#RSVFLG ;TEST FLAG SHOULD GO TO 000000
7768 012760 001402 BEQ A0136 ;BR IF TRAP SPRUNG
7769
7770 012762 000000 E0136: HALT ;RSVD INSTR. TRAP FAILED
7771 012764 000750 BR T0136 ;LOCK ON HARD ERROR
7772
7773 012766 012737 065050 000010 A0136: MOV #RSERR,@#10 ;SET UP RSVD INSTR TRAP VECTOR TO POINT
7774 012774 012737 000340 000012 MOV #340,@#12 ;TO ERROR SERVICE ROUTINE
7775
```

```
7776 ; *****  
7777 ; .SBTTL T0137 BASIC TEST OF BUS TIMEOUT TRAP LINKAGE  
7778 ; *****  
7779 ;MICROPROGRAMMING / LOGIC INFORMATION  
7780 ;ROM SEQ: [150,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6,10  
7781 ;ACT BUTS: 37[004]100,126 / 04[140]REG EXAM / 01[332]122,123 / 03[333]REG DEP  
7782 ; / 26[123]010,013  
7783 ;EXEC: N / A  
7784 ;CODES: N / A  
7785 ;SYNC: B05J2 (-) / T= 6 USEC  
7786 ;KEY SIG: K4-6 NOSACK(1)L  
7787  
7788  
7789  
7790  
7791  
7792  
7793  
7794  
7795 013002 012700 000137 T0137: MOV #0137,R0 ;LOAD R0 WITH TEST NO.  
7796 013006 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
7797 013014 013701 013046 MOV @#I0137,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
7798 013020 010605 MOV SP,R5 ;SAVE THE SP  
7799 013022 012737 065152 000004 MOV #BETST,@#4 ;SET UP THE BUS ERROR VECTOR  
7800 013030 012737 000340 000006 R0137: MOV #340,@#6 ;RESET SP FOR ERROR LOOP  
7801 013036 010506 MOV R5,SP ;INITIALIZE TEST FLAG THAT WILL GET  
7802 013040 005037 066714 CLR @#BERFLG ;COMPLEMENTED BY TRAP SERVICE  
7803 ;SCOPE SYNC  
7804 013044 000257 CCC  
7805  
7806 013046 005737 177700 I0137: TST @#177700 ;FORCE BUS TIMEOUT USING R0 ADDR.  
7807  
7808 013052 005137 066714 COM @#BERFLG ;TEST FLAG SHOULD GO TO 000000  
7809 013056 001402 BEQ T0140 ;BR IF TRAP SPRUNG  
7810  
7811 013060 000000 E0137: HALT ;BUS ERROR FAILED TO SPRING TRAP  
7812 013062 000747 BR T0137 ;LOCK ON HARD ERROR  
7813  
7814
```

T0137 BASIC TEST OF BUS TIMEOUT TRAP LINKAGE

```
7815 ; *****
7816 ; .SBTTL T0140 BASIC TEST FOR ACCESSING DL11 REGISTERS
7817 ; *****
7818 ;MICROPROGRAMMING / LOGIC INFORMATION
7819 ;ROM SEQ: [162,260,267,220,211,367,375,016] FC 1,3,9,8
7820 ;ACT BUTS: 37[004]100,162 / 33[260]220,220 / 16[367]016,016
7821 ;EXEC: [220]ALUC=LLLLL :[211]D=N / A
7822 ;CODES: N / A
7823 ;SYNC: B05J2 (-)
7824 ;KEY SIG: K3-8 BIT+CMP+TSTH / K3-3 DM=2L
7825
7826
7827
7828
7829
7830
7831
7832
7833 013064 012700 000140 T0140: MOV #0140,R0 ;LOAD R0 WITH TEST NO.
7834 013070 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
7835 013076 005067 054456 CLR MBUF0 ;INIT STALL COUNTER
7836 013102 005367 054452 1S: DEC MBUF0 ;COUNT THE TIMER
7837 013106 001375 BNE 1S ;BR IF NO TIMEOUT
7838 013110 012737 013150 000004 MOV #E0140,@#4 ;SET UP BUS TIMEOUT VECTOR
7839 013116 012737 000340 000006 MOV #340,@#6
7840 013124 010605 MOV SP,R5 ;SAVE TH SP
7841 013126 010506 R0140: MOV R5,SP ;RESET SP FOR ERROR LOOP
7842 013130 012702 177560 MOV #RCSR,R2 ;[R2] = STARTING DL11 ADDR.
7843 013134 000257 CCC ;SCOPE SYNC
7844
7845 013136 005722 I0140: TST (R2)+ ;REFERENCE DL11 - RCSR
7846 013140 005722 TST (R2)+ ;REFERENCE DL11 - RDBR
7847 013142 005722 TST (R2)+ ;REFERENCE DL11 - XCSR
7848 013144 005712 TST (R2) ;REFERENCE DL11 - XDBR
7849
7850 013146 000403 BR A0140 ;GO TO NEXT TEST
7851
7852 013150 005742 E0140: TST -(R2) ;BAD ADDRESS IN R2
7853 013152 000000 HALT ;ONE OF DL11 ADDR'S CAUSED TIME OUT
7854 013154 000764 BR R0140 ;LOCK ON HARD ERROR
7855
7856 013156 012737 065160 000004 A0140: MOV #BERR,@#4 ;SET UP BUS ERROR VECTOR TO POINT
7857 013164 012737 000340 000006 MOV #340,@#6 ;TO ERROR SERVICE ROUTINE
```

```
7858 ; *****
7859 ; .SBTTL T0141 BASIC TEST OF DL11 - XCSR - READY(1)
7860 ; *****
7861 ;MICROPROGRAMMING / LOGIC INFORMATION
7862 ;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8
7863 ;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016
7864 ;EXEC: [224]ALUC=LLHHL :[367]D=000000
7865 ;CODES: [367] SPS=3 / N:C=0100
7866 ;SYNC: B05J2 (-)
7867 ;KEY SIG: K3-3 SM=0L / K3-3 DM=1L / K5-2 PS(Z)(1)H / K3-8 BIT+CMP+TSTH
7868
7869
7870
7871
7872
7873
7874
7875
7876 013172 012700 000141 T0141: MOV #0141,R0 ;LOAD R0 WITH TEST NO.
7877 013176 112760 000377 070140 MOVB #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
7878 013204 012702 177564 MOV #XCSR,R2 ;DEST ADDR = XCSR
7879 013210 012704 000200 MOV #200,R4 ;RESULT S / B = 200
7880 013214 005012 R0141: CLR (R2) ;CLEAR [DEST]
7881 013216 012701 000000 MOV #0,R1 ;SET UP TIMEOUT COUNTER
7882 013222 000257 CCC ;SCOPE SYNC
7883
7884 013224 020412 I0141: CMP R4,(R2) ;TEST READY BIT - IT SHOULD BE SET
7885
7886 013226 001405 BEQ T0142 ;BR IF IT WAS
7887 013230 005301 DEC R1 ;TICK-TOCK GOES THE TIMER
7888 013232 001374 BNE I0141 ;BR IF NOT A TIMEOUT
7889
7890 013234 011203 E0141: MOV (R2),R3 ;GET THE WAS DATA
7891 013236 000000 HALT ;READY BIT IN XCSR FAILED ON A (0)
7892 013240 000765 BR R0141 ;LOCK ON HARD ERROR
7893
```

```
7894 ; *****  
7895 ; .SBTTL T0142 BASIC TEST OF DL11 - XCSR - MAINT BIT (0)  
7896 ; *****  
7897  
7898 ;MICROPROGRAMMING / LOGIC INFORMATION  
7899  
7900 ;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8  
7901  
7902 ;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016  
7903  
7904 ;EXEC: [224]ALUC=LLHHL :[367]D=000000  
7905  
7906 ;CODES: [367] SPS=3 / N:C=0100  
7907  
7908 ;SYNC: B05J2 (-)  
7909  
7910 ;KEY SIG: K3-3 SM=0L / K3-3 DM=1L / K5-2 PS(?) (1)H / K3-8 BIT+CMPT+TSTH  
7911  
7912 013242 012700 000142 T0142: MOV #0142,R0 ;LOAD R0 WITH TEST NO.  
7913 013246 112760 000377 070140 MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
7914 013254 012702 177564 MOV #XCSR,R2 ;DEST ADDR = XCSR  
7915 013260 012704 000200 MOV #200,R4 ;RESULT S / B = 200  
7916 013264 005012 R0142: CLR (R2) ;CLEAR MAINT. BIT  
7917 013266 000257 CCC ;SCOPE SYNC  
7918  
7919 013270 020412 I0142: CMP R4,(R2) ;TEST MAINT(0)  
7920  
7921 013272 001403 BEQ T0143 ;BR IF MAINT BIT CLEAR  
7922  
7923 013274 011203 E0142: MOV (R2),R3 ;GET THE WAS DATA  
7924 013276 000000 HALT ;CAN'T CLEAR MAINT BIT  
7925 013300 000771 BR R0142 ;LOCK ON HARD ERROR  
7926
```

7927
7928
7929
7930
7931
7932
7933
7934
7935
7936
7937
7938
7939
7940
7941
7942
7943
7944
7945
7946
7947
7948
7949
7950
7951
7952
7953
7954
7955
7956
7957
7958

013302 012700 000143
013306 112760 000377 070140
013314 012702 177564
013320 012704 000204
013324 012712 000004
013330 000257

013332 020412

013334 001403

013336 011203
013340 000000
013342 000770

```
; *****  
; .SBTTL T0143 BASIC TEST OF DL11 XCSR - MAINT BIT = 1  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016  
;EXEC: [224]SLUC=LLHHL :[367]D=000000  
;CODES: [367] SPS=3 / N:C=0100  
;SYNC: B05J2 (-)  
;KEY SIG: K3-3 SM=0L / K3-3 DM=1L / K5-2 PS(Z)(1)H / K3-8 BIT+COMP+TSTH  
  
T0143: MOV #0143,R0 ;LOAD R0 WITH TEST NO.  
MOV #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE  
MOV #XCSR,R2 ;DEST ADDR = XCSR  
MOV #204,R4 ;RESULT S / B = 204  
R0143: MOV #4,(R2) ;SET THE MAINT. BIT  
CCC ;SCOPE SYNC  
  
I0143: CMP R4,(R2) ;TEST MAINT.(1)  
  
BEQ T0144 ;BR IF IT WAS  
  
E0143: MOV (R2),R3 ;GET THE WAS DATA  
HALT ;CAN'T SET MAINT BIT IN XCSR  
BR R0143 ;LOCK O HARD ERROR
```

7959
7960
7961
7962
7963
7964
7965
7966
7967
7968
7969
7970
7971
7972
7973
7974
7975
7976
7977
7978
7979
7980
7981
7982
7983
7984
7985
7986
7987
7988
7989
7990
7991
7992
7993
7994
7995
7996
7997
7998
7999
8000
8001
8002
8003
8004
8005
8006
8007
8008
8009
8010
8011
8012
8013
8014

: *****
: .SBTTL T0144 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.

```
T0144:  MOV    #0144,R0          ;LOAD R0 WITH TEST NO.
        MOVB  #377,STAB1(R0) ;SET FLAG FOR THIS TEST IN MISSED TABLE
        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$:     MOV   #0,R1          ;R1 USED AS TIMEOUT COUNTER
        MOVB  (R4)+,6(R2)   ;LOAD OUTPUT BUFFER IN DL11
2$:     TSTB  (R2)          ;RECEIVER DONE SET ?
        BMI   3$           ;BR IF YES
        DEC  R1             ;COUNT THE TIMER
        BNE  2$           ;BR IF NO TIMEOUT

        HALT                ;DL11 FAILED TO RESPOND IN TIME
        BR   T0144         ;LOCK ON HARD ERROR

3$:     MOVB  2(R2),(R3)+   ;READ THE DL11 INPUT BUFFER INTO CORE
        DEC  R5             ;COUNT ONE CHAR
        BNE  1$           ;BR IF NOT DONE 8 CHARS

        CLR   4(R2)         ;TURN OFF MAINT. MODE
        MOV   #10,R5        ;RESET CHAR COUNTER
        MOV   #IBUF,R3     ;RESET INBUF POINTER
        MOV   #OBUF,R4     ;RESET OUTBUF POINTER

4$:     CMPB  (R3)+,(R4)+   ;INPUT = OUTPUT ??
        BNE  5$           ;BR IF NOT
        DEC  R5             ;COUNT ONE CHECKED
        BNE  4$           ;BR UNTIL 8 DONE
        BR   CITST        ;GO TO NEXT TEST

5$:     MOVB  -(R3),R3      ;WAS DATA IN R3 [BITS 7:0]
        MOVB  -(R4),R4      ;S / B DATA IN R4 [BITS 7:0]
        BIC  #177400,R3     ;STRIP OFF BITS <15:08>
        BIC  #177400,R4     ;
        HALT                ;RECEIVED DATA NOT EQUAL TO OUTPUT DATA
        BR   T0144         ;LOCK ON HARD ERROR
```

070140

000004

177400
177400

8015
8016

```
8017 :XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8018 :////////////////COMPREHENSIVE INSTRUCTION TESTS////////////////////////////////
8019 :XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
8020
8021 013522 012737 065266 000020 CITST: MOV #SCOPEB,@#20 ;SET UP IOT VECTOR
8022 013530 005037 000022 CLR @#22
8023 013534 012737 065406 000030 MOV #ERRB,@#30 ;SET UP EMT VECTOR
8024 013542 012737 000340 000032 MOV #340,@#32
8025 013550 012737 066050 000034 MOV #PRINT,@#34 ;SET UP TRAP VECTOR
8026 013556 012737 000340 000036 MOV #340,@#36
8027 013564 012737 064706 000024 MOV #PDWN,@#24 ;SET UP POWER FAIL VECTOR
8028 013572 012737 000340 000026 MOV #340,@#26
8029 013600 032737 010000 177580 BIT #SW12,@#SR ;INHIBIT PRINTING INTRO. I.D. MESSAGE
8030 013606 001007 BNE 1$ ;BR IF YES
8031 013610 005737 066722 TSJ @#ONCE ;FIRST TIME INTO 'CIT' TESTS ?
8032 013614 001004 BNE 1$ ;BR IF NOT - PRINT ID ONLY ONCE
8033 013616 005137 066722 COM @#ONCE ;SET FLAG TO INHIBIT PRINTING AGAIN
8034 013622 104400 TYPE ;IDENTIFY THIS PROGRAM
8035 013624 067212 IDENT1 ;ADDR OF THE ID MESSAGE
8036 013626 004737 066210 1$: JSR PC,@#TSTOP4 ;GO TEST FOR 11/40 OPTIONS
8037 013632 005037 7776 CLR @#PSW ;SET CPU PRIORITY TO LEVEL 000
8038 013636 012737 013636 066654 2$: MOV #2$,@#RETURN ;INITIALIZE SCOPE LOOP RETURN
8039
```

8040
8041
8042
8043
8044
8045
8046
8047
8048
8049
8050
8051
8052
8053
8054
8055
8056
8057
8058
8059
8060
8061
8062
8063
8064
8065
8066
8067
8068
8069
8070
8071
8072
8073

013644 012700 000145
013650 013701 013656
013654 000257

013656 001404
013660 100403
013662 102402
013664 103401
013666 000402

013670 104005
013672 013654

013674 000004

```
; *****  
      .SBTTL T0145 QUICK VERIFY TEST FOR BMI,BEQ,BVS,BCS-FLG=0  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ:      [110,347,016] FC 1,7  
  
;ACT BUTS:     37[004]100,110 / 16[110]016,016  
  
;EXEC:         N / A  
  
;CODES:        N / A  
  
;SYNC:         B05J2 (-) T = 1.5 USEC  
  
;KEY SIG:      K5-3 BR INSTR L / K3-5 BUBC3(BUT37) H  
  
T0145: MOV      #0145,R0          ;LOAD R0 WITH TEST NO.  
      MOV      @#I0145,R1        ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0145: CCC  
      ;CLEAR ALL FLAGS  
  
I0145: BEQ      E0145            ;NO BR SHOULD OCCUR-FLAG=0  
      BMI      E0145            ;NO BR SHOULD OCCUR-FLAG=0  
      BVS      E0145            ;NO BR SHOULD OCCUR-FLAG=0  
      BCS      E0145            ;NO BR SHOULD OCCUR-FLAG=0  
      BR       00145            ;GO CALL SCOPE  
  
E0145: ERROR5          ;ONE OF ABOVE BR'S FAILED  
      R0145          ;ERROR LOOP RETURN  
  
00145: SCOPE          ;CALL SCOPE LOOP UTILITY
```

```
8074 ; *****  
8075 ; .SBTTL T0146 QUICK VERIFY TEST FOR BMI,BEQ,BVS,BCS-FLAG=1  
8076 ; *****  
8077  
8078 ;MICROPROGRAMMING / LOGIC INFORMATION  
8079  
8080 ;ROM SEQ: [111,340,341,016] FC 1,7  
8081  
8082 ;ACT BUTS: 37[004]100,111 / 16[340]016,016  
8083  
8084 ;EXEC: [341]ALUC=LHLLH :[016]D=#I20146 OR #I30146 OR #I40146 OR #O0146 DEPENDEN  
8085  
8086 ;CODES: N / A  
8087  
8088 ;SYNC: B05J2 (-) T = 2 USEC  
8089  
8090 ;KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR L / K3-5 BUBC3(BUT37) H / K3-  
8091  
8092 013676 012700 000146 T0146: MOV #0146,R0 ;LOAD R0 WITH TEST NO.  
8093 013702 013701 013710 MOV @#I10146,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
8094  
8095 013706 000277 R0146: SCC ;MAKE N:C = 1111  
8096  
8097 013710 001402 I10146: BEQ I20146 ;TEST THE BEQ-IT SHOULD BR  
8098  
8099 013712 104005 E10146: ERRORS ;BEQ FAILED  
8100 013714 013706 R0146 ;ERROR LOOP RETURN  
8101  
8102 013716 100402 I20146: BMI I30146 ;TEST THE BMI-IT SHOULD BR  
8103  
8104 013720 104005 E20146: ERRORS ;BMI FAILED  
8105 013722 013706 R0146 ;ERROR LOOP RETURN  
8106  
8107 013724 102402 I30146: BVS I40146 ;TEST THE BVS-IT SHOULD BR  
8108  
8109 013726 104005 E30146: ERRORS ;BVS FAILED  
8110 013730 013706 R0146 ;ERROR LOOP RETURN  
8111  
8112 013732 103402 I40146: BCS 00146 ;TEST THE BCS-IT SHOULD BR  
8113  
8114 013734 104005 E40146: ERRORS ;BCS FAILED  
8115 013736 013706 R0146 ;ERROR LOOP RETURN  
8116  
8117 013740 000004 00146: SCOPE ;CALL SCOPE LOOP UTILITY  
8118  
8119
```

T0146 QUICK VERIFY TEST FOR BMI,BEQ,BVS,BCS-FLAG=1

SEQ 0220

8120
8121
8122
8123
8124
8125
8126
8127
8128
8129
8130
8131
8132
8133
8134
8135
8136
8137
8138
8139
8140
8141
8142
8143
8144
8145
8146
8147
8148
8149
8150
8151

: *****
: .SBTTL T0147 BNE TEST WITH Z=1
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [110,347,016] FC 1,7
:ACT BUTS: 37[004]100,110 / 16[110]016,016
:EXEC: N / A
:CODES: N / A
:SYNC: B05J2 (-) T = 1.5 USEC
:KEY SIG: K5-3 BR INSTR L / K3-5 BUBC3(BUT37) H

013742 012700 000147
013746 013701 013754
013752 000264
013754 001001
013756 000402
013760 104005
013762 013752
013764 000004

T0147: MOV #0147,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0147,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0147: SEZ ;MAKE Z=1
I0147: BNE E0147 ;TEST THE BNE-IT SHOULDN'T BR
BR 00147 ;GO TO SCOPE EXIT
E0147: ERRORS ;BNE FAILED
R0147 ;ERROR LOOP RETURN
00147: SCOPE ;CALL SCOPE LOOP UTILITY

8152
8153
8154
8155
8156
8157
8158
8159
8160
8161
8162
8163
8164
8165
8166
8167
8168
8169
8170 013766 012700 000150
8171 013772 013701 014000
8172
8173 013776 000244
8174 014000 001002
8175 014002 104005
8176 014004 013776
8177 014006 000004
8178

```
; *****  
; .SBTTL T0150 BNE TEST WITH Z=0  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [111,340,341,016] FC 1,7  
;ACT BUTS: 37[004]100,111 / 16[340]016,016  
;EXEC: [016] D = #00150  
;CODES: N / A  
;SYNC: B05J2 (-) T = 1.8 USEC  
;KEY SIG: K5-3 BR INSTR L / K5-3 FALSE BR L / K3-3 SM=1L  
T0150: MOV #0150,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0150,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0150: CLZ ;MAKE Z=0  
I0150: BNE 00150 ;TEST THE BNE-IT SHOULD BR  
E0150: ERRORS ;BNE FAILED  
R0150 ;ERROR LOOP RETURN  
00150: SCOPE ;CALL SCOPE LOOP UTILITY
```

8179
8180
8181
8182
8183
8184
8185
8186
8187
8188
8189
8190
8191
8192
8193
8194
8195
8196
8197
8198
8199
8200
8201
8202
8203
8204
8205
8206
8207
8208
8209
8210

```
; *****  
; .SBTTL T0151 BPL TEST WITH N=1  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [110,347,016] FC 1,7  
;ACT BUTS: 37[004]100,110 / 16[110]016,016  
;EXEC: NO BRANCH  
;CODES: N / A  
;SYNC: B05J2 (-) T= 1.4 USEC  
;KEY SIG: K5-3 BR INSTR L / K3-3 SM=0L / K3-4 IR15 L  
T0151: MOV #0151,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0151,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0151: SEN ;MAKE N=1  
I0151: BPL E0151 ;TEST THE BPL-IT SHOULDN'T BR  
BR 00151 ;GO TO SCOPE EXIT  
E0151: ERROR5 ;BPL FAILED  
R0151 ;ERROR LOOP RETURN  
00151: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
8211 ; *****  
8212 ; .SBTTL T0152 BPL TEST WITH N=0  
8213 ; *****  
8214  
8215 ;MICROPROGRAMMING / LOGIC INFORMATION  
8216  
8217 ;ROM SEQ: [111,340,341,016] FC 1,7  
8218  
8219 ;ACT BUTS: 37[004]100,111 / 16[340]016,016  
8220  
8221 ;EXEC: [016] D = #00152  
8222  
8223 ;CODES: N / A  
8224  
8225 ;SYNC: B05J2 (-) T = 1.8 USEC  
8226  
8227 ;KEY SIG: K5-3 BR INSTR L / K5-3 FALSE BR L / K3-3 SM=0L / K3-4 IR15 L  
8228  
8229 014034 012700 000152 T0152: MOV #0152,R0 ;LOAD R0 WITH TEST NO.  
8230 014040 013701 014046 MOV @#I0152,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
8231  
8232 014044 000250 R0152: CLN ;MAKE N=0  
8233  
8234 014046 100002 I0152: BPL 00152 ;TEST THE BPL-IT SHOULD BR  
8235  
8236 014050 104005 E0152: ERRORS ;BPL FAILED  
8237 014052 014044 R0152 ;ERROR LOOP RETURN  
8238  
8239 014054 000004 00152: SCOPE ;CALL SCOPE LOOP UTILITY  
8240  
8241
```

```
8242 ; *****  
8243 ; .SBTTL T0153 BVC TEST WITH V=1  
8244 ; *****  
8245  
8246 ;MICROPROGRAMMING / LOGIC INFORMATION  
8247  
8248 ;ROM SEQ: [110,347,016] FC 1,7  
8249  
8250 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
8251  
8252 ;EXEC: NO BRANCH  
8253  
8254 ;CODES: N / A  
8255  
8256 ;SYNC: B05J2 (-) T = 1.4 USEC  
8257  
8258 ;KEY SIG: K5-3 BR INSTR L / K3-3 SM=2L / K3-4 IR15 L  
8259  
8260 014056 012700 000153 T0153: MOV #0153,R0 ;LOAD R0 WITH TEST NO.  
8261 014062 013701 014070 MOV @#I0153,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
8262  
8263 014066 000262 R0153: SEV ;MAKE V=1  
8264  
8265 014070 102001 I0153: BVC E0153 ;TEST THE BVC-IT SHOULDN'T BR  
8266 014072 000402 BR 00153 ;GO TO SCOPE EXIT  
8267  
8268 014074 104005 E0153: ERROR5 ;BVC FAILED  
8269 014076 014066 R0153 ;ERROR LOOP RETURN  
8270  
8271 014100 000004 00153: SCOPE ;CALL SCOPE LOOP UTILITY  
8272  
8273
```

```
8274 ; *****  
8275 ; .SBTTL T0154 BVC TEST WITH V=0  
8276 ; *****  
8277  
8278 ;MICROPROGRAMMING / LOGIC INFORMATION  
8279  
8280 ;ROM SEQ: [111,340,341,016] FC 1,7  
8281  
8282 ;ACT BUTS: 37[004]100,111 / 16[340]016,016  
8283  
8284 ;EXEC: [016] D = #00154  
8285  
8286 ;CODES: N / A  
8287  
8288 ;SYNC: B05J2 (-) T = 1.8 USEC  
8289  
8290 ;KEY SIG: K5-3 BR INSTR L / KK5-3 FALSE BR L / K3-3 SM-2L / K3-4 IR15  
8291  
8292  
8293 ;CODES: N / A  
8294 014102 012700 000154 T0154: MOV #0154,R0 ;LOAD R0 WITH TEST NO.  
8295 014106 013701 014114 MOV @#I0154,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
8296  
8297 014112 000242 R0154: CLV ;MAKE V=0  
8298  
8299 014114 102002 I0154: BVC 00154 ;TEST THE BVC-IT SHOULD BR  
8300  
8301 014116 104005 E0154: ERROR5 ;BVC FAILED  
8302 014120 014112 R0154 ;ERROR LOOP RETURN  
8303  
8304 014122 000004 00154: SCOPE ;CALL SCOPE LOOP UTILITY  
8305  
8306
```

```
8307 ; *****  
8308 ; .SBTTL T0155 BCC TEST WITH C=1  
8309 ; *****  
8310 ;MICROPROGRAMMING / LOGIC INFORMATION  
8311 ;ROM SEQ: [110,347,016] FC 1,7  
8312 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
8313 ;EXEC: NO BRANCH  
8314 ;CODES: N / A  
8315 ;SYNC: B05J2 (-) T = 1.4 USEC  
8316 ;KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L / K3-4 IR15 L  
8317  
8318  
8319  
8320  
8321  
8322  
8323  
8324  
8325 014124 012700 000155 T0155: MOV #0155,R0 ;LOAD R0 WITH TEST NO.  
8326 014130 013701 014136 MOV @#I0155,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
8327  
8328 014134 000261 R0155: SEC ;MAKE C=1  
8329  
8330 014136 103001 I0155: BCC E0155 ;TEST THE BCC, IT SHOULDN'T BR  
8331 014140 000402 BR 00155 ;GO TO SCOPE EXIT  
8332  
8333 014142 104005 E0155: ERROR5 ;BCC FAILED  
8334 014144 014134 R0155 ;ERROR LOOP RETURN  
8335  
8336 014146 000004 00155: SCOPE ;CALL SCOPE LOOP UTILITY  
8337
```

```
8338 ; *****  
8339 ; .SBTTL T0156 BCC TEST WITH C=0  
8340 ; *****  
8341  
8342 ;MICROPROGRAMMING / LOGIC INFORMATION  
8343  
8344 ;ROM SEQ: [111,340,341,016] FC 1,7  
8345  
8346 ;ACT BUTS: 37[004]100,111 / 16[340]016,016  
8347  
8348 ;EXEC: [016] D = #00156  
8349  
8350 ;CODES: N / A  
8351  
8352 ;SYNC: B05J2 (-) T = 1.8 USEC  
8353  
8354 ;KEY SIG: K5-3 BR INSTR L / K5-3 FALSE BR L / K3-3 SM=3L / K3-4 IR15 L  
8355  
8356  
8357  
8358 014150 012700 000156 T0156: MOV #0156,R0 ;LOAD R0 WITH TEST NO.  
8359 014154 013701 014162 MOV @#I0156,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
8360  
8361 014160 000241 R0156: CLC ;MAKE C=0  
8362  
8363 014162 103002 I0156: BCC 00156 ;TEST THE BCC-IT SHOULD BR  
8364  
8365 014164 104005 E0156: ERRORS ;BCC FAILED  
8366 014166 014160 R0156 ;ERROR LOOP RETURN  
8367  
8368 014170 000004 00156: SCOPE ;CALL SCOPE LOOP UTILITY  
8369  
8370
```

```
8371 ; *****  
8372 ; .SBTTL T0157 VERIFY NO BRANCH MICROROUTINE DOES NOT CLR FLAGS  
8373 ; *****  
8374 ;MICROPROGRAMMING / LOGIC INFORMATION  
8375 ;ROM SEQ: [110,347,016] FC 1,7  
8376 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
8377 ;EXEC: NO BRANCH  
8378 ;CODES: N:C = 1111 (NO CHANGE)  
8379 ;SYNC: B05J2 (-) T = 1.4 USEC  
8380 ;KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L / K3-4 IR15 L  
8381  
8382  
8383  
8384  
8385  
8386  
8387  
8388  
8389 014172 012700 000157 T0157: MOV #0157,R0 ;LOAD R0 WITH TEST NO.  
8390 014176 013701 014204 MOV @#I0157,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
8391  
8392 014202 000277 R0157: SCC ;MAKE N:C = 1111  
8393  
8394 014204 103007 I0157: BCC E0157 ;TEST THE BCC-IT SHOULDN'T BR  
8395  
8396 014206 013702 177776 MOV @#PSW,R2 ;GET WAS FLAGS  
8397 014212 022702 000017 CMP #17,R2 ;N:C = 1111?  
8398 014216 001404 BEQ 00157 ;BR IF YES  
8399  
8400 014220 010237 177776 MOV R2,@#PSW ;RESTORE N:C  
8401  
8402 014224 104005 E0157: ERRORS ;NO, BRANCH MICROROUTINE ALTERED CODES  
8403 014226 014202 R0157 ;ERROR LOOP RETURN  
8404  
8405 014230 000004 00157: SCOPE ;CALL SCOPE LOOP UTILITY  
8406  
8407
```

8408
8409
8410
8411
8412
8413
8414
8415
8416
8417
8418
8419
8420
8421
8422
8423
8424
8425
8426
8427
8428
8429
8430
8431
8432
8433
8434
8435
8436
8437
8438
8439
8440
8441
8442
8443
8444
8445
8446
8447

; *****
; .SBTTL T0160 VERIFY BRANCH MICROROUTINE DOES NOT CLR FLAGS
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [111,340,341,016] FC 1,7
;ACT BUTS: 37[004]100,111 / 16[340]016,016
;EXEC: [016] D = #A0160
;CODES: N:C = 1111 (NO CHANGE)
;SYNC: B05J2 (-) T = 1.8 USEC
;KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR / K3-3 SM=0L / K3-3 IR(14:12) = 0 L

T0160: MOV #0160,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0160,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0160: SCC ;MAKE N:C = 1111
I0160: BR A0160 ;TEST THE BR
E10160: ERROR5 ;JUST IN CASE THE BR DIDN'T WORK
R0160 ;ERROR LOOP RETURN
A0160: MOV @#PSW,R2 ;GET THE FLAGS
CMP #17,R2 ;N:C = 1111?
BEQ 00160 ;BR IF YES
MOV R2,@#PSW ;RESTORE FLAGS
E20160: ERROR5 ;BRANCH MICROROUTINE ALTERED CODES
R0160 ;ERROR LOOP RETURN
00160: SCOPE ;CALL SCOPE LOOP UTILITY

014232 012700 000160
014236 013701 014244
014242 000277
014244 000402
014246 104005
014250 014242
014252 013702 177776
014256 022702 000017
014262 001404
014264 010237 177776
014270 104005
014272 014242
014274 000004

8448
8449
8450
8451
8452
8453
8454
8455
8456
8457
8458
8459
8460
8461
8462
8463
8464
8465
8466
8467
8468
8469
8470
8471
8472
8473
8474
8475
8476
8477
8478
8479
8480
8481
8482
8483
8484

```
; *****  
      .SBTTL T0161 VERIFY NO BRANCH MICROROUTINE DOES NOT SET FLAGS  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ:      [110,347,016] FC 1,7  
  
;ACT BUTS:     37[004]100,110 / 16[110]016,016  
  
;EXEC:         NO BRANCH  
  
;CODES:        N:C = 0000      (NO CHANGE)  
  
;SYNC:         B05J2 (-) T = 1.4 USEC  
  
;KEY SIG:      K5-3 BR INSTR L / K3-3 SM=3L / K3-4 IR15 L  
  
T0161:  MOV      #0161,R0          ;LOAD R0 WITH TEST NO.  
        MOV      @#I0161,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
  
R0161:  CCC  
        ;MAKE N:C = 0000  
  
I0161:  BCS      E0161          ;TEST THE BCS-IT SHOULDN'T BR  
        MOV      @#PSW,R2          ;GET FLAGS  
        TST      R2              ;N:C = 0000  
        BEQ      00161          ;BR IF YES  
  
        MOV      R2,@#PSW        ;RESTORE FLAGS  
  
E0161:  ERROR5  
        R0161          ;NO BRANCH MICROROUTINE-ALTERED CODES  
        ;ERROR LOOP RETURN  
  
00161:  SCOPE          ;CALL SCOPE LOOP UTILITY
```

T0161 VERIFY NO BRANCH MICROROUTINE DOES NOT SET FLAGS

SEQ 0231

```
8485 ; *****
8486 ; .SBTTL T0162 VERIFY BRANCH MICROROUTINE DOES NOT SET FLAGS
8487 ; *****
8488 ;MICROPROGRAMMING / LOGIC INFORMATION
8489 ;ROM SEQ: [111,340,341,016] FC 1,7
8490 ;ACT BUTS: 37[004]100,111 / 16[340]016,016
8491 ;EXEC: [016] D = #A0162
8492 ;CODES: N:C = 0000 (NO CHANGE)
8493 ;SYNC: B05J2 (-) T = 1.8 USEC
8494 ;KEY SIG: K5-3 BIT INSTR L / K5-3 TRUE BR L / K3-3 SM=0L / K3-4 IR(14:12)=0
8495
8496
8497
8498
8499
8500
8501 ;KEY SIG: K5-3 BIT INSTR L / K5-3 TRUE BR L / K3-3 SM=0L / K3-4 IR(14:12)=0
8502
8503 014334 012700 000162 T0162: MOV #0162,R0 ;LOAD R0 WITH TEST NO.
8504 014340 013701 014346 MOV @#I0162,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
8505
8506 014344 000257 R0162: CCC ;MAKE N:C = 0000
8507
8508 014346 000402 I0162: BR A0162 ;TEST THE BR
8509
8510 014350 104005 E10162: ERROR5 ;JUST IN CASE THE BR DIDN'T WORK
8511 014352 014344 R0162 ;ERROR LOOP RETURN
8512
8513 014354 013702 177776 A0162: MOV @#PSW,R2 ;GET FLAGS
8514 014360 005702 TST R2 ;N:C = 0000
8515 014362 001404 BEQ 00162 ;BR IF YES
8516
8517 014364 010237 177776 MOV R2,@#PSW ;RESTORE FLAGS
8518
8519 014370 104005 E20162: ERROR5 ;BRANCH MICROROUTINE ALTERED CODES.
8520 014372 014344 R0162 ;ERROR LOOP RETURN
8521
8522 014374 000004 00162: SCOPE ;CALL SCOPE LOOP UTILITY
8523
8524
```

```

8525 ; *****
8526 ; .SBTTL T0163 BGE TEST WITH N,V = 00
8527 ; *****
8528
8529 ;MICROPROGRAMMING / LOGIC INFORMATION
8530
8531 ;ROM SEQ: [111,340,341,016] FC 1,7
8532
8533 ;ACT BUTS: 37[004]100,111 / 16[340]016,016
8534
8535 ;EXEC: [016] D = #00163
8536
8537 ;CODES: N:C = 0000
8538
8539 ;SYNC: B05J2 (-) T = 1.8 USEC
8540
8541 ;KEY SIG: K5-3 BR INSTR L / K5-3 FALSE BR L / K3-3 SM=2L
8542
8543 014376 012700 000163 T0163: MOV #0163,R0 ;LOAD R0 WITH TEST NO.
8544 014402 013701 014410 MOV @#I0163,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
8545
8546 014406 000257 R0163: CCC ;MAKE N:C = 0000
8547
8548 014410 002002 I0163: BGE 00163 ;TEST THE BGE-IT SHOULD BR
8549
8550 014412 104005 E0163: ERROR5 ;BGE FAILED
8551 014414 014406 R0163 ;ERROR LOOP RETURN
8552
8553 014416 000004 00163: SCOPE ;CALL SCOPE LOOP UTILITY
8554
8555

```

T0165 BGE TEST WITH N,V = 00

SEQ 0233

8556
8557
8558
8559
8560
8561
8562
8563
8564
8565
8566
8567
8568
8569
8570
8571
8572
8573
8574 014420 012700 000164
8575 014424 013701 014434
8576
8577 014430 000257
8578 014432 000262
8579
8580 014434 002001
8581 014436 000402
8582
8583 014440 104005
8584 014442 014430
8585 014444 000004
8586

```
; *****  
; .SBTTL T0164 BGE TEST WITH N,V = 01  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [110,347,016] FC 1,7  
;ACT BUTS: 37[004]100,110 / 16[110]016,016  
;EXEC: NO BRANCH  
;CODES: N:C = 0010  
;SYNC: B05J2 (-) T = 1.4 USEC  
;KEY SIG: K5-3 BR INSTR L / K3-3 SM=2L / K3-3 IR(14:12)=0 L  
T0164: MOV #0164,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0164,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0164: CCC ;CLEAR FLAGS  
SEV ;MAKE N,V = 01  
I0164: BGE E0164 ;TEST THE BGE-IT SHOULDN'T BR  
BR 00164 ;GO TO SCOPE EXIT  
E0164: ERROR5 ;BGE FAILED  
R0164 ;ERROR LOOP RETURN  
00164: SCOPE ;CALL SCOPE LOOP UTILITY
```

8587
8588
8589
8590
8591
8592
8593
8594
8595
8596
8597
8598
8599
8600
8601
8602
8603
8604
8605
8606
8607
8608
8609
8610
8611
8612
8613
8614
8615
8616
8617
8618
8619

; *****
; .SBTTL T0165 BGE TEST WITH N,V = 10
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [110,347,016] FC 1,7
;ACT BUTS: 37[004]100,110 / 16[110]016,016
;EXEC: NO BRANCH
;CODES: N:C = 1000
;SYNC: B05J2 (-) T = 1.4 USEC
;KEY SIG: K5-3 BR INSTR L / K3-3 SM=2L

014446 012700 000165
014452 013701 014462
014456 000257
014460 000270
014462 002001
014464 000402
014466 104005
014470 014456
014472 000004

T0165: MOV #0165,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0165,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0165: CCC ;CLEAR FLAGS
SEN ;MAKE N,V = 10
I0165: BGE E0165 ;TEST THE BGE-IT SHOULDN'T BR
BR 00165 ;GO TO SCOPE EXIT
E0165: ERROR5 ;BGE FAILED
R0165 ;ERROR LOOP RETURN
00165: SCOPE ;CALL SCOPE LOOP UTILITY

```
8620 ; *****
8621 ; .SBTTL T0166 BGE TEST WITH N,V = 11
8622 ; *****
8623 ;MICROPROGRAMMING / LOGIC INFORMATION
8624 ;ROM SEQ: [111,340,341,016] FC 1,7
8625 ;ACT BUTS: 37[004]100,111 / 16[340]016,016
8626 ;EXEC: [016] D = #00166
8627 ;CODES: N:C = 1010
8628 ;SYNC: B05J2 (-) T = 1.8 USEC
8629 ;KEY SIG: K5-3 BR INSTR L / K5-3 FALSE BR L / K3-3 SM=2L / K3-3 IR(14:
8630
8631
8632
8633
8634
8635
8636
8637
8638 014474 012700 000166 T0166: MOV #0166,R0 ;LOAD R0 WITH TEST NO.
8639 014500 013701 014510 MOV @#I0166,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
8640
8641 014504 000257 R0166: CCC ;CLEAR FLAGS
8642 014506 000272 272 ;MAKE N,V = 11
8643
8644 014510 002002 I0166: BGE 00166 ;TEST THE BGE-IT SHOULD BR
8645
8646 014512 104005 E0166: ERRORS ;BGE FAILED
8647 014514 014504 R0166 ;ERROR LOOP RETURN
8648
8649 014516 000004 00166: SCOPE ;CALL SCOPE LOOP UTILITY
8650
8651
```

8652
8653
8654
8655
8656
8657
8658
8659
8660
8661
8662
8663
8664
8665
8666
8667
8668
8669
8670
8671
8672
8673
8674
8675
8676
8677
8678
8679
8680
8681
8682
8683

```
; *****  
; .SBTTL T0167 BLT TEST WITH N,V = 00  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [110,347,016] FC 1,7  
;ACT BUTS: 37[004]100,110 / 16[110]016,016  
;EXEC: NO BRANCH  
;CODES: N:C = 0000  
;SYNC: B05J2 (-) T = 1.4 USEC  
;KEY SIG: K5-3 BR INSTR L / K3-3 SM=2L  
T0167: MOV #0167,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0167,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0167: CCC ;CLEAR FLAGS  
I0167: BLT E0167 ;TEST THE BLT-IT SHOULDN'T BR  
BR 00167 ;GO TO SCOPE EXIT  
E0167: ERROR5 ;BLT FAILED  
R0167 ;ERROR LOOP RETURN  
00167: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
8684 ; *****  
8685 ; .SBTTL T0170 BLT TEST WITH N,V = 01  
8686 ; *****  
8687  
8688 ;MICROPROGRAMMING / LOGIC INFORMATION  
8689  
8690 ;ROM SEQ: [111,340,341,016] FC 1,7  
8691  
8692 ;ACT BUTS: 37[004]100,111 / 16[340]016,016  
8693  
8694 ;EXEC: [016] D = #00170  
8695  
8696 ;CODES: N:C = 0010  
8697  
8698 ;SYNC: B05J2 (-) T = 1.8 USEC  
8699  
8700 ;KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR L / K3-3 SM=2L / K3-3 IR(14:1  
8701  
8702 014544 012700 000170 T0170: MOV #0170,R0 ;LOAD R0 WITH TEST NO.  
8703 014550 013701 014560 MOV @#I0170,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
8704  
8705 014554 000257 R0170: CCC ;CLEAR FLAGS  
8706 014556 000262 SEV ;MAKE N,V = 01  
8707  
8708 014560 002402 I0170: BLT 00170 ;TEST THE BLT-IT SHOULD BR  
8709  
8710 014562 104005 E0170: ERRORS ;BLT FAILED  
8711 014564 014554 R0170 ;ERROR LOOP RETURN  
8712  
8713 014566 000004 00170: SCOPE ;CALL SCOPE LOOP UTILITY  
8714  
8715
```

8716
8717
8718
8719
8720
8721
8722
8723
8724
8725
8726
8727
8728
8729
8730
8731
8732
8733
8734
8735
8736
8737
8738
8739
8740
8741
8742
8743
8744
8745
8746
8747

```
; *****  
; .SBTTL T0171 BLT TEST WITH N,V = 10  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [111,340,341,016] FC 1,7  
;ACT BUTS: 37[004]100,111 / 16[340]016,016  
;EXEC: [016] D = #00171  
;CODES: N:C = 1000  
;SYNC: B05J2 (-) T = 1.8 USEC  
;KEY SIG: K5-3 BR INSTR L / K3-3 SM=2L / K5-3 TRUE L  
T0171: MOV #0171,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0171,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0171: CCC ;CLEAR FLAGS  
SEN ;SET N - N,V = 10  
I0171: BLT 00171 ;TEST THE BLT-IT SHOULD BR  
E0171: ERROR5 ;BLT FAILED  
R0171 ;ERROR LOOP RETURN  
00171: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
014570 012700 000171  
014574 013701 014604  
014600 000257  
014602 000270  
014604 002402  
014606 104005  
014610 014600  
014612 000004
```

8748
8749
8750
8751
8752
8753
8754
8755
8756
8757
8758
8759
8760
8761
8762
8763
8764
8765
8766
8767
8768
8769
8770
8771
8772
8773
8774
8775
8776
8777
8778
8779
8780

014614 012700 000172
014620 013701 014630
014624 000257
014626 000272
014630 002401
014632 000402
014634 104005
014636 014624
014640 000004

```
; *****  
; .SBTTL T0172 BLT TEST WITH N,V = 11  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [110,347,016] FC 1,7  
;ACT BUTS: 37[004]100,110 / 16[110]016,016  
;EXEC: NO BRANCH  
;CODES: N:C = 1010  
;SYNC: B05J2 (-) T = 1.4 USEC  
;KEY SIG: K5-3 BR INSTR I. / K3-3 SM=2L / K3-3 IR(14:12)=0 L  
T0172: MOV #0172,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0172,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0172: CCC ;CLEAR FLAGS  
272 ;MAKE N,V = 11  
I0172: BLT E0172 ;TEST THE BLT-IT SHOULDN'T BR  
BR 00172 ;GO TO SCOPE EXIT  
E0172: ERROR5 ;BLT FAILED  
R0172 ;ERROR LOOP RETURN  
00172: SCOPE ;CALL SCOPE LOOP UTILITY
```

8781
8782
8783
8784
8785
8786
8787
8788
8789
8790
8791
8792
8793
8794
8795
8796
8797
8798
8799
8800
8801
8802
8803
8804
8805
8806
8807
8808
8809
8810
8811
8812
8813

```
; *****  
      .SBTTL T0173 BGT TEST WITH Z = 1 AND N,V = 01  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ:      [110,347,016] FC 1,7  
;ACT BUTS:     37[004]100,110 / 16[110]016,016  
;EXEC:         NO BRANCH  
;CODES:        N:C = 0110  
;SYNC:         B05J2 (-) T = 1.4 USEC  
;KEY SIG:      K5-3 BR INSTR L / K3-3 SM=3L  
  
T0173:  MOV      #0173,R0          ;LOAD R0 WITH TEST NO.  
        MOV      @#I0173,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
  
R0173:  CCC  
        266                      ;CLEAR FLAGS  
                          ;SET Z AND V  
  
I0173:  BGT      E0173          ;TEST THE BGT-IT SHOULDN'T BR  
        BR       00173          ;GO TO SCOPE EXIT  
  
E0173:  ERROR5  
        R0173                    ;BGT FAILED  
                          ;ERROR LOOP RETURN  
  
00173:  SCOPE                    ;CALL SCOPE LOOP UTILITY
```

```
8814 ; *****  
8815 ; .SBTTL T0174 BGT TEST WITH Z = 0 AND N,V = 01  
8816 ; *****  
8817  
8818 ;MICROPROGRAMMING / LOGIC INFORMATION  
8819  
8820 ;ROM SEQ: [110,347,016] FC 1,7  
8821  
8822 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
8823  
8824 ;EXEC: NO BRANCH  
8825  
8826 ;CODES: N:C = 0010  
8827  
8828 ;SYNC: B05J2 (-) T = 1.4 USEC  
8829  
8830 ;KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L / K3-3 IR(14:12)=0 L  
8831  
8832 014670 012700 000174 T0174: MOV #0174,R0 ;LOAD R0 WITH TEST NO.  
8833 014674 013701 014716 MOV @#I0174,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
8834  
8835 014700 032737 000010 066642 BIT #10,@#BPTLOC ;BREAKPOINT HALT SET ??  
8836 014706 001401 BEQ .+4 ;BR IF NOT  
8837 014710 000000 HALT ;BREAK - DEPRESS CONTINUE TO RESTART  
8838 014712 000257 R0174: CCC ;CLEAR FLAGS  
8839 014714 000262 SEV ;SET V  
8840  
8841 014716 003001 I0174: BGT E0174 ;TEST THE BGT-IT SHOULD NOT BR  
8842 014720 000402 BR 00174 ;GO TO SCOPE LOOP EXIT  
8843  
8844 014722 104005 E0174: ERROR5 ;BGT FAILED  
8845 014724 014712 R0174 ;ERROR LOOP RETURN  
8846  
8847 014726 000004 00174: SCOPE ;CALL SCOPE LOOP UTILITY  
8848  
8849
```

8850
8851
8852
8853
8854
8855
8856
8857
8858
8859
8860
8861
8862
8863
8864
8865
8866
8867
8868 014730 012700 000175
8869 014734 013701 014744
8870
8871 014740 000257
8872 014742 000264
8873
8874 014744 003001
8875 014746 000402
8876
8877 014750 104005
8878 014752 014740
8879
8880 014754 000004

```
; *****  
; .SBTTL T0175 BGT TEST WITH Z = 1 AND N,V = 00  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [110,347,016] FC 1,7  
;ACT BUTS: 37[004]100,110 / 16[110]016,016  
;EXEC: NO BRANCH  
;CODES: N:C = 0100  
;SYNC: B05J2 (-) T = 1.4 USEC  
;KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L  
  
T0175: MOV #0175,R0 ;LOAD R0 WITH TEST NO.  
MOV @I0175,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
  
R0175: CCC ;CLEAR FLAGS  
SEZ ;SET Z  
  
I0175: BGT E0175 ;TEST THE BGT-IT SHOULD NOT BR  
BR 00175 ;GO TO SCOPE LOOP EXIT  
  
E0175: ERRORS ;BGT FAILED  
R0175 ;ERROR LOOP RETURN  
  
00175: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
8881 ; *****
8882 ; .SBTTL T0176 BGT TEST WITH Z = 0 AND N,V = 00
8883 ; *****
8884
8885 ;MICROPROGRAMMING / LOGIC INFORMATION
8886
8887 ;ROM SEQ: [111,340,347,016] FC 1,7
8888
8889 ;ACT BUTS: 37[004]100,111 / 16[340]016,016
8890
8891 ;EXEC: [016] D = #00176
8892
8893 ;CODE N:C = 0000
8894
8895 ;SYNC: B05J2 (-) T = 1.8 USEC
8896
8897 ;KEY SIG: K5-3 BR INSTR L / K5-3 FALSE BR L / K3-3 SM=3L / K3-3 IR(14:12)=0
8898
8899 014756 012700 000176 T0176: MOV #0176,R0 ;LOAD R0 WITH TEST NO.
8900 014762 013701 014770 MOV @#I0176,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
8901
8902 014766 000257 R0176: CCC ;CLEAR FLAGS
8903
8904 014770 003002 I0176: BGT 00176 ;TEST THE BGT - IT SHOULD BR
8905
8906 014772 104005 E0176: ERROR5 ;BGT FAILED
8907 014774 014766 R0176 ;ERROR LOOP RETURN
8908
8909 014776 000004 00176: SCOPE ;CALL SCOPE LOOP UTILITY
8910
8911
8912
```

8913
8914
8915
8916
8917
8918
8919
8920
8921
8922
8923
8924
8925
8926
8927
8928
8929
8930
8931
8932
8933
8934
8935
8936
8937
8938
8939
8940
8941
8942
8943
8944
8945

: *****
: .SBTTL T0177 BGT TEST WITH Z = 1 AND N,V = 01
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [110,347,016] FC 1,7
:ACT BUTS: 37[004]100,110 / 16[110]016,016
:EXEC: NO BRANCH
:CODES: N:C = 0110
:SYNC: B05J2 (-) T = 1.4 USEC
:KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L

015000 012700 000177
015004 013701 015014

015010 000257
015012 000266

015014 003001
015016 000402

015020 104005
015022 015010

015024 000004

T0177: MOV #0177,R0 ;LOAD R0 WITH TEST NO.
MOV @#10177,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

R0177: CCC ;CLEAR FLAGS
266 ;MAKE N,V = 01 AND Z = 1

I0177: BGT E0177 ;TEST THE BGT-IT SHOULDN'T BR
BR 00177 ;GO TO SCOPE EXIT

E0177: ERROR5 ;BGT FAILED
R0177 ;ERROR LOOP RETURN

00177: SCOPE ;CALL SCOPE LOOP UTILITY

8946
8947
8948
8949
8950
8951
8952
8953
8954
8955
8956
8957
8958
8959
8960
8961
8962
8963
8964
8965
8966
8967
8968
8969
8970
8971
8972
8973
8974
8975
8976
8977
8978

; *****
; .SBTTL T0200 BGT TEST WITH Z = 1 AND N,V = 10
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [110,347,016] FC 1,7
;ACT BUTS: 37[004]100,110 / 16[110]016,016
;EXEC: NO BRANCH
;CODES: N:C = 1100
;SYNC: B05J2 (-) T = 1.4 USEC
;KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L / K3-3 IR(14:12)=0 L

015026 012700 000200
015032 013701 015042

015036 000257
015040 000274

015042 003001
015044 000402

015046 104005
015050 015036

015052 000004

T0200: MOV #0200,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0200,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

R0200: CCC ;CLEAR FLAGS
274 ;MAKE Z = 1 AND N,V = 10

I0200: BGT E0200 ;TEST THE BLT-IT SHOULDN'T BR
BR 00200 ;GO TO SCOPE EXIT

E0200: ERROR5 ;BLT FAILED
R0200 ;ERROR LOOP RETURN

00200: SCOPE ;CALL SCOPE LOOP UTILITY

8979
8980
8981
8982
8983
8984
8985
8986
8987
8988
8989
8990
8991
8992
8993
8994
8995
8996
8997
8998
8999
9000
9001
9002
9003
9004
9005
9006
9007
9008
9009

015054 012700 000201
015060 013701 015070
015064 000257
015066 000276
015070 003001
015072 000402
015074 104005
015076 015064
015100 000004

T0200 BGT TEST WITH Z = 1 AND N,V = 10
: *****
: .SBTTL T0201 BGT TEST WITH Z = 1 AND N,V = 11
: *****
:MICROPROGRAMMING / LOGIC INFORMATION
:ROM SEQ: [110,347,016] FC 1,7
:ACT BUTS: 37[004]100,110 / 16[110]016,016
:EXEC: NO BRANCH
:CODES: N:C = 1110
:SYNC: B05J2 (-) T = 1.4 USEC
:KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L
T0201: MOV #0201,R0 ;LOAD R0 WITH TEST NO.
MOV @I0201,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0201: CCC ;CLEAR FLAGS
276 ;MAKE Z = 1 AND N,V = 11
I0201: BGT E0201 ;TEST THE BGT-IT SHOULD NOT BR
BR 00201 ;GO TO SCOPE EXIT
E0201: ERRORS ;BLT FAILED
R0201 ;ERROR LOOP RETURN
00201: SCOPE ;CALL SCOPE LOOP UTILITY

```
9010 ; *****  
9011 ; .SBTTL T0202 BGT TEST WITH Z=0 AND N,V=11  
9012 ; *****  
9013 ;MICROPROGRAMMING / LOGIC INFORMATION  
9014 ;ROM SEQ: [111,340,341,016] FC 1,7  
9015 ;ACT BUTS: 37[004]100,111 / 16[340]016,016  
9016 ;EXEC: [016] D = #00202  
9017 ;CODES: N:C = 1010  
9018 ;SYNC: B05J2 (-) T = 1.8 USEC  
9019 ;KEY SIG: K5-3 BR INSTR L / K5-3 FALSE BR L / K3-3 SM=3L / K3-3 IR(14:12)=0  
9020  
9021  
9022  
9023  
9024  
9025  
9026  
9027  
9028  
9029 015102 012700 000202 T0202: MOV #0202,R0 ;LOAD R0 WITH TEST NO.  
9030 015106 013701 015116 MOV @#I0202,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9031  
9032 015112 000257 R0202: CCC ;CLEAR FLAGS  
9033 015114 000272 272 ;MAKE N:C=1010  
9034  
9035 015116 003002 I0202: BGT 00202 ;TEST THE BGT - IT SHOULD BR  
9036  
9037 015120 104005 E0202: ERROR5 ;BGT FAILED  
9038 015122 015112 R0202 ;ERROR LOOP RETURN  
9039  
9040 015124 000004 00202: SCOPE ;CALL SCOPE LOOP UTILITY  
9041  
9042  
9043
```

```
9044 ; *****  
9045 ; .SBTTL T0203 BLE TEST WITH Z = 0, AND N,V = 00  
9046 ; *****  
9047  
9048 ;MICROPROGRAMMING / LOGIC INFORMATION  
9049  
9050 ;ROM SEQ: [110,347,016] FC 1,7  
9051  
9052 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
9053  
9054 ;EXEC: NO BRANCH  
9055  
9056 ;CODES: N:C = 0000  
9057  
9058 ;SYNC: B05J2 (-) T = 1.4 USEC  
9059  
9060 ;KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L  
9061  
9062 015126 012700 000203 T0203: MOV #0203,R0 ;LOAD R0 WITH TEST NO.  
9063 015132 013701 015140 MOV @#I0203,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9064  
9065 015136 000257 R0203: CCC ;CLEAR FLAGS  
9066  
9067 015140 003401 I0203: BLE E0203 ;TEST THE BLE-IT SHOULDN'T BR  
9068 015142 000402 BR 00203 ;GO TO SCOPE EXIT  
9069  
9070 015144 104005 E0203: ERROR5 ;BLE FAILED  
9071 015146 015136 R0203 ;ERROR LOOP RETURN  
9072  
9073 015150 000004 00203: SCOPE ;CALL SCOPE LOOP UTILITY  
9074  
9075
```

```
9076 ; *****  
9077 ; .SBTTL T0204 BLE TEST WITH Z = 1 AND N,V = 00  
9078 ; *****  
9079  
9080 ;MICROPROGRAMMING / LOGIC INFORMATION  
9081  
9082 ;ROM SEQ: [111,340,341,016] FC 1,7  
9083  
9084 ;ACT BUTS: 37[004]100,111 / 16[340]016,016  
9085  
9086 ;EXEC: [016] D = #00204  
9087  
9088 ;CODES: N:C = 0100  
9089  
9090 ;SYNC: B05J2 (-) T = 1.8 USEC  
9091  
9092 ;KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR L / K3-3 SM=3L / K3-3 IR(14:12)=0 L  
9093  
9094 015152 012700 000204 T0204: MOV #0204,R0 ;LOAD R0 WITH TEST NO.  
9095 015156 013701 015166 MOV @#I0204,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9096  
9097 015162 000257 R0204: CCC ;CLEAR FLAGS  
9098 015164 000264 SEZ ;SET Z = 1  
9099  
9100 015166 003402 I0204: BLE 00204 ;TEST THE BLE-IT SHOULD BR  
9101  
9102 015170 104005 E0204: ERROR5 ;BLE FAILED  
9103 015172 015162 R0204 ;ERROR LOOP RETURN  
9104  
9105 015174 000004 00204: SCOPE ;CALL SCOPE LOOP UTILITY  
9106  
9107
```

```

9108 ; *****
9109 ; .SBTTL T0205 BLE TEST WITH Z = 0 AND N,V = 01
9110 ; *****
9111
9112 ;MICROPROGRAMMING / LOGIC INFORMATION
9113
9114 ;ROM SEQ: [111,340,341,016] FC 1,7
9115
9116 ;ACT BUTS: 37[004]100,111 / 16[340]016,016
9117
9118 ;EXEC: [016] D = #00205
9119
9120 ;CODES: N:C = 0010
9121
9122 ;SYNC: B05J2 (-) T = 1.8 USEC
9123
9124 ;KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR L / K3-3 SM=3L / K5-2 PS(V)(1)H
9125
9126 015176 012700 000205 T0205: MOV #0205,R0 ;LOAD R0 WITH TEST NO.
9127 015202 013701 015212 MOV @#I0205,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
9128
9129 015206 000257 R0205: CCC ;CLEAR FLAGS
9130 015210 000262 SEV ;MAKE Z = 0 AND N,V = 01
9131
9132 015212 003402 I0205: BLE 00205 ;TEST THE BLE-IT SHOULD BR
9133
9134 015214 104005 E0205: ERRORS ;BLE FAILED
9135 015216 015206 R0205 ;ERROR LOOP RETURN
9136
9137 015220 000004 00205: SCOPE ;CALL SCOPE LOOP UTILITY
9138
9139

```

```

9140 ; *****
9141 ; .SBTTL T0206 BLE TEST WITH Z = 0 AND N,V = 10
9142 ; *****
9143
9144 ;MICROPROGRAMMING / LOGIC INFORMATION
9145
9146 ;ROM SEQ: [111,340,341,016] FC 1,7
9147
9148 ;ACT BUTS: 37[004]100,111 / 16[340]016,016
9149
9150 ;EXEC: [016] D = #00206
9151
9152 ;CODES: N:C = 1000
9153
9154 ;SYNC: B05J2 (-) T = 1.8 USEC
9155
9156 ;KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR L / K3-3 SM=3L / K3-3 IR(14:12)=0 L
9157 ; K5-2 PS(N)(1)H
9158
9159 015222 012700 000206 T0206: MOV #0206,R0 ;LOAD R0 WITH TEST NO.
9160 015226 013701 015236 MOV @#I0206,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
9161
9162 015232 000257 R0206: CCC ;CLEAR FLAGS
9163 015234 000270 SEN ;MAKE Z = 0 AND N,V = 10
9164
9165 015236 003402 I0206: BLE 00206 ;TEST THE BLE-IT SHOULD BR
9166
9167 015240 104005 E0206: ERRORS ;BLE FAILED
9168 015242 015232 R0206 ;ERROR LOOP RETURN
9169
9170 015244 000004 00206: SCOPE ;CALL SCOPE LOOP UTILITY
9171
9172

```

```
9173 ; *****  
9174 ; .SBTTL T0207 BLE TEST WITH Z = 0 AND N,V = 11  
9175 ; *****  
9176  
9177 ;MICROPROGRAMMING / LOGIC INFORMATION  
9178  
9179 ;ROM SEQ: [110,347,016] FC 1,7  
9180  
9181 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
9182  
9183 ;EXEC: NO BRANCH  
9184  
9185 ;CODES: N:C = 1010  
9186  
9187 ;SYNC: B05J2 (-) T = 1.4 USEC  
9188  
9189 ;KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L  
9190  
9191 015246 012700 000207 T0207: MOV #0207,R0 ;LOAD R0 WITH TEST NO.  
9192 015252 013701 015262 MOV @#I0207,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9193  
9194 015256 000257 R0207: CCC ;CLEAR FLAGS  
9195 015260 000272 272 ;MAKE Z = 0 AND N,V = 11  
9196  
9197 015262 003401 I0207: BLE E0207 ;TEST THE BLE-IT SHOULDN'T BR  
9198 015264 000402 BR 00207 ;GO TO SCOPE EXIT  
9199  
9200 015266 104005 E0207: ERROR5 ;BLE FAILED  
9201 015270 015256 R0207 ;ERROR LOOP RETURN  
9202  
9203 015272 000004 00207: SCOPE ;CALL SCOPE LOOP UTILITY  
9204  
9205
```

```
9206 ; *****  
9207 ; .SBTTL T0210 BHI TEST WITH Z,C = 00  
9208 ; *****  
9209  
9210 ;MICROPROGRAMMING / LOGIC INFORMATION  
9211  
9212 ;ROM SEQ: [111,340,341,016] FC 1,7  
9213  
9214 ;ACT BUTS: 37[004]100,111 / 16[340]016,016  
9215  
9216 ;EXEC: [016] D = #00210  
9217  
9218 ;CODES: N:C = 0000  
9219  
9220 ;SYNC: B05J2 (-) T = 1.8 USEC  
9221  
9222 ;KEY SIG: K5-3 BR INSR L / K5-3 FALSE BR L / K3-3 SM=1L / K3-4 IR15 L  
9223  
9224 015274 012700 000210 T0210: MOV #0210,R0 ;LOAD R0 WITH TEST NO.  
9225 015300 013701 015306 MOV @#I0210,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9226  
9227 015304 000257 R0210: CCC ;MAKE Z,C = 00  
9228  
9229 015306 101002 I0210: BHI 00210 ;TEST THE BHI-IT SHOULD BR  
9230  
9231 015310 104005 E0210: ERRORS ;BHI FAILED  
9232 015312 015304 R0210 ;ERROR LOOP RETURN  
9233  
9234 015314 000004 00210: SCOPE ;CALL SCOPE LOOP UTILITY  
9235  
9236
```

```
9237 ; *****  
9238 ; .SBTTL T0211 BHI TEST WITH Z,C = 01  
9239 ; *****  
9240 ;MICROPROGRAMMING / LOGIC INFORMATION  
9241 ;ROM SEQ: [110,347,016] FC 1,7  
9242 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
9243 ;EXEC: NO BRANCH  
9244 ;CODES: N:C = 0001  
9245 ;SYNC: B05J2 (-) T = 1.4 USEC  
9246 ;KEY SIG: K5-3 BR INSTR L / K3-3 SM=1L / K3-4 IR15 L  
9247  
9248  
9249  
9250  
9251  
9252  
9253  
9254  
9255 015316 012700 000211 T0211: MOV #0211,R0 ;LOAD R0 WITH TEST NO.  
9256 015322 013701 015332 MOV @#I0211,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9257  
9258 015326 000257 R0211: CCC ;CLEAR FLAGS  
9259 015330 000261 SEC ;MAKE Z,C = 01  
9260  
9261 015332 101001 I0211: BHI E0211 ;TEST THE BHI-IT SHOULD NOT BR  
9262 015334 000402 BR 00211 ;GO TO SCOPE EXIT  
9263  
9264 015336 104005 E0211: ERRORS ;BHI FAILED  
9265 015340 015326 R0211 ;ERROR LOOP RETURN  
9266  
9267 015342 000004 00211: SCOPE ;CALL SCOPE LOOP UTILITY  
9268  
9269
```

```
9270 ; *****  
9271 ; .SBTTL T0212 BHI TEST WITH Z,C = 10  
9272 ; *****  
9273  
9274 ;MICROPROGRAMMING / LOGIC INFORMATION  
9275  
9276 ;ROM SEQ: [110,347,016] FC 1,7  
9277  
9278 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
9279  
9280 ;EXEC: NO BRANCH  
9281  
9282 ;CODES: N:C = 0100  
9283  
9284 ;SYNC: B05J2 (-) T = 1.4 USEC  
9285  
9286 ;KEY SIG: K5-3 BR INSTR L / K3-3 SM=1L / K3-4 IR15 L  
9287  
9288 015344 012700 000212 T0212: MOV #0212,R0 ;LOAD R0 WITH TEST NO.  
9289 015350 013701 015360 MOV @#I0212,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9290  
9291 015354 000257 R0212: CCC ;CLEAR FLAGS  
9292 015356 000264 SEZ ;MAKE Z,C = 10  
9293  
9294 015360 101001 I0212: BHI E0212 ;TEST THE BHI-IT SHOULD NOT BR  
9295 015362 000402 BR 00212 ;GO TO SCOPE EXIT  
9296  
9297 015364 104005 E0212: ERRORS ;BHI FAILED  
9298 015366 015354 R0212 ;ERROR LOOP RETURN  
9299  
9300 015370 000004 00212: SCOPE ;CALL SCOPE LOOP UTILITY  
9301
```

```
9302 ; *****  
9303 ; .SBTTL T0213 BHI TEST WITH Z,C = 11  
9304 ; *****  
9305  
9306 ;MICROPROGRAMMING / LOGIC INFORMATION  
9307  
9308 ;ROM SEQ: [110,347,016] FC 1,7  
9309  
9310 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
9311  
9312 ;EXEC: NO BRANCH  
9313  
9314 ;CODES: N:C = 0101  
9315  
9316 ;SYNC: B05J2 (-) T = 1.4 USEC  
9317  
9318 ;KEY SIG: K5-3 BR INSTR L / K3-3 SM=1L / K3-4 IR15 L  
9319  
9320 015372 012700 000213 T0213: MOV #0213,R0 ;LOAD R0 WITH TEST NO.  
9321 015376 013701 015406 MOV @#I0213,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9322  
9323 015402 000257 R0213: CCC ;CLEAR FLAGS  
9324 015404 000265 265 ;MAKE Z,C = 11  
9325  
9326 015406 101001 I0213: BHI E0213 ;TEST THE BHI-IT SHOULDN'T BR  
9327 015410 000402 BR 00213 ;GO TO SCOPE EXIT  
9328  
9329  
9330 015412 104005 E0213: ERRORS ;BHI FAILED  
9331 015414 015402 R0213 ;ERROR LOOP RETURN  
9332  
9333 015416 000004 00213: SCOPE ;CALL SCOPE LOOP UTILITY  
9334  
9335
```

```
9336 ; *****  
9337 ; .SBTTL T0214 BLOS TEST WITH Z,C = 00  
9338 ; *****  
9339  
9340 ;MICROPROGRAMMING / LOGIC INFORMATION  
9341  
9342 ;ROM SEQ: [110,347,016] FC 1,7  
9343  
9344 ;ACT BUTS: 37[004]100,110 / 16[110]016,016  
9345  
9346 ;EXEC: NO BRANCH  
9347  
9348 ;CODES: N:C = 0000  
9349  
9350 ;SYNC: B05J2 (-) T = 1.4 USEC  
9351  
9352 ;KEY SIG: K5-3 BR INSTR H / K3-3 SM=3L / K3-4 IR15 L  
9353  
9354 015420 012700 000214 T0214: MOV #0214,R0 ;LOAD R0 WITH TEST NO.  
9355 015424 013701 015432 MOV @#I0214,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9356  
9357 015430 000257 R0214: CCC ;MAKE Z,C = 00  
9358  
9359 015432 101401 I0214: BLOS E0214 ;TEST THE BLOS-IT SHOULDN'T BR  
9360 015434 000402 BR 00214 ;GO TO SCOPE EXIT  
9361  
9362 015436 104005 E0214: ERRORS ;BLOS FAILED  
9363 015440 015430 R0214 ;ERROR LOOP RETURN  
9364  
9365 015442 000004 00214: SCOPE ;CALL SCOPE LOOP UTILITY  
9366  
9367
```

```
9368 ; *****  
9369 ; .SBTTL T0215 BLOS TEST WITH Z,C = 01  
9370 ; *****  
9371  
9372 ;MICROPROGRAMMING / LOGIC INFORMATION  
9373  
9374 ;ROM SEQ: [111,340,341,016] FC 1,7  
9375  
9376 ;ACT BUTS: 37[004]100,111 / 16[340]016,016  
9377  
9378 ;EXEC: [016] D = #00215  
9379  
9380 ;CODES: N:C = 0001  
9381  
9382 ;SYNC: B05J2 (-) T = 1.8 USEC  
9383  
9384 ;KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR L / K3-3 SM=1L / K3-4 IR15 L  
9385 ; K5-2 PS(C)(1)L  
9386  
9387 015444 012700 000215 T0215: MOV #0215,R0 ;LOAD R0 WITH TEST NO.  
9388 015450 013701 015460 MOV @#I0215,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9389  
9390 015454 000257 R0215: CCC ;CLEAR FLAGS  
9391 015456 000261 SEC ;MAKE Z,C = 01  
9392  
9393 015460 101402 I0215: BLOS 00215 ;TEST THE BLOS-IT SHOULD BR  
9394  
9395 015462 104005 E0215: ERRORS ;BLOS FAILED  
9396 015464 015454 R0215 ;ERROR LOOP RETURN  
9397  
9398 015466 000004 00215: SCOPE ;CALL SCOPE LOOP UTILITY  
9399  
9400
```

9%01
9%02
9%03
9%04
9%05
9%06
9%07
9%08
9%09
9%10
9%11
9%12
9%13
9%14
9%15
9%16
9%17
9%18
9%19
9%20
9%21
9%22
9%23
9%24
9%25
9%26
9%27
9%28
9%29
9%30
9%31
9%32

: *****
: .SBTTL T0216 BLOS TEST WITH Z,C = 10
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [111,340,341,016] FC 1,7
:ACT BUTS: 37[004]100,111 / 16[340]016,016
:EXEC: [016] D = #00216
:CODES: N:C = 0100
:SYNC: B05J2 (-) T = 1.8 USEC
:KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR L / K3-3 SM=1L / K3-4 IR15 L
: K5-2 PS(Z)(1)L

015470 012700 000216
015474 013701 015504
015500 000257
015502 000264
015504 101402
015506 104005
015510 015500
015512 000004

T0216: - MOV #0216,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0216,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0216: CCC ;CLEAR FLAGS
SEZ ;MAKE Z,C = 10
I0216: BLOS 00216 ;TEST THE BLOS-IT SHOULD BR
E0216: ERRORS ;BLOS FAILED
R0216 ;ERROR LOOP RETURN
00216: SCOPE ;CALL SCOPE LOOP UTILITY

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

: *****
.SBTTL T0217 BLOS TEST WITH Z,C = 11
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [111,340,341,016] FC 1,7
:ACT BUTS: 37[004]100,111 / 16[340]016,016
:EXEC: [016] D = #00217
:CODES: N:C = 0101
:SYNC: B05J2 (-) T = 1.8 USEC
:KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR L / K3-3 SM=1L / K3-4 IR15 L
: K5-2 PS(C)(1)L

015514 012700 000217
015520 013701 015530
015524 000257
015526 000265
015530 101402
015532 104005
015534 015524
015536 000004

T0217: MOV #0217,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0217,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0217: CCC ;CLEAR FLAGS
265 ;MAKE Z,C = 11
I0217: BLOS 00217 ;TEST THE BLOS-IT SHOULD BR
E0217: ERRORS ;BLOS FAILED
R0217 ;ERROR LOOP RETURN
00217: SCOPE ;CALL SCOPE LOOP UTILITY

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

; *****
; .SBTTL T0220 BHIS TEST WITH C = 0
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [111,340,341,016] FC 1,7

;ACT BUTS: 37[004]100,111 / 16[340]016,016

;EXEC: [016] D = #00220

;CODES: N:C = 0000

;SYNC: B05J2 (-) T = 1.8 USEC

;KEY SIG: K5-3 BR INSTR L / K5-3 FALSE BR L / K3-3 SM=3L / K3-4 IR15 L

015540 012700 000220
015544 013701 015552

T0220: MOV #0220,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0220,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

015550 000257

R0220: CCC ;CLEAR FLAGS

015552 103002

I0220: BHIS 00220 ;TEST THE BHIS-IT SHOULD BR

015554 104000
015556 015550

E0220: ERROR ;BHIS FAILED
R0220 ;ERROR LOOP RETURN

015560 000004

O0220: SCOPE ;CALL SCOPE LOOP UTILITY

9497
9498
9499
9500
9501
9502
9503
9504
9505
9506
9507
9508
9509
9510
9511
9512
9513
9514
9515
9516
9517
9518
9519
9520
9521
9522
9523
9524
9525
9526
9527
9528
9529

015562 012700 000221
015566 013701 015576
015572 000257
015574 000261
015576 103001
015600 000402
015602 104000
015604 015572
015606 000004

```
; *****  
; .SBTTL T0221 BHIS TEST WITH C = 1  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [110,347,016] FC 1,7  
;ACT BUTS: 37[004]100,110 / 16[110]016,016  
;EXEC: NO BRANCH  
;CODES: N:C = 0001  
;SYNC: B05J2 (-) T = 1.4 USEC  
;KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L / K3-4 IR15 L  
T0221: MOV #0221,R0 ;LOAD R0 WITH TEST NO.  
MOV @I0221,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0221: CCC ;CLEAR FLAGS  
SEC ;MAKE C = 1  
I0221: BHIS E0221 ;TEST THE BHIS-IT SHOULDN'T BR  
BR 00221 ;GO TO SCOPE EXIT  
E0221: ERROR ;BHIS FAILED  
R0221 ;ERROR LOOP RETURN  
00221: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

9530 ; *****
9531 ; .SBTTL T0222 BLO TEST WITH C = 0
9532 ; *****
9533
9534 ;MICROPROGRAMMING / LOGIC INFORMATION
9535
9536 ;ROM SEQ: [110,347,016] FC 1,7
9537
9538 ;ACT BUTS: 37[004]100,111 / 16[110]016,016
9539
9540 ;EXEC: NO BRANCH
9541
9542 ;CODES: N:C = 0000
9543
9544 ;SYNC: B05J2 (-) T = 1.4 USEC
9545
9546 ;KEY SIG: K5-3 BR INSTR L / K3-3 SM=3L / K3-4 IR15 L
9547
9548 015610 012700 000222 T0222: MOV #0222,R0 ;LOAD R0 WITH TEST NO.
9549 015614 013701 015622 MOV @#I0222,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
9550
9551 015620 000257 R0222: CCC ;CLEAR FLAGS
9552
9553 015622 103401 I0222: BLO E0222 ;TEST THE BLO-IT SHOULDN'T BR
9554 015624 000402 BR 00222 ;GO TO SCOPE EXIT
9555
9556 015626 104005 E0222: ERRORS ;BLO FAILED
9557 015630 015620 R0222 ;ERROR LOOP RETURN
9558
9559 015632 000004 00222: SCOPE ;CALL SCOPE LOOP UTILITY
9560

```

```
9561 ; *****  
9562 ; .SBTTL T0223 BLO TEST WITH C = 1  
9563 ; *****  
9564  
9565 ;MICROPROGRAMMING / LOGIC INFORMATION  
9566  
9567 ;ROM SEQ: [111,340,341,016] FC 1,7  
9568  
9569 ;ACT BUTS: 37[004]100,111 / 16[340]016,016  
9570  
9571 ;EXEC: [016] D = #00223  
9572  
9573 ;CODES: N:C = 0001  
9574  
9575 ;SYNC: B05J2 (-) T = 1.8 USEC  
9576  
9577 ;KEY SIG: K5-3 BR INSTR L / K5-3 TRUE BR L / K5-2 PS(C)(1)H / K3-3 SM=3L  
9578 ; K3-4 IR15 L  
9579  
9580 015634 012700 000223 T0223: MOV #0223,R0 ;LOAD R0 WITH TEST NO.  
9581 015640 013701 015650 MOV @#I0223,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9582  
9583 015644 000257 R0223: CCC ;CLEAR FLAGS  
9584 015646 000261 SEC ;MAKE C = 1  
9585  
9586 015650 103402 I0223: BLO 00223 ;TEST THE BLO-IT SHOULD BR  
9587  
9588 015652 104005 E0223: ERRORS ;BLO FAILED  
9589 015654 015644 R0223 ;ERROR LOOP RETURN  
9590  
9591 015656 000004 00223: SCOPE ;CALL SCOPE LOOP UTILITY  
9592  
9593
```

```
9594 ; *****
9595 ; .SATTTL T0224 SXT MODE 0 TEST WITH N = 0 AND C = 1
9596 ; *****
9597
9598 ;MICROPROGRAMMING / LOGIC INFORMATION
9599
9600 ;ROM SEQ: [132,360,001] FC 1,8
9601
9602 ;ACT BUTS: 37[004]100,132 / 27[132]000,001
9603
9604 ;EXEC: [132]ALUC=HLLHH :[360] D = 000000
9605
9606 ;CODES: [360] SPS=3 / N:C = 0101
9607
9608 ;SYNC: B05J2 (-) T = 1 USEC
9609
9610 ;KEY SIG: K3-3 DM=0L / K3-4 ONLAP INSTR H / K3-5 SXT L / K3-8 ALUM H
9611
9612 015660 012700 000224 T0224: MOV #0224,R0 ;LOAD R0 WITH TEST NO.
9613 015664 013701 015706 MOV @#10224,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
9614 015670 005004 CLR R4 ;RESULT S / B = 0
9615 015672 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
9616 015676 012703 177777 R0224: MOV #-1,R3 ;INITIAL DEST. OP = 177777
9617 015702 000257 CCC ;CLEAR CODES
9618 015704 000263 263 ;N:C = 0011
9619
9620 015706 006703 I0224: SXT R3 ;TEST THE SXT
9621
9622 015710 100403 BMI E10224
9623 015712 001002 BNE E10224 ;DID SXT MAKE N:C = 0101?
9624 015714 102401 BVS E10224
9625 015716 103402 BCS A0224
9626
9627 015720 104000 E10224: ERROR ;SXT FAILED TO ALTER CODES PROPERLY
9628 015722 015676 R0224 ;ERROR LOOP RETURN
9629
9630 015724 005703 A0224: TST R3 ;DID RESULT = 0?
9631 015726 001402 BEQ 00224 ;BR IF IT DID
9632
9633 015730 104000 E20224: ERROR ;SXT DELIVERED WRONG RESULT TO R3
9634 015732 015676 R0224 ;ERROR LOOP RETURN
9635
9636 015734 000004 00224: SCOPE ;CALL SCOPE LOOP UTILITY
9637
9638
```

9639
9640
9641
9642
9643
9644
9645
9646
9647
9648
9649
9650
9651
9652
9653
9654
9655
9656
9657
9658
9659
9660
9661
9662
9663
9664
9665
9666
9667
9668
9669
9670
9671
9672
9673

015736 012700 000225
015742 013701 015762

015746 005004
015750 012702 177703
015754 012703 177777
015760 000257

015762 006703
015764 103002

015766 104000
015770 015754

015772 000004

```
; *****  
; .SBTTL T0225 SXT MODE 0 TEST WITH N = 0 AND C = 0  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [132,360,001] FC 1,8  
;ACT BUTS: 37[004]100,132 / 27[132]000,001  
;EXEC: [132]ALUC=HLLHH :[360] D = 000000  
;CODES: [360] SPS=3 / N:C = 0100  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-3 DM=0L / K3-4 OVL INSTR H / K3-5 SXT L / K3-8 ALUM H  
T0225: MOV #0225,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0225,R1 ;LOAD RT WITH TEST INSTRUCTION WORD  
  
CLR R4 ;RESULT S / B = 0  
MOV #177703,R2 ;DEST ADDR = R3  
R0225: MOV #-1,R3 ;INITIAL DEST OP = 177777  
CCC ;CLEAR N:C  
  
I0225: SXT R3 ;TEST THE SXT  
BCC 00225 ;BR IF 'C' STILL CLEAR  
  
E0225: ERROR ;SXT AFFECTED 'C' BIT  
R0225 ;ERROR LOOP RETURN  
  
00225: SCOPE ;CALL SCOPE LOOP UTILITY
```

9674
9675
9676
9677
9678
9679
9680
9681
9682
9683
9684
9685
9686
9687
9688
9689
9690
9691
9692
9693
9694
9695
9696
9697
9698
9699
9700
9701
9702
9703
9704
9705
9706
9707
9708
9709
9710
9711
9712
9713
9714
9715
9716
9717
9718
9719

015774 012700 000226
016000 013701 016020

016004 012704 177777
016010 012703 177703
016014 005003
016016 000277

016020 006703

016022 100003
016024 001402
016026 102401
016030 103402

016032 104000
016034 016014

016036 010305
016040 005105
016042 001402

016044 104000
016046 016014

016050 000004

; *****
; .SBTTL T0226 SXT MODE 0 TEST WITH N = 1 AND C = 1
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [132,360,001] FC 1,8
;ACT BUTS: 37[004]100,132 / 27[132]000,001
;EXEC: [132]ALUC=LLLHH :[360] D = 177777
;CODES: [360] SPS=3 / N:C = 1001
;SYNC: B05J2 (-) T = 1 USEC
;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-5 SXT L / K5-2 PS(N)(1)H

T0226: MOV #0226,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0226,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

R0226: MOV #-1,R4 ;RESULT S / B = 177777
MOV #177703,R3 ;DEST ADDR = 177703
CLR R3 ;INITIAL DEST OP = 0
SCC ;MAKE N:C = 1111

I0226: SXT R3 ;TEST THE SXT

BPL E10226
BEQ E10226 ;N:C = 1001?
BVS E10226
BCS A0226

E10226: ERROR ;SXT FAILED TO ALTER CODES PROPERLY
R0226 ;ERROR LOOP RETURN

A0226: MOV R3,R5 ;GET RESULT
COM R5 ;COMPLEMENT IT-SHOULD GO TO 0
BEQ 00226 ;BR IF RESULT OF SXT = 1

E20226: ERROR ;SXT DELIVERED WRONG RESULT.
R0226 ;ERROR LOOP RETURN

00226: SCOPE ;CALL SCOPE LOOP UTILITY

9720
9721
9722
9723
9724
9725
9726
9727
9728
9729
9730
9731
9732
9733
9734
9735
9736
9737
9738
9739
9740
9741
9742
9743
9744
9745
9746
9747
9748
9749
9750
9751
9752
9753
9754
9755

016052 012700 000227
016056 013701 016100

016062 012704 177777
016066 012702 177703

016072 005003
016074 000257
016076 000276

016100 006703
016102 103002

016104 104000
016106 016072

016110 000004

```
; *****  
; .SBTTL T0227 SXT MODE 0 TEST WITH N = 1 AND C = 0  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [132,360,001] FC 1,8  
;ACT BUTS: 37[004]100,132 / 27[132]000,001  
;EXEC: [132]ALUC=LLLHH :[360] D = 177777  
;CODES: [360] SPS=3 / N:C = 1000  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-3 DM = 0L / K3-4 OVLAP INSTR H / K3-5 SXT L / K5-2 PS(N)(1) H  
T0227: MOV #0227,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0227,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
  
MOV #-1,R4 ;RESULT S / B = 177777  
MOV #177703,R2 ;DEST ADDR = 177703  
R0227: CLR R3 ;INITIAL DEST OP = 0  
CCC ;CLEAR FLAGS  
276 ;MAKE N:C = 1110  
  
I0227: SXT R3 ;TEST THE SXT  
BCC 00227 ;BR IF 'C' UNAFFECTED  
  
E0227: ERROR ;SXT SET 'C' BIT  
R0227 ;ERROR LOOP RETURN  
  
00227: SCOPE ;CALL SCOPE LOOP UTILITY
```

9756
9757
9758
9759
9760
9761
9762
9763
9764
9765
9766
9767
9768
9769
9770
9771
9772
9773
9774
9775
9776
9777
9778
9779
9780
9781
9782
9783
9784
9785
9786
9787
9788
9789
9790
9791
9792
9793
9794
9795
9796
9797
9798
9799
9800
9801
9802
9803
9804
9805
9806
9807
9808
9809
9810
9811

016112 012700 000230
016116 013701 016140
016122 012702 067560
016126 005004
016130 012712 177777
016134 000257
016136 000263
016140 006712
016142 100403
016144 001002
016146 102401
016150 103402
016152 104000
016154 016130
016156 005712
016160 001402
016162 104000
016164 016130
016166 012702 067560
016172 013701 016206
016176 012712 177777
016202 000257
016204 000263
016206 006722
016210 100403
016212 001002
016214 102401

```
; *****  
.SBTTL T0230 SXT MODE 1 AND 2 TEST WITH N = 0 AND C = 1  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ:(DM1) [161,266,267,234,367,375,016] FC 1,3,8  
;      :(DM2) [162,260,267,234,367,375,016] FC 1,3,8  
;ACT BUTS:(DM1) 37[004]100,161 / 33[266]220,234 / 16[367]016,016  
;      :(DM2) 37[004]100,162 / 33[260]220,234 / 16[367]016,016  
;EXEC: [234]ALUC=HLLHM :[367] D = 000000  
;CODES: [367] SPS=3 / N:C = 0101  
;SYNC: B05J2 (-) T = 2.5 USEC  
;KEY SIG: K3-3 DM=1L / K3-5 SXT L / K3-8 ALUM H  
T0230: MOV #0230,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I10230,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
R10230: 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  
I10230: SXT (R2) ;TEST THE SXT - DM1  
BMI E10230  
BNE E10230 ;N:C = 0101  
BVS E10230  
BCS A0230  
E10230: ERROR ;SXT FAILED TO ALTER CODES PROPERLY  
R10230 ;ERROR LOOP RETURN  
A0230: TST (R2) ;DID RESULT = 0?  
BEQ R20230 ;BR IF YES  
E20230: ERROR ;SXT SHOULD HAVE ZEROED [DEST]  
R10230 ;ERROR LOOP RETURN  
R20230: MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV @#I20230,R1 ;LOAD R1 WITH TEST INSTR WORD  
MOV #-1,(R2) ;INITIAL [DEST] = 177777  
CCC ;CLEAR CODES  
263 ;MAKE N:C = 0011  
I20230: SXT (R2)+ ;TEST SXT - DM2  
BMI E30230  
BNE E30230 ;N:C = 0101 ?  
BVS E30230
```

9812	016216	103402		BCS	B0230	
9813						
9814	016220	104000		E30230:	ERROR	:SXT FAILED TO ALTER CODES PROPERLY
9815	016222	016166			R20230	:ERROR LOOP RETURN ADDRESS
9816						
9817	016224	005737	067560	B0230:	TST @#MBUF0	:DID RESULT GET ZEROED ?
9818	016230	001402			BEQ C0230	:BR IF YES
9819						
9820	016232	104000		E40230:	ERROR	:SXT FAILED TO ZERO [DEST]
9821	016234	016166			R20230	:ERROR LOOP RETURN ADDRESS
9822						
9823	016236	020227	067562	C0230:	CMP R2,#MBUF0+2	:WAS IT REALLY MODE 2 ?
9824	016242	001402			BEQ 00230	:BR IF YES
9825						
9826	016244	104000		E50230:	ERROR	:SXT FAILED TO AUTO INCREMENT
9827	016246	016166			R20230	:ERROR LOOP RETURN ADDRESS
9828						
9829	016250	000004		00230:	SCOPE	:CALL SCOPE LOOP UTILITY
9830						
9831						

```
9832 ; *****  
9833 ; .SBTTL T0231 SXT MODE 1 TEST WITH N = 0 AND C = 0  
9834 ; *****  
9835  
9836 ;MICROPROGRAMMING / LOGIC INFORMATION  
9837  
9838 ;ROM SEQ: [161,266,267,234,367,375,016] FC 1,3,8  
9839  
9840 ;ACT BUTS: 37[004]100,161 / 33[266]220,234 / 16[367]016,016  
9841  
9842 ;EXEC: [234]ALUC=HLLHH :[367] D = 000000  
9843  
9844 ;CODES: [367] SPS=3 / N:C = 0100  
9845  
9846 ;SYNC: B05J2 (-) T = 2.5 USEC  
9847  
9848 ;KEY SIG: K3-3 DM=1L / K3-5 SXT L / K3-8 ALUM H  
9849  
9850 016252 012700 000231 T0231: MOV #0231,R0 ;LOAD R0 WITH TEST NO.  
9851 016256 013701 016276 MOV @#I0231,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
9852  
9853 016262 005004 CLR R4 ;RESULT S / B = 0  
9854 016264 012702 067560 MOV #MBUF0,R2 ;R2 POINTS TO DEST OP  
9855 016270 012712 177777 R0231: MOV #-1,(R2) ;INITIAL [DEST] = 177777  
9856 016274 000257 CCC ;CLEAR 'C' BIT  
9857  
9858 016276 006712 I0231: SXT (R2) ;TEST THE SXT  
9859 016300 103002 BCC 00231 ;BR IF 'C' UNDISTURBED  
9860  
9861 016302 104000 E0231: ERROR ;SXT SET THE 'C' BIT  
9862 016304 016270 R0231 ;ERROR LOOP RETURN  
9863  
9864 016306 000004 00231: SCOPE ;CALL SCOPE LOOP UTILITY  
9865  
9866
```

9867
9868
9869
9870
9871
9872
9873
9874
9875
9876
9877
9878
9879
9880
9881
9882
9883
9884
9885
9886
9887
9888
9889
9890
9891
9892
9893
9894
9895
9896
9897
9898
9899
9900
9901
9902
9903
9904
9905
9906
9907
9908
9909
9910
9911

016310 012700 000232
016314 013701 016334

016320 012704 177777
016324 012702 067560
016330 005012
016332 000277

016334 006712

016336 100003
016340 001402
016342 102401
016344 103402

016346 104000
016350 016330

016352 021204
016354 001402

016356 104000
016360 016330

016362 000004

```
; *****  
; .SBTTL T0232 SXT MODE 1 TEST WITH N = 1 AND C = 1  
; *****  
  
:MICROPROGRAMMING / LOGIC INFORMATION  
  
:ROM SEQ: [161,266,267,234,367,375,016] FC 1,3,8  
:ACT BUTS: 37[004]100,161 / 33[266]220,234 / 16[367]016,016  
:EXEC: [234]ALUC=LLLHM :[367] D = 177777  
:CODES: [367] SPS=3 / N:C =1001  
:SYNC: B05J2 (-) T = 2.5 USEC  
:KEY SIG: K3-3 DM=1L / K3-5 SXT L / K5-2 PS(N)(1)H  
  
T0232: MOV #0232,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0232,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
  
MOV #-1,R4 ;RESULT S / B = 177777  
MOV #MBUF0,R2 ;R2 POINTS TO DEST OP  
R0232: CLR (R2) ;INITIAL [DEST] = 0  
SCC ;MAKE N:C = 1111  
  
I0232: SXT (R2) ;TEST THE SXT  
  
BPL E10232  
BEQ E10232 ;N:C = 1001?  
BVS E10232  
BCS A0232  
  
E10232: ERROR ;SXT FAILED TO ALTER CODES PROPERLY  
R0232 ;ERROR LOOP RETURN  
  
A0232: CMP (R2),R4 ;RESULT = 177777?  
BEQ 00232 ;BR IF YES  
  
E20232: ERROR ;SXT DELIVERED WRONG RESULT  
R0232 ;ERROR LOOP RETURN  
  
00232: SCOPE ;CALL SCOPE LOOP UTILITY
```

9912
9913
9914
9915
9916
9917
9918
9919
9920
9921
9922
9923
9924
9925
9926
9927
9928
9929
9930 016364 012700 000233
9931 016370 013701 016412
9932
9933 016374 012704 177777
9934 016400 012702 067560
9935 016404 005012
9936 016406 000257
9937 016410 000276
9938
9939 016412 006712
9940 016414 103002
9941
9942 016416 104000
9943 016420 016404
9944
9945 016422 000004
9946
9947

; *****
; .SBTTL T0233 SXT MODE 1 TEST WITH N = 1 AND C = 0
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,234,367,375,016] FC 1,3,8
;ACT BUTS: 37[004]100,161 / 33[266]220,234 / 16[367]016,016
;EXEC: [234]ALUC=LLLHM :[367] D = 177777
;CODES: [367] SPS=3 / N:C = 1000
;SYNC: B05J2 (-) T = 2.5 USEC
;KEY SIG: K3-3 DM=1L / K3-5 SXT L / K5-2 PS(N)(1)H

T0233: MOV #0233,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0233,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0233: MOV #-1,R4 ;RESULT S / B = 177777
MOV #MBUF0,R2 ;R2 POINTS TO DEST OP
CLR (R2) ;INITIAL [DEST] = 0
CCC ;CLEAR FLAGS
276 ;MAKE N:C = 1110
I0233: SXT (R2) ;TEST THE SXT
BCC 00233 ;BR IF 'C' UNAFFECTED
E0233: ERROR ;SXT SET THE 'C' BIT
R0233 ;ERROR LOOP RETURN
00233: SCOPE ;CALL SCOPE LOOP UTILITY

9948
9949
9950
9951
9952
9953
9954
9955
9956
9957
9958
9959
9960
9961
9962
9963
9964
9965
9966
9967
9968
9969
9970
9971
9972
9973
9974
9975
9976
9977
9978
9979
9980
9981
9982
9983
9984
9985
9986
9987
9988
9989
9990
9991
9992
9993
9994
9995
9996
9997

016424 012700 000234
016430 013701 016466
016434 032737 000020 066642
016442 001401
016444 000000
016446 012704 177400
016452 012702 177703
016456 012703 000377
016462 000257
016464 000273
016466 000303
016470 100403
016472 001002
016474 102401
016476 103002
016500 104000
016502 016456
016504 020403
016506 001402
016510 104000
016512 016456
016514 000004

: *****
.SBTTL T0234 SWAB MODE 0 TEST WITH POS. RESULT
: *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [134,135,360,001] FC 1,7,8
;ACT BUTS: 37[004]100,134 / 27[135]000,001
;EXEC: [135]ALUC=HHLHL :[360] D = 177400
;CODES: [360] SPS=3 / N:C = 0100
;SYNC: B05J2 (-) T = 1 USEC
;KEY SIG: K2-5 SBML1 (1) H / K2-5 SMBL0 (1) L / K2-5 SBMH1 (1) H
: / K2-5 SBMH0 (1) L / K3-3 DM=0L / K3-5 SWAB H / K3-4 OVLAP INS

T0234: MOV #0234,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0234,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

R0234: BIT #20,@#BPTLOC ;BREAKPOINT HALT SET ??
BEQ .+4 ;BR IF NOT
HALT ;BREAK - DEPRESS CONTINUE TO RESTART
MOV #177400,R4 ;RESULT S / B = 177400
MOV #177703,R2 ;DEST ADDR = 177703
MOV #377,R3 ;INITIAL DEST OP = 377
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011

I0234: SWAB R3 ;TEST THE SWAB
BMI E10234
BNE E10234 ;N:C = 0100
BVS E10234
BCC A0234

E10234: ERROR ;SWAB FAILED TO ALTER CODES PROPERLY
R0234 ;ERROR LOOP RETURN

A0234: CMP R4,R3 ;CORRECT RESULT?
BEQ 00234 ;BR IF YES

E20234: ERROR ;SWAB DELIVERED WRONG RESULT
R0234 ;ERROR LOOP RETURN

00234: SCOPE ;CALL SCOPE LOOP UTILITY

```
9998 ; *****  
9999 ; .SBTTL T0235 SWAB MODE 0 TEST WITH NEG. RESULT  
10000 ; *****  
10001 ;  
10002 ;MICROPROGRAMMING / LOGIC INFORMATION  
10003 ;  
10004 ;ROM SEQ: [134,135,360,001] FC 1,7,8  
10005 ;  
10006 ;ACT BUTS: 37[004]100,134 / 27[135]000,001  
10007 ;  
10008 ;EXEC: [135]ALUC=HMLHL :[360] D = 000377  
10009 ;  
10010 ;CODES: [360] SPS=3 / N:C = 1000  
10011 ;  
10012 ;SYNC: B05J2 (-) T = 1 USEC  
10013 ;  
10014 ;KEY SIG: K2-5 SBML1 (1) H / K2-5 SBML0 (1) L / K2-5 SBMH1 (1) H  
10015 ; / K2-5 SBMH0 (1) L / K3-3 DM=0L / K3-5 SWAB H / K3-4 0  
10016 ;  
10017 016516 012700 000235 T0235: MOV #0235,R0 ;LOAD R0 WITH TEST NO.  
10018 016522 013701 016546 MOV @#I0235,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
10019 ;  
10020 016526 012704 000377 MOV #377,R4 ;RESULT S / B = 377  
10021 016532 012702 177703 MOV #177703,R2 ;DEST ADDR = 177703  
10022 016536 012703 177400 R0235: MOV #177400,R3 ;INITIAL DEST OP = 177400  
10023 016542 000257 CCC ;CLEAR FLAGS  
10024 016544 000267 267 ;MAKE N:C = 0111  
10025 ;  
10026 016546 000303 I0235: SWAB R3 ;TEST THE SWAB  
10027 ;  
10028 016550 100003 BPL E10235  
10029 016552 001402 BEQ E10235 ;DID SWAB MAKE N:C = 1000  
10030 016554 102401 BVS E10235  
10031 016556 103002 BCC A0235  
10032 ;  
10033 016560 104000 E10235: ERROR ;SWAB FAILED TO ALTER CODES PROPERLY  
10034 016562 016536 R0235 ;ERROR LOOP RETURN  
10035 ;  
10036 016564 020403 A0235: CMP R4,R3 ;DID SWAB DELIVER CORRECT RESULT?  
10037 016566 001402 BEQ 00235 ;BR IF OK  
10038 ;  
10039 016570 104000 E20235: ERROR ;SWAB DELIVERED WRONG RESULT  
10040 016572 016536 R0235 ;ERROR LOOP RETURN  
10041 ;  
10042 016574 000004 00235: SCOPE ;CALL SCOPE LOOP UTILITY  
10043 ;  
10044 ;
```

10045
10046
10047
10048
10049
10050
10051
10052
10053
10054
10055
10056
10057
10058
10059
10060
10061
10062
10063
10064
10065
10066
10067
10068
10069
10070
10071
10072
10073
10074
10075
10076
10077
10078
10079
10080
10081
10082
10083
10084
10085
10086
10087
10088
10089
10090
10091
10092
10093
10094
10095
10096
10097
10098
10099
10100

016576 012700 000236
016602 013701 016626
016606 012704 177400
016612 012702 067560
016616 012712 000377
016622 000257
016624 000273
016626 000312
016630 100403
016632 001002
016634 102401
016636 103002
016640 104000
016642 016616
016644 020412
016646 001402
016650 104000
016652 016616
016654 013701 016674
016660 012702 067560
016664 012712 000377
016670 000257
016672 000273
016674 000322
016676 100403
016700 001002

; *****
; .SBTTL T0236 SWAB MODE 1 AND 2 TEST WITH POS. RESULT
; *****
;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ:(DM1) [161,266,267,236,367,375,016] FC 1,3,9,8
; : (DM2) [162,260,267,236,367,375,016] FC 1,3,9,8
;ACT BUTS:(DM1) 37[004]100,161 / 33[266]220,236 / 16[367]016,016
; : (DM2) 37[004]100,162 / 33[260]220,236 / 16[367]016,016
;EXEC: [236]ALUC=HHLHL :[367] D = 177400
;CODES: [367] SPS=3 / N:C = 0100
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K2-5 SBML1 (1) H / K2-5 SBML0 (1) L / K2-5 SBMH1 (1) H / K2-
; K3-3 DM=1L / K3-5 SWAB H

T0236: MOV #0236,R0 ;LOAD R0 WITH TEST NO.
MOV @#I10236,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177400,R4 ;RESULT S / B = 177400
MOV #MBUF0,R2 ;R2 POINTS TO DEST OP
R10236: MOV #377,(R2) ;SET UP DEST OP = 377
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
I10236: SWAB (R2) ;TEST THE SWAB - DM1
BMI E10236
BNE E10236 ;N:C = 0100
BVS E10236
BCC A0236
E10236: ERROR ;SWAB FAILED TO ALTER CODES PROPERLY
R10236 ;ERROR LOOP RETURN
A0236: CMP R4,(R2) ;CORRECT RESULT?
BEQ R20236 ;BR IF OK
E20236: ERROR ;SWAB DELIVERED WRONG RESULT
R10236 ;ERROR LOOP RETURN
R20236: MOV @#I20236,R1 ;LOAD R1 WITH TEST INSTR. WORD
MOV #MBUF0,R2 ;R2 POINTS TO DEST OP
MOV #377,(R2) ;[DEST] = 000377
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011
I20236: SWAB (R2)+ ;TEST THE SWAB - DM2
BMI E30236
BNE E30236 ;N:C = 0100

10101	016702	102401		BVS	E30236	
10102	016704	103002		BCC	B0236	
10103						
10104	016706	104000		E30236:	ERROR	:SWAB FAILED TO SET CODES PROPERLY
10105	016710	016654			R20236	:ERROR LOOP RETURN ADDRESS
10106						
10107	016712	020437	067560	B0236:	CMP	:CORRECT RESULT ?
10108	016716	001402			R4,@#MBUF0	:BR IF YES
10109					C0236	
10110	016720	104000		E40236:	ERROR	:SWAB DELIVERED THE WRONG RESULT
10111	016722	016654			R20236	:ERROR LOOP RETURN ADDRESS
10112						
10113	016724	020227	067562	C0236:	CMP	:DID AUTO INCREMENT OCCUR ?
10114	016730	001402			R2,@#MBUF0+2	:BR IF YES
10115					C0236	
10116	016732	104000		E50236:	ERROR	:SWAB FAILED TO AUTO INC REG.
10117	016734	016654			R20236	:ERROR LOOP RETURN ADDRESS
10118						
10119	016736	000004		00236:	SCOPE	:CALL SCOPE LOOP UTILITY
10120						
10121						

10122
10123
10124
10125
10126
10127
10128
10129
10130
10131
10132
10133
10134
10135
10136
10137
10138
10139
10140
10141
10142
10143
10144
10145
10146
10147
10148
10149
10150
10151
10152
10153
10154
10155
10156
10157
10158
10159
10160
10161
10162
10163
10164
10165
10166
10167
10168

016740 . 012700 000237
016744 013701 016770
016750 012704 000377
016754 012702 067560
016760 012712 177400
016764 000257
016766 000267
016770 000312
016772 100003
016774 001402
016776 102401
017000 103002
017002 104000
017004 016760
017006 020412
017010 001402
017012 104000
017014 016760
017016 000004

```
; *****  
; .SBTTL T0237 SWAB MODE 1 TEST WITH NEG. RESULT  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,236,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,134 / 33[266]220,236 / 16[367]016,016  
;EXEC: [236]ALUC=HHLHL :[367] D = 000377  
;CODES: [367] SPS=3 / N:C = 1000  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: K2-5 SBML1 (1) H / K2-5 SBML0 (1) L / K2-5 SBMH2 (1) H / K2-5 SBMH  
; K3-3 DM=1L / K3-5 SWAB H  
T0237: MOV #0237,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0237,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #377,R4 ;RESULT S / B = 377  
MOV #MBUF0,R2 ;R2 POINTS TO DEST OP  
R0237: MOV #177400,(R2) ;SET UP DEST. OP = 177400  
CCC ;CLEAR FLAGS  
267 ;MAKE N:C = 0111  
I0237: SWAB (R2) ;TEST THE SWAB  
BPL E10237  
BEQ E10237 ;N:C = 1000?  
BVS E10237  
BCC A0237  
E10237: ERROR ;SWAB FAILED TO ALTER CODES PROPERLY  
R0237 ;ERROR LOOP RETURN  
A0237: CMP R4,(R2) ;CORRECT RESULT?  
BEQ 00237 ;BR IF YES  
E20237: ERROR ;SWAB DELIVERED WRONG RESULT  
R0237 ;ERROR LOOP RETURN  
00237: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
10169 ; *****
10170 ; .SBTTL T0240 NEG MODE 0 TEST : [DEST] = 0
10171 ; *****
10172
10173 ;MICROPROGRAMMING / LOGIC INFORMATION
10174
10175 ;ROM SEQ: [105,372,360,001] FC 1,7,8
10176
10177 ;ACT BUTS: 37[004]100,105 / 31[105]360,360 / 27[372]000,001
10178
10179 ;EXEC: [372]ALUC=LLHHL :[360] D = 000000
10180
10181 ;CODES: [360] SPS=3 / N:C = 0100
10182
10183 ;SYNC: B05J2 (-) T = 1 USEC
10184
10185 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=0L / K3-4 NEG L / K3-4 OVLAP INSTR H
10186
10187 017020 012700 000240 T0240: MOV #0240,R0 ;LOAD R0 WITH TEST NO.
10188 017024 013701 017044 MOV @#I0240,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
10189
10190 017030 012702 177703 MOV #177703,R2 ;DEST ADDR = 177703
10191 017034 005004 CLR R4 ;RESULT S / B = 0
10192 017036 005003 R0240: CLR R3 ;INITIAL [DEST] = 0
10193 017040 000257 CCC ;CLEAR FLAGS
10194 017042 000273 273 ;MAKE N:C = 1011
10195
10196 017044 005403 I0240: NEG R3 ;TEST THE NEG
10197
10198 017046 100403 BMI E10240
10199 017050 001002 BNE E10240 ;N:C = 0100 ONLY 'Z' SET?
10200 017052 102401 BVS E10240
10201 017054 103002 BCC A0240
10202
10203 017056 104000 E10240: ERROR ;NEG FAILED TO ALTER CODES PROPERLY
10204 017060 017036 R0240 ;ERROR LOOP RETURN
10205
10206 017062 020304 A0240: CMP R3,R4 ;WAS RESULT = 0
10207 017064 001402 BEQ 00240 ;BR IF YES
10208
10209 017066 104000 E20240: ERROR ;NEG DELIVERED WRONG RESULT
10210 017070 017036 R0240 ;ERROR LOOP RETURN
10211
10212 017072 000004 00240: SCOPE ;CALL SCOPE LOOP UTILITY
10213
10214
```

```

10215 ; *****
10216 ; .SBTTL T0241 NEG MODE 0 TEST : [DEST] > 0
10217 ; *****
10218
10219 ;MICROPROGRAMMING / LOGIC INFORMATION
10220
10221 ;ROM SEQ: [105,372,360,001] FC 1,7,8
10222
10223 ;ACT BUTS: 37[004]100,105 / 31[105]360,360 / 27[372]000,001
10224
10225 ;EXEC: [372]ALUC=LLHHL :[360] D = 177776
10226
10227 ;CODES: [360] SPS=3 / N:C = 1001
10228
10229 ;SYNC: B05J2 (-) T = 1 USEC
10230
10231 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=0L / K3-4 NEG L / K3-4 OVLAP INSTR H
10232
10233 017074 012700 000241 T0241: MOV #0241,R0 ;LOAD R0 WITH TEST NO.
10234 017100 013701 017124 MOV @#I0241,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
10235
10236 017104 012702 177703 MOV #177703,R2 ;DEST ADDR = 177703
10237 017110 012704 177776 MOV #-2,R4 ;RESULT S / B = 177776
10238 017114 012703 000002 R0241: MOV #2,R3 ;INITIAL [DEST] = 2
10239 017120 000257 CCC ;CLEAR FLAGS
10240 017122 000266 266 ;MAKE N:C = 0110
10241
10242 017124 005403 I0241: NEG R3 ;TEST THE NEG
10243
10244 017126 100003 BPL E10241
10245 017130 001402 BEQ E10241 ;DID N:C = 1001?
10246 017132 102401 BVS E10241
10247 017134 103402 BCS A0241
10248
10249 017136 104000 E10241: ERROR ;NEGATE FAILED TO ALTER CODES PROPERLY
10250 017140 017114 R0241 ;ERROR LOOP RETURN
10251
10252 017142 020304 A0241: CMP R3,R4 ;CORRECT RESULT?
10253 017144 001402 BEQ 00241 ;BR IF YES
10254
10255 017146 104000 E20241: ERROR ;NEG DELIVERED WRONG RESULT
10256 017150 017114 R0241 ;ERROR LOOP RETURN
10257
10258 017152 000004 00241: SCOPE ;CALL SCOPE LOOP UTILITY
10259
10260

```

```
10261 ; *****  
10262 ; .SBTTL T0242 NEG MODE 0 TEST : [DEST] < 0  
10263 ; *****  
10264  
10265 ;MICROPROGRAMMING / LOGIC INFORMATION  
10266  
10267 ;ROM SEQ: [105,372,360,001] FC 1,7,8  
10268  
10269 ;ACT BUTS: 37[004]100,105 / 31[105]360,360 / 27[372]000,001  
10270  
10271 ;EXEC: [372]ALUC=LLHHL :[360] D = 000002  
10272  
10273 ;CODES: [360] SPS=3 / N:C = 0001  
10274  
10275 ;SYNC: B05J2 (-) T = 1 USEC  
10276  
10277 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=0L / K3-4 NEG L / K3-4 OVLAP INSTR H  
10278  
10279 017154 012700 000242 T0242: MOV #0242,R0 ;LOAD R0 WITH TEST NO.  
10280 017160 013701 017204 MOV @#I0242,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
10281  
10282 017164 012702 177703 MOV #177703,R2 ;DEST ADDR = 177703  
10283 017170 012704 000002 MOV #2,R4 ;RESULT S / B = 2  
10284 017174 012703 177776 R0242: MOV #-2,R3 ;INITIAL [DEST] = 177776  
10285 017200 000257 CCC ;CLEAR FLAGS  
10286 017202 000276 276 ;MAKE N:C = 1110  
10287  
10288 017204 005403 I0242: NEG R3 ;TEST THE NEG  
10289  
10290 017206 100403 BMI E10242  
10291 017210 001402 BEQ E10242 ;N:C = 0001?  
10292 017212 102401 BVS E10242  
10293 017214 103402 BCS A0242  
10294  
10295 017216 104000 E10242: ERROR ;NEG FAILED TO ALTER CODES PROPERLY  
10296 017220 017174 R0242 ;ERROR LOOP RETURN  
10297  
10298 017222 020304 A0242: CMP R3,R4 ;RESULT = 2?  
10299 017224 001402 BEQ 00242 ;BR IF YES  
10300  
10301 017226 104000 E20242: ERROR ;NEG DELIVERED WRONG RESULT  
10302 017230 017174 R0242 ;ERROR LOOP RETURN  
10303  
10304 017232 000004 00242: SCOPE ;CALL SCOPE LOOP UTILITY  
10305  
10306  
10307
```

10308
10309
10310
10311
10312
10313
10314
10315
10316
10317
10318
10319
10320
10321
10322
10323
10324
10325
10326
10327
10328
10329
10330
10331
10332
10333
10334
10335
10336
10337
10338
10339
10340
10341
10342
10343
10344
10345
10346
10347
10348
10349
10350
10351
10352
10353

017234 012700 000243
017240 013701 017262

017244 012702 177703
017250 012704 100000
017254 010403
017256 000257
017260 000264

017262 005403

017264 100003
017266 001402
017270 102001
017272 103402

017274 104000
017276 017254

017300 020304
017302 001402

017304 104000
017306 017254

017310 000004

: *****
.SBTTL T0243 NEG MODE 0 TEST : [DEST] = 100000 (8)
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [105,372,360,001] FC 1,7,8
:ACT BUTS: 37[004]100,105 / 31[105]360,360 / 27[372]000,001
:EXEC: [372]ALUC=LLHHL :[360] D = 100000
:CODES: [360] SPS=3 / N:C = 1011
:SYNC: B05J2 (-) T = 1 USEC
:KEY SIG: K3-8 CIN00 L / K3-3 DM=0L / K3-4 NEG L / K3-4 OVLAP INSTR H

T0243: MOV #0243,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0243,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

MOV #177703,R2 ;DEST ADDR = 177703
MOV #100000,R4 ;RESULT S / B = 100000
R0243: MOV R4,R3 ;INITIAL [DEST] = 100000
CCC ;CLEAR FLAGS
SEZ ;MAKE N:C = 01000

I0243: NEG R3 ;TEST THE NEG

BPL E10243
BEQ E10243 ;N:C = 1011?
BVC E10243
BCS A0243

E10243: ERROR ;NEG FAILED TO ALTER CODES PROPERLY
R0243 ;ERROR LOOP RETURN

A0243: CMP R3,R4 ;RESULT STILL 100000?
BEQ 00243 ;BR IF YES

E20243: ERROR ;NEG DELIVERED WRONG RESULT
R0243 ;ERROR LOOP RETURN

00243: SCOPE ;CALL SCOPE LOOP UTILITY

10354
10355
10356
10357
10358
10359
10360
10361
10362
10363
10364
10365
10366
10367
10368
10369
10370
10371
10372
10373
10374
10375
10376
10377
10378
10379
10380
10381
10382
10383
10384
10385
10386
10387
10388
10389
10390
10391
10392
10393
10394
10395
10396
10397
10398
10399

017312 012700 000244
017316 013701 017336

017322 012702 067560
017326 005004
017330 005012
017332 000257
017334 000273

017336 005412

017340 100403
017342 001002
017344 102401
017346 103002

017350 104000
017352 017330

017354 021204
017356 001402

017360 104000
017362 017330

017364 000004

```
; *****  
; .SBTTL T0244 NEG MODE 1 TEST : [DEST] = 0  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ:      [161,266,267,221,367,375,016] FC 1,3,9,8  
;ACT BUTS:     37[004]100,161 / 33[266]220,221 / 16[367]016,016  
;EXEC:         [221]ALUC=LLHHL :[367] D = 000000  
;CODES:        [367] SPS=3 / N:C = 0100  
;SYNC:         B05J2 (-) T = 2 USEC  
;KEY SIG:      K3-8 CIN00 L / K3-3 DM=1L / K3-4 NEG L  
  
T0244:  MOV    #0244,R0          ;LOAD R0 WITH TEST NO.  
        MOV    @#I0244,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
  
        MOV    #MBUF0,R2      ;R2 POINTS TO DEST OP  
R0244:  CLR    R4              ;RESULT S / B = 0  
        CLR    (R2)           ;INITIAL [DEST] = 0  
        CCC    273            ;CLEAR FLAGS  
        ;MAKE N:C = 1011  
  
I0244:  NEG    (R2)           ;TEST THE NEG  
  
        BMI    E10244  
        BNE    E10244        ;N:C = 0100?  
        BVS    E10244  
        BCC    A0244  
  
E10244:  ERROR  R0244        ;NEG FAILED TO ALTER CODES PROPERLY  
        ;ERROR LOOP RETURN  
  
A0244:  CMP    (R2),R4        ;RESULT = 0?  
        BEQ    00244         ;BR IF YES  
  
E20244:  ERROR  R0244        ;NEG DELIVERED WRONG RESULT  
        ;ERROR LOOP RETURN  
  
00244:  SCOPE                ;CALL SCOPE LOOP UTILITY
```

```
10400 ; *****
10401 ; .SBTTL T0245 NEG MODE 1 TEST : [DEST] > 0
10402 ; *****
10403
10404 ;MICROPROGRAMMING / LOGIC INFORMATION
10405
10406 ;ROM SEQ: [161,266,267,221,367,375,016] FC 1,3,9,8
10407
10408 ;ACT BUTS: 37[004]100,161 / 33[266]220,221 / 16[367]016,016
10409
10410 ;EXEC: [221]ALUC=LLHHL :[367] D = 177776
10411
10412 ;CODES: [367] SPS=3 / N:C = 1001
10413
10414 ;SYNC: B05J2 (-) T = 2 USEC
10415
10416 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=1L / K3-4 NEG L
10417
10418 017366 012700 000245 T0245: MOV #0245,R0 ;LOAD R0 WITH TEST NO.
10419 017372 013701 017416 MOV @#I0245,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
10420
10421 017376 012702 067560 MOV #MBUF0,R2 ;R2 POINTS TO DEST OP
10422 017402 012704 177776 MOV #-2,R4 ;RESULT S / B = 177776
10423 017406 012712 000002 R0245: MOV #2,(R2) ;INITIAL [DEST] = 2
10424 017412 000257 CCC ;CLEAR FLAGS
10425 017414 000266 266 ;MAKE N:C = 0110
10426
10427 017416 005412 I0245: NEG (2) ;TEST THE NEG
10428
10429 017420 100003 BPL E10245
10430 017422 001402 BEQ E10245 ;N:C = 1001?
10431 017424 102401 BVS E10245
10432 017426 103402 BCS A0245
10433
10434 017430 104000 E10245: ERROR ;NEG FAILED TO ALTER CODES PROPERLY
10435 017432 017406 R0245 ;ERROR LOOP RETURN
10436
10437 017434 021204 A0245: CMP (R2),R4 ;CORRECT RESULT?
10438 017436 001402 BEQ 00245 ;BR IF YES
10439
10440 017440 104000 E20245: ERROR ;NEG DELIVERED WRONG RESULT
10441 017442 017406 R0245 ;ERROR LOOP RETURN
10442
10443 017444 000004 00245: SCOPE ;CALL SCOPE LOOP UTILITY
10444
10445
```

```
10446 ; *****
10447 ; .SBTTL T0246 NEG MODE 1 TEST : [DEST] < 0
10448 ; *****
10449
10450 ;MICROPROGRAMMING / LOGIC INFORMATION
10451
10452 ;ROM SEQ: [161,266,267,221,367,375,016] FC 1,3,9,8
10453
10454 ;ACT BUTS: 37[004]100,161 / 33[266]220,221 / 16[367]016,016
10455
10456 ;EXEC: [221]ALUC=LLHHL :[367] D = 000002
10457
10458 ;CODES: [367] SPS=3 /. N:C = 0001
10459
10460 ;SYNC: B05J2 (-) T = 2 USEC
10461
10462 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=1L / K3-4 NEG L
10463
10464 017446 012700 000246 T0246: MOV #0246,R0 ;LOAD R0 WITH TEST NO.
10465 017452 013701 017476 MOV @#I0246,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
10466
10467 017456 012702 067560 MOV #MBUF0,R2 ;R2 POINTS TO DEST OP
10468 017462 012704 000002 MOV #2,R4 ;RESULT S / B = 2
10469 017466 012712 177776 R0246: MOV #2,(R2) ;INITIAL [DEST] = 177776
10470 017472 000257 CCC ;CLEAR FLAGS
10471 017474 000276 276 ;MAKE N:C = 1110
10472
10473 017476 005412 I0246: NEG (R2) ;TEST THE NEG
10474
10475 017500 100403 BMI E10246
10476 017502 001402 BEQ E10246 ;N:C = 0001?
10477 017504 102401 BVS E10246
10478 017506 103402 BCS A0246
10479
10480 017510 104000 E10246: ERROR ;NEG FAILED TO ALTER CODES PROPERLY
10481 017512 017466 R0246 ;ERROR LOOP RETURN
10482
10483 017514 021204 A0246: CMP (R2),R4 ;CORRECT RESULT = 2?
10484 017516 001402 BEQ 00246 ;BR IF YES
10485
10486 017520 104000 E20246: ERROR ;NEG DELIVERED WRONG RESULT
10487 017522 017466 R0246 ;ERROR LOOP RETURN
10488
10489 017524 000004 00246: SCOPE ;CALL SCOPE LOOP UTILITY
10490
10491
```

10492
10493
10494
10495
10496
10497
10498
10499
10500
10501
10502
10503
10504
10505
10506
10507
10508
10509
10510
10511
10512
10513
10514
10515
10516
10517
10518
10519
10520
10521
10522
10523
10524
10525
10526
10527
10528
10529
10530
10531
10532
10533
10534
10535
10536
10537

017526 012700 000247
017532 013701 017554
017536 012702 067560
017542 012704 100000
017546 010412
017550 000257
017552 000264
017554 005412
017556 100003
017560 001402
017562 102001
017564 103402
017566 104000
017570 017546
017572 021204
017574 001402
017576 104000
017600 017546
017602 000004

```
; *****  
; .SBTTL T0247 NEG MODE 1 TEST: [DEST] = 100000 (8)  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
:ROM SEQ: [161,266,267,221,367,375,016] FC 1,3,9,8  
:ACT BUTS: 37[004]100,161 / 33[266]220,221 / 16[367]016,016  
:EXEC: [221]ALUC=LLHHL :[367] D = 100000  
:CODES: [367] SPS=3 / N:C = 1011  
:SYNC: B05J2 (-) T = 2 USEC  
:KEY SIG: K3-8 CIN00 L / K3-3 DM=1L / K3-4 NEG L  
T0247: MOV #0247,R0 ;LOAD R0 WITH TEST NO.  
MOV @#10247,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUFO,R2 ;R2 POINTS TO DEST OP  
MOV #100000,R4 ;RESULT S / B = 100000  
R0247: MOV R4,(R2) ;INITIAL [DEST] = 100000  
CCC ;CLEAR FLAGS  
SEZ ;MAKE N:Z = 0100  
I0247: NEG (R2) ;TEST THE NEG  
BPL E10247  
BEQ E10247 ;N:C = 1011?  
BVC E10247  
BCS A0247  
E10247: ERROR ;NEG FAILED TO ALTER CODES PROPERLY  
R0247 ;ERROR LOOP RETURN  
A0247: CMP (R2),R4 ;CORRECT RESULT = 100000?  
BEQ 00247 ;BR IF YES  
E20247: ERROR ;NEG DELIVERED WRONG RESULT  
R0247 ;ERROR LOOP RETURN  
O0247: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
10538 ; *****  
10539 ; .SBTTL T0250 ROR TEST - DMO - <N:C> = 1110  
10540 ; *****  
10541 ;MICROPROGRAMMING / LOGIC INFORMATION  
10542 ;ROM SEQ: [106,271,274,001] FC 1,9  
10543 ;ACT BUTS: 37[004]100,106 / 27[271]000,001  
10544 ;EXEC: [106]ALUC=LLLLL :[271] D = 052525  
10545 ;CODES: [106] SPS=1, [274] SPS=2 / N:C = 0000  
10546 ;SYNC: B05J2 (-) T = 1.25 USEC  
10547 ;KEY SIG: K2-5 SDM1 (1) H / K2-5 SDMO (1) H / K3-3 DM=0L / K3-4 OVLAP INSTR  
10548 ; K3-8 ROT (R)L / K3-8 ROTSHF (R)H  
10549  
10550  
10551  
10552  
10553  
10554  
10555  
10556  
10557 017604 012700 000250 T0250: MOV #0250,R0 ;LOAD R0 WITH TEST NO.  
10558 017610 013701 017634 MOV @#I0250,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
10559 017614 012702 177703 MOV #177703,R2 ;DEST ADDR = R3  
10560 017620 012704 052525 MOV #52525,R4 ;RESULT S / B = 52525  
10561 017624 012703 125252 R0250: MOV #125252,R3 ;[DEST] = 125252  
10562 017630 000257 CCC ;CLEAR FLAGS  
10563 017632 000276 276 ;N:C = 1111  
10564  
10565 017634 006003 I0250: ROR R3 ;TEST THE ROR  
10566  
10567 017636 100403 BMI E10250 ;N:C = 0000 ?  
10568 017640 001402 BEQ E10250  
10569 017642 102401 BVS E10250  
10570 017644 103002 BCC A0250  
10571  
10572 017646 104005 E10250: ERROR5 ;ROR FAILED TO ALTER CODES PROPERLY  
10573 017650 017624 R0250 ;ERROR LOOP RETURN ADDRESS  
10574  
10575 017652 020403 A0250: CMP R4,R3 ;CORRECT RESULT ?  
10576 017654 001402 BEQ 00250 ;BR IF YES  
10577  
10578 017656 104000 E20250: ERROR ;ROR DELIVERED THE WRONG RESULT  
10579 017660 017624 R0250 ;ERROR LOOP RETURN  
10580  
10581 017662 000004 00250: SCOPE ;CALL THE SCOPE LOOP UTILITY  
10582
```

10583
10584
10585
10586
10587
10588
10589
10590
10591
10592
10593
10594
10595
10596
10597
10598
10599
10600
10601
10602
10603
10604
10605
10606
10607
10608
10609
10610
10611
10612
10613
10614
10615
10616
10617
10618
10619
10620
10621
10622
10623
10624
10625
10626
10627

017664 012700 000251
017670 013701 017712
017674 012702 177703
017700 005004
017702 012703 000001
017706 000257
017710 000270
017712 006003
017714 100403
017716 001002
017720 102001
017722 103402
017724 104005
017726 017702
017730 020403
017732 001402
017734 104000
017736 017702
017740 000004

; *****
; .SBTTL T0251 ROR TEST - DMO - <N:C> = 1000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [106,271,274,00i] FC 1,9
;ACT BUTS: 37[004]100,106 / 27[271]000,001
;EXEC: [106]ALUC=LLLLL :[271] D = 000000
;CODES: [106] SPS=1, [274] SPS=2 / N:C = 0111
;SYNC: B05J2 (-) T = 1.25 USEC
;KEY SIG: K2-5 SDM1 (1) H / K2-5 SDMO (1) H / K3-3 DM=0L / K3-8 ROT (R)H
; K3-8 ROTSHF (R)H / K3-4 OVLAP INSTR H

T0251: MOV #0251,R0 ;LOAD R0 WITH TEST NO.
MOV @#10251,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
CLR R4 ;RESULT S / B = 000000
R0251: MOV #1,R3 ;[DEST] = 1
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

I0251: ROR R3 ;TEST THE ROR

BMI E10251 ;N:C = 0111 ?
BNE E10251
BVC E10251
BCS A0251

E10251: ERROR5 ;ROR FAILED TO ALTER CODES PROPERLY
R0251 ;ERROR LOOP RETURN ADDRESS

A0251: CMP R4,R3 ;CORRECT RESULT ?
BEQ 00251 ;BR IF YES

E20251: ERROR ;ROR DELIVERED THE WRONG RESULT
R0251 ;ERROR LOOP RETURN

00251: SCOPE ;CALL THE SCOPE LOOP UTILITY

10628
10629
10630
10631
10632
10633
10634
10635
10636
10637
10638
10639
10640
10641
10642
10643
10644
10645
10646
10647
10648
10649
10650
10651
10652
10653
10654
10655
10656
10657
10658
10659
10660
10661
10662
10663
10664
10665
10666
10667
10668
10669
10670
10671
10672

017742 012700 000252
017746 013701 017772
017752 012702 177703
017756 012704 125252
017762 012703 052525
017766 000257
017770 000267
017772 006003
017774 100003
017776 001402
020000 102401
020002 103402
020004 104005
020006 017762
020010 020403
020012 001402
020014 104000
020016 017762
020020 000004

; *****
; .SBTTL T0252 ROR TEST - DMO - <N:C> = 0111
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [106,271,274,001] FC 1,9
;ACT BUTS: 37[004]100,106 / 27[271]000,001
;EXEC: [106]ALUC=LLLLL :[271] D = 125252
;CODES: [106] SPS=1, [274] SPS=2 / N:C = 1001
;SYNC: B05J2 (-) T = 1.25 USEC
;KEY SIG: K1-5 D(C)(1) H / K2-5 SDM1 (1) H / K2-5 SDMO (1) H / K3-8 ROT (R)
; / K3-8 ROTSHF (R)H / K3-3 DM=OL / K3-4 OVLAP INSTR H

T0252: MOV #0252,R0 ;LOAD R0 WITH TEST NO.
MOV @#10252,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
MOV #125252,R4 ;RESULT S / B = 125252
R0252: MOV #52525,R3 ;[DEST] = 052525
CCC ;CLEAR FLAGS
267 ;N:C = 0111
I0252: ROR R3 ;TEST THE ROR
BPL E10252 ;N:C = 1001 ?
BEQ E10252
BVS E10252
BCS A0252
E10252: ERROR5 ;ROR FAILED TO ALTER CODES PROPERLY
R0252 ;ERROR LOOP RETURN ADDRESS
A0252: CMP R4,R3 ;CORRECT RESULT ?
BEQ 00252 ;BR IF YES
E20252: ERROR ;ROR DELIVERED THE WRONG RESULT
R0252 ;ERROR LOOP RETURN
00252: SCOPE ;CALL THE SCOPE LOOP UTILITY

10673
10674
10675
10676
10677
10678
10679
10680
10681
10682
10683
10684
10685
10686
10687
10688
10689
10690
10691
10692
10693
10694
10695
10696
10697
10698
10699
10700
10701
10702
10703
10704
10705
10706
10707
10708
10709
10710
10711
10712
10713
10714
10715
10716
10717

; *****
; .SBTTL T0253 ASR TEST - DMO - <N:C> = 1000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [106,271,274,001] FC 1,9
;ACT BUTS: 37[004]100,106 / 27[271]000,001
;EXEC: [106]ALUC=LLLLL :[271] D = 000000
;CODES: [106] SPS=1, [274] SPS=2 / N:C = 0111
;SYNC: B05J2 (-) T = 1.25 USEC
;KEY SIG: K2-5 SDM1 (1) H / K2-5 SDMO (1) H / K3-3 DM=0L / K3-8 SHF (R)L
; K3-8 ROTSHF (R)H / K3-4 OVLAP INSTR H

T0253: MOV #0253,R0 ;LOAD R0 WITH TEST NO.
MOV @#10253,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
CLR R4 ;RESULT S / B = 000000
R0253: MOV #1,R3 ;[DEST] = 1
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

I0253: ROR R3 ;TEST THE ROR

BMI E10253 ;N:C = 0111 ?
BNE E10253
BVC E10253
BCS A0253

E10253: ERROR5 ;ROR FAILED TO ALTER CODES PROPERLY
R0253 ;ERROR LOOP RETURN ADDRESS

A0253: CMP R4,R3 ;CORRECT RESULT ?
BEQ 00253 ;BR IF YES

E20253: ERROR ;ROR DELIVERED THE WRONG RESULT
R0253 ;ERROR LOOP RETURN

00253: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

10718 ; *****
10719 ; .SBTTL T0254 ASR TEST - DMO - <N:C> = 0101
10720 ; *****
10721 ;MICROPROGRAMMING / LOGIC INFORMATION
10722 ;ROM SEQ: [106,271,274,001] FC 1,9
10723 ;ACT BUTS: 37[004]100,106 / 27[271]000,001
10724 ;EXEC: [106]ALUC=LLLLL :[271] D = 152525
10725 ;CODES: [106] SPS=1, [274] SPS=2 / N:C = 1010
10726 ;SYNC: B05J2 (-) T = 1.25 USEC
10727 ;KEY SIG: K2-5 SDM1 (1) H / K2-5 SDMO (1) H / K3-3 DM=0L / K1-5 D(C)(1)H
10728 ; / K3-4 OVLAP INSTR H / K3-8 SHF (R)L / K3-8 ROTSHF (R) H
10729
10730 T0254: MOV #0254,R0 ;LOAD R0 WITH TEST NO.
10731 MOV @#I0254,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
10732 MOV #177703,R2 ;DEST ADDR = R3
10733 MOV #152525,R4 ;RESULT S / B = 152525
10734 R0254: MOV #125252,R3 ;[DEST] = 125252
10735 CCC ;CLEAR FLAGS
10736 265 ;N:C = 0101
10737
10738 I0254: ROR R3 ;TEST THE ROR
10739 BPL E10254 ;N:C = 1010 ?
10740 BEQ E10254
10741 BVC E10254
10742 BCC A0254
10743
10744 E10254: ERROR5 ;ROR FAILED TO ALTER CODES PROPERLY
10745 R0254 ;ERROR LOOP RETURN ADDRESS
10746
10747 A0254: CMP R4,R3 ;CORRECT RESULT ?
10748 BEQ 00254 ;BR IF YES
10749
10750 E20254: ERROR ;ROR DELIVERED THE WRONG RESULT
10751 R0254 ;ERROR LOOP RETURN
10752
10753 00254: / SCOPE ;CALL THE SCOPE LOOP UTILITY
10754
10755
10756
10757
10758
10759
10760
10761
10762
    
```

```
10763 ; *****
10764 ; .SBTTL T0255 ASR TEST - DMO - <N:C> = 1100
10765 ; *****
10766
10767 ;MICROPROGRAMMING / LOGIC INFORMATION
10768
10769 ;ROM SEQ: [106,271,274,001] FC 1,9
10770
10771 ;ACT BUTS: 37[004]100,106 / 27[271]000,001
10772
10773 ;EXEC: [106]JALUC=LLLLL :[271] D = 025252
10774
10775 ;CODES: [106] SPS=1, [274] SPS=2 / N:C = 0011
10776
10777 ;SYNC: B05J2 (-) T = 1.25 USEC
10778
10779 ;KEY SIG: K2-5 SDM1 (1) H / K2-5 SDMO (1) H / K3-3 DM=0L / K3-8 SHF (R)L
10780 ; K3-8 ROTSHF (R)H / K3-4 OVLAP INSTR H
10781
10782 020160 012700 000255 T0255: MOV #0255,R0 ;LOAD R0 WITH TEST NO.
10783 020164 013701 020210 MOV @#10255,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
10784 020170 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
10785 020174 012704 025252 MOV #25252,R4 ;RESULT S / B = 25252
10786 020200 012703 052525 R0255: MOV #52525,R3 ;[DEST] = 52525
10787 020204 000257 CCC ;CLEAR FLAGS
10788 020206 000274 274 ;N:C = 1100
10789
10790 020210 006003 .I0255: ROR R3 ;TEST THE ROR
10791
10792 020212 100403 BMI E10255 ;N:C = 0011 ?
10793 020214 001402 BEQ E10255
10794 020216 102001 BVC E10255
10795 020220 103402 BCS A0255
10796
10797 020222 104005 E10255: ERROR5 ;ROR FAILED TO ALTER CODES PROPERLY
10798 020224 020200 R0255 ;ERROR LOOP RETURN ADDRESS
10799
10800 020226 020403 A0255: CMP R4,R3 ;CORRECT RESULT ?
10801 020230 001402 BEQ 00255 ;BR IF YES
10802
10803 020232 104000 E20255: ERROR ;ROR DELIVERED THE WRONG RESULT
10804 020234 020200 R0255 ;ERROR LOOP RETURN
10805
10806 020236 000004 00255: SCOPE ;CALL THE SCOPE LOOP UTILITY
10807
```

```
10808 ; *****
10809 ; .SBTTL T0256 ROR TEST - DM1 - <N:C> = 1110
10810 ; *****
10811 ;MICROPROGRAMMING / LOGIC INFORMATION
10812 ;ROM SEQ: [161,266,267,232,275,277,376,016] FC 1,3,9
10813 ;ACT BUTS: 37[004]100,161 / 33[266]220,232 / 16[277]016,016
10814 ;EXEC: [232]ALUC=HHLHL :[275] D = 052525
10815 ;CODES: [232] SPS=1, [277] SPS=2 / N:C = 0000
10816 ;SYNC: B05J2 (-) T = 2 USEC
10817 ;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF L / K3-8 ROT (R) L / K3-8 ROTSHF (R) L
10818 ; K2-5 SDM1 (1) H / K2-5 SDM0 (1) L
10819
10820
10821
10822
10823
10824
10825
10826
10827 020240 012700 000256 T0256: MOV #0256,R0 ;LOAD R0 WITH TEST NO.
10828 020244 013701 020270 MOV @#10256,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
10829 020250 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
10830 020254 012704 052525 MOV #52525,R4 ;RESULT S / B = 52525
10831 020260 012712 125252 R0256: MOV #125252,(R2) ;[DEST] = 125252
10832 020264 000257 CCC ;CLEAR FLAGS
10833 020266 000276 276 ;N:C = 1110
10834
10835 020270 006012 I0256: ROR (R2) ;TEST THE ROR
10836
10837 020272 100403 BMI E10256 ;N:C = 0000 ?
10838 020274 001402 BEQ E10256
10839 020276 102401 BVS E10256
10840 020300 103002 BCC A0256
10841
10842 020302 104005 E10256: ERROR5 ;:ROR FAILED TO ALTER CODES PROPERLY
10843 020304 020260 R0256 ;:ERROR LOOP RETURN
10844
10845 020306 020412 A0256: CMP R4,(R2) ;:CORRECT RESULT ?
10846 020310 001403 BEQ 00256 ;:BR IF YES
10847
10848 020312 011203 MOV (R2),R3 ;:GET THE WAS DATA
10849 020314 104000 E20256: ERROR ;:ROR DELIVERED WRONG RESULT
10850 020316 020260 R0256 ;:ERROR LOOP RETURN ADDRESS
10851
10852 020320 000004 00256: SCOPE ;:CALL THE SCOPE LOOP UTILITY
10853
```

```
10854 ; *****  
10855 ; .SBTTL T0257 ROR TEST - DM1 - <N:C> = 1000  
10856 ; *****  
10857  
10858 ;MICROPROGRAMMING / LOGIC INFORMATION  
10859  
10860 ;ROM SEQ: [161,266,267,232,275,277,376,016] FC 1,3,9  
10861 ;ACT BUTS: 37[004]100,161 / 33[266]220,232 / 16[277]016,016  
10862 ;EXEC: [232]ALUC=HHLHL :[275] D = 000000  
10863 ;CODES: [232] SPS=1, [277] SPS=2 / N:C = 0111  
10864 ;SYNC: B05J2 (-) T = 2 USEC  
10865 ;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF L / K3-8 ROT (R) L / K3-8 ROTSHF (R) L  
10866 ; K2-5 SDM1 (1) H / K2-5 SDMO (1) L  
10867  
10868  
10869  
10870  
10871  
10872  
10873 020322 012700 000257 T0257: MOV #0257,R0 ;LOAD R0 WITH TEST NO.  
10874 020326 013701 020350 MOV @#10257,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
10875 020332 012702 067560 MOV #M000,R2 ;DEST ADDR = M000  
10876 020336 005004 CLR R4 ;RESULT S / B = 000000  
10877 020340 012712 000001 R0257: MOV #1,(R2) ;[DEST] = 1  
10878 020344 000257 CCC ;CLEAR FLAGS  
10879 020346 000270 SEN ;N:C = 1000  
10880  
10881 020350 006012 I0257: ROR (R2) ;TEST THE ROR  
10882  
10883 020352 100403 BMI E10257 ;N:C = 0111 ?  
10884 020354 001002 BNE E10257  
10885 020356 102001 BVC E10257  
10886 020360 103402 BCS A0257  
10887  
10888 020362 104005 E10257: ERROR5 ;ROR FAILED TO ALTER CODES PROPERLY  
10889 020364 020340 R0257 ;ERROR LOOP RETURN  
10890  
10891 020366 020412 A0257: CMP R4,(R2) ;CORRECT RESULT ?  
10892 020370 001403 BEQ 00257 ;BR IF YES  
10893  
10894 020372 011203 MOV (R2),R3 ;GET THE WAS DATA  
10895 020374 104000 E20257: ERROR ;ROR DELIVERED WRONG RESULT  
10896 020376 020340 R0257 ;ERROR LOOP RETURN ADDRESS  
10897  
10898 020400 000004 00257: SCOPE ;CALL THE SCOPE LOOP UTILITY  
10899
```

```
10900 ; *****
10901 ; .SBTTL T0260 ROR TEST - DM1 - <N:C> = 0111
10902 ; *****
10903 ;MICROPROGRAMMING / LOGIC INFORMATION
10904 ;ROM SEQ: [161,266,267,232,275,277,376,016] FC 1,3,9
10905 ;ACT BUTS: 37[004]100,161 / 33[266]220,232 / 16[277]016,016
10906 ;EXEC: [232]ALUC=HHLHL :[275] D = 125252
10907 ;CODES: [232] SPS=1, [277] SPS=2 / N:C = 1001
10908 ;SYNC: B05J2 (-) T = 2 USEC
10909 ;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF L / K3-8 ROT (R) L / K3-8 ROTSHF (R) L
10910 ; / K2-5 SDM1 (1) H / K2-5 SDM0 (1) L / K1-5 D(C)(1) H
10911
10912
10913
10914
10915
10916
10917
10918
10919 020402 012700 000260 T0260: MOV #0260,R0 ;LOAD R0 WITH TEST NO.
10920 020406 013701 020432 MOV @#I0260,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
10921 020412 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
10922 020416 012704 125252 MOV #125252,R4 ;RESULT S / B = 125252
10923 020422 012712 052525 R0260: MOV #52525,(R2) ;[DEST] = 52525
10924 020426 000257 CCC ;CLEAR FLAGS
10925 020430 000267 267 ;N:C = 0111
10926
10927 020432 006012 I0260: ROR (R2) ;TEST THE ROR
10928
10929 020434 100003 BPL E10260 ;N:C = 1001 ?
10930 020436 001402 BEQ E10260
10931 020440 102401 BVS E10260
10932 020442 103402 BCS A0260
10933
10934 020444 104005 E10260: ERROR5 ;ROR FAILED TO ALTER CODES PROPERLY
10935 020446 020422 R0260 ;ERROR LOOP RETURN
10936
10937 020450 020412 A0260: CMP R4,(R2) ;CORRECT RESULT ?
10938 020452 001403 BEQ 00260 ;BR IF YES
10939
10940 020454 011203 MOV (R2),R3 ;GET THE WAS DATA
10941 020456 104000 E20260: ERROR ;ROR DELIVERED WRONG RESULT
10942 020460 020422 R0260 ;ERROR LOOP RETURN ADDRESS
10943
10944 020462 000004 00260: SCOPE ;CALL THE SCOPE LOOP UTILITY
10945
```

```
10946 ; *****  
10947 ; .SBTTL T0261 ASR TEST - DM1 - <N:C> = 1000  
10948 ; *****  
10949  
10950 ;MICROPROGRAMMING / LOGIC INFORMATION  
10951  
10952 ;ROM SEQ: [161,266,267,232,275,277,376,016] FC 1,3,9  
10953 ;ACT BUTS: 37[004]100,161 / 33[266]220,232 / 16[277]016,016  
10954 ;EXEC: [232]ALUC=HHLHL :[275] D = 000000  
10955 ;CODES: [232] SPS=1, [277] SPS=2 / N:C = 0111  
10956 ;SYNC: B05J2 (-) T = 2 USEC  
10957 ;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF L / K3-8 SHF (R) L / K3-8 ROTSHF (R  
10958 ; K2-5 SDM1 (1) H / K2-5 SDMO (1) L  
10959  
10960  
10961  
10962  
10963  
10964  
10965 020464 012700 000261 T0261: MOV #0261,R0 ;LOAD R0 WITH TEST NO.  
10966 020470 013701 020512 MOV @#10261,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
10967 020474 012702 067560 MOV #M00F0,R2 ;DEST ADDR = M00F0  
10968 020500 005004 CLR R4 ;RESULT S / B = 000000  
10969 020502 012712 000001 R0261: MOV #1,(R2) ;[DEST] = 1  
10970 020506 000257 CCC ;CLEAR FLAGS  
10971 020510 000270 SEN ;N:C = 1000  
10972  
10973 020512 006012 I0261: ROR (R2) ;TEST THE ROR  
10974  
10975 020514 100403 BMI E10261 ;N:C = 0111 ?  
10976 020516 001002 BNE E10261  
10977 020520 102001 BVC E10261  
10978 020522 103402 BCS A0261  
10979  
10980 020524 104005 E10261: ERROR5 ;ROR FAILED TO ALTER CODES PROPERLY  
10981 020526 020502 R0261 ;ERROR LOOP RETURN  
10982  
10983 020530 020412 A0261: CMP R4,(R2) ;CORRECT RESULT ?  
10984 020532 001403 BEQ 00261 ;BR IF YES  
10985  
10986 020534 011203 MOV (R2),R3 ;GET THE WAS DATA  
10987 020536 104000 E20261: ERROR ;ROR DELIVERED WRONG RESULT  
10988 020540 020502 R0261 ;ERROR LOOP RETURN ADDRESS  
10989  
10990 020542 000004 00261: SCOPE ;CALL THE SCOPE LOOP UTILITY  
10991
```

```
10992 ; *****  
10993 ; .SBTTL T0262 ASR TEST - DM1 - <N:C> = 1100  
10994 ; *****  
10995  
10996 ;MICROPROGRAMMING / LOGIC INFORMATION  
10997  
10998 ;ROM SEQ: [161,266,267,232,275,277,376,016] FC 1,3,9  
10999  
11000 ;ACT BUTS: 37[004]100,161 / 33[266]220,232 / 16[277]016,016  
11001  
11002 ;EXEC: [232]ALUC=HHLHL :[275] D= 025252  
11003  
11004 ;CODES: [232] SPS=1, [277] SPS=2 / N:C = 0011  
11005  
11006 ;SYNC: B05J2 (-) T = 2 USEC  
11007  
11008 ;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF L / K3-8 SHF (R) L / K3-8 ROTSHF (R  
11009 ; K2-5 SDM1(1) H / K2-5 SDMO (1) L  
11010  
11011 020544 012700 000262 T0262: MOV #0262,R0 ;LOAD R0 WITH TEST NO.  
11012 020550 013701 020574 MOV @#I0262,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
11013 020554 012702 067560 MOV #M000,R2 ;DEST ADDR = M000  
11014 020560 012704 025252 MOV #25252,R4 ;RESULT S / B = 25252  
11015 020564 012712 052525 R0262: MOV #52525,(R2) ;[DEST] = 52525  
11016 020570 000257 CCC ;CLEAR FLAGS  
11017 020572 000274 274 ;N:C = 1100  
11018  
11019 020574 006012 I0262: ROR (R2) ;TEST THE ROR  
11020  
11021 020576 100403 BMI E10262 ;N:C = 0011 ?  
11022 020600 001402 BEQ E10262  
11023 020602 102001 BVC E10262  
11024 020604 103402 BCS A0262  
11025  
11026 020606 104005 E10262: ERROR5 ;ROR FAILED TO ALTER CODES PROPERLY  
11027 020610 020564 R0262 ;ERROR LOOP RETURN  
11028  
11029 020612 020412 A0262: CMP R4,(R2) ;CORRECT RESULT ?  
11030 020614 001403 BEQ 00262 ;BR IF YES  
11031  
11032 020616 011203 MOV (R2),R3 ;GET THE WAS DATA  
11033 020620 104000 E20262: ERROR ;ROR DELIVERED WRONG RESULT  
11034 020622 020564 R0262 ;ERROR LOOP RETURN ADDRESS  
11035  
11036 020624 000004 00262: SCOPE ;CALL THE SCOPE LOOP UTILITY  
11037
```

```
11038 ; *****  
11039 ; .SBTTL T0263 ASR TEST - DM1 - <N:C> = 0101  
11040 ; *****  
11041  
11042 ;MICROPROGRAMMING / LOGIC INFORMATION  
11043  
11044 ;ROM SEQ: [161,266,267,232,275,277,376,016] FC 1,3,9  
11045 ;ACT BUTS: 37[004]100,161 / 33[266]220,232 / 16[277]016,016  
11046 ;EXEC: [232]ALUC=MHLHL :[275] D= 152525  
11047  
11048 ;CODES: [232] SPS=1, [277] SPS=2 / N:C = 1010  
11049  
11050 ;SYNC: B05J2 (-) T = 2 USEC  
11051  
11052 ;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF L / K3-8 SHF (R) L / K3-8 ROTSHF (R) L  
11053 ; / K2-5 SDM1 (1) H / K2-5 SDMO (1) L / K1-5 D(C)(1) H  
11054  
11055  
11056  
11057 020626 012700 000263 T0263: MOV #0263,R0 ;LOAD R0 WITH TEST NO.  
11058 020632 013701 020656 MOV @#I0263,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD *  
11059 020636 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
11060 020642 012704 152525 MOV #152525,R4 ;RESULT S / B = 152525  
11061 020646 012712 125252 R0263: MOV #125252,(R2) ;[DEST] = 125252  
11062 020652 000257 CCC ;CLEAR FLAGS  
11063 020654 000265 265 ;N:C = 0101  
11064  
11065 020656 006012 I0263: ROR (R2) ;TEST THE ROR  
11066  
11067 020660 100003 BPL E10263 ;N:C = 1010 ?  
11068 020662 001402 BEQ E10263  
11069 020664 102001 BVC E10263  
11070 020666 103002 BCC A0263  
11071  
11072 020670 104005 E10263: ERRORS ;ROR FAILED TO ALTER CODES PROPERLY  
11073 020672 020646 R0263 ;ERROR LOOP RETURN  
11074  
11075 020674 020412 A0263: CMP R4,(R2) ;CORRECT RESULT ?  
11076 020676 001403 BEQ 00263 ;BR IF YES  
11077  
11078 020700 011203 E20263: MOV (R2),R3 ;GET THE WAS DATA  
11079 020702 104000 ERROR ;ROR DELIVERED WRONG RESULT  
11080 020704 020646 R0263 ;ERROR LOOP RETURN ADDRESS  
11081  
11082 020706 0C0004 00263: SCOPE ;CALL THE SCOPE LOOP UTILITY  
11083
```

```
11084 ; *****  
11085 ; .SBTTL T0264 RORB TEST - DM2 - EVEN ADDRESS  
11086 ; *****  
11087  
11088 ;MICROPROGRAMMING / LOGIC INFORMATION  
11089  
11090 ;ROM SEQ: [162,260,267,233,276,277,376,016] FC 1,3,9  
11091  
11092 ;ACT BUTS: 37[004]100,162 / 33[260]220,233 / 16[277]016,016  
11093  
11094 ;EXEC: [233] DMUX SHIFT RIGHT :[277] D = 077777  
11095  
11096 ;CODES: [233] SPS=1, [277] SPS=2 / N:C = 0000  
11097  
11098 ;SYNC: B05J2 (-) T = 2.4 USEC  
11099  
11100 ;KEY SIG: K3-6 BYTE INSTR H / K1-5 D(C)(1) H  
11101  
11102 020710 012700 000264 T0264: MOV #0264,R0 ;LOAD R0 WITH TEST NO.  
11103 020714 013701 020740 MOV @#I0264,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
11104 020720 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
11105 020724 012704 000177 MOV #177,R4 ;RESULT S / B = 177  
11106 020730 010203 R0264: MOV R2,R3 ;R3 CONTAINS DEST ADDR  
11107 020732 012712 000377 MOV #377,(R2) ;[DEST] = 377  
11108 020736 000257 CCC ;SCOPE SYNC 'C' = 0  
11109  
11110 020740 106023 I0264: RORB (R3)+ ;TEST THE RORB  
11111  
11112 020742 103402 BCS A0264 ;BR IF ROR SET 'C'  
11113  
11114 020744 104005 E10264: ERROR5 ;ROR FAILED TO SET 'C'  
11115 020746 020730 R0264 ;ERROR LOOP RETURN ADDRESS  
11116  
11117 020750 022703 067561 A0264: CMP #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED ?  
11118 020754 001402 BEQ B0264 ;BR IF YES  
11119  
11120 020756 104005 E20264: ERROR5 ;RORB FAILED TO UPDATE DEST REG  
11121 020760 020730 R0264 ;ERROR LOOP RETURN ADDRESS  
11122  
11123 020762 020412 B0264: CMP R4,(R2) ;CORRECT RESULT ?  
11124 020764 001403 BEQ 00264 ;BR IF YES  
11125  
11126 020766 011203 E30264: MOV (R2),R3 ;GET THE WAS DATA  
11127 020770 104000 ERROR ;RORB DELIVERED WRONG RESULT  
11128 020772 020730 R0264 ;ERROR LOOP RETURN ADDRESS  
11129  
11130 020774 000004 O0264: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
11131 ; *****  
11132 ; .SBTTL T0265 RORB TEST - DM1 - EVEN ADDRESS  
11133 ; *****  
11134 ;MICROPROGRAMMING / LOGIC INFORMATION  
11135 ;ROM SEQ: [161,266,267,233,276,277,376,016] FC 1,3,9  
11136 ;ACT BUTS: 37[004]100,161 / 33[266]220,233 / 16[277]016,016  
11137 ;EXEC: [233] DMUX SHIFT RIGHT :[277] D = 077777  
11138 ;CODES: [233] SPS=1, [277] SPS= / N:C = 1010  
11139 ;SYNC: B05J2 (-) T = 2.4 USEC  
11140 ;KEY SIG: K3-6 BYTE INSTR H / K1-5 D(C)(1) H  
11141  
11142  
11143  
11144  
11145  
11146  
11147  
11148  
11149 020776 012700 000265 T0265: MOV #0265,R0 ;LOAD R0 WITH TEST NO.  
11150 021002 013701 021030 MOV @#I0265,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
11151 021006 012702 067560 MOV #M000,R2 ;DEST ADDR = M000  
11152 021012 012704 000377 MOV #377,R4 ;RESULT S / B = 377  
11153 021016 010203 R0265: MOV R2,R3 ;R3 CONTAINS DEST ADDR  
11154 021020 012712 000376 MOV #376,(R2) ;[DEST] = 376  
11155 021024 000257 CCC ;CLEAR FLAGS  
11156 021026 000261 SEC ;SCOPE SYNC - SET 'C'  
11157  
11158 021030 106013 I0265: RORB (R3) ;TEST THE RORB  
11159  
11160 021032 103002 BCC A0265 ;BR IF 'C' CLR - IT SHOULD BE  
11161  
11162 021034 104005 E10265: ERROR5 ;RORB FAILED TO CLR 'C'  
11163 021036 021016 R0265 ;ERROR LOOP RETURN ADDRESS  
11164  
11165 021040 020412 A0265: CMP R4,(R2) ;CORRECT RESULT ?  
11166 021042 001403 BEQ 00265 ;BR IF YES  
11167  
11168 021044 011203 E20265: MOV (R2),R3 ;GET THE WAS DATA  
11169 021046 104000 ERROR ;RORB DELIVERED WRONG RESULT  
11170 021050 021016 R0265 ;ERROR LOOP RETURN ADDRESS  
11171  
11172 021052 000004 00265: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
11173 ; *****  
11174 ; .SBTTL T0266 RORB TEST - DM2 - ODD ADDRESS  
11175 ; *****  
11176 ;MICROPROGRAMMING / LOGIC INFORMATION  
11177  
11178 ;ROM SEQ: [162,260,267,237,270,233,276,277,376,016] FC 1,3,9  
11179 ;ACT BUTS: 37[004]100,162 / 33[260]220,237 / 34[237]220,233 / 16[277]016,016  
11180  
11181 ;EXEC: [233] DMUX SHIFT RIGHT : [277] D = 77577  
11182  
11183 ;CODES: [233] SPS=1, [277] SPS=3 / N:C = 0011  
11184  
11185 ;SYNC: B05J2 (-) T = 2.8 USEC  
11186  
11187 ;KEY SIG: K3-6 BYTE INSTR H / K3-7 ODD BYTE H / K1-6 BA00(1) H  
11188  
11189  
11190  
11191 021054 012700 000266 T0266: MOV #0266,R0 ;LOAD R0 WITH TEST NO.  
11192 021060 013701 021110 MOV @#I0266,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
11193 021064 012702 067561 MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1  
11194 021070 012704 077777 MOV #77777,R4 ;RESULT S / B = 77777  
11195 021074 012705 067560 MOV #MBUF0,R5 ;POINT R5 TO CHECK RESULT  
11196 021100 010203 R0266: MOV R2,R3 ;R3 CONTAINS DEST ADDR  
11197 021102 012715 177777 MOV #-1,(R5) ;[DEST] = 177777  
11198 021106 000257 CCC ;SCOPE SYNC - 'C' =0  
11199  
11200 021110 106023 I0266: RORB (R3)+ ;TEST THE RORB  
11201  
11202 021112 103402 BCS A0266 ;BR IF 'C' IS SET - IT SHOULD BE  
11203  
11204 021114 104005 E10266: ERROR5 ;RORB FAILED TO SET 'C'  
11205 021116 021100 R0266 ;ERROR LOOP RETURN ADDRESS  
11206  
11207 021120 022703 067562 A0266: CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED ?  
11208 021124 001402 BEQ B0266 ;BR IF YES  
11209  
11210 021126 104005 E20266: ERROR5 ;RORB FAILED TO UPDATE DEST REG  
11211 021130 021100 R0266 ;ERROR LOOP RETURN ADDRESS  
11212  
11213 021132 020415 B0266: CMP R4,(R5) ;CORRECT RESULT ?  
11214 021134 001403 BEQ 00266 ;BR IF YES  
11215  
11216 021136 011503 E30266: MOV (R5),R3 ;GET THE WAS DATA  
11217 021140 104000 ERROR ;RORB DELIVERED WRONG RESULT  
11218 021142 021100 R0266 ;ERROR LOOP RETURN ADDRESS  
11219  
11220 021144 000004 00266: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
11221 ; *****
11222 ; .SBTTL T0267 RORB TEST - DM1 - ODD ADDRESS
11223 ; *****
11224
11225 ;MICROPROGRAMMING / LOGIC INFORMATION
11226
11227 ;ROM SEQ: [161,266,267,237,270,233,276,277,376,016] FC 1,3,9
11228
11229 ;ACT BUTS: 37[004]100,161 / 33[266]220,237 / 34[237]220,233 / 16[277]016,016
11230
11231 ;EXEC: [233] DMUX SHIFT RIGHT :[277] D = 177777
11232
11233 ;CODES: [233] SPS=1, [277] SPS=2 / N:C - 1010
11234
11235 ;SYNC: B05J2 (-) T = 2.8 USEC
11236
11237 ;KEY SIG: K1-5 D(C)(1) H / K3-6 BYTE INSTR H / K3-7 ODD BYTE H / K1-6 BA00(1)
11238
11239 021146 012700 000267 T0267: MOV #0267,R0 ;LOAD R0 WITH TEST NO.
11240 021152 013701 021202 MOV @#I0267,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
11241 021156 012702 067561 MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
11242 021162 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777
11243 021166 012705 067560 MOV #MBUF0,F,5 ;POINT R5 TO CHECK RESULT
11244 021172 010203 R0267: MOV R2,R3 ;R3 CONTAINS DEST ADDR
11245 021174 012715 177377 MOV #177377,(R5) ;[DEST] = 177377
11246 021200 000261 SEC ;SCOPE SYNC - SET 'C'
11247
11248 021202 106023 I0267: RORB (R3)+ ;TEST THE RORB
11249
11250 021204 103002 BCC A0267 ;BR IF 'C' CLEAR - IT SHOULD BE
11251
11252 021206 104005 E10267: ERROR5 ;RORB FAILED TO CLEAR 'C'
11253 021210 021172 R0267 ;ERROR LOOP RETURN ADDRESS
11254
11255 021212 020415 A0267: CMP R4,(R5) ;CORRECT RESULT ?
11256 021214 001403 BEQ 00267 ;BR IF YES
11257
11258 021216 011503 E20267: MOV (R5),R3 ;GET THE WAS DATA
11259 021220 104000 ERROR ;RORB DELIVERED WRONG RESULT
11260 021222 021172 R0267 ;ERROR LOOP RETURN ADDRESS
11261
11262 021224 000004 00267: SCOPE ;CALL SCOPE LOOP UTILITY
```

11263
11264
11265
11266
11267
11268
11269
11270
11271
11272
11273
11274
11275
11276
11277
11278
11279
11280
11281
11282
11283
11284
11285
11286
11287
11288
11289
11290
11291
11292
11293
11294
11295
11296
11297
11298
11299
11300
11301
11302
11303
11304
11305
11306
11307
11308
11309
11310
11311

021226 012700 000270
021232 013701 021262
021236 012702 067561
021242 012704 000377
021246 012705 067560
021252 010203
021254 012715 000777
021260 000257
021262 106223
021264 103402
021266 104005
021270 021252
021272 022703 067562
021276 001402
021300 104005
021302 021252
021304 020415
021306 001403
021310 011503
021312 104000
021314 021252
021316 000004

; *****
; .SBTTL T0270 ASRB TEST - DM2 - ODD ADDRESS
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [162,260,267,237,270,233,276,277,376,016] FC 1,3,9
;ACT BUTS: 37[004]100,162 / 33[260]220,237 / 34[237]220,233 / 16[277]016,016
;EXEC: [233] DMUX SHIFT RIGHT :[277] D = 000000
;CODES: [233] SPS=1, [277] SPS=2 / N:C = 0111
;SYNC: B05J2 (-) T = 2.8 USEC
;KEY SIG: K3-6 BYTE INSTR H / K3-7 ODD BYTE H / K1-6 BA00(1) H

T0270: MOV #0270,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0270,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0270: 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
I0270: ASRB (R3)+ ;TEST THE ASRB
BCS A0270 ;BR IF CARRY SET - IT SHOULD BE
E10270: ERROR5 ;ASRB FAILED TO SET THE CARRY
R0270 ;ERROR LOOP RETURN ADDRESS
A0270: CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED ?
BEQ B0270 ;BR IF YES
E20270: ERROR5 ;ASRB FAILED TO UPDATE DEST REG
R0270 ;ERROR LOOP RETURN ADDRESS
B0270: CMP R4,(R5) ;CORRECT RESULT ?
BEQ 00270 ;BR IF YES
E30270: MOV (R5),R3 ;GET THE WAS DATA
ERROR ;ASRB DELIVERED WRONG RESULT
R0270 ;ERROR LOOP RETURN ADDRESS
00270: SCOPE ;CALL SCOPE LOOP UTILITY

```
11312 ; *****
11313 ; .SBTTL T0271 ASRB TEST - DM1 - ODD ADDRESS
11314 ; *****
11315 ;MICROPROGRAMMING / LOGIC INFORMATION
11316 ;ROM SEQ: [161,266,267,237,270,233,276,277,376,016] FC 1,3,9
11317 ;ACT BUTS: 37[004]100,161 / 33[266]220,237 / 34[237]220,233 / 16[277]016,016
11318 ;EXEC: [233] DMUX SHIFT RIGHT :[277] D = 140300
11319 ;CODES: [233] SPS=1, [277] SPS=2 / N:C = 1010
11320 ;SYNC: B05J2 (-) T = 2.8 USEC
11321 ;KEY SIG: K1-5 D(C)(1) H / K3-6 BYTE INSTR H / K3-7 ODD BYTE H / K1-6 BA00(1)
11322
11323
11324
11325
11326
11327
11328
11329
11330 021320 012700 000271 T0271: MOV #0271,R0 ;LOAD R0 WITH TEST NO.
11331 021324 013701 021354 MOV @#I0271,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
11332 021330 012702 067561 MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
11333 021334 012704 140377 MOV #140377,R4 ;RESULT S / B = 140377
11334 021340 012705 067560 MOV #MBUF0,R5 ;POINT R5 TO CHECK RESULT
11335 021344 010203 R0271: MOV R2,R3 ;R3 CONTAINS DEST ADDR
11336 021346 012715 100377 MOV #100377,(R5) ;[DEST] = 100377
11337 021352 000261 SEC ;SCOPE SYNC - 'C' = 1
11338
11339 021354 106213 I0271: ASRB (R3) ;TEST THE ASRB
11340
11341 021356 103002 BCC A0271 ;BR IF CARRY CLEAR - IT SHOULD BE
11342
11343 021360 104005 E10271: ERRORS ;ASRB FAILED TO CLEAR THE CARRY
11344 021362 021344 R0271 ;ERROR LOOP RETURN ADDRESS
11345
11346 021364 020415 A0271: CMP R4,(R5) ;CORRECT RESULT ?
11347 021366 001403 BEQ 00271 ;BR IF YES
11348
11349 021370 011503 E20271: MOV (R5),R3 ;GET THE WAS DATA
11350 021372 104000 ERROR ;ASRB DELIVERED WRONG RESULT
11351 021374 021344 R0271 ;ERROR LOOP RETURN ADDRESS
11352
11353 021376 000004 00271: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
11354 ; *****  
11355 ; .SBTTL T0272 ASRB TEST - DM2 - EVEN ADDRESS  
11356 ; *****  
11357  
11358 ;MICROPROGRAMMING / LOGIC INFORMATION  
11359  
11360 ;ROM SEQ: [162,260,267,233,276,277,376,016] FC 1,3,9  
11361  
11362 ;ACT BUTS: 37[004]100,162 / 33[260]220,233 / 16[277]016,016  
11363  
11364 ;EXEC: [233] DMUX SHIFT RIGHT :[277] D =  
11365  
11366 ;CODES: [233] SPS=1, [277] SPS=2 / N:C =  
11367  
11368 ;SYNC: B05J2 (-) T = 2.8 USEC  
11369  
11370 ;KEY SIG: K3-6 BYTE INSTR H / K3-3 DM=2L  
11371  
11372 021400 012700 000272 T0272: MOV #0272,R0 ;LOAD R0 WITH TEST NO.  
11373 021404 013701 021430 MOV @#I0272,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
11374 021410 012702 067560 MOV #M0UF0,R2 ;DEST ADDR = M0UF0  
11375 021414 012704 000077 MOV #77,R4 ;RESULT S / B = 77  
11376 021420 010203 R0272: MOV R2,R3 ;R3 CONTAINS DEST ADDR  
11377 021422 012712 000177 MOV #177,(R2) ;[DEST] = 177  
11378 021426 000257 CCC ;SCOPE SYNC - 'C' = 0  
11379  
11380 021430 106223 I0272: ASRB (R3)+ ;TEST THE ASRB  
11381  
11382 021432 103402 BCS A0272 ;BR IF 'C' = 1 - IT SHOULD BE  
11383  
11384 021434 104005 E10272: ERROR5 ;ASRB FAILED TO SET 'C'  
11385 021436 021420 R0272 ;ERROR LOOP RETURN ADDRESS  
11386  
11387 021440 022703 067561 A0272: CMP #M0UF0+1,R3 ;DID DEST REG GET INCREMENTED ?  
11388 021444 001402 BEQ B0272 ;BR IF YES  
11389  
11390 021446 104005 E20272: ERROR5 ;ASRB FAILED TO UPDATE DEST REG  
11391 021450 021420 R0272 ;ERROR LOOP RETURN ADDRESS  
11392  
11393 021452 020412 B0272: CMP R4,(R2) ;CORRECT RESULT ?  
11394 021454 001403 BEQ 00272 ;BR IF YES  
11395  
11396 021456 011203 E30272: MOV (R2),R3 ;GET THE WAS DATA  
11397 021460 104000 ERROR ;ASRB DELIVERED WRONG RESULT  
11398 021462 021420 R0272 ;ERROR LOOP RETURN ADDRESS  
11399  
11400 021464 000004 00272: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
11401 ; *****  
11402 ; .SBTTL T0273 ASRB TEST - DM1 - EVEN ADDRESS  
11403 ; *****  
11404  
11405 ;MICROPROGRAMMING / LOGIC INFORMATION  
11406  
11407 ;ROM SEQ: [161,266,267,233,276,277,376,016] FC 1,3,9  
11408  
11409 ;ACT BUTS: 37[004]100,161 / 33[266]220,233 / 16[277]016,016  
11410  
11411 ;EXEC: [233] DMUX SHIFT RIGHT :[277] D = 141703  
11412  
11413 ;CODES: [233] SPS=1, [277] SPS=2 / N:C = 1010  
11414  
11415 ;SYNC: @05J2 (-) T = 2.4 USEC  
11416  
11417 ;KEY SIG: K1-5 D(C)(1) H / K3-6 BYTE INSTR H  
11418  
11419 021466 012700 000273 T0273: MOV #0273,R0 ;LOAD R0 WITH TEST NO.  
11420 021472 013701 021516 MOV @#I0273,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
11421 021476 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
11422 021502 012704 000303 MOV #303,R4 ;RESULT S / B = 303  
11423 021506 010203 R0273: MOV R2,R3 ;R3 CONTAINS DEST ADDR  
11424 021510 012712 000206 MOV #206,(R2) ;[DEST] = 206  
11425 021514 000261 SEC ;SCOPE SYNC - 'C' = 1  
11426  
11427 021516 106213 I0273: ASRB (R3) ;TEST THE CLRASRB  
11428  
11429 021520 103002 BCC A0273 ;BR IF CARRY CLEAR - IT SHOULD BE  
11430  
11431 021522 104005 E10273: ERROR5 ;ASRB FAILED TO CLEAR THE CARRY  
11432 021524 021506 R0273 ;ERROR LOOP RETURN ADDRESS  
11433  
11434 021526 020412 A0273: CMP R4,(R2) ;CORRECT RESULT ?  
11435 021530 001403 BEQ 00273 ;BR IF YES  
11436  
11437 021532 011203 E20273: MOV (R2),R3 ;GET THE WAS DATA  
11438 021534 104000 ERROR ;ASRB DELIVERED WRONG RESULT  
11439 021536 021506 R0273 ;ERROR LOOP RETURN ADDRESS  
11440  
11441 021540 000004 00273: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
11442 ; *****  
11443 ; .SBTTL T0274 TST DMO TEST - <N:C> = 1011  
11444 ; *****  
11445 ;  
11446 ;MICROPROGRAMMING / LOGIC INFORMATION  
11447 ;  
11448 ;ROM SEQ: [104,373,362,001] FC 1,7,8  
11449 ;  
11450 ;ACT BUTS: 37[004]100,104 / 31[104]360,362 / 27[373]000,001  
11451 ;  
11452 ;EXEC: [104]ALUC=LLLLL :[373] D = 000000  
11453 ;  
11454 ;CODES: [373] SPS=1, [362] SPS=2 / N:C = 0100  
11455 ;  
11456 ;SYNC: B05J2 (-) T = 1 USEC  
11457 ;  
11458 ;KEY SIG: K3-3 DM=0L / K1-7 D(15:00)=0 H / K3-4 TST L / K3-4 OVLAP INSTR H  
11459 ;  
11460 021542 012700 000274 T0274: MOV #0274,R0 ;LOAD R0 WITH TEST NO.  
11461 021546 013701 021600 MOV @#10274,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
11462 021552 032737 000040 066642 BIT #40,@#BPTLOC ;BREAKPOINT HALT SET ??  
11463 021560 001401 BEQ .+4 ;BR IF NOT  
11464 021562 000000 HALT ;BREAK - DEPRESS CONTINUE TO RESTART  
11465 021564 012702 177703 MOV #177703,R2 ;DEST ADDR = R3  
11466 021570 005004 CLR R4 ;RESULT S / B = 000000  
11467 021572 005003 R0274: CLR R3 ;[DEST] = 000000  
11468 021574 000257 CCC ;CLEAR CODES  
11469 021576 000273 273 ;N:C=1011  
11470 ;  
11471 021600 005703 I0274: TST R3 ;TEST THE TST  
11472 ;  
11473 021602 100403 BMI E10274 ;N:C = 0100 ?  
11474 021604 001002 BNE E10274  
11475 021606 102401 BVS E10274  
11476 021610 103002 BCC A0274  
11477 ;  
11478 021612 104005 E10274: ERROR5 ;TST FAILED TO ALTER CODES PROPERLY  
11479 021614 021572 R0274 ;ERROR LOOP RETURN  
11480 ;  
11481 021616 020403 A0274: CMP R4,R3 ;RESULT OK ?  
11482 021620 001402 BEQ 00274 ;BR IF YES  
11483 ;  
11484 021622 104000 E20274: ERROR ;TST ALTERED THE [DEST]  
11485 021624 021572 R0274 ;ERROR LOOP RETURN  
11486 ;  
11487 021626 000004 00274: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
11488 ; *****
11489 ; .SBTTL T0275 TST DMO TEST - <N:C> = 0100
11490 ; *****
11491 ;MICROPROGRAMMING / LOGIC INFORMATION
11492 ;ROM SEQ: [104,373,362,001] FC 1,7,8
11493 ;ACT BUTS: 37[004]100,104 / 31[104]360,362 / 27[373]000,001
11494 ;EXEC: [104]ALUC=LLLLL :[373] D = 177777
11495 ;CODES: [373] SPS=1, [362] SPS=3 / N:C = 1000
11496 ;SYNC: B05J2 (-) T = 1 USEC
11497 ;KEY SIG: K3-3 DM=OL / K3-4 OVLAP INSTR H / K3-4 TST L
11498
11499
11500
11501
11502
11503
11504
11505
11506 021630 012700 000275 T0275: MOV #0275,R0 ;LOAD R0 WITH TEST NO.
11507 021634 013701 021656 MOV @#I0275,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
11508 021640 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
11509 021644 005004 CLR R4
11510 021646 005104 COM R4 ;RESULT S / B = 177777
11511 021650 010403 R0275: MOV R4,R3 ;[DEST] = 177777
11512 021652 000257 CCC ;CLEAR CODES
11513 021654 000264 264 ;N:C=0100
11514
11515 021656 005703 I0275: TST R3 ;TEST THE TST
11516
11517 021660 100003 BPL E10275 ;N:C = 1000 ?
11518 021662 001402 BEQ E10275
11519 021664 102401 BVS E10275
11520 021666 103002 BCC A0275
11521
11522 021670 104005 E10275: ERROR5 ;TST FAILED TO ALTER CODES PROPERLY
11523 021672 021650 R0275 ;ERROR LOOP RETURN
11524
11525 021674 020403 A0275: CMP R4,R3 ;RESULT OK ?
11526 021676 001402 BEQ 00275 ;BR IF YES
11527
11528 021700 104000 E20275: ERROR ;TST ALTERED THE [DEST]
11529 021702 021650 R0275 ;ERROR LOOP RETURN
11530
11531 021704 000004 00275: SCOPE ;CALL SCOPE LOOP UTILITY
11532
```

```

11533 ; *****
11534 ; .SBTTL T0276 CLR DMO TEST - <N:C> = 1011
11535 ; *****
11536 ;MICROPROGRAMMING / LOGIC INFORMATION
11537 ;ROM SEQ: [104,373,360,001] FC 1,7,8
11538 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001
11539 ;EXEC: [104]ALUC=HLLHH :[373] D = 000000
11540 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0100
11541 ;SYNC: B05J2 (-) T = 1 USEC
11542 ;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 CLR L
11543
11544
11545
11546
11547
11548
11549
11550
11551 021706 012700 000276 T0276: MOV #0276,R0 ;LOAD R0 WITH TEST NO.
11552 021712 013701 021734 MOV @#I0276,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
11553 021716 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
11554 021722 005004 CLR R4 ;RESULT S / B = 000000
11555 021724 012703 177777 R0276: MOV #-1,R3 ;[DEST] = 177777
11556 021730 000257 CCC ;CLEAR CODES
11557 021732 000273 273 ;N:C = 1011
11558
11559 021734 005003 I0276: CLR R3 ;TEST THE CLR
11560
11561 021736 100403 BMI E10276 ;N:C = 0100 ?
11562 021740 001002 BNE E10276
11563 021742 102401 BVS E10276
11564 021744 103002 BCC A0276
11565
11566 021746 104005 E10276: ERROR5 ;CLR FAILED TO ALTER THE CODES PROPERLY
11567 021750 021724 R0276 ;ERROR LOOP RETURN
11568 021752 020403 A0276: CMP R4,R3 ;RESULT OK ?
11569 021754 001402 BEQ 00276 ;BR IF YES
11570
11571 021756 104000 E20276: ERROR ;CLR DELIVERED THE WRONG RESULT
11572 021760 021724 R0276 ;ERROR LOOP RETURN
11573
11574 021762 000004 00276: SCOPE ;CALL SCOPE LOOP UTILITY
11575

```

```
11576 ; *****  
11577 ; .SBTTL T0277 CLR DMO TEST - <N:C> = 0000  
11578 ; *****  
11579  
11580 ;MICROPROGRAMMING / LOGIC INFORMATION  
11581  
11582 ;ROM SEQ: [104,373,360,001] FC 1,7,8  
11583  
11584 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
11585  
11586 ;EXEC: [104]ALUC=HLLMH :[373] D = 000000  
11587  
11588 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0100  
11589  
11590 ;SYNC: B05J2 (-) T = 1 USEC  
11591  
11592 ;KEY SIG: K3-3 DM=OL / K3-4 OVLAP INSTR H / K3-4 CLR L  
11593  
11594 021764 012700 000277 T0277: MOV #0277,R0 ;LOAD R0 WITH TEST NO.  
11595 021770 013701 022010 MOV @#I0277,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
11596 021774 012702 177703 MOV #177703,R2 ;DEST ADDR = R3  
11597 022000 005004 CLR R4 ;RESULT S / B = 000000  
11598 022002 012703 177777 R0277: MOV #-1,R3 ;[DEST] = 177777  
11599 022006 000257 CCC ;CLEAR CODES  
11600  
11601 022010 005003 I0277: CLR R3 ;TEST THE CLR  
11602  
11603 022012 100403 BMI E10277 ;N:C = 0100 ?  
11604 022014 001002 BNE E10277  
11605 022016 102401 BVS E10277  
11606 022020 103002 BCC A0277  
11607  
11608 022022 104005 E10277: ERROR5 ;CLR FAILED TO ALTER THE CODES PROPERLY  
11609 022024 022002 R0277 ;ERROR LOOP RETURN  
11610 022026 020403 A0277: CMP R4,R3 ;RESULT OK ?  
11611 022030 001402 BEQ 00277 ;BR IF YES  
11612  
11613 022032 104000 E20277: ERROR ;CLR DELIVERED THE WRONG RESULT  
11614 022034 022002 R0277 ;ERROR LOOP RETURN  
11615  
11616 022036 000004 00277: SCOPE ;CALL SCOPE LOOP UTILITY  
11617
```

11618
11619
11620
11621
11622
11623
11624
11625
11626
11627
11628
11629
11630
11631
11632
11633
11634
11635
11636
11637
11638
11639
11640
11641
11642
11643
11644
11645
11646
11647
11648
11649
11650
11651
11652
11653
11654
11655
11656
11657
11658
11659
11660
11661

022040 012700 000300
022044 013701 022070
022050 012702 177703
022054 012704 125252
022060 012703 052525
022064 000257
022066 000266
022070 005103
022072 100003
022074 001402
022076 102401
022100 103402
022102 104005
022104 022060
022106 020403
022110 001402
022112 104000
022114 022060
022116 000004

```
; *****  
; .SBTTL T0300 COM DMO TEST - <N:C> = 0110  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [104,373,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
;EXEC: [104]ALUC=HLLLL :[373] D = 125252  
;CODES: [373] SPS=1, [360] SPS=3 / N:C = 1001  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 COM L .  
T0300: MOV #0300,R0 ;LOAD R0 WITH TEST NO.  
MOV @#10300,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #125252,R4 ;RESULT S / B = 125252  
R0300: MOV #52525,R3 ;[DEST] = 52525  
CCC ;CLEAR CODES  
266 ;N:C = 0110  
I0300: COM R3 ;TEST THE COM  
BPL E10300 ;N:C = 1001 ?  
BEQ E10300  
BVS E10300  
BCS A0300  
E10300: ERROR5 ;COM FAILED TO ALTER THE CODES PROPERLY  
R0300 ;ERROR LOOP RETURN  
A0300: CMP R4,R3 ;RESULT OK ?  
BEQ 00300 ;BR IF YES  
E20300: ERROR ;COM DELIVERED THE WRONG RESULT  
R0300 ;ERROR LOOP RETURN  
00300: SCOPE ;CALL SCOPE LOOP UTILITY
```

11662
11663
11664
11665
11666
11667
11668
11669
11670
11671
11672
11673
11674
11675
11676
11677
11678
11679
11680
11681
11682
11683
11684
11685
11686
11687
11688
11689
11690
11691
11692
11693
11694
11695
11696
11697
11698
11699
11700
11701
11702
11703
11704

022120 012700 000301
022124 013701 022146
022130 012702 177703
022134 005004
022136 012703 177777
022142 000257
022144 000271
022146 005103
022150 100403
022152 001002
022154 102401
022156 103402
022160 104005
022162 022136
022164 020403
022166 001402
022170 104000
022172 022136
022174 000004

```
; *****  
; .SBTTL T0301 COM DMO TEST - <N:C> = 1001  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [104,373,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
;EXEC: [104]ALUC=HLLLL :[373] D = 000000  
;CODES: [373] SPS=1, [360] SPS=3 / N:C =0101  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 COM L  
T0301: MOV #0301,R0 ;LOAD R0 WITH TEST NO.  
MOV @#10301,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
CLR R4 ;RESULT S / B = 000000  
R0301: MOV #-1,R3 ;[DEST] = 177777  
CCC ;CLEAR CODES  
271 ;N:C = 1001  
I0301: COM R3 ;TEST THE COM  
BMI E10301 ;N:C = 0101 ?  
BNE E10301  
BVS E10301  
BCS A0301  
E10301: ERROR5 ;COM FAILED TO ALTER THE CODES PROPERLY  
R0301 ;ERROR LOOP RETURN  
A0301: CMP R4,R3 ;RESULT OK ?  
BEQ 00301 ;BR IF YES  
E20301: ERROR ;COM DELIVERED THE WRONG RESULT  
R0301 ;ERROR LOOP RETURN  
00301: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
11705 ; *****
11706 ; .SBTTL T0302 INC DMO TEST - <N:C> = 1011
11707 ; *****
11708
11709 ;MICROPROGRAMMING / LOGIC INFORMATION
11710
11711 ;ROM SEQ: [104,373,360,001] FC 1,7,8
11712
11713 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001
11714
11715 ;EXEC: [104]ALUC=LLLLL :[373] D = 000000
11716
11717 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0101
11718
11719 ;SYNC: B05J2 (-) T = 1 USEC
11720
11721 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 INC L
11722
11723 022176 012700 000302 T0302: MOV #0302,R0 ;LOAD R0 WITH TEST NO.
11724 022202 013701 022224 MOV @#10302,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
11725 022206 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
11726 022212 005004 CLR R4 ;RESULT S / B = 000000
11727 022214 012703 177777 R0302: MOV #-1,R3 ;[DEST] = 177777
11728 022220 000257 CCC ;CLEAR CODES
11729 022222 000273 273 ;N:C = 1011
11730
11731 022224 005203 I0302: INC R3 ;TEST THE INC
11732
11733 022226 100403 BMI E10302 ;N:C = 0101 ?
11734 022230 001002 BNE E10302
11735 022232 102401 BVS E10302
11736 022234 103402 BCS A0302
11737
11738 022236 104005 E10302: ERROR5 ;INC FAILED TO ALTER THE CODES PROPERLY
11739 022240 022214 R0302 ;ERROR LOOP RETURN
11740 022242 020403 A0302: CMP R4,R3 ;RESULT OK ?
11741 022244 001402 BEQ 00302 ;BR IF YES
11742
11743 022246 104000 E20302: ERROR ;INC DELIVERED THE WRONG RESULT
11744 022250 022214 R0302 ;ERROR LOOP RETURN
11745
11746 022252 000004 00302: SCOPE ;CALL SCOPE LOOP UTILITY
11747
```

11748
11749
11750
11751
11752
11753
11754
11755
11756
11757
11758
11759
11760
11761
11762
11763
11764
11765
11766
11767
11768
11769
11770
11771
11772
11773
11774
11775
11776
11777
11778
11779
11780
11781
11782
11783
11784
11785
11786
11787
11788
11789
11790

022254 012700 000303
022260 013701 022304
022264 012702 177703
022270 012704 100000
022274 012703 077777
022300 000257
022302 000264
022304 005203
022306 100003
022310 001402
022312 102001
022314 103002
022316 104005
022320 022274
022322 020403
022324 001402
022326 104000
022330 022274
022332 000004

```
; *****  
; .SBTTL T0303 INC DMO TEST - <N:C> = 0100  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [104,373,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
;EXEC: [104]ALUC=LLLLL :[373] D = 100000  
;CODES: [373] SPS=1, [360] SPS=3 / N:C = 1010  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-8 CIN00 L / K3-3 DM=0L / K3-4 INC L / K3-4 OVLAP INSTR H  
T0303: MOV #0303,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0303,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #100000,R4 ;RESULT S / B = 100000  
R0303: MOV #77777,R3 ;[DEST] = 77777  
CCC ;CLEAR CODES  
264 ;N:C = 0100  
I0303: INC R3 ;TEST THE INC  
BPL E10303 ;N:C = 1010 ?  
BEQ E10303  
BVC E10303  
BCC A0303  
E10303: ERROR5 ;INC FAILED TO ALTER THE CODES PROPERLY  
R0303 ;ERROR LOOP RETURN  
A0303: CMP R4,R3 ;RESULT OK ?  
BEQ 00303 ;BR IF YES  
E20303: ERROR ;INC DELIVERED THE WRONG RESULT  
R0303 ;ERROR LOOP RETURN  
00303: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

11791 ; *****
11792 ; .SBTTL T0304 DEC DMO TEST - <N:C> = 1011
11793 ; *****
11794
11795 ;MICROPROGRAMMING / LOGIC INFORMATION
11796
11797 ;ROM SEQ: [104,373,360,001] FC 1,7,8
11798
11799 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001
11800
11801 ;EXEC: [104]ALUC=LHHH :[373] D = 000000
11802
11803 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0101
11804
11805 ;SYNC: B05J2 (-) T = 1 USEC
11806
11807 ;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 DEC L
11808
11809 022334 012700 000304 T0304: MOV #0304,R0 ;LOAD R0 WITH TEST NO.
11810 022340 013701 022362 MOV @#10304,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
11811 022344 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
11812 022350 005004 CLR R4 ;RESULT S / B = 000000
11813 022352 012703 000001 R0304: MOV #1,R3 ;[DEST] = 1
11814 022356 000257 CCC ;CLEAR CODES
11815 022360 000273 273 ;N:C = 1011
11816
11817 022362 005303 I0304: DEC R3 ;TEST THE DEC
11818
11819 022364 100403 BMI E10304 ;N:C = 0101 ?
11820 022366 001002 BNE E10304
11821 022370 102401 BVS E10304
11822 022372 103402 BCS A0304
11823
11824 022374 104005 E10304: ERROR5 ;DEC FAILED TO ALTER THE CODES PROPERLY
11825 022376 022352 R0304 ;ERROR LOOP RETURN
11826 022400 020403 A0304: CMP R4,R3 ;RESULT OK ?
11827 022402 001402 BEQ 00304 ;BR IF YES
11828
11829 022404 104000 E20304: ERROR ;DEC DELIVERED THE WRONG RESULT
11830 022406 022352 R0304 ;ERROR LOOP RETURN
11831
11832 022410 000004 00304: SCOPE ;CALL SCOPE LOOP UTILITY
11833

```

```

11834 ; *****
11835 ; .SBTTL T0305 DEC DMO TEST - <N:C> = 1100
11836 ; *****
11837
11838 ;MICROPROGRAMMING / LOGIC INFORMATION
11839
11840 ;ROM SEQ: [104,373,360,001] FC 1,7,8
11841
11842 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001
11843
11844 ;EXEC: [104]ALUC=LHHMH :[373] D = 77777
11845
11846 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0010
11847
11848 ;SYNC: B05J2 (-) T = 1 USEC
11849
11850 ;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 DEC L
11851
11852 022412 012700 000305 T0305: MOV #0305,R0 ;LOAD R0 WITH TEST NO.
11853 022416 013701 022442 MOV @#10305,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
11854 022422 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
11855 022426 012704 077777 MOV #77777,R4 ;RESULT S / B = 77777
11856 022432 012703 100000 R0305: MOV #100000,R3 ;[DEST] = 100000
11857 022436 000257 CCC ;CLEAR CODES
11858 022440 000274 274 ;N:C = 1100
11859
11860 022442 005303 I0305: DEC R3 ;TEST THE DEC
11861
11862 022444 100403 BMI E10305 ;N:C = 0010 ?
11863 022446 001402 BEQ E10305
11864 022450 102001 BVC E10305
11865 022452 103002 BCC A0305
11866
11867 022454 104005 E10305: ERROR5 ;DEC FAILED TO ALTER THE CODES PROPERLY
11868 022456 022432 R0305 ;ERROR LOOP RETURN
11869 022460 020403 A0305: CMP R4,R3 ;RESULT OK ?
11870 022462 001402 BEQ 00305 ;BR IF YES
11871
11872 022464 104000 E20305: ERROR ;DEC DELIVERED THE WRONG RESULT
11873 022466 022432 R0305 ;ERROR LOOP RETURN
11874
11875 022470 000004 00305: SCOPE ;CALL SCOPE LOOP UTILITY
11876

```

11877
11878
11879
11880
11881
11882
11883
11884
11885
11886
11887
11888
11889
11890
11891
11892
11893
11894
11895
11896
11897
11898
11899
11900
11901
11902
11903
11904
11905
11906
11907
11908
11909
11910
11911
11912
11913
11914
11915
11916
11917
11918

022472 012700 000306
022476 013701 022516
022502 012702 177703
022506 012704 177777
022512 005003
022514 000257

022516 005303

022520 100003
022522 001402
022524 102401
022526 103002

022530 104005
022532 022512
022534 020403
022536 001402

022540 104000
022542 022512

022544 000004

```
; *****  
; .SBTTL T0306 DEC DMO TEST - <N:C> = 0000  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [104,373,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
;EXEC: [104]ALUC=LHHMH :[373] D = 177777  
;CODES: [373] SPS=1, [360] SPS=3 / N:C = 1000  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-3 DM=OL / K3-4 OVLAP INSTR H / K3-4 DEC L  
T0306: MOV #0306,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0306,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #-1,R4 ;RESULT S / B = 177777  
R0306: CLR R3 ;[DEST] = 000000  
;CLEAR CODES  
;CCL  
I0306: DEC R3 ;TEST THE DEC  
;N:C = 1000 ?  
BPL E10306  
BEQ E10306  
BVS E10306  
BCC A0306  
E10306: ERROR5 ;DEC FAILED TO ALTER THE CODES PROPERLY  
R0306 ;ERROR LOOP RETURN  
A0306: CMP R4,R3 ;RESULT OK ?  
BEQ 00306 ;BR IF YES  
E20306: ERROR ;DEC DELIVERED THE WRONG RESULT  
R0306 ;ERROR LOOP RETURN  
00306: SCOPE ;CALL SCOPE LOOP UTILITY
```

11919
11920
11921
11922
11923
11924
11925
11926
11927
11928
11929
11930
11931
11932
11933
11934
11935
11936
11937 022546 012700 000307
11938 022552 013701 022574
11939 022556 012702 177703
11940 022562 005004
11941 022564 012703 100000
11942 022570 000257
11943 022572 000270
11944
11945 022574 006303
11946
11947 022576 100403
11948 022600 001002
11949 022602 102001
11950 022604 103402
11951
11952 022606 104005
11953 022610 022564
11954 022612 020403
11955 022614 001402
11956
11957 022616 104000
11958 022620 022564
11959
11960 022622 000004
11961

; *****
; .SBTTL T0307 ASL DMO TEST - <N:C> = 1000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [104,373,360,001] FC 1,7,8
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001
;EXEC: [104]ALUC=LHLL :[373] D = 000000
;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0111
;SYNC: B05J2 (-) T = 1 USEC
;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-6 ROTSHF (L) L / K3-5 ROTSHF

T0307: MOV #0307,R0 ;LOAD R0 WITH TEST NO.
MOV @#10307,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
CLR R4 ;RESULT S / B = 000000
R0307: MOV #100000,R3 ;[DEST] = 100000
CCC ;CLEAR CODES
SEN ;N:C = 1000

I0307: ASL R3 ;TEST THE ASL

BMI E10307 ;N:C = 0111 ?
BNE E10307
BVC E10307
BCS A0307

E10307: ERROR5 ;ASL FAILED TO ALTER THE CODES PROPERLY
R0307 ;ERROR LOOP RETURN
A0307: CMP R4,R3 ;RESULT OK ?
BEQ 00307 ;BR IF YES

E20307: ERROR ;ASL DELIVERED THE WRONG RESULT
R0307 ;ERROR LOOP RETURN

00307: SCOPE ;CALL SCOPE LOOP UTILITY

11962
11963
11964
11965
11966
11967
11968
11969
11970
11971
11972
11973
11974
11975
11976
11977
11978
11979
11980
11981
11982
11983
11984
11985
11986
11987
11988
11989
11990
11991
11992
11993
11994
11995
11996
11997
11998
11999
12000
12001
12002
12003
12004

022624 012700 000310
022630 013701 022654
022634 012702 177703
022640 012704 100000
022644 012703 040000
022650 000257
022652 000265
022654 006303
022656 100003
022660 001402
022662 102001
022664 103002
022666 104005
022670 022644
022672 020403
022674 001402
022676 104000
022700 022644
022702 000004

: *****
: .SBTTL T0310 ASL DMO TEST - <N:C> = 0101
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [104,373,360,001] FC 1,7,8
:ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001
:EXEC: [104]ALUC=LHLL :[373] D = 100000
:CODES: [373] SPS=1, [360] SPS=3 / N:C = 1010
:SYNC: B05J2 (-) T = 1 USEC
:KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-6 ROTSHF (L) L / K3-5 ROTSHF

T0310: MOV #0310,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0310,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
MOV #100000,R4 ;RESULT S / B = 100000
R0310: MOV #40000,R3 ;[DEST] = 40000
CCC ;CLEAR CODES
265 ;N:C = 0101

I0310: ASL R3 ;TEST THE ASL
BPL E10310 ;N:C = 1010 ?
BEQ E10310
BVC E10310
BCC A0310

E10310: ERROR5 ;ASL FAILED TO ALTER THE CODES PROPERLY
R0310 ;ERROR LOOP RETURN
A0310: CMP R4,R3 ;RESULT OK ?
BEQ 00310 ;BR IF YES

E20310: ERROR ;ASL DELIVERED THE WRONG RESULT
R0310 ;ERROR LOOP RETURN

00310: SCOPE ;CALL SCOPE LOOP UTILITY

```

12005 ; *****
12006 ; .SBTTL T0311 ASL DMO TEST - <N:C> = 0010
12007 ; *****
12008
12009 ;MICROPROGRAMMING / LOGIC INFORMATION
12010
12011 ;ROM SEQ: [104,373,360,001] FC 1,7,8
12012
12013 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001
12014
12015 ;EXEC: [104]ALUC=LHLL :[373] D = 000000
12016
12017 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0100
12018
12019 ;SYNC: B05J2 (-) T = 1 USEC
12020
12021 ;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-6 ROTSHF (L) L / K3-5 R
12022
12023 022704 012700 000311 T0311: MOV #0311,R0 ;LOAD R0 WITH TEST NO.
12024 022710 013701 022730 MOV @#10311,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
12025 022714 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
12026 022720 005004 CLR R4 ;RESULT S / B = 000000
12027 022722 005003 R0311: CLR R3 ;[DEST] = 000000
12028 022724 000257 CCC ;CLEAR CODES
12029 022726 000262 SEV ;N:C = 0010
12030
12031 022730 006303 I0311: ASL R3 ;TEST THE ASL
12032
12033 022732 100403 BMI E10311 ;N:C = 0100 ?
12034 022734 001002 BNE E10311
12035 022736 102401 BVS E10311
12036 022740 103002 BCC A0311
12037
12038 022742 104005 E10311: ERROR5 ;ASL FAILED TO ALTER THE CODES PROPERLY
12039 022744 022722 R0311 ;ERROR LOOP RETURN
12040 022746 020403 A0311: CMP R4,R3 ;RESULT OK ?
12041 022750 001402 BEQ 00311 ;BR IF YES
12042
12043 022752 104000 E20311: ERROR ;ASL DELIVERED THE WRONG RESULT
12044 022754 022722 R0311 ;ERROR LOOP RETURN
12045
12046 022756 000004 00311: SCOPE ;CALL SCOPE LOOP UTILITY
12047

```

```

12048 ; *****
12049 ; .SBTTL T0312 ROL DMO TEST - <N:C> = 1101
12050 ; *****
12051 ;MICROPROGRAMMING / LOGIC INFORMATION
12052 ;ROM SEQ: [104,373,360,001] FC 1,7,8
12053 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001
12054 ;EXEC: [104]ALUC=LHLL :[373] D = 052525
12055 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0011
12056 ;SYNC: B05J2 (-) T = 1 USEC
12057 ;KEY SIG: K3-3 DM=OL / K3-4 OVLAP INSTR H / K3-6 ROTSHF (L) L / K3-5 R
12058 ; K3-8 ROT (L) H
12059
12060 T0312: MOV #0312,R0 ;LOAD R0 WITH TEST NO.
12061 MOV @#10312,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
12062 MOV #177703,R2 ;DEST ADDR = R3
12063 MOV #52525,R4 ;RESULT S / B = 52525
12064 R0312: MOV #125252,R3 ;[DEST] = 125252
12065 CCC ;CLEAR CODES
12066 275 ;N:C = 1101
12067
12068 10312: ROL R3 ;TEST THE ROL
12069 BMI E10312 ;N:C = 0011 ?
12070 BEQ E10312
12071 BVC E10312
12072 BCS A0312
12073
12074 E10312: ERROR5 ;ROL FAILED TO ALTER THE CODES PROPERLY
12075 R0312 ;ERROR LOOP RETURN
12076 A0312: CMP R4,R3 ;RESULT OK ?
12077 BEQ 00312 ;BR IF YES
12078
12079 E20312: ERROR ;ROL DELIVERED THE WRONG RESULT
12080 R0312 ;ERROR LOOP RETURN
12081
12082 00312: SCOPE ;CALL SCOPE LOOP UTILITY
12083
12084
12085
12086
12087
12088
12089
12090
12091

```

12092
12093
12094
12095
12096
12097
12098
12099
12100
12101
12102
12103
12104
12105
12106
12107
12108
12109
12110
12111
12112
12113
12114
12115
12116
12117
12118
12119
12120
12121
12122
12123
12124
12125
12126
12127
12128
12129
12130
12131
12132
12133
12134
12135

023040 012700 000313
023044 013701 023070
023050 012702 177703
023054 012704 125253
023060 012703 052525
023064 000257
023066 000265
023070 006103
023072 100003
023074 001402
023076 102001
023100 103002
023102 104005
023104 023060
023106 020403
023110 001402
023112 104000
023114 023060
023116 000004

```
; *****  
; .SBTTL T0313 ROL DMO TEST - <N:C> = 0101  
; *****  
:MICROPROGRAMMING / LOGIC INFORMATION  
:ROM SEQ: [104,373,360,001] FC 1,7,8  
:ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
:EXEC: [104]ALUC=LHLL :[373] D = 125252  
:CODES: [373] SPS=1, [360] SPS=3 / N:C = 1010  
:SYNC: B05J2 (-) T = 1 USEC  
:KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-6 ROTSHF (L) L / K3-5 ROTSHF  
; K3-8 ROT (L) H  
  
T0313: MOV #0313,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0313,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #125253,R4 ;RESULT S / B = 125253  
R0313: MOV #52525,R3 ;[DEST] = 52525  
CCC ;CLEAR CODES  
265 ;N:C = 0101  
  
I0313: ROL R3 ;TEST THE ROL  
  
BPL E10313 ;N:C = 1010 ?  
BEQ E10313  
BVC E10313  
BCC A0313  
  
E10313: ERROR5 ;ROL FAILED TO ALTER THE CODES PROPERLY  
R0313 ;ERROR LOOP RETURN  
A0313: CMP R4,R3 ;RESULT OK ?  
BEQ 00313 ;BR IF YES  
  
E20313: ERROR ;ROL DELIVERED THE WRONG RESULT  
R0313 ;ERROR LOOP RETURN  
  
00313: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
12136 ; *****  
12137 ; .SBTTL T0314 ROL DMO TEST - <N:C> = 0010  
12138 ; *****  
12139  
12140 ;MICROPROGRAMMING / LOGIC INFORMATION  
12141  
12142 ;ROM SEQ: [104,373,360,001] FC 1,7,8  
12143  
12144 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
12145  
12146 ;EXEC: [104]ALUC=LHLL :[373] D = 000000  
12147  
12148 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0100  
12149  
12150 ;SYNC: B05J2 (-) T = 1 USEC  
12151  
12152 ;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-6 ROTSHF (L) L / K3-5 ROTSHF  
12153 ; K3-8 ROT (L) H  
12154  
12155 023120 012700 000314 T0314: MOV #0314,R0 ;LOAD R0 WITH TEST NO.  
12156 023124 013701 023144 MOV @#I0314,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
12157 023130 012702 177703 MOV #177703,R2 ;DEST ADDR = R3  
12158 023134 005004 CLR R4 ;RESULT S / B = 000000  
12159 023136 005003 R0314: CLR R3 ;[DEST] = 000000  
12160 023140 000257 CCC ;CLEAR CODES  
12161 023142 000262 SEV ;N:C = 0010  
12162  
12163 023144 006103 I0314: ROL R3 ;TEST THE ROL  
12164  
12165 023146 100403 BMI E10314 ;N:C = 0100 ?  
12166 023150 001002 BNE E10314  
12167 023152 102401 BVS E10314  
12168 023154 103002 BCC A0314  
12169  
12170 023156 104005 E10314: ERROR5 ;ROL FAILED TO ALTER THE CODES PROPERLY  
12171 023160 023136 R0314 ;ERROR LOOP RETURN  
12172 023162 020403 A0314: CMP R4,R3 ;RESULT OK ?  
12173 023164 001402 BEQ 00314 ;BR IF YES  
12174  
12175 023166 104000 E20314: ERROR ;ROL DELIVERED THE WRONG RESULT  
12176 023170 023136 R0314 ;ERROR LOOP RETURN  
12177  
12178 023172 000004 00314: SCOPE ;CALL SCOPE LOOP UTILITY  
12179
```

```

12180 ; *****
12181 ; .SBTTL T0315 ADC DMO TEST - <N:C> = 0101
12182 ; *****
12183
12184 ;MICROPROGRAMMING / LOGIC INFORMATION
12185
12186 ;ROM SEQ: [104,373,360,001] FC 1,7,8
12187
12188 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001
12189
12190 ;EXEC: [104]ALUC=LLLLL :[373] D = 100000
12191
12192 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 1010
12193
12194 ;SYNC: B05J2 (-) T = 1 USEC
12195
12196 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 ADC L
12197
12198 023174 012700 000315 T0315: MOV #0315,R0 ;LOAD R0 WITH TEST NO.
12199 023200 013701 023224 MOV @#10315,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
12200 023204 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
12201 023210 012704 100000 MOV #100000,R4 ;RESULT S / B = 100000
12202 023214 012703 077777 R0315: MOV #77777,R3 ;[DEST] = 77777
12203 023220 000257 CCC ;CLEAR CODES
12204 023222 000265 265 ;N:C = 0101
12205
12206 023224 005503 I0315: ADC R3 ;TEST THE ADC
12207
12208 023226 100003 BPL E10315 ;N:C = 1010 ?
12209 023230 001402 BEQ E10315
12210 023232 102001 BVC E10315
12211 023234 103002 BCC A0315
12212
12213 023236 104005 E10315: ERROR5 ;ADC FAILED TO ALTER THE CODES PROPERLY
12214 023240 023214 R0315 ;ERROR LOOP RETURN
12215 023242 020403 A0315: CMP R4,R3 ;RESULT OK ?
12216 023244 001402 BEQ 00315 ;BR IF YES
12217
12218 023246 104000 E20315: ERROR ;ADC DELIVERED THE WRONG RESULT
12219 023250 023214 R0315 ;ERROR LOOP RETURN
12220
12221 023252 000004 00315: SCOPE ;CALL SCOPE LOOP UTILITY
12222

```

```
12223 ; *****  
12224 ; .SBTTL T0316 ADC DMO TEST - <N:C> = 1011  
12225 ; *****  
12226 ;MICROPROGRAMMING / LOGIC INFORMATION  
12227  
12228 ;ROM SEQ: [104,373,360,001] FC 1,7,8  
12229 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
12230 ;EXEC: [104]ALUC=LLLLL :[373] D = 000000  
12231 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0101  
12232 ;SYNC: B05J2 (-) T = 1 USEC  
12233 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 ADC L  
12234  
12235  
12236  
12237  
12238  
12239  
12240  
12241 023254 012700 000316 T0316: MOV #0316,R0 ;LOAD R0 WITH TEST NO.  
12242 023260 013701 023302 MOV @#I0316,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
12243 023264 012702 177703 MOV #177703,R2 ;DEST ADDR = R3  
12244 023270 005004 CLR R4 ;RESULT S / B = 000000  
12245 023272 012703 177777 R0316: MOV #-1,R3 ;[DEST] = 177777  
12246 023276 000257 CCC ;CLEAR CODES  
12247 023300 000273 273 ;N:C = 1011  
12248  
12249 023302 005503 I0316: ADC R3 ;TEST THE ADC  
12250  
12251 023304 100403 BMI E10316 ;N:C = 0101 ?  
12252 023306 001002 BNE E10316  
12253 023310 102401 BVS E10316  
12254 023312 103402 BCS A0316  
12255  
12256 023314 104005 E10316: ERRORS ;ADC FAILED TO ALTER THE CODES PROPERLY  
12257 023316 023272 R0316 ;ERROR LOOP RETURN  
12258 023320 020403 A0316: CMP R4,R3 ;RESULT OK ?  
12259 023322 001402 BEQ 00316 ;BR IF YES  
12260  
12261  
12262 023324 104000 E20316: ERROR ;ADC DELIVERED THE WRONG RESULT  
12263 023326 023272 R0316 ;ERROR LOOP RETURN  
12264  
12265 023330 000004 00316: SCOPE ;CALL SCOPE LOOP UTILITY
```

12266
12267
12268
12269
12270
12271
12272
12273
12274
12275
12276
12277
12278
12279
12280
12281
12282
12283
12284 023332 012700 000317
12285 023336 013701 023362
12286 023342 012702 177703
12287 023346 012704 177777
12288 023352 012703 177777
12289 023356 000257
12290 023360 000272
12291
12292 023362 005503
12293
12294 023364 100003
12295 023366 001402
12296 023370 102401
12297 023372 103002
12298
12299 023374 104005
12300 023376 023352
12301 023400 020403
12302 023402 001402
12303
12304 023404 104000
12305 023406 023352
12306
12307 023410 000004
12308

```

; *****
; .SBTTL T0317 ADC DMO TEST - <N:C> = 1010
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ:      [104,373,360,001] FC 1,7,8
;ACT BUTS:     37[004]100,104 / 31[104]360,360 / 27[373]000,001
;EXEC:         [104]ALUC=LLLLL :[373] D = 177777
;CODES:        [373] SPS=1, [360] SPS=3 / N:C = 1000
;SYNC:         B05J2 (-) T = 1 USEC
;KEY SIG:      / K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 ADC L

T0317:  MOV    #0317,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0317,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV    #177703,R2     ;DEST ADDR = R3
        MOV    #-1,R4         ;RESULT S / B = 177777
R0317:  MOV    #-1,R3         ;[DEST] = 177777
        CCC
        272                 ;CLEAR CODES
                               ;N:C = 1010

I0317:  ADC    R3              ;TEST THE ADC
                               ;N:C = 1000 ?

        BPL    E10317
        BEQ    E10317
        BVS    E10317
        BCC    A0317

E10317:  ERRORS R0317         ;ADC FAILED TO ALTER THE CODES PROPERLY
        R0317                ;ERROR LOOP RETURN
A0317:  CMP    R4,R3          ;RESULT OK ?
        BEQ    00317         ;BR IF YES

E20317:  ERROR  R0317         ;ADC DELIVERED THE WRONG RESULT
        R0317                ;ERROR LOOP RETURN

00317:  SCOPE                  ;CALL SCOPE LOOP UTILITY

```

```
12309 ; *****  
12310 ; .SBTTL T0320 SBC DMO TEST - <N:C> = 1011  
12311 ; *****  
12312 ;MICROPROGRAMMING / LOGIC INFORMATION  
12313  
12314 ;ROM SEQ: [104,373,360,001] FC 1,7,8  
12315 ;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
12316  
12317 ;EXEC: [104]ALUC=LHHH :[373] D = 000000  
12318  
12319 ;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0100  
12320  
12321 ;SYNC: B05J2 (-) T = 1 USEC  
12322  
12323 ;KEY SIG: K3-3 DM=OL / K3-4 OVLAP INSTR H / K3-4 SBC L  
12324  
12325  
12326  
12327 023412 012700 000320 T0320: MOV #0320,R0 ;LOAD R0 WITH TEST NO.  
12328 023416 013701 023440 MOV @#I0320,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
12329 023422 012702 177703 MOV #177703,R2 ;DEST ADDR = R3  
12330 023426 005004 CLR R4 ;RESULT S / B = 000000  
12331 023430 012703 000001 R0320: MOV #1,R3 ;[DEST0321  
12332 023434 000257 CCC ;CLEAR CODES  
12333 023436 000273 273 ;N:C = 1011  
12334  
12335 023440 005603 I0320: SBC R3 ;TEST THE SBC  
12336  
12337 023442 100403 BMI E10320 ;N:C = 0100 ?  
12338 023444 001002 BNE E10320  
12339 023446 102401 BVS E10320  
12340 023450 103002 BCC A0320  
12341  
12342 023452 104005 E10320: ERROR5 ;SBC FAILED TO ALTER THE CODES PROPERLY  
12343 023454 023430 R0320 ;ERROR LOOP RETURN  
12344 023456 020403 A0320: CMP R4,R3 ;RESULT OK ?  
12345 023460 001402 BEQ 00320 ;BR IF YES  
12346  
12347 023462 104000 E20320: ERROR ;SBC DELIVERED THE WRONG RESULT  
12348 023464 023430 R0320 ;ERROR LOOP RETURN  
12349  
12350 023466 000004 00320: SCOPE ;CALL SCOPE LOOP UTILITY  
12351
```

12352
12353
12354
12355
12356
12357
12358
12359
12360
12361
12362
12363
12364
12365
12366
12367
12368
12369
12370
12371
12372
12373
12374
12375
12376
12377
12378
12379
12380
12381
12382
12383
12384
12385
12386
12387
12388
12389
12390
12391
12392
12393
12394

023470 012700 000321
023474 013701 023520
023500 012702 177703
023504 012704 077777
023510 012703 100000
023514 000257
023516 000265
023520 005603
023522 100403
023524 001402
023526 102001
023530 103002
023532 104005
023534 023510
023536 020403
023540 001402
023542 104000
023544 023510
023546 000004

```
; *****  
; .SBTTL T0321 SBC DMO TEST - <N:C> = 0101  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [104,373,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
;EXEC: [104]ALUC=LHHH :[373] D = 077777  
;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0010  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-3 DM=OL / K3-4 OVLAP INSTR H / K3-4 SBC L  
T0321: MOV #0321,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0321,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #077777,R4 ;RESULT S / B = 077777  
R0321: MOV #100000,R3 ;[DEST] = 100000  
CCC ;CLEAR CODES  
265 ;N:C = 0101  
I0321: SBC R3 ;TEST THE SBC  
BMI E10321 ;N:C = 0010 ?  
BEQ E10321  
BVC E10321  
BCC A0321  
E10321: ERROR5 ;SBC FAILED TO ALTER THE CODES PROPERLY  
R0321 ;ERROR LOOP RETURN  
A0321: CMP R4,R3 ;RESULT OK ?  
BEQ 00321 ;BR IF YES  
E20321: ERROR ;SBC DELIVERED THE WRONG RESULT  
R0321 ;ERROR LOOP RETURN  
00321: SCOPE ;CALL SCOPE LOOP UTILITY
```

12395
12396
12397
12398
12399
12400
12401
12402
12403
12404
12405
12406
12407
12408
12409
12410
12411
12412
12413 023550 012700 000322
12414 023554 013701 023600
12415 023560 012702 177703
12416 023564 012704 000001
12417 023570 012703 000001
12418 023574 000257
12419 023576 000276
12420
12421 023600 005603
12422
12423 023602 100403
12424 023604 001402
12425 023606 102401
12426 023610 103002
12427
12428 023612 104005
12429 023614 023570
12430 023616 020403
12431 023620 001402
12432
12433 023622 104000
12434 023624 023570
12435
12436 023626 000004

```
; *****  
; .SBTTL T0322 SBC DMO TEST - <N:C> = 1110  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [104,373,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
;EXEC: [104]ALUC=LLLLL :[373] D = 000001  
;CODES: [373] SPS=1, [360] SPS=3 / N:C = 0000  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 SBC L  
T0322: MOV #0322,R0 ;LOAD R0 WITH TEST NO.  
MOV @#10322,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
R0322: MOV #1,R4 ;RESULT S / B = 1  
MOV #1,R3 ;[DEST] = 1  
CCC ;CLEAR CODES  
276 ;N:C = 1110  
I0322: SBC R3 ;TEST THE SBC  
BMI E10322 ;N:C = 0000 ?  
BEQ E10322  
BVS E10322  
BCC A0322  
E10322: ERROR5 ;SBC FAILED TO ALTER THE CODES PROPERLY  
R0322 ;ERROR LOOP RETURN  
A0322: CMP R4,R3 ;RESULT OK ?  
BEQ 00322 ;BR IF YES  
E20322: ERROR ;SBC DELIVERED THE WRONG RESULT  
R0322 ;ERROR LOOP RETURN  
00322: SCOPE ;CALL SCOPE LOOP UTILITY
```

12437
12438
12439
12440
12441
12442
12443
12444
12445
12446
12447
12448
12449
12450
12451
12452
12453
12454
12455
12456
12457
12458
12459
12460
12461
12462
12463
12464
12465
12466
12467
12468
12469
12470
12471
12472
12473
12474
12475
12476
12477
12478
12479

023630 012700 000323
023634 013701 023656
023640 012702 177703
023644 012704 177777
023650 005003
023652 000257
023654 000267
023656 005603
023660 100003
023662 001402
023664 102401
023666 103402
023670 104005
023672 023650
023674 020403
023676 001402
023700 104000
023702 023650
023704 000004

```
; *****  
; .SBTTL T0323 SBC DMO TEST - <N:C> = 0111  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [104,373,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,104 / 31[104]360,360 / 27[373]000,001  
;EXEC: [104]ALUC=LHHH :[373] D = 177777  
;CODES: [373] SPS=1, [360] SPS=3 / N:C 1001  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-3 DM=0L / K3-4 OVLAP INSTR H / K3-4 SBC L  
T0323: MOV #0323,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0323,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #-1,R4 ;RESULT S / B = 177777  
R0323: CLR R3 ;[DEST] = 000000  
CCC ;CLEAR CODES  
267 ;N:C = 0111  
I0323: SBC R3 ;TEST THE SBC  
BPL E10323 ;N:C = 1001 ?  
BEQ E10323  
BVS E10323  
BCS A0323  
E10323: ERROR5 ;SBC FAILED TO ALTER THE CODES PROPERLY  
R0323 ;ERROR LOOP RETURN  
A0323: CMP R4,R3 ;RESULT OK ?  
BEQ 00323 ;BR IF YES  
E20323: ERROR ;SBC DELIVERED THE WRONG RESULT  
R0323 ;ERROR LOOP RETURN ADDRESS  
00323: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

12480
12481
12482
12483
12484
12485
12486
12487
12488
12489
12490
12491
12492
12493
12494
12495
12496
12497
12498
12499
12500
12501
12502
12503
12504
12505
12506
12507
12508
12509
12510
12511
12512
12513
12514
12515
12516
12517
12518
12519
12520
12521
12522
12523
12524
12525

023706 012700 000324
023712 013701 023732
023716 012702 067560
023722 005004
023724 005012
023726 000257
023730 000273

023732 005712

023734 100403
023736 001002
023740 102401
023742 103002

023744 104005
023746 023724

023750 020412
023752 001403

023754 011203
023756 104000
023760 023724

023762 000004

; *****
; .SBTTL T0324 TST DM1 TEST - <N:C> = 1011
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016
;EXEC: [220]ALUC=LLLLL :[211] D = 000000
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0100
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K3-3 DM=1L / K3-4 TST L / K1-7 D(15:00)=0 H

T0324: MOV #0324,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0324,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
R0324: CLR (R2) ;[DEST] = 000000
CCC ;CLEAR CODES
273 ;N:C=1011

I0324: TST (R2) ;TEST THE TST

BMI E10324 ;N:C = 0100 ?
BNE E10324
BVS E10324
BCC A0324

E10324: ERROR5 ;TST FAILED TO ALTER CODES PROPERLY
R0324 ;ERROR LOOP RETURN

A0324: CMP R4,(R2) ;RESULT OK ?
BEQ 00324 ;BR IF YES

E20324: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;TST ALTERED THE [DEST]
R0324 ;ERROR LOOP RETURN

00324: SCOPE ;CALL SCOPE LOOP UTILITY

12526
12527
12528
12529
12530
12531
12532
12533
12534
12535
12536
12537
12538
12539
12540
12541
12542
12543
12544
12545
12546
12547
12548
12549
12550
12551
12552
12553
12554
12555
12556
12557
12558
12559
12560
12561
12562
12563
12564
12565
12566
12567
12568
12569
12570
12571

023764 012700 000325
023770 013701 024014
023774 012702 067560
024000 005004
024002 005104
024004 012712 177777
024010 000257
024012 000264
024014 005712
024016 100003
024020 001402
024022 102401
024024 103002
024026 104005
024030 024004
024032 020412
024034 001403
024036 011203
024040 104000
024042 024004
024044 000004

```
; *****  
; .SBTTL T0325 TST DM1 TEST - <N:C> = 0100  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=LLLLL :[211] D = 177777  
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 1000  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: K3-3 DM=1L / K3-4 TST L  
T0325: MOV #0325,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0325,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #M0325,R2 ;DEST ADDR = M0325  
CLR R4  
COM R4 ;RESULT S / B = 177777  
R0325: MOV #-1,(R2) ;[DEST] = 177777  
CCC ;CLEAR CODES  
264 ;N:C=0100  
I0325: TST (R2) ;TEST THE TST  
BPL E10325 ;N:C = 1000 ?  
BEQ E10325  
BVS E10325  
BCC A0325  
E10325: ERROR5 ;TST FAILED TO ALTER CODES PROPERLY  
R0325 ;ERROR LOOP RETURN  
A0325: CMP R4,(R2) ;RESULT OK ?  
BEQ 00325 ;BR IF YES  
E20325: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;TST ALTERED THE [DEST]  
R0325 ;ERROR LOOP RETURN  
00325: SCOPE ;CALL SCOPE LOOP UTILITY
```

12572
12573
12574
12575
12576
12577
12578
12579
12580
12581
12582
12583
12584
12585
12586
12587
12588
12589
12590
12591
12592
12593
12594
12595
12596
12597
12598
12599
12600
12601
12602
12603
12604
12605
12606
12607
12608
12609
12610
12611
12612
12613
12614
12615

024046 012700 000326
024052 013701 024074
024056 012702 067560
024062 005004
024064 012712 177777
024070 000257
024072 000273
024074 005012
024076 100403
024100 001002
024102 102401
024104 103002
024106 104005
024110 024064
024112 020412
024114 001403
024116 011203
024120 104000
024122 024064
024124 000004

```
; *****  
; .SBTTL T0326 CLR DM1 TEST - <N:C> = 1011  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=HLLHH :[211] D = 000000  
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0100  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: K3-3 DM=1L / K3-4 CLR L  
T0326: MOV #0326,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0326,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
CLR R4 ;RESULT S / B = 000000  
R0326: MOV #-1,(R2) ;[DEST] = 177777  
CCC ;CLEAR CODES  
273 ;N:C = 1011  
I0326: CLR (R2) ;TEST THE CLR  
BMI E10326 ;N:C = 0100 ?  
BNE E10326  
BVS E10326  
BCC A0326  
E10326: ERROR5 ;CLR FAILED TO ALTER THE CODES PROPERLY  
R0326 ;ERROR LOOP RETURN  
A0326: CMP R4,(R2) ;RESULT OK ?  
BEQ 00326 ;BR IF YES  
E20326: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;CLR DELIVERED THE WRONG RESULT  
R0326 ;ERROR LOOP RETURN  
00326: SCOPE ;CALL SCOPE LOOP UTILITY
```

12616
12617
12618
12619
12620
12621
12622
12623
12624
12625
12626
12627
12628
12629
12630
12631
12632
12633
12634
12635
12636
12637
12638
12639
12640
12641
12642
12643
12644
12645
12646
12647
12648
12649
12650
12651
12652
12653
12654
12655
12656
12657
12658
12659
12660
12661
12662
12663
12664

024126 012700 000327
024132 013701 024152
024136 012702 067560
024142 005004
024144 013712 067572
024150 000257
024152 005022
024154 100403
024156 001002
024160 102401
024162 103002
024164 104005
024166 024144
024170 022702 067562
024174 001402
024176 104005
024200 024144
024202 020442
024204 001403
024206 011203
024210 104000
024212 024144
024214 000004

```
; *****  
; .SBTTL T0327 CLR DM2 TEST - <N:C> = 0000  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=HLLHH :[211] D = 000000  
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0100  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: K3-3 DM=1L / K3-4 CLR L  
T0327: MOV #0327,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0327,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
CLR R4 ;RESULT S / B = 000000  
R0327: MOV @#DWTA+2,(R2) ;[DEST] = 177777  
CCC ;CLEAR CODES  
I0327: CLR (R2)+ ;TEST THE CLR  
BMI E10327 ;N:C = 0100 ?  
BNE E10327  
BVS E10327  
BCC A0327  
E10327: ERROR5 ;CLR FAILED TO ALTER THE CODES PROPERLY  
R0327 ;ERROR LOOP RETURN  
A0327: CMP #MBUF0+2,R2 ;DID CLR INCREMENT DEST REG  
BEQ B0327 ;BR IF YES  
E20327: ERROR5 ;CLR FAILED TO UPDATE DEST REG  
R0327 ;ERROR LOOP RETURN  
B0327: CMP R4,-(R2) ;RESULT OK ?  
BEQ 00327 ;BR IF YES  
E30327: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;CLR DELIVERED THE WRONG RESULT  
R0327 ;ERROR LOOP RETURN  
00327: SCOPE ;CALL SCOPE LOOP UTILITY
```

12665
12666
12667
12668
12669
12670
12671
12672
12673
12674
12675
12676
12677
12678
12679
12680
12681
12682
12683
12684
12685
12686
12687
12688
12689
12690
12691
12692
12693
12694
12695
12696
12697
12698
12699
12700
12701
12702
12703
12704
12705
12706
12707
12708
12709

024216 012700 000330
024222 013701 024246
024226 012702 067560
024232 012704 125252
024236 012712 052525
024242 000257
024244 000266
024246 005112
024250 100003
024252 001402
024254 102401
024256 103402
024260 104005
024262 024236
024264 020412
024266 001403
024270 011203
024272 104000
024274 024236
024276 000004

```
; *****  
; .SBTTL T0330 COM DM1 TEST - <N:C> = 0110  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=HLLLL :[211] D = 125252  
;CODES: [211] SPS=1, [367] SPS=3 / N:C=1001  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: K3-3 DM=1L / K3-4 COM L  
T0330: MOV #0330,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0330,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #125252,R4 ;RESULT S / B = 125252  
R0330: MOV #52525,(R2) ;[DEST] = 52525  
CCC ;CLEAR CODES  
266 ;N:C = 0110  
I0330: COM (R2) ;TEST THE CLR  
BPL E10330 ;N:C = 1001 ?  
BEQ E10330  
BVS E10330  
BCS A0330  
E10330: ERROR5 ;COM FAILED TO ALTER THE CODES PROPERLY  
R0330 ;ERROR LOOP RETURN  
A0330: CMP R4,(R2) ;RESULT OK ?  
FEQ 00330 ;BR IF YES  
MOV (R2),R3 ;GET THE WAS DATA  
E20330: ERROR ;COM DELIVERED THE WRONG RESULT  
R0330 ;ERROR LOOP RETURN  
00330: SCOPE ;CALL SCOPE LOOP UTILITY
```

12710
12711
12712
12713
12714
12715
12716
12717
12718
12719
12720
12721
12722
12723
12724
12725
12726
12727
12728 024300 012700 000331
12729 024304 013701 024326
12730 024310 012702 067560
12731 024314 005004
12732 024316 012712 177777
12733 024322 000257
12734 024324 000271
12735
12736 024326 005112
12737
12738 024330 100403
12739 024332 001002
12740 024334 102401
12741 024336 103402
12742
12743 024340 104005
12744 024342 024316
12745 024344 020412
12746 024346 001403
12747
12748 024350 011203
12749 024352 104000
12750 024354 024316
12751
12752 024356 000004
12753

; *****
; .SBTTL T0331 COM DM1 TEST - <N:C> = 1001
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016
;EXEC: [220]ALUC=HLLLL :[211] D = 000000
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0101
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K3-3 DM=1L / K3-4 COM L

T0331: MOV #0331,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0331,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
R0331: MOV #-1,(R2) ;[DEST] = 177777
CCC ;CLEAR CODES
271 ;N:C = 1001

I0331: COM (R2) ;TEST THE COM

BMI E10331 ;N:C = 0101 ?
BNE E10331
BVS E10331
BCS A0331

E10331: ERROR5 ;COM FAILED TO ALTER THE CODES PROPERLY
R0331 ;ERROR LOOP RETURN
A0331: CMP R4,(R2) ;RESULT OK ?
BEQ 00331 ;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
E20331: ERROR ;COM DELIVERED THE WRONG RESULT
R0331 ;ERROR LOOP RETURN

00331: SCOPE ;CALL SCOPE LOOP UTILITY

12754
12755
12756
12757
12758
12759
12760
12761
12762
12763
12764
12765
12766
12767
12768
12769
12770
12771
12772 024360 012700 000332
12773 024364 013701 024406
12774 024370 012702 067560
12775 024374 005004
12776 024376 012712 177777
12777 024402 000257
12778 024404 000273
12779
12780 024406 005212
12781
12782 024410 100403
12783 024412 001002
12784 024414 102401
12785 024416 103402
12786
12787 024420 104005
12788 024422 024376
12789 024424 020412
12790 024426 001403
12791
12792 024430 011203
12793 024432 104000
12794 024434 024376
12795
12796 024436 000004
12797

```
; *****  
; .SBTTL T0332 INC DM1 TEST - <N:C> = 1011  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [161,266,267,220,211,367,275,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=L LLLL :[211] D = 000000  
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0101  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: K3-3 DM=1L / K3-4 INC L / K3-8 CIN00 L  
  
T0332: MOV #0332,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0332,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
CLR R4 ;RESULT S / B = 000000  
R0332: MOV #-1,(R2) ;[DEST] = 177777  
CCC ;CLEAR CODES  
273 ;N:C = 1011  
  
I0332: INC (R2) ;TEST THE INC  
  
BMI E10332 ;N:C = 0101 ?  
BNE E10332  
BVS E10332  
BCS A0332  
  
E10332: ERROR5 ;INC FAILED TO ALTER THE CODES PROPERLY  
R0332 ;ERROR LOOP RETURN  
A0332: CMP R4,(R2) ;RESULT OK ?  
BEQ 00332 ;BR IF YES  
  
MOV (R2),R3 ;GET THE WAS DATA  
E20332: ERROR ;INC DELIVERED THE WRONG RESULT  
R0332 ;ERROR LOOP RETURN  
  
00332: SCOPE ;CALL SCOPE LOOP UTILITY
```

12798
12799
12800
12801
12802
12803
12804
12805
12806
12807
12808
12809
12810
12811
12812
12813
12814
12815
12816 024440 012700 000333
12817 024444 013701 024470
12818 024450 012702 067560
12819 024454 012704 100000
12820 024460 012712 077777
12821 024464 000257
12822 024466 000264
12823
12824 024470 005212
12825
12826 024472 100003
12827 024474 001402
12828 024476 102001
12829 024500 103002
12830
12831 024502 104005
12832 024504 024460
12833 024506 020412
12834 024510 001403
12835
12836 024512 011203
12837 024514 104000
12838 024516 024460
12839
12840 024520 000004
12841

; *****
; .SBTTL T0333 INC DM1 TEST - <N:C> = 0100
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016
;EXEC: [220]ALUC=LLLLL :[211] D = 100000
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 1010
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K3-3 DM=1L / K3-4 INC L / K3-8 CIN00 L

T0333: MOV #0333,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0333,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #M000,R2 ;DEST ADDR = M000
MOV #100000,R4 ;RESULT S / B = 100000
R0333: MOV #77777,(R2) ;[DEST] = 77777
CCC ;CLEAR CODES
264 ;N:C = 0100

I0333: INC (R2) ;TEST THE INC
BPL E10333 ;N:C = 1010 ?
BEQ E10333
BVC E10333
BCC A0333

E10333: ERROR5 ;INC FAILED TO ALTER THE CODES PROPERLY
R0333 ;ERROR LOOP RETURN
A0333: CMP R4,(R2) ;RESULT OK ?
BEQ 00333 ;BR IF YES

E20333: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;INC DELIVERED THE WRONG RESULT
R0333 ;ERROR LOOP RETURN

00333: SCOPE ;CALL SCOPE LOOP UTILITY

12842
12843
12844
12845
12846
12847
12848
12849
12850
12851
12852
12853
12854
12855
12856
12857
12858
12859
12860
12861
12862
12863
12864
12865
12866
12867
12868
12869
12870
12871
12872
12873
12874
12875
12876
12877
12878
12879
12880
12881
12882
12883
12884
12885
12886
12887
12888

024522 012700 000334
024526 013701 024562
024532 032737 000100 066642
024540 001401
024542 000000
024544 012702 067560
024550 005004
024552 012712 000001
024556 000257
024560 000273

024562 005312

024564 100403
024566 001002
024570 102401
024572 103402

024574 104005
024576 024552
024600 020412
024602 001403

024604 011203
024606 104000
024610 024552

024612 000004

```
; *****  
; .SBTTL T0334 DEC DM1 TEST - <N:C> = 1011  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=LHHH [211] D = 000000  
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0101  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: K3-3 DM=1L / K3-4 DEC L  
  
T0334: MOV #0334,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0334,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
BIT #100,@#BPTLOC ;BREAKPOINT HALT SET  
BEQ .+4 ;BR IF NOT  
HALT ;BREAK-DEPRESS CONTINUE TO RESTART  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
CLR R4 ;RESULT S / B = 000000  
R0334: MOV #1,(R2) ;[DEST] = 1  
CCC ;CLEAR CODES  
273 ;N:C = 1011  
  
I0334: DEC (R2) ;TEST THE DEC  
  
BMI E10334 ;N:C = 0101 ?  
BNE E10334  
BVS E10334  
BCS A0334  
  
E10334: ERROR5 ;DEC FAILED TO ALTER THE CODES PROPERLY  
R0334 ;ERROR LOOP RETURN  
A0334: CMP R4,(R2) ;RESULT OK ?  
BEQ 00334 ;BR IF YES  
  
E20334: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;DEC DELIVERED THE WRONG RESULT  
R0334 ;ERROR LOOP RETURN  
  
00334: SCOPE ;CALL SCOPE LOOP UTILITY
```

12889
12890
12891
12892
12893
12894
12895
12896
12897
12898
12899
12900
12901
12902
12903
12904
12905
12906
12907
12908
12909
12910
12911
12912
12913
12914
12915
12916
12917
12918
12919
12920
12921
12922
12923
12924
12925
12926
12927
12928
12929
12930
12931
12932

024614 012700 000335
024620 013701 024644
024624 012702 067560
024630 012704 077777
024634 012712 100000
024640 000257
024642 000274
024644 005312
024646 100403
024650 001402
024652 102001
024654 103002
024656 104005
024660 024634
024662 020412
024664 001403
024666 011203
024670 104000
024672 024634
024674 000004

```
; *****  
; .SBTTL T0335 DEC DM1 TEST - <N:C> = 1100  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
:ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
:ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
:EXEC: [220]ALUC=LHHH [211] D = 77777  
:CODES: [211] SPS=1, [367] SPS=3 / N:C = 0010  
:SYNC: B05J2 (-) T = 2 USEC  
:KEY SIG: K3-3 DM=1L / K3-4 DEC L  
T0335: MOV #0335,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0335,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
MOV #77777,R4 ;RESULT S / B = 77777  
R0335: MOV #100000,(R2) ;[DEST] = 100000  
CCC ;CLEAR CODES  
274 ;N:C = 1100  
I0335: DEC (R2) ;TEST THE DEC  
BMI E10335 ;N:C = 0010 ?  
BEQ E10335  
BVC E10335  
BCC A0335  
E10335: ERROR5 ;DEC FAILED TO ALTER THE CODES PROPERLY  
R0335 ;ERROR LOOP RETURN  
A0335: CMP R4,(R2) ;RESULT OK ?  
BEQ 00335 ;BR IF YES  
E20335: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;DEC DELIVERED THE WRONG RESULT  
R0335 ;ERROR LOOP RETURN  
00335: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
12933 ; *****  
12934 ; .SBTTL T0336 DEC DM1 TEST - <N:C> = 0000  
12935 ; *****  
12936  
12937 ;MICROPROGRAMMING / LOGIC INFORMATION  
12938  
12939 ;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
12940  
12941 ;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
12942  
12943 ;EXEC: [220]ALUC=LHHHH :[211] D = 177777  
12944  
12945 ;CODES: [211] SPS=1, [367] SPS=3 / N:C 1000  
12946  
12947 ;SYNC: B05J2 (-) T = 2 USEC  
12948  
12949 ;KEY SIG: K3-3 DM=1L / K3-4 DEC L  
12950  
12951 024676 012700 000336 T0336: MOV #0336,R0 ;LOAD R0 WITH TEST NO.  
12952 024702 013701 024722 MOV @#I0336,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
12953 024706 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
12954 024712 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777  
12955 024716 005012 R0336: CLR (R2) ;[DEST] = 000000  
12956 024720 000257 CCC ;CLEAR CODES  
12957  
12958 024722 005312 I0336: DEC (R2) ;TEST THE DEC  
12959  
12960 024724 100003 BPL E10336 ;N:C = 1000 ?  
12961 024726 001402 BEQ E10336  
12962 024730 102401 BVS E10336  
12963 024732 103002 BCC A0336  
12964  
12965 024734 104005 E10336: ERROR5 ;DEC FAILED TO ALTER THE CODES PROPERLY  
12966 024736 024716 R0336 ;ERROR LOOP RETURN  
12967 024740 020412 A0336: CMP R4,(R2) ;RESULT OK ?  
12968 024742 001403 BEQ 00336 ;BR IF YES  
12969  
12970 024744 011203 MOV (R2),R3 ;GET THE WAS DATA  
12971 024746 104000 E20336: ERROR ;DEC DELIVERED THE WRONG RESULT  
12972 024750 024716 R0336 ;ERROR LOOP RETURN  
12973  
12974 024752 000004 00336: SCOPE ;CALL SCOPE LOOP UTILITY  
12975
```

12976
12977
12978
12979
12980
12981
12982
12983
12984
12985
12986
12987
12988
12989
12990
12991
12992
12993
12994 024754 012700 000337
12995 024760 013701 025002
12996 024764 012702 067560
12997 024770 005004
12998 024772 012712 100000
12999 024776 000257
13000 025000 000270
13001
13002 025002 006312
13003
13004 025004 100403
13005 025006 001002
13006 025010 102001
13007 025012 103402
13008
13009 025014 104005
13010 025016 024772
13011 025020 020412
13012 025022 001403
13013
13014 025024 011203
13015 025026 104000
13016 025030 024772
13017
13018 025032 000004
13019

; *****
; .SBTTL T0337 ASL DM1 TEST - <N:C> = 1000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016
;EXEC: [220]ALUC=LHLL :[211] D = 000000
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0111
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF (L) L / K3-5 ROTSHF H

T0337: MOV #0337,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0337,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
CLR R4 ;RESULT S / B = 000000
R0337: MOV #100000,(R2) ;[DEST] = 100000
CCC ;CLEAR CODES
SEN ;N:C = 1000

I0337: ASL (R2) ;TEST THE ASL

BMI E10337 ;N:C = 0111 ?
BNE E10337
BVC E10337
BCS A0337

E10337: ERROR5 ;ASL FAILED TO ALTER THE CODES PROPERLY
R0337 ;ERROR LOOP RETURN
A0337: CMP R4,(R2) ;RESULT OK ?
BEQ 00337 ;BR IF YES

MOV (R2),R3 ;GET THE WAS DATA
E20337: ERROR ;ASL DELIVERED THE WRONG RESULT
R0337 ;ERROR LOOP RETURN

00337: SCOPE ;CALL SCOPE LOOP UTILITY

13020
13021
13022
13023
13024
13025
13026
13027
13028
13029
13030
13031
13032
13033
13034
13035
13036
13037
13038
13039
13040
13041
13042
13043
13044
13045
13046
13047
13048
13049
13050
13051
13052
13053
13054
13055
13056
13057
13058
13059
13060
13061
13062
13063

; *****
; .SBTTL T0340 ASL DM1 TEST - <N:C> = 0101
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016
;EXEC: [220]ALUC=LHLL :[211] D = 100000
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 1010
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF (L) L / K3-5 ROTSHF H

T0340: MOV #0340,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0340,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #100000,R4 ;RESULT S / B = 100000
R0340: MOV #40000,(R2) ;[DEST] = 40000
CCC ;CLEAR CODES
265 ;N:C = 0101

I0340: ASL (R2) ;TEST THE ASL
BPL E10340 ;N:C = 1010 ?
BEQ E10340
BVC E10340
BCC A0340

E10340: ERROR5 ;ASL FAILED TO ALTER THE CODES PROPERLY
R0340 ;ERROR LOOP RETURN
A0340: CMP R4,(R2) ;RESULT OK ?
BEQ 00340 ;BR IF YES

E20340: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;ASL DELIVERED THE WRONG RESULT
R0340 ;ERROR LOOP RETURN

00340: SCOPE ;CALL SCOPE LOOP UTILITY

025034 012700 000340
025040 013701 025064
025044 012702 067560
025050 012704 100000
025054 012712 040000
025060 000257
025062 000265
025064 006312
025066 100003
025070 001402
025072 102001
025074 103002
025076 104005
025100 025054
025102 020412
025104 001403
025106 011203
025110 104000
025112 025054
025114 000004

13064
13065
13066
13067
13068
13069
13070
13071
13072
13073
13074
13075
13076
13077
13078
13079
13080
13081
13082
13083
13084
13085
13086
13087
13088
13089
13090
13091
13092
13093
13094
13095
13096
13097
13098
13099
13100
13101
13102
13103
13104
13105
13106
13107

025116 012700 000341
025122 013701 025142
025126 012702 067560
025132 005004
025134 005012
025136 000257
025140 000262
025142 006312
025144 100403
025146 001002
025150 102401
025152 103002
025154 104005
025156 025134
025160 020412
025162 001403
025164 011203
025166 104000
025170 025134
025172 000004

```
; *****  
; .SBTTL T0341 ASL DM1 TEST - <N:C> = 0010  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=LHLL :[211] D = 000000  
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0100  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF (L) L / K3-5 ROTSHF F  
T0341: MOV #0341,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0341,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
R0341: CLR R4 ;RESULT S / B = 000000  
CLR (R2) ;[DEST] = 000000  
CCC ;CLEAR CODES  
SEV ;N:C = 0010  
I0341: ASL (R2) ;TEST THE ASL  
BMI E10341 ;N:C = 0100 ?  
BNE E10341  
BVS E10341  
BCC A0341  
E10341: ERROR5 ;ASL FAILED TO ALTER THE CODES PROPERLY  
R0341 ;ERROR LOOP RETURN  
A0341: CMP R4,(R2) ;RESULT OK ?  
BEQ 00341 ;BR IF YES  
E20341: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;ASL DELIVERED THE WRONG RESULT  
R0341 ;ERROR LOOP RETURN  
00341: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
13108 ; *****  
13109 ; .SBTTL T0342 ROL DM1 TEST - <N:C> = 1101  
13110 ; *****  
13111  
13112 ;MICROPROGRAMMING / LOGIC INFORMATION  
13113  
13114 ;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
13115  
13116 ;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
13117  
13118 ;EXEC: [220]JALUC=LHLL :[211] D = 052525  
13119  
13120 ;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0011  
13121  
13122 ;SYNC: B05J2 (-) T = 2 USEC  
13123  
13124 ;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF (L) L / K3-5 ROTSHF H / K3-8 ROT (L) H  
13125  
13126 025174 012700 000342 T0342: MOV #0342,R0 ;LOAD R0 WITH TEST NO.  
13127 025200 013701 025224 MOV @#I0342,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
13128 025204 012702 067560 MOV #M0342,R2 ;DEST ADDR = M0342  
13129 025210 012704 052525 MOV #52525,R4 ;RESULT S / B = 52525  
13130 025214 012712 125252 R0342: MOV #125252,(R2) ;[DEST] = 125252  
13131 025220 000257 CCC ;CLEAR CODES  
13132 025222 000275 275 ;N:C = 1101  
13133  
13134 025224 006112 I0342: ROL (R2) ;TEST THE ROL  
13135  
13136 025226 100403 BMI E10342 ;N:C = 0011 ?  
13137 025230 001402 BEQ E10342  
13138 025232 102001 BVC E10342  
13139 025234 103402 BCS A0342  
13140  
13141 025236 104005 E10342: ERROR5 ;ROL FAILED TO ALTER THE CODES PROPERLY  
13142 025240 025214 R0342 ;ERROR LOOP RETURN  
13143 025242 020412 A0342: CMP R4,(R2) ;RESULT OK ?  
13144 025244 001403 BEQ 00342 ;BR IF YES  
13145  
13146 025246 011203 MOV (R2),R3 ;GET THE WAS DATA  
13147 025250 104000 E20342: ERROR ;ROL DELIVERED THE WRONG RESULT  
13148 025252 025214 R0342 ;ERROR LOOP RETURN  
13149  
13150 025254 000004 00342: SCOPE ;CALL SCOPE LOOP UTILITY  
13151
```

```
13152 ; *****  
13153 ; .SBTTL T0343 ROL DM1 TEST - <N:C> = 0101  
13154 ; *****  
13155  
13156 ;MICROPROGRAMMING / LOGIC INFORMATION  
13157  
13158 ;ROM SEQ: [161,266,267,220,211,367,075,016] FC 1,3,9,8  
13159  
13160 ;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
13161  
13162 ;EXEC: [220]ALUC=LHLL :[211] D = 125253  
13163  
13164 ;CODES: [211] SPS=1, [367] SPS=3 / N:C = 1010  
13165  
13166 ;SYNC: B05J2 (-) T = 2 USEC  
13167  
13168 ;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF (L) L / K3-5 ROTSHF H / K3-8 ROT (L) H  
13169  
13170 025256 012700 000343 T0343: MOV #0343,R0 ;LOAD R0 WITH TEST NO.  
13171 025262 013701 025306 MOV @#I0343,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
13172 025266 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
13173 025272 012704 125253 MOV #125253,R4 ;RESULT S / B = 125253  
13174 025276 012712 052525 R0343: MOV #52525,(R2) ;[DEST] = 52525  
13175 025302 000257 CCC ;CLEAR CODES  
13176 025304 000265 265 ;N:C = 0101  
13177  
13178 025306 006112 I0343: ROL (R2) ;TEST THE ROL  
13179  
13180 025310 100003 BPL E10343 ;N:C = 1010 ?  
13181 025312 001402 BEQ E10343  
13182 025314 102001 BVC E10343  
13183 025316 103002 BCC A0343  
13184  
13185 025320 104005 E10343: ERROR5 ;ROL FAILED TO ALTER THE CODES PROPERLY  
13186 025322 025276 R0343 ;ERROR LOOP RETURN  
13187 025324 020412 A0343: CMP R4,(R2) ;RESULT OK ?  
13188 025326 001403 BEQ 00343 ;BR IF YES  
13189  
13190 025330 011203 MOV (R2),R3 ;GET THE WAS DATA  
13191 025332 104000 E20343: ERROR ;ROL DELIVERED THE WRONG RESULT  
13192 025334 025276 R0343 ;ERROR LOOP RETURN  
13193  
13194 025336 000004 00343: SCOPE ;CALL SCOPE LOOP UTILITY  
13195
```

13196
13197
13198
13199
13200
13201
13202
13203
13204
13205
13206
13207
13208
13209
13210
13211
13212
13213
13214
13215
13216
13217
13218
13219
13220
13221
13222
13223
13224
13225
13226
13227
13228
13229
13230
13231
13232
13233
13234
13235
13236
13237
13238
13239
13240

025340 012700 000344
025344 013701 025364
025350 012702 067560
025354 005004
025356 005012
025360 000257
025362 000262
025364 006112
025366 100403
025370 001002
025372 102401
025374 103002
025376 104005
025400 025356
025402 020412
025404 001403
025406 011203
025410 104000
025412 025356
025414 000004

```
; *****  
; .SBTTL T0344 ROL DM1 TEST - <N:C> = 0010  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=LHLL :[211] D = 000000  
;CODES: [211] SPS=1, [367] SPS=3 / N:C =0100  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: K3-3 DM=1L / K3-6 ROTSHF (L) L / K3-5 ROTSHF H / K3-8 ROT (L) H  
T0344: MOV #0344,R0 ;LOAD R0 WITH TEST NO.  
MOV @#10344,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
CLR R4 ;RESULT S / B = 000000  
R0344: CLR (R2) ;[DEST] = 000000  
CCC ;CLEAR CODES  
SEV ;N:C = 0010  
I0344: ROL (R2) ;TEST THE ROL  
BMI E10344 ;N:C = 0100 ?  
BNE E10344  
BVS E10344  
BCC A0344  
E10344: ERROR5 ;ROL FAILED TO ALTER THE CODES PROPERLY  
R0344 ;ERROR LOOP RETURN  
A0344: CMP R4,(R2) ;RESULT OK ?  
BEQ 00344 ;BR IF YES  
E20344: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;ROL DELIVERED THE WRONG RESULT  
R0344 ;ERROR LOOP RETURN  
00344: SCOPE ;CALL SCOPE LOOP UTILITY
```

13241
13242
13243
13244
13245
13246
13247
13248
13249
13250
13251
13252
13253
13254
13255
13256
13257
13258
13259 025416 012700 000345
13260 025422 013701 025446
13261 025426 012702 067560
13262 025432 012704 100000
13263 025436 012712 077777
13264 025442 000257
13265 025444 000265
13266
13267 025446 005512
13268
13269 025450 100003
13270 025452 001402
13271 025454 102001
13272 025456 103002
13273
13274 025460 104005
13275 025462 025436
13276 025464 020412
13277 025466 001403
13278
13279 025470 011203
13280 025472 104000
13281 025474 025436
13282
13283 025476 000004
13284

; *****
; .SBTTL T0345 ADC DM1 TEST - <N:C> = 0101
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016
;EXEC: [220]ALUC=LLLLL :[211] D = 100000
;CODES: [211] SPS=1, [367] SPS=3 / N:C =1010
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K3-8 CIN00 L / K3-3 DM=1L / K3-4 ADC L

T0345: MOV #0345,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0345,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #M0345,R2 ;DEST ADDR = M0345
R0345: MOV #100000,R4 ;RESULT S / B = 100000
MOV #77777,(R2) ;[DEST] = 77777
CCC ;CLEAR CODES
265 ;N:C = 0101

I0345: ADC (R2) ;TEST THE ADC
BPL E10345 ;N:C = 1010 ?
BEQ E10345
BVC E10345
BCC A0345

E10345: ERROR5 ;ADC FAILED TO ALTER THE CODES PROPERLY
R0345 ;ERROR LOOP RETURN
A0345: CMP R4,(R2) ;RESULT OK ?
BEQ 00345 ;BR IF YES

E20345: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;ADC DELIVERED THE WRONG RESULT
R0345 ;ERROR LOOP RETURN

00345: SCOPE ;CALL SCOPE LOOP UTILITY

```

13285 ; *****
13286 ; .SBTTL T0346 ADC DM1 TEST - <N:C> = 1011
13287 ; *****
13288 ;MICROPROGRAMMING / LOGIC INFORMATION
13289 ;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8
13290 ;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016
13291 ;EXEC: [220]ALUC=LHLL :[211] D = 000000
13292 ;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0101
13293 ;SYNC: B05J2 (-) T = 2 USEC
13294 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=1L / K3-4 ADC L
13295
13296 T0346: MOV #0346,R0 ;LOAD R0 WITH TEST NO.
13297 MOV @#I0346,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13298 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
13299 CLR R4 ;RESULT S / B = 000000
13300 R0346: MOV #-1,(R2) ;[DEST] = 177777
13301 CCC ;CLEAR CODES
13302 273 ;N:C = 1011
13303
13304 I0346: ADC (R2) ;TEST THE ADC
13305 BMI E10346 ;N:C = 0101 ?
13306 BNE E10346
13307 BVS E10346
13308 BCS A0346
13309
13310 E10346: ERROR5 ;ADC FAILED TO ALTER THE CODES PROPERLY
13311 R0346 ;ERROR LOOP RETURN
13312 A0346: CMP R4,(R2) ;RESULT OK ?
13313 BEQ 00346 ;BR IF YES
13314
13315 E20346: MOV (R2),R3 ;GET THE WAS DATA
13316 ERROR ;ADC DELIVERED THE WRONG RESULT
13317 R0346 ;ERROR LOOP RETURN
13318
13319 00346: SCOPE ;CALL SCOPE LOOP UTILITY
13320
13321
13322
13323
13324
13325
13326
13327

```

```

025500 012700 000346
025504 013701 025526
025510 012702 067560
025514 005004
025516 012712 177777
025522 000257
025524 000273
025526 005512
025530 100403
025532 001002
025534 102401
025536 103402
025540 104005
025542 025516
025544 020412
025546 001403
025550 011203
025552 104000
025554 025516
025556 000004

```

13328
13329
13330
13331
13332
13333
13334
13335
13336
13337
13338
13339
13340
13341
13342
13343
13344
13345
13346 025560 012700 000347
13347 025564 013701 025610
13348 025570 012702 067560
13349 025574 012704 177777
13350 025600 012712 177777
13351 025604 000257
13352 025606 000272
13353
13354 025610 005512
13355
13356 025612 100003
13357 025614 001402
13358 025616 102401
13359 025620 103002
13360
13361 025622 104005
13362 025624 025600
13363 025626 020412
13364 025630 001403
13365
13366 025632 011203
13367 025634 104000
13368 025636 025600
13369
13370 025640 000004
13371

```
; *****  
; .SBTTL T0347 ADC DM1 TEST - <N:C> = 1010  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=LHLL :[211] D = 177777  
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 1000  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: / K3-3 DM=1 / K3-4 ADC L  
T0347: MOV #0347,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0347,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #-1,R4 ;RESULT S / B = 177777  
R0347: MOV #-1,(R2) ;[DEST] = 177777  
CCC ;CLEAR CODES  
272 ;N:C = 1010  
I0347: ADC (R2) ;TEST THE ADC  
BPL E10347 ;N:C = 1000 ?  
BEQ E10347  
BVS E10347  
BCC A0347  
E10347: ERROR5 ;ADC FAILED TO ALTER THE CODES PROPERLY  
R0347 ;ERROR LOOP RETURN  
A0347: CMP R4,(R2) ;RESULT OK ?  
BEQ 00347 ;BR IF YES  
E20347: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;ADC DELIVERED THE WRONG RESULT  
R0347 ;ERROR LOOP RETURN  
00347: SCOPE ;CALL SCOPE LOOP UTILITY
```

13372
13373
13374
13375
13376
13377
13378
13379
13380
13381
13382
13383
13384
13385
13386
13387
13388
13389
13390
13391
13392
13393
13394
13395
13396
13397
13398
13399
13400
13401
13402
13403
13404
13405
13406
13407
13408
13409
13410
13411
13412
13413
13414
13415

; *****
; .SBTTL T0350 SBC DM1 TEST - <N:C> = 1011
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016
;EXEC: [220]ALUC=LHHH :[211] D = 000000
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0100
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K3-3 DM=1L / K3-4 SBC L

T0350: MOV #0350,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0350,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
R0350: MOV #1,(R2) ;[DEST0351
CCC ;CLEAR CODES
273 ;N:C = 1011
I0350: SBC (R2) ;TEST THE SBC
;N:C = 0100 ?
BMI E10350
BNE E10350
BVS E10350
BCC A0350
E10350: ERROR5 ;SBC FAILED TO ALTER THE CODES PROPERLY
R0350 ;ERROR LOOP RETURN
A0350: CMP R4,(R2) ;RESULT OK ?
BEQ 00350 ;BR IF YES
E20350: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;SBC DELIVERED THE WRONG RESULT
R0350 ;ERROR LOOP RETURN
00350: SCOPE ;CALL SCOPE LOOP UTILITY

13416
13417
13418
13419
13420
13421
13422
13423
13424
13425
13426
13427
13428
13429
13430
13431
13432
13433
13434
13435
13436
13437
13438
13439
13440
13441
13442
13443
13444
13445
13446
13447
13448
13449
13450
13451
13452
13453
13454
13455
13456
13457
13458
13459

025722 012700 000351
025726 013701 025752
025732 012702 067560
025736 012704 077777
025742 012712 100000
025746 000257
025750 000265

025752 005612

025754 100403
025756 001402
025760 102001
025762 103002

025764 104005
025766 025742
025770 020412
025772 001403

025774 011203
025776 104000
026000 025742

026002 000004

```
; *****  
; .SBTTL T0351 SBC DM1 TEST - <N:C> = 0101  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=LHLL :[211] D = 077777  
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0010  
;SYNC: B05J2 (-) T = 2 USEC  
;KEY SIG: K3-3 DM=1L / K3-4 SBC L  
  
T0351: MOV #0351,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0351,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #M0351,R2 ;DEST ADDR = M0351  
MOV #077777,R4 ;RESULT S / B = 077777  
R0351: MOV #100000,(R2) ;[DEST] = 100000  
CCC ;CLEAR CODES  
265 ;N:C = 0101  
  
I0351: SBC (R2) ;TEST THE SBC  
  
BMI E10351 ;N:C = 0010 ?  
BEQ E10351  
BVC E10351  
BCC A0351  
  
E10351: ERROR5 ;SBC FAILED TO ALTER THE CODES PROPERLY  
R0351 ;ERROR LOOP RETURN  
A0351: CMP R4,(R2) ;RESULT OK ?  
BEQ 00351 ;BR IF YES  
  
MOV (R2),R3 ;GET THE WAS DATA  
E20351: ERROR ;SBC DELIVERED THE WRONG RESULT  
R0351 ;ERROR LOOP RETURN  
  
00351: SCOPE ;CALL SCOPE LOOP UTILITY
```

13460
13461
13462
13463
13464
13465
13466
13467
13468
13469
13470
13471
13472
13473
13474
13475
13476
13477
13478 026004 012700 000352
13479 026010 013701 026034
13480 026014 012702 067560
13481 026020 012704 000001
13482 026024 012712 000001
13483 026030 000257
13484 026032 000276
13485
13486 026034 005612
13487
13488 026036 100403
13489 026040 001402
13490 026042 102401
13491 026044 103002
13492
13493 026046 104005
13494 026050 026024
13495 026052 020112
13496 026054 001403
13497
13498 026056 011203
13499 026060 104000
13500 026062 026024
13501
13502 026064 000004

```
; *****  
; .SBTTL T0352 SBC DM1 TEST - <N:C> = 1110  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=LLLLL :[211] D = 000001  
;CODES: [211] SPS=1, [367] SPS=3 / N:C 0001  
;SYNC: B05J2 (-) T = 0001  
;KEY SIG: K3-3 DM=1L / K3-4 SBC L  
  
T0352: MOV #0352,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0352,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #M0352,R2 ;DEST ADDR = M0352  
MOV #1,R4 ;RESULT S / B = 1  
R0352: MOV #1,(R2) ;[DEST] = 1  
CCC ;CLEAR CODES  
276 ;N:C = 1110  
  
I0352: SBC (R2) ;TEST THE SBC  
  
BMI E10352 ;N:C = 0000 ?  
BEQ E10352  
BVS E10352  
BCC A0352  
  
E10352: ERROR5 ;SBC FAILED TO ALTER THE CODES PROPERLY  
R0352 ;ERROR LOOP RETURN  
A0352: CMP R4,(R2) ;RESULT OK ?  
BEQ 00352 ;BR IF YES  
  
MOV (R2),R3 ;GET THE WAS DATA  
E20352: ERROR ;SBC DELIVERED THE WRONG RESULT  
R0352 ;ERROR LOOP RETURN  
  
00352: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

13503 ; *****
13504 ; .SBTTL T0353 SBC DM1 TEST - <N:C> = 0111
13505 ; *****
13506
13507 ;MICROPROGRAMMING / LOGIC INFORMATION
13508
13509 ;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8
13510
13511 ;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016
13512
13513 ;EXEC: [220]ALUC=LHLL :[211] D = 177777
13514
13515 ;CODES: [211] SPS=1, [367] SPS=3 / N:C = 1001
13516
13517 ;SYNC: B05J2 (-) T = 2 USEC
13518
13519 ;KEY SIG: K3-3 DM=1L / K3-4 SBC L
13520
13521 026066 012700 000353 T0353: MOV #0353,R0 ;LOAD R0 WITH TEST NO.
13522 026072 013701 026114 MOV @#I0353,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13523 026076 012702 067560 MOV #M0353,R2 ;DEST ADDR = M0353
13524 026102 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777
13525 026106 005012 R0353: CLR (R2) ;[DEST] = 000000
13526 026110 000257 CCC ;CLEAR CODES
13527 026112 000267 267 ;N:C = 0111
13528
13529 026114 005612 I0353: SBC (R2) ;TEST THE SBC
13530
13531 026116 100003 BPL E10353 ;N:C = 1001 ?
13532 026120 001402 BEQ E10353
13533 026122 102401 BVS E10353
13534 026124 103402 BCS A0353
13535
13536 026126 104005 E10353: ERROR5 ;SBC FAILED TO ALTER THE CODES PROPERLY
13537 026130 026106 R0353 ;ERROR LOOP RETURN
13538 026132 020412 A0353: CMP R4,(R2) ;RESULT OK ?
13539 026134 001403 BEQ 00353 ;BR IF YES
13540
13541 026136 011203 MOV (R2),R3 ;GET THE WAS DATA
13542 026140 104000 E20353: ERROR ;SBC DELIVERED THE WRONG RESULT
13543 026142 026106 R0353 ;ERROR LOOP RETURN
13544
13545 026144 000004 00353: SCOPE ;CALL SCOPE LOOP UTILITY
13546
    
```

13547 : *****
13548 : .SBTTL T0354 NEGB - MODE 0 TEST - <N:C> = 0110
13549 : *****
13550

:MICROPROGRAMMING / LOGIC INFORMATION

13551
13552
13553 :ROM SEQ: [105,372,361,001] FC 1,7,8
13554
13555 :ACT BUTS: 37[004]100,105 / 31[105]360,361 / 27[372]000,001
13556
13557 :EXEC: [372]ALUC=LLHHL :[361]D= 000376
13558
13559 :CODES: [361]SPS=3 / N:C=1001
13560
13561 :SYNC: B05J2 (-) T= 1 USEC
13562
13563 :KEY SIG: K3-4 NEG L / K3-4 OVLAP INSTR H / K3-8 CIN00 L
13564 :K3-3 DM=0 L / K3-6 BYTE INSTR H
13565

13566 026146 012700 000354
13567 026152 013701 026176
13568 026156 012702 177703
13569 026162 012704 177776
13570 026166 012703 177402
13571 026172 000257
13572 026174 000266
13573

T0354: MOV #0354,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0354,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
MOV #177776,R4 ;RESULT S / B = 376 (LO BYTE)
R0354: MOV #177402,R3 ;[DEST] = 177402
CCC ;CLEAR FLAGS
266 ;N:C = 0110

13574 026176 105403
13575
13576 026200 100003
13577 026202 001402
13578 026204 102401
13579 026206 103402
13580

I0354: NEGB R3 ;TEST THE NEGB
BPL E10354 ;N:C = 1001
BEQ E10354
BVS E10354
BCS A0354

13581 026210 104000
13582 026212 026166
13583
13584 026214 020403
13585 026216 001402
13586

E10354: ERROR ;NEGB FAILED TO ALTER CODES PROPERLY
R0354 ;ERROR LOOP RETURN ADDRESS
A0354: CMP R4,R3 ;CORRECT RESULT ?
BEQ 00354 ;BR IF YES

13587 026220 104000
13588 026222 026166
13589
13590 026224 000004

E20354: ERROR ;NEGB DELIVERED THE WRONG RESULT
R0354 ;ERROR LOOP RETURN ADDRESS
00354: SCOPE ;CALL THE SCOPE LOOP UTILITY

13591
13592
13593
13594
13595
13596
13597
13598
13599
13600
13601
13602
13603
13604
13605
13606
13607
13608
13609
13610 026226 012700 000355
13611 026232 013701 026256
13612 026236 012702 177703
13613 026242 012704 177400
13614 026246 012703 177400
13615 026252 000257
13616 026254 000263
13617
13618 026256 105403
13619
13620 026260 100403
13621 026262 001002
13622 026264 102401
13623 026266 103002
13624
13625 026270 104000
13626 026272 026246
13627
13628 026274 020403
13629 026276 001402
13630
13631 026300 104000
13632 026302 026246
13633
13634 026304 000004
13635

```
; *****  
; .SBTTL T0355 NEGB - MODE 0 TEST - <N:C> = 0011  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [105,372,361,001] FC 1,7,8  
;ACT BUTS: 37[004]100,105 / 31[105]360,361 / 27[372]000,001  
;EXEC: [372]ALUC=LLHHL :[361]D=000400  
;CODES: [361]SPS=3 / N:C=0100  
;SYNC: B05J2 (-) T= 1 USEC  
;KEY SIG: K3-4 NEG L / K3-4 OVLAP INSTR H / K3-8 CIN00 L  
;K3-3 DM=0 L / K3-6 BYTE INSTR H  
T0355: MOV #0355,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0355,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #177400,R4 ;RESULT S / B = 000 (LO BYTE)  
R0355: MOV #177400,R3 ;[DEST] = 177400  
CCC ;CLEAR FLAGS  
263 ;N:C = 0011  
I0355: NEGB R3 ;TEST THE NEGB  
BMI E10355 ;N:C = 0100  
BNE E10355  
BVS E10355  
BCC A0355  
E10355: ERROR ;NEGB FAILED TO ALTER CODES PROPERLY  
R0355 ;ERROR LOOP RETURN ADDRESS  
A0355: CMP R4,R3 ;CORRECT RESULT ?  
BEQ 00355 ;BR IF YES  
E20355: ERROR ;NEGB DELIVERED THE WRONG RESULT  
R0355 ;ERROR LOOP RETURN ADDRESS  
00355: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

13636
13637
13638
13639
13640
13641
13642
13643
13644
13645
13646
13647
13648
13649
13650
13651
13652
13653
13654
13655 026306 012700 000356
13656 026312 013701 026336
13657 026316 012702 177703
13658 026322 012704 177600
13659 026326 012703 177600
13660 026332 000257
13661 026334 000275
13662
13663 026336 105403
13664
13665 026340 100003
13666 026342 001402
13667 026344 102001
13668 026346 103402
13669
13670 026350 104000
13671 026352 026326
13672
13673 026354 020403
13674 026356 001402
13675
13676 026360 104000
13677 026362 026326
13678
13679 026364 000004
13680

```
; *****  
; .SBTTL T0356 NEGB - MODE 0 TEST - <N:C> = 1101  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [105,372,361,001] FC 1,7,8  
;ACT BUTS: 37[004]100,105 / 31[105]360,361 / 27[372]000,001  
;EXEC: [372]ALUC=LLHHL :[361]D=000200  
;CODES: [361]SPS=3 / N:C=1011  
;SYNC: B05J2 (-) T= 1 USEC  
;KEY SIG: K3-4 NEG L / K3-4 OVLAP INSTR H / K3-8 CIN00 L  
;K3-3 DM=0 L / K3-6 BYTE INSTR H  
T0356: MOV #0356,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0356,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #177600,R4 ;RESULT S / B = 200 (LO BYTE)  
R0356: MOV #177600,R3 ;[DEST] = 177600  
CCC ;CLEAR FLAGS  
275 ;N:C = 1101  
I0356: NEGB R3 ;TEST THE NEGB  
BPL E10356 ;N:C = 1011  
BEQ E10356  
BVC E10356  
BCS A0356  
E10356: ERROR ;NEGB FAILED TO ALTER CODES PROPERLY  
R0356 ;ERROR LOOP RETURN ADDRESS  
A0356: CMP R4,R3 ;CORRECT RESULT ?  
BEQ 00356 ;BR IF YES  
E20356: ERROR ;NEGB DELIVERED THE WRONG RESULT  
R0356 ;ERROR LOOP RETURN ADDRESS  
00356: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

13681
13682
13683
13684
13685
13686
13687
13688
13689
13690
13691
13692
13693
13694
13695
13696
13697
13698
13699 026366 012700 000357
13700 026372 013701 026416
13701 026376 012702 177703
13702 026402 012704 177400
13703 026406 012703 177777
13704 026412 000257
13705 026414 000273
13706
13707 026416 105003
13708
13709 026420 100403
13710 026422 001002
13711 026424 102401
13712 026426 103002
13713
13714 026430 104000
13715 026432 026406
13716
13717 026434 020403
13718 026436 001402
13719
13720 026440 104000
13721 026442 026406
13722
13723 026444 000004

; *****
; .SBTTL T0357 CLR B - MODE 0 TEST - <N:C> = 1011
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [104,373,361,001] FC 1,7,8
;ACT BUTS: 37[004]100,104 / 31[105]360,361 / 27[372]000,001
;EXEC: [104]ALUC=HLLHH :[373]D=000000
;CODES: [373]SPS=1,[361]SPS=3 / N:C=0100
;SYNC: B05J2 (-) T= 1 USEC
;KEY SIG: K3-4 CLR L / K3-6 BYTE INSTR H / K3-4 OVLAP INSTR H / K3-3 DM=0 L

T0357: MOV #0357,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0357,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
MOV #177400,R4 ;RESULT S / B = 000 (LO BYTE)
R0357: MOV #-1,R3 ;[DEST] = 177777
CCC ;CLEAR FLAGS
273 ;N:C = 1011

I0357: CLR B R3 ;TEST THE CLR B
BMI E10357 ;N:C = 0100 ?
BNE E10357
BVS E10357
BCC A0357

E10357: ERROR ;CLR B FAILED TO SET CODES PROPERLY
R0357 ;ERROR LOOP RETURN ADDRESS

A0357: CMP R4,R3 ;RESULT CORRECT ?
BEQ 00357 ;BR IF YES

E20357: ERROR ;CLR B DELIVERED THE WRONG RESULT
R0357 ;ERROR LOOP RETURN ADDRESS

00357: SCOPE ;CALL THE SCOPE LOOP UTILITY

13724
13725
13726
13727
13728
13729
13730
13731
13732
13733
13734
13735
13736
13737
13738
13739
13740
13741
13742
13743
13744
13745
13746
13747
13748
13749
13750
13751
13752
13753
13754
13755
13756
13757
13758
13759
13760
13761
13762
13763
13764
13765
13766
13767
13768

026446 012700 000360
026452 013701 026476
026456 012702 177703
026462 012704 177400
026466 012703 177777
026472 000257
026474 000264

026476 105003

026500 100403
026502 001002
026504 102401
026506 103002

026510 104000
026512 026466

026514 020403
026516 001402

026520 104000
026522 026466

026524 000004

: *****
: .SBTTL T0360 CLRB - MODE 0 TEST - <N:C> = 0100
: *****
:MICROPROGRAMMING / LOGIC INFORMATION
:ROM SEQ: [104,373,361,001] FC 1,7,8
:ACT BUTS: 37[004]100,104 / 31[360,361 / 27[373]000,001
:EXEC: [104]ALUC=HLLHH :[373]D=000000
:CODES: [373]SPS=1,[361]SPS=3 / N:C=0100
:SYNC: B05J2 (-) T= 1 USEC
:KEY SIG: K3-4 CLR L / K3-4 OVLAP INSTR H / K3-6 BYTE INSTR H / K3-8 CIN00 L
: K3-3 DM=0 L

T0360: MOV #0360,R0 ;LOAD R0 WITH TEST NO.
MOV @#10360,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
MOV #177400,R4 ;RESULT S / B = 000 (LO BYTE)
R0360: MOV #-1,R3 ;[DEST] = 177777
CCC ;CLEAR FLAGS
SEZ ;N:C = 0100

I0360: CLRB R3 ;TEST THE CLRB

BMI E10360 ;N:C = 0100 ?
BNE E10360
BVS E10360
BCC A0360

E10360: ERROR ;CLRB FAILED TO SET CODES PROPERLY
R0360 ;ERROR LOOP RETURN ADDRESS

A0360: CMP R4,R3 ;RESULT CORRECT ?
BEQ 00360 ;BR IF YES

E20360: ERROR ;CLRB DELIVERED THE WRONG RESULT
R0360 ;ERROR LOOP RETURN ADDRESS

00360: SCOPE ;CALL THE SCOPE LOOP UTILITY

13769
13770
13771
13772
13773
13774
13775
13776
13777
13778
13779
13780
13781
13782
13783
13784
13785
13786
13787 026526 012700 000361
13788 026532 013701 026562
13789 026536 012702 067561
13790 026542 012704 000377
13791 026546 012705 067560
13792 026552 010203
13793 026554 012715 177777
13794 026560 000257
13795
13796 026562 105023
13797
13798 026564 022703 067562
13799 026570 001402
13800
13801 026572 104005
13802 026574 026552
13803
13804 026576 020415
13805 026600 001403
13806
13807 026602 011503
13808 026604 104000
13809 026606 026552
13810
13811 026610 000004
13812

```
; *****  
; .SBTTL T0361 CLRB TEST - DM2 - ODD ADDRESS  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [162,260,267,237,270,222,253,075,374,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,162 / 33[260]220,237 / 34[237]220,222 / 16[374]016,016  
;EXEC: [222]ALUC=HLLHM :[375] D = 000000  
;CODES: [253] SP=1, [075] SP=3 / N:C = 0100  
;SYNC: B05J2 (-) T = 1.9 USEC  
;KEY SIG: K3-6 BYTE INSTR H / K1-6 BA00(1) H / K3-7 ODD BYTE H / K3-3 DM=2L  
T0361: MOV #0361,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0361,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  
R0361: MOV R2,R3 ;R3 CONTAINS DEST ADDR  
MOV #-1,(R5) ;[DEST] = 177777  
CCC ;SCOPE SYNC  
I0361: CLRB (R3)+ ;TEST THE CLRB  
CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED ?  
BEQ A0361 ;BR IF YES  
E10361: ERROR5 ;CLRB FAILED TO UPDATE DEST REG  
R0361 ;ERROR LOOP RETURN ADDRESS  
A0361: CMP R4,(R5) ;CORRECT RESULT ?  
BEQ 00361 ;BR IF YES  
MOV (R5),R3 ;GET THE WAS DATA  
E20361: ERROR ;CLRB DELIVERED WRONG RESULT  
R0361 ;ERROR LOOP RETURN ADDRESS  
00361: SCOPE ;CALL SCOPE LOOP UTILITY
```

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

026612 012700 000362
026616 013701 026646
026622 012702 067561
026626 012704 000377
026632 012705 067560
026636 010203
026640 012715 177777
026644 000257
026646 105013
026650 020415
026652 001403
026654 011503
026656 104000
026660 026636
026662 000004

: *****
: .SBTTL T0362 CLRB TEST - DM1 - ODD ADDRESS
: *****
:MICROPROGRAMMING / LOGIC INFORMATION
:ROM SEQ: [161,266,267,237,270,222,253,075,374,375,016] FC 1,3,9,8
:ACT BUTS: 37[004]100,161 / 33[266]220,237 / 34[237]220,222 / 16[374]016,016
:EXEC: [222]ALUC=HLLHM :[375] D = 000000
:CODES: [253] SPS=1, [075] SPS=3 / N:C = 0100
:SYNC: B05J2 (-) T = 1.9 USEC
:KEY SIG: K3-6 BYTE INSTR H / K1-6 BA00 (1) H / K3-7 ODD BYTE H / K3-3 DM=2L

T0362: MOV #0362,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0362,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
R0362: MOV R2,R3 ;R3 CONTAINS DEST ADDR
MOV #-1,(R5) ;[DEST] = 177777
CCC ;SCOPE SYNC
I0362: CLRB (R3) ;TEST THE CLRB
CMP R4,(R5) ;CORRECT RESULT ?
BEQ 00362 ;BR IF YES
E0362: MOV (R5),R3 ;GET THE WAS DATA
ERROR ;CLRB DELIVERED WRONG RESULT
R0362 ;ERROR LOOP RETURN ADDRESS
00362: SCOPE ;CALL SCOPE LOOP UTILITY

13850
13851
13852
13853
13854
13855
13856
13857
13858
13859
13860
13861
13862
13863
13864
13865
13866
13867
13868 026664 012790 000363
13869 026670 013701 026714
13870 026674 012702 067560
13871 026700 012704 177400
13872 026704 010203
13873 026706 012712 177777
13874 026712 000257
13875
13876 026714 105023
13877
13878 026716 022703 067561
13879 026722 001402
13880
13881 026724 104005
13882 026726 026704
13883
13884 026730 020412
13885 026732 001403
13886
13887 026734 011203
13888 026736 104000
13889 026740 026704
13890
13891 026742 000004

```

; *****
; .SBTTL T0363 CLRB TEST - DM2 - EVEN ADDRESS
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ:      [162,260,267,220,211,367,375,016] FC 1,3,9,8
;ACT BUTS:     37[004]100,162 / 33[260]220,220 / 16[367]016,016
;EXEC:         [220]JALUC=HLLHH :[211] D = 000000
;CODES:        [211] SPS=1, [367] SPS=3 / N:C = 0100
;SYNC:         B05J2 (-) T = 1.9 USEC
;KEY SIG:      K3-6 BYTE INSTR H / K3-3 DM=1L / K3-4 CLR L

T0363:  MOV    #0363,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0363,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV    #M0363,R2      ;DEST ADDR = M0363
        MOV    #177400,R4     ;RESULT S / B = 177400
R0363:  MOV    R2,R3           ;R3 CONTAINS DEST ADDR
        MOV    #-1,(R2)       ;[DEST] = 177777
        CCC                   ;SCOPE SYNC

I0363:  CLRB   (R3)+          ;TEST THE CLRB
        CMP    #M0363+1,R3    ;DID DEST REG GET INCREMENTED ?
        BEQ   A0363          ;BR IF YES

E10363: ERROR5
        R0363                ;CLRB FAILED TO UPDATE DEST REG
                                ;ERROR LOOP RETURN ADDRESS

A0363:  CMP    R4,(R2)        ;CORRECT RESULT ?
        BEQ   00363          ;BR IF YES

E20363:  MOV    (R2),R3       ;GET THE WAS DATA
        ERROR R0363          ;CLRB DELIVERED WRONG RESULT
                                ;ERROR LOOP RETURN ADDRESS

00363:  SCOPE                ;CALL SCOPE LOOP UTILITY

```

13892
13893
13894
13895
13896
13897
13898
13899
13900
13901
13902
13903
13904
13905
13906
13907
13908
13909
13910 026744 012700 000364
13911 026750 013701 026774
13912 026754 012702 067560
13913 026760 012704 177400
13914 026764 010203
13915 026766 012712 177777
13916 026772 000257
13917
13918 026774 105013
13919
13920 026776 020412
13921 027000 001403
13922
13923 027002 011203
13924 027004 104000
13925 027006 026764
13926
13927 027010 000004

```
; *****  
; .SBTTL T0364 CLRB TEST - DM1 - EVEN ADDRESS  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,220,211,367,375,016] FC 1,3,9,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,220 / 16[367]016,016  
;EXEC: [220]ALUC=HLLMH :[367] D = 000000  
;CODES: [211] SPS=1, [367] SPS=3 / N:C = 0100  
;SYNC: B05J2 (-) T = 1.8 USEC  
;KEY SIG: K3-6 BYTE INSTR H  
T0364: MOV #0364,R0 ;LOAD R0 WITH TEST NO.  
MOV @I0364,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #M0364,R2 ;DEST ADDR = M0364  
MOV #177400,R4 ;RESULT S / B = 177400  
R0364: MOV R2,R3 ;R3 CONTAINS DEST ADDR  
MOV #-1,(R ;[DEST] = 177777  
CCC ;SCOPE SYNC  
I0364: CLRB (R3) ;TEST THE CLRB  
CMP R4,(R2) ;CORRECT RESULT ?  
BEQ 00364 ;BR IF YES  
E0364: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;CLRB DELIVERED WRONG RESULT  
R0364 ;ERROR LOOP RETURN ADDRESS  
00364: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

13928 ; *****
13929 ; .SBTTL T0365 NEGB TEST - DM2 - ODD ADDRESS
13930 ; *****
13931 ;MICROPROGRAMMING / LOGIC INFORMATION
13932 ;ROM SEQ: [162,260,267,237,270,223,253,075,374,375,016] FC 1,3,9,8
13933 ;ACT BUTS: 37[004]100,162 / 33[260]220,237 / 34[237]220,223 / 16[374]016,016
13934 ;EXEC: [223]ALUC=LLHHL :[375] D = 000400
13935 ;CODES: [253] SPS=1, [075] SPS=3 / N:C = 0001
13936 ;SYNC: B05J2 (-) T = 1.9 USEC
13937 ;KEY SIG: K3-6 BYTE INSTR H / K1-6 BA00(1) H / K3-7 ODD BYTE H / K3-8 CIN00
13938
13939
13940
13941
13942
13943
13944
13945
13946 027012 012700 000365 T0365: MOV #0365,R0 ;LOAD R0 WITH TEST NO.
13947 027016 013701 027046 MOV @#I0365,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
13948 027022 012702 067561 MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0+1
13949 027026 012704 000777 MOV #777,R4 ;RESULT S / B = 777
13950 027032 012705 067560 MOV #MBUF0,R5 ;POINT R5 TO CHECK RESULT
13951 027036 010203 R0365: MOV R2,R3 ;R3 CONTAINS DEST ADDR
13952 027040 012715 177777 MOV #-1,(R5) ;[DEST] = 177777
13953 027044 000257 CCC ;SCOPE SYNC
13954
13955 027046 105423 I0365: NEGB (R3)+ ;TEST THE NEGB
13956
13957 027050 022703 067562 CMP #MBUF0+2,R3 ;DID DEST REG GET INCREMENTED ?
13958 027054 001402 BEQ A0365 ;BR IF YES
13959
13960 027056 104005 E10365: ERROR5 ;NEGB FAILED TO UPDATE DEST REG
13961 027060 027036 R0365 ;ERROR LOOP RETURN ADDRESS
13962
13963 027062 020415 A0365: CMP R4,(R5) ;CORRECT RESULT ?
13964 027064 001403 BEQ 00365 ;BR IF YES
13965
13966 027066 011503 E20365: MOV (R5),R3 ;GET THE WAS DATA
13967 027070 104000 ERROR ;NEGB DELIVERED WRONG RESULT
13968 027072 027036 R0365 ;ERROR LOOP RETURN ADDRESS
13969
13970 027074 000004 00365: SCOPE ;CALL SCOPE LOOP UTILITY
    
```

13971
13972
13973
13974
13975
13976
13977
13978
13979
13980
13981
13982
13983
13984
13985
13986
13987
13988
13989 027076 012700 000366
13990 027102 013701 027132
13991 027106 012702 067561
13992 027112 012704 000777
13993 027116 012705 067560
13994 027122 010203
13995 027124 012715 177777
13996 027130 000257
13997
13998 027132 105413
13999
14000 027134 020415
14001 027136 001403
14002
14003 027140 011503
14004 027142 104000
14005 027144 027122
14006
14007 027146 000004

```

; *****
; .SBTTL T0366 NEGB TEST - DM1 - ODD ADDRESS
; *****

```

:MICROPROGRAMMING / LOGIC INFORMATION

```

:ROM SEQ:      [161,266,267,237,270,223,253,075,374,375,016] FC 1,3,9,8
:ACT BUTS:     37[004]100,161 / 33[266]220,237 / 34[237]220,223 / 16[374]016,016
:EXEC:         [223]ALUC=LLHHL :[375] D = 000400
:CODES:        [253] SPS=1, [075] SPS=3 / N:C = 0001
:SYNC:         B05J2 (-) T = 1.9 USEC
:KEY SIG:      K3-6 BYTE INSTR H / K1-6 BA00 (1) H / K3-7 ODD BYTE H / K3-8 CIN00

```

```

T0366:  MOV    #0366,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0366,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
R0366:  MOV    R2,R3           ;R3 CONTAINS DEST ADDR
        MOV    #-1,(R5)        ;[DEST] = 177777
        CCC                    ;SCOPE SYNC

I0366:  NEGB   (R3)           ;TEST THE NEGB

        CMP    R4,(R5)         ;CORRECT RESULT ?
        BEQ   00366           ;BR IF YES

E0366:  MOV    (R5),R3         ;GET THE WAS DATA
        ERROR R0366           ;NEGB DELIVERED WRONG RESULT
        ERROR R0366           ;ERROR LOOP RETURN ADDRESS

00366:  SCOPE                    ;CALL SCOPE LOOP UTILITY

```

```

14008 ; *****
14009 ; .SBTTL T0367 NEGB TEST - DM2 - EVEN ADDRESS
14010 ; *****
14011
14012 ;MICROPROGRAMMING / LOGIC INFORMATION
14013
14014 ;ROM SEQ: [162,260,267,221,367,375,016] FC 1,3,9,8
14015
14016 ;ACT BUTS: 37[004]100,162 / 33[260]220,221 / 16[367]016,016
14017
14018 ;EXEC: [221]ALUC=LLHHL :[367] D = 000001
14019
14020 ;CODES: [367] SPS=3 / N:C = 0001
14021
14022 ;SYNC: B05J2 (-) T = 1.8 USEC
14023
14024 ;KEY SIG: K3-6 BYTE INSTR H / K3-8 CIN00 L / K3-4 NEG L / K3-3 DM=2L
14025
14026 027150 012700 000367 T0367: MOV #0367,R0 ;LOAD R0 WITH TEST NO.
14027 027154 013701 027200 MOV @#I0367,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
14028 027160 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
14029 027164 012704 177401 MOV #177401,R4 ;RESULT S / B = 177401
14030 027170 010203 R0367: MOV R2,R3 ;R3 CONTAINS DEST ADDR
14031 027172 012712 177777 MOV #-1,(R2) ;[DEST] = 177777
14032 027176 000257 CCC ;SCOPE SYNC
14033
14034 027200 105423 I0367: NEGB (R3)+ ;TEST THE NEGB
14035
14036 027202 022703 067561 CMP #MBUF0+1,R3 ;DID DEST REG GET INCREMENTED ?
14037 027206 001402 BEQ A0367 ;BR IF YES
14038
14039 027210 104005 E10367: ERROR5 ;NEGB FAILED TO UPDATE DEST REG
14040 027212 027170 R0367 ;ERROR LOOP RETURN ADDRESS
14041
14042 027214 020412 A0367: CMP R4,(R2) ;CORRECT RESULT ?
14043 027216 001403 BEQ 00367 ;BR IF YES
14044
14045 027220 011203 E20367: MOV (R2),R3 ;GET THE WAS DATA
14046 027222 104000 ERROR R0367 ;NEGB DELIVERED WRONG RESULT
14047 027224 027170 ;ERROR LOOP RETURN ADDRESS
14048
14049 027226 000004 00367: SCOPE ;CALL SCOPE LOOP UTILITY

```

```
14050 ; *****  
14051 ; .SBTTL T0370 NEGB TEST - DM1 - EVEN ADDRESS  
14052 ; *****  
14053  
14054 ;MICROPROGRAMMING / LOGIC INFORMATION  
14055  
14056 ;ROM SEQ: [161,266,267,221,367,375,016] FC 1,3,9,8  
14057  
14058 ;ACT BUTS: 37[004]100,161 / 33[266]220,221 / 16[367]016,016  
14059  
14060 ;EXEC: [221]JALUC=LLHHL :[367] D = 000001  
14061  
14062 ;CODES: [367] SPS=3 / N:C = 0001  
14063  
14064 ;SYNC: B05J2 (-) T = 1.8 USEC  
14065  
14066 ;KEY SIG: K3-6 BYTE INSTR H / K3-8 CIN00 L / K3-4 NEG L / K3-3 DM=2L  
14067  
14068 027230 012700 000370 T0370: MOV #0370,R0 ;LOAD R0 WITH TEST NO.  
14069 027234 013701 027260 MOV @#I0370,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
14070 027240 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
14071 027244 012704 177401 MOV #177401,R4 ;RESULT S / B = 177401  
14072 027250 010203 R0370: MOV R2,R3 ;R3 CONTAINS DEST ADDR  
14073 027252 012712 177777 MOV #-1,(R2) ;[DEST] = 177777  
14074 027256 000257 CCC ;SCOPE SYNC  
14075  
14076 027260 105413 I0370: NEGB (R3) ;TEST THE NEGB  
14077  
14078 027262 020412 CMP R4,(R2) ;CORRECT RESULT ?  
14079 027264 001403 BEQ 00370 ;BR IF YES  
14080  
14081 027266 011203 MOV (R2),R3 ;GET THE WAS DATA  
14082 027270 104000 E0370: ERROR ;NEGB DELIVERED WRONG RESULT  
14083 027272 027250 R0370 ;ERROR LOOP RETURN ADDRESS  
14084  
14085 027274 000004 00370: SCOPE ;CALL SCOPE LOOP UTILITY
```

14086
14087
14088
14089
14090
14091
14092
14093
14094
14095
14096
14097
14098
14099
14100
14101
14102
14103
14104
14105
14106
14107
14108
14109
14110
14111
14112
14113
14114
14115
14116
14117
14118
14119
14120
14121
14122
14123
14124
14125
14126
14127
14128
14129

027276 012700 000371
027302 013701 027330
027306 012702 177703
027312 005004
027314 012705 177777
027320 012703 000001
027324 000257
027326 000272
027330 060503
027332 100403
027334 001002
027336 102401
027340 103402
027342 104005
027344 027320
027346 020403
027350 001402
027352 104000
027354 027320
027356 000004

```
; *****  
; .SBTTL T0371 ADD TEST - SMO,DMO - <N:C> = 1010  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [102,364,300,001] FC 1,7,8  
;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001  
;EXEC: [364]ALUC=LHLLH :[360] D = 000000  
;CODES: [360] SPS=3 / N:C = 0101  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=0L / K3-3 DM=0L  
T0371: MOV #0371,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0371,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
CLR R4 ;RESULT S / B = 000000  
R0371: MOV #-1,R5 ;SRC OPR = 177777  
MOV #+1,R3 ;[DEST0372  
CCC ;CLEAR FLAGS  
272 ;N:C = 1010  
I0371: ADD R5,R3 ;TEST THE ADD  
BMI E10371 ;N:C = 0101  
BNE E10371  
BVS E10371  
BCS A0371  
E10371: ERROR5 ;ADD FAILED TO ALTER CODES PROPERLY  
R0371 ;ERROR LOOP RETURN ADDRESS  
A0371: CMP R4,R3 ;CORRECT RESULT ?  
BEQ 00371 ;BR IF YES  
E20371: ERROR ;ADD DELIVERED THE WRONG RESULT  
R0371 ;ERROR LOOP RETURN ADDRESS  
00371: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
14130 ; *****
14131 ; .SBTTL T0372 ADD TEST - SMO,DMO - <N:C> = 0101
14132 ; *****
14133 ;MICROPROGRAMMING / LOGIC INFORMATION
14134 ;ROM SEQ: [102,364,360,001] FC 1,7,8
14135 ;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001
14136 ;EXEC: [364]ALUC=LHLLH :[360] D = 100006
14137 ;CODES: [360] SPS=3 / N:C = 1010
14138 ;SYNC: B05J2 (-) T = 1 USEC
14139 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=0L / K3-3 DM=0L
14140
14141
14142
14143
14144
14145
14146
14147
14148 027360 012700 000372 T0372: MOV #0372,R0 ;LOAD R0 WITH TEST NO.
14149 027364 013701 027414 MOV @#10372,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
14150 027370 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
14151 027374 012704 100006 MOV #100006,R4 ;RESULT S / B = 100006
14152 027400 012705 077777 MOV #77777,R5 ;SRC OPR = 77777
14153 027404 012703 000007 R0372: MOV #7,R3 ;[DEST] = 7
14154 027410 000257 CCC ;CLEAR FLAGS
14155 027412 000265 265 ;N:C = 0101
14156
14157 027414 060503 I0372: ADD R5,R3 ;TEST THE ADD
14158
14159 027416 100003 BPL E10372 ;N:C = 1010
14160 027420 001402 BEQ E10372
14161 027422 102001 BVC E10372
14162 027424 103002 BCC A0372
14163
14164 027426 104005 E10372: ERROR5 ;ADD FAILED TO ALTER CODES PROPERLY
14165 027430 027404 R0372 ;ERROR LOOP RETURN ADDRESS
14166
14167 027432 020403 A0372: CMP R4,R3 ;CORRECT RESULT ?
14168 027434 001402 BEQ 00372 ;BR IF YES
14169
14170 027436 104000 E20372: ERROR ;ADD DELIVERED THE WRONG RESULT
14171 027440 027404 R0372 ;ERROR LOOP RETURN ADDRESS
14172
14173 027442 000004 00372: SCOPE ;CALL SCOPE LOOP UTILITY
```

14174
14175
14176
14177
14178
14179
14180
14181
14182
14183
14184
14185
14186
14187
14188
14189
14190
14191
14192
14193
14194
14195
14196
14197
14198
14199
14200
14201
14202
14203
14204
14205
14206
14207
14208
14209
14210
14211
14212
14213
14214
14215
14216

027444 012700 000373
027450 013701 027474
027454 012702 177703
027460 012704 067570
027464 012705 067544
027470 005003
027472 000257
027474 061503
027476 020403
027500 001402
027502 104000
027504 027464
027506 022705 067544
027512 001402
027514 104005
027516 027464
027520 000004

; *****
; .SBTTL T0373 ADD SM1,DMO TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,120,371,360,000] FC 1,2,8
;ACT BUTS: 37[004]100,141 / 35[247]120,120 / 31[120]360,360 / 27[371]000,000
;EXEC: [371]ALUC=LHLLH :[360] D = #DWTA
;CODES: [360] SPS=3
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=1L

T0373: MOV #0373,R0 ;LOAD R0 WITH THE TEST NO.
MOV @#I0373,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
MOV #DWTA,R4 ;RESULT S / B = #DWTA
R0373: MOV #ATA,R5 ;SOURCE ADDR = ATA
CLR R3 ;[DEST] = 0
CCI ;SCOPE SYNC
I0373: ADD (R5),R3 ;TEST THE ADD - SM1,DMO
CMP R4,R3 ;RESULT = #DWTA?
BEQ A0373 ;BR IF YES
E10373: ERROR R0373 ;ADD DELIVERED WRONG RESULT
;ERROR LOOP RETURN
A0373: CMP #ATA,R5 ;DID ADD CHANGE REG.
BEQ 00373 ;BR IF NOT
E20373: ERROR5 R0373 ;REG GOT MODIFIED
;ERROR LOOP RETURN
00373: SCOPE ;CALL SCOPE LOOP UTILITY

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

027522 012700 000374
027526 013701 027564
027532 032737 000200 066642
027540 001401
027542 000000
027544 012702 177703
027550 012704 067570
027554 012705 067544
027560 005003
027562 000257
027564 062503
027566 020403
027570 001402
027572 104000
027574 027554
027576 022705 067546
027602 001402
027604 104005
027606 027554
027610 000004

```
; *****  
      .SBTTL T0374 ADD SM2,DMO TEST  
; *****  
:MICROPROGRAMMING / LOGIC INFORMATION  
:ROM SEQ:      [142,240,250,120,371,360,000] FC 1,2,8  
:ACT BUTS:     37[004]100,142 / 35[240]120,120 / 31[120]360,360 / 27[371]000,000  
:EXEC:         [371]JALUC=LHLLH :[360] D = #DWTA  
:CODES:        [360] SPS=3  
:SYNC:         B05J2 (-) T = USEC  
:KEY SIG:      K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=2L / K5-5 BCON (1+2) H  
T0374:  MOV    #0374,R0          ;LOAD R0 WITH THE TEST NO.  
        MOV    @#I0374,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD  
        BIT    #200,@#BPTLOC    ;BREAKPOINT HALT SET ??  
        BEQ    .+4              ;BR IF NOT  
        HALT                    ;BREAK- DEPRESS CONTINUE TO RESTART  
        MOV    #177703,R2       ;DEST ADDR = R3  
        MOV    #DWTA,R4         ;RESULT S / B = #DWTA  
R0374:  MOV    #ATA,R5          ;SOURCE ADDR = ATA  
        CLR    R3               ;[DEST] = 0  
        CCC                    ;SCOPE SYNC  
I0374:  ADD    (R5)+,R3         ;TEST THE ADD - SM2,DMO  
        CMP    R4,R3            ;RESULT = #DWTA  
        BEQ    A0374           ;BR IF YES  
E10374: ERROR R0374           ;ADD DELIVERED WRONG RESULT  
        ;ERROR LOOP RETURN  
A0374:  CMP    #ATA+2,R5        ;DID ADD AUTO INCREMENT SOURCE REG?  
        BEQ    00374           ;BR IF YES  
E20374: ERROR5 R0374         ;ADD FAILED TO UPDATE SOURCE REG.  
        ;ERROR LOOP RETURN  
00374:  SCOPE                  ;CALL SCOPE LOOP UTILITY
```

```

14262 ; *****
14263 ; .SBTTL T0375 ADD SM3,DMO TEST
14264 ; *****
14265 ;MICROPROGRAMMING / LOGIC INFORMATION
14266 ;ROM SEQ: [143,245,246,247,250,120,371,360,000] FC 1,2,8
14267 ;ACT BUTS: 37[004]100,143 / 35[247]120,120 / 31[120]360,360 / 27[371]000,000
14268 ;EXEC: [371]ALUC=LHLLH :[360] D = #DWTA
14269 ;CODES: [360] SPS=3
14270 ;SYNC: B05J2 (-) T = 2.75 USEC
14271 ;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=2L / K5-5 BC01 H
14272
14273 T0375: MOV #0375,R0 ;LOAD R0 WITH THE TEST NO.
14274 MOV @#I0375,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
14275 MOV #177703,R2 ;DEST ADDR = R3
14276 MOV #DWTA,R4 ;RESULT S / B = #DWTA
14277 R0375: MOV #ATA+10,R5 ;R5 POINTS TO SOURCE ADDR
14278 MOV R4,@#MBUF0 ;[SOURCE] = #DWTA
14279 CLR R3 ;[DEST] = 0
14280 CCC ;SCOPE SYNC
14281
14282 I0375: ADD @(R5)+,R3 ;TEST THE ADD - SM3,DMO
14283
14284 CMP R4,@#MBUF0 ;RESULT = #DWTA?
14285 BEQ A0375 ;BR IF YES
14286
14287 E10375: ERROR R0375 ;ADD DELIVERED WRONG RESULT
14288 ;ERROR LOOP RETURN
14289
14290 A0375: CMP #ATA+12,R5 ;DID ADD AUTO INCREMENT SOURCE REG?
14291 BEQ 00375 ;BR IF YES
14292
14293 E20375: ERROR5 R0375 ;ADD FAILED TO UPDATE SOURCE REG.
14294 ;ERROR LOOP RETURN
14295
14296 00375: SCOPE ;CALL SCOPE LOOP UTILITY
14297
14280 027612 012700 000375
14281 027616 013701 027646
14282 027622 012702 177703
14283 027626 012704 067570
14284 027632 012705 067554
14285 027636 010437 067560
14286 027642 005003
14287 027644 000257
14288
14289 027646 063503
14290
14291 027650 020437 067560
14292 027654 001402
14293
14294 027656 104000
14295 027660 027632
14296
14297 027662 022705 067556
14298 027666 001402
14299
14300 027670 104005
14301 027672 027632
14302
14303 027674 000004
14304

```

```

14305 ; *****
14306 ; .SBTTL T0376 ADD SM4,DMO TEST
14307 ; *****
14308
14309 ;MICROPROGRAMMING / LOGIC INFORMATION
14310
14311 ;ROM SEQ: [144,240,250,120,371,360,000] FC 1,2,8
14312
14313 ;ACT BUTS: 37[004]100,144 / 35[240]120,120 / 31[120]360,360 / 27[371]000,000
14314
14315 ;EXEC: [371]ALUC=LHLLH :[360] D = #DWTA
14316
14317 ;CODES: [360] SPS=3
14318
14319 ;SYNC: B05J2 (-) T = 2 USEC
14320
14321 ;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=4L / K5-5 BCON (1+2) H
14322
14323 027676 012700 000376 T0376: MOV #0376,R0 ;LOAD R0 WITH THE TEST NO.
14324 027702 013701 027726 MOV @#I0376,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
14325 027706 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
14326 027712 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA
14327 027716 012705 067546 R0376: MOV #ATA+2,R5 ;SOURCE ADDR = ATA
14328 027722 005003 CLR R3 ;[DEST] = 0
14329 027724 000257 CCC ;SCOPE SYNC
14330
14331 027726 064503 I0376: ADD -(R5),R3 ;TEST THE ADD - SM4,DMO
14332
14333 027730 020403 CMP R4,R3 ;RESULT = #DWTA?
14334 027732 001402 BEQ A0376 ;BR IF YES
14335
14336 027734 104000 E10376: ERROR ;ADD DELIVERED WRONG RESULT
14337 027736 027716 R0376 ;ERROR LOOP RETURN
14338
14339 027740 022705 067544 A0376: CMP #ATA,R5 ;DID SOURCE REG GET DECREMENTED?
14340 027744 001402 BEQ A0376 ;BR IF YES
14341
14342 027746 104005 E20376: ERROR5 ;ADD FAILED TO UPDATE SOURCE REG
14343 027750 027716 R0376 ;ERROR LOOP RETURN
14344
14345 027752 000004 00376: SCOPE ;CALL SCOPE LOOP UTILITY
14346

```

```
14347 ; *****  
14348 ; .SBTTL T0377 ADD SM5,DMO TEST  
14349 ; *****  
14350  
14351 ;MICROPROGRAMMING / LOGIC INFORMATION  
14352  
14353 ;ROM SEQ: [145,245,246,247,250,120,371,360,000] FC 1,2,8  
14354  
14355 ;ACT BUTS: 37[004]100,145 / 35[247]120,120 / 31[120]360,360 / 27[371]000,000  
14356  
14357 ;EXEC: [371]ALUC=LHLLH :[360] D = #DWTA  
14358  
14359 ;CODES: [360] SPS=3  
14360  
14361 ;SYNC: B05J2 (-) T = 2.75 USEC  
14362  
14363 ;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=5L / K5-5 BC01 H  
14364  
14365 027754 012700 000377 T0377: MOV #0377,R0 ;LOAD R0 WITH THE TEST NO.  
14366 027760 013701 030010 MOV @#I0377,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
14367 027764 012702 177703 MOV #177703,R2 ;DEST ADDR = R3  
14368 027770 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA  
14369 027774 012705 067556 R0377: MOV #ATA+12,R5 ;R5 POINTS TO SOURCE ADDR  
14370 030000 010437 067560 MOV R4,@#MBUF0 ;[SOURCE] = #DWTA  
14371 030004 005003 CLR R3 ;[DEST] = 0  
14372 030006 000257 CCC ;SCOPE SYNC  
14373  
14374 030010 065503 I0377: ADD @-(R5),R3 ;TEST THE ADD - SM5,DMO  
14375  
14376 030012 020437 067560 CMP R4,@#MBUF0 ;RESULT = #DWTA?  
14377 030016 001402 BEQ A0377 ;BR IF YES  
14378  
14379 030020 104000 E10377: ERROR ;ADD DELIVERED WRONG RESULT  
14380 030022 027774 R0377 ;ERROR LOOP RETURN  
14381  
14382 030024 022705 067554 A0377: CMP #ATA+10,R5 ;DID ADD DECREMENT SOURCE REG?  
14383 030030 001402 BEQ 00377 ;BR IF YES  
14384  
14385 030032 104005 E20377: ERROR5 ;ADD FAILED TO UPDATE SOURCE REG.  
14386 030034 027774 R0377 ;ERROR LOOP RETURN  
14387  
14388 030036 000004 00377: SCOPE ;CALL SCOPE LOOP UTILITY  
14389
```

```

14390 ; *****
14391 ; .SBTTL T0400 ADD SM6,DMO TEST
14392 ; *****
14393
14394 ;MICROPROGRAMMING / LOGIC INFORMATION
14395
14396 ;ROM SEQ: [146,241,242,247,250,120,371,360,000] FC 1,2,8
14397
14398 ;ACT BUTS: 37[004]100,146 / 35[247]120,120 / 31[120]360,360 / 27[371]000,000
14399
14400 ;EXEC: [371]ALUC=LHLLH :[360] D = #MBUFO
14401
14402 ;CODES: [360] SPS=3
14403
14404 ;SYNC: B05J2 (-) T = 2.5 USEC
14405
14406 ;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=6L / K3-4 OVLAP CYCLE L
14407
14408 030040 012700 000400 T0400: MOV #0400,R0 ;LOAD R0 WITH THE TEST NO.
14409 030044 013701 030070 MOV @#10400,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
14410 030050 012703 177703 MOV #177703,R3 ;DEST ADDR = R3
14411 030054 012704 067560 MOV #MBUFO,R4 ;RESULT S / B = MBUFO
14412 030060 012705 067544 R0400: MOV #ATA,R5 ;BASE SOURCE ADDR = ATA
14413 030064 005003 CLR R3 ;[DEST] = 0
14414 030066 000257 CCC ;SCOPE SYNC
14415
14416 030070 066503 000010 I0400: ADD 10(R5),R3 ;TEST THE ADD - SM6,DMO
14417
14418 030074 020403 CMP R4,R3 ;RESULT =MBUFO?
14419 030076 001402 BEQ 00400 ;BR IF YES
14420
14421 030100 104000 E0400: ERROR ;ADD DELIVERED WRONG RESULT
14422 030102 030060 R0400 ;ERROR LOOP RETURN
14423
14424 030104 000004 O0400: SCOPE ;CALL SCOPE LOOP UTILITY
14425

```

14426
14427
14428
14429
14430
14431
14432
14433
14434
14435
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

```
: *****  
      .SBTTL T0401 ADD SM7,DMO TEST  
: *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ:      [147,243,244,245,246,247,250,120,371,360,000] FC 1,2,8  
;ACT BUTS:     37[004]100,147 / 35[247]120,120 / 31[120]360,360 / 27[371]000,000  
;EXEC:         [371]ALUC=LHLLH :[360] D = #DWTA  
;CODES:        [360] SPS=3  
;SYNC:         B05J2 (-) T = 3.5 USEC  
;KEY SIG:      K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=7L / K3-4 OVLAP CYCLE L  
T0401: MOV      #0401,R0          ;LOAD R0 WITH THE TEST NO.  
        MOV      @#I0401,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
        MOV      #177703,R2     ;DEST ADDR = R3  
        MOV      #DWTA,R4       ;RESULT S / B = #DWTA  
R0401:  MOV      #ATA,R5        ;BASE SOURCE ADDR = ATA  
        MOV      R4,@#MBUF0     ;[SOURCE] = #DWTA  
        CLR      R3            ;[DEST] = 0  
        CCC                     ;SCOPE SYNC  
I0401:  ADD      @10(R5),R3      ;TEST THE ADD - SM7,DMO  
        CMP      R4,R3          ;RESULT = #DWTA?  
        BEQ     00401          ;BR IF YES  
E0401:  ERROR   R0401          ;ADD DELIVERED WRONG RESULT  
        ;ERROR LOOP RETURN  
00401:  SCOPE                    ;CALL SCOPE LOOP UTILITY
```

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

030160 012700 000402
030164 013701 030210
030170 012702 067560
030174 012704 067570
030200 012705 067544
030204 005012
030206 000257
030210 061512
030212 020412
030214 001403
030216 011203
030220 104000
030222 030200
030224 000004

```
; *****  
; .SBTTL T0402 ADD SM1,DM1 TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,161,266,267,225,367,375,016] FC 1,2,3,8  
;ACT BUTS: 37[004]100,141 / 35[247]120,161 / 33[266]220,225 / 16[367]016,016  
;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA  
;CODES: [367] SPS=3  
;SYNC: B05J2 (-) T = 2.6 USEC  
;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=1L / K3-3 SM=1L  
T0402: MOV #0402,R0 ;LOAD R0 WITH THE TEST NO.  
MOV @#I0402,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
MOV #DWTA,R4 ;RESULT S / B = #DWTA  
R0402: MOV #ATA,R5 ;SOURCE ADDR = ATA  
CLR (R2) ;[DEST] = 0  
CCC ;SCOPE SYNC  
I0402: ADD (R5),(R2) ;TEST THE ADD - SM1,DM1  
CMP R4,(R2) ;RESULT = #DWTA?  
BEQ 00402 ;BR IF YES  
E0402: MOV (R2),R3 ;GET WAS DATA  
ERROR ;ADD DELIVERED WRONG RESULT  
R0402 ;ERROR LOOP RETURN  
00402: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
14500 ; *****  
14501 ; .SBTTL T0403 ADD SM2,DM1 TEST  
14502 ; *****  
14503 ;  
14504 ;MICROPROGRAMMING / LOGIC INFORMATION  
14505 ;  
14506 ;ROM SEQ: [142,240,250,161,266,267,225,367,375,016] FC 1,2,3,8  
14507 ;  
14508 ;ACT BUTS: 37[004]100,142 / 35[240]120,161 / 35[266]220,225 / 16[367]016,016  
14509 ;  
14510 ;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA  
14511 ;  
14512 ;CODES: [367] SPS=3  
14513 ;  
14514 ;SYNC: B05J2 (-) T = 2.7 USEC  
14515 ;  
14516 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=1L / K3-3 DM=1L  
14517 ;  
14518 030226 012700 000403 T0403: MOV #0403,R0 ;LOAD R0 WITH THE TEST NO.  
14519 030232 013701 030256 MOV @#I0403,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
14520 030236 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
14521 030242 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA  
14522 030246 012705 067544 R0403: MOV #ATA,R5 ;SOURCE ADDR = ATA  
14523 030252 005012 CLR (R2) ;[DEST] = 0  
14524 030254 000257 CCC ;SCOPE SYNC  
14525 ;  
14526 030256 062512 I0403: ADD (R5)+,(R2) ;TEST THE ADD - SM2,DM1  
14527 ;  
14528 030260 020412 CMP R4,(R2) ;RESULT = #DWTA?  
14529 030262 001403 BEQ 00403 ;BR IF YES  
14530 ;  
14531 030264 011203 MOV (R2),R3 ;GET WAS DATA  
14532 030266 104000 E0403: ERROR ;ADD DELIVERED WRONG RESULT  
14533 030270 030246 R0403 ;ERROR LOOP RETURN  
14534 ;  
14535 030272 000004 00403: SCOPE ;CALL SCOPE LOOP UTILITY  
14536
```

14537
14538
14539
14540
14541
14542
14543
14544
14545
14546
14547
14548
14549
14550
14551
14552
14553
14554
14555 030274 012700 000404
14556 030300 013701 030326
14557 030304 012702 067560
14558 030310 012704 067570
14559 030314 012705 067544
14560 030320 010203
14561 030322 005012
14562 030324 000257
14563
14564 030326 061523
14565
14566 030330 020412
14567 030332 001407
14568
14569 030334 010337 067564
14570 030340 011203
14571 030342 104000
14572 030344 030314
14573
14574 030346 013703 067564
14575 030352 022703 067562
14576 030356 001402
14577
14578 030360 104005
14579 030362 030314
14580
14581 030364 000004
14582

```
; *****  
; .SBTTL T0404 ADD SM1,DM2 TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,162,260,267,225,367,375,016] FC 1,2,3,8  
;ACT BUTS: 37[004]100,141 / 35[247]120,162 / 33[260]220,225 / 16[367]016,016  
;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA  
;CODES: [367] SPS=3  
;SYNC: B05J2 (-) T = 2.7 USEC  
;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=2L / K3-3 SM=1L  
T0404: MOV #0404,R0 ;LOAD R0 WITH THE TEST NO.  
MOV @#I0404,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #DWTA,R4 ;RESULT S / B = #DWTA  
R0404: MOV #ATA,R5 ;SOURCE ADDR = ATA  
MOV R2,R3 ;[R3] = DEST ADDR  
CLR (R2) ;[DEST] = 0  
CCC ;SCOPE SYNC  
I0404: ADD (R5),(R3)+ ;TEST THE ADD - SM1,DM2  
CMP R4,(R2) ;RESULT = #DWTA?  
BEQ A0404 ;BR IF YES  
MOV R3,@#MBUF1 ;SAVE UPDATED DEST ADDR  
MOV (R2),R3 ;GET WAS DATA  
E10404: ERROR ;ADD DELIVERED WRONG RESULT  
R0404 ;ERROR LOOP RETURN  
MOV @#MBUF1,R3 ;RESTORE UPDATED DEST ADDR  
A0404: CMP #MBUF0+2,R3 ;DID ADD INCREMENT DEST REG  
BEQ 00404 ;BR IF YES  
E20404: ERROR5 ;ADD FAILED TO UPDATE DEST REG  
R0404 ;ERROR LOOP RETURN  
00404: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
14583 ; *****  
14584 ; .SBTTL T0405 ADD SM2,DM2 TEST  
14585 ; *****  
14586  
14587 ;MICROPROGRAMMING / LOGIC INFORMATION  
14588  
14589 ;ROM SEQ: [142,240,250,162,260,267,225,367,375,016] FC 1,2,3,8  
14590  
14591 ;ACT BUTS: 37[004]100,142 / 35[240]120,162 / 33[260]220,225 / 16[367]016,016  
14592  
14593 ;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA  
14594  
14595 ;CODES: [367] SPS=3  
14596  
14597 ;SYNC: B05J2 (-) T = 2.7 USEC  
14598  
14599 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=2L / K3-3 DM=2L  
14600  
14601 030366 012700 000405 T0405: MOV #0405,R0 ;LOAD R0 WITH THE TEST NO.  
14602 030372 013701 030420 MOV @#I0405,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
14603 030376 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
14604 030402 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA  
14605 030406 012705 067544 R0405: MOV #ATA,R5 ;SOURCE ADDR = ATA  
14606 030412 010203 MOV R2,R3 ;[R3] = DEST ADDR  
14607 030414 005012 CLR (R2) ;[DEST] = 0  
14608 030416 000257 CCC ;SCOPE SYNC  
14609  
14610 030420 062523 I0405: ADD (R5)+,(R3)+ ;TEST THE ADD - SM2,DM2  
14611  
14612 030422 020412 CMP R4,(R2) ;RESULT = #DWTA  
14613 030424 001407 BEQ A0405 ;BR IF YES  
14614  
14615 030426 010337 067564 MOV R3,@#MBUF1 ;SAVE UPDATED DEST ADDR  
14616 030432 011203 MOV (R2),R3 ;GET WAS DATA  
14617 030434 104000 E10405: ERROR ;ADD DELIVERED WRONG RESULT  
14618 030436 030406 R0405 ;ERROR LOOP RETURN  
14619  
14620 030440 013703 067564 A0405: MOV @#MBUF1,R3 ;RESTORE UPDATED DEST ADDR  
14621 030444 022703 067562 CMP #MBUF0+2,R3 ;DID ADD INCREMENT DEST REG?  
14622 030450 001402 BEQ 00405 ;BR IF YES  
14623  
14624 030452 104005 E20405: ERROR5 ;ADD FAILED TO UPDATE DEST REG  
14625 030454 030406 R0405 ;ERROR LOOP RETURN  
14626  
14627 030456 000004 00405: SCOPE ;CALL SCOPE LOOP UTILITY  
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

030460 012700 000406
030464 013701 030514
030470 012702 067560
030474 012704 067570
030500 012705 067544
030504 012703 067554
030510 005012
030512 000257

030514 061533

030516 020412
030520 001407

030522 010337 067564
030526 011203
030530 104000
030532 030500

030534 013703 067564
030540 022703 067556
030544 001402

030546 104005
030550 030500

030552 000004

```
; *****  
; .SBTTL T0406 ADD SM1,DM3 TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,163,264,265,266,267,225,367,375,016] FC 1,2,3,8  
;ACT BUTS: 37[004]100,141 / 35[247]120,163 / 33[266]220,225 / 16[367]016,016  
;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA  
;CODES: [367] SPS=3  
;SYNC: B05J2 (-) T = 4.5 USEC  
;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=1L / K3-3 DM=3L  
T0406: MOV #0406,R0 ;LOAD R0 WITH THE TEST NO.  
MOV @#I0406,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #DWTA,R4 ;RESULT S / B = #DWTA  
R0406: MOV #ATA,R5 ;SOURCE ADDR = ATA  
MOV #ATA+10,R3 ;[R3] = ADDR OF DEST ADDR  
CLR (R2) ;[DEST] = 0  
CCC ;SCOPE SYNC  
  
I0406: ADD (R5),@(R3)+ ;TEST THE ADD - SM1,DM3  
  
CMP R4,(R2) ;RESULT = #DWTA?  
BEQ A0406 ;BR IF YES  
  
MOV R3,@#MBUF1 ;SAVE R3  
MOV (R2),R3 ;GET WAS DATA  
E10406: ERROR R0406 ;ADD DELIVERED WRONG RESULT  
;ERROR LOOP RETURN  
  
MOV @#MBUF1,R3 ;RESTORE R3  
A0406: CMP #ATA+12,R3 ;DID ADD INCREMENT DEST REG  
BEQ 00406 ;BR IF YES  
  
E20406: ERROR R0406 ;ADD FAILED TO UPDATE DEST REG  
;ERROR LOOP RETURN  
  
00406: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
14675 ; *****  
14676 ; .SBTTL T0407 ADD SM2,DM3 TEST  
14677 ; *****  
14678 ;MICROPROGRAMMING / LOGIC INFORMATION  
14679 ;ROM SEQ: [142,240,250,163,264,265,266,267,225,367,375,016] FC 1,2,3,8  
14680 ;ACT BUTS: 37[004]100,142 / 35[240]120,163 / 33[266]220,225 / 16[367]016,016  
14681 ;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA  
14682 ;CODES: [367] SPS=3  
14683 ;SYNC: B05J2 (-) T = 3.6 USEC  
14684 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=2L / K3-3 DM=3L  
14685  
14686  
14687  
14688  
14689  
14690  
14691  
14692  
14693 030554 012700 000407 T0407: MOV #0407,R0 ;LOAD R0 WITH THE TEST NO.  
14694 030560 013701 030610 MOV @#I0407,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
14695 030564 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
14696 030570 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA  
14697 030574 012705 067544 R0407: MOV #ATA,R5 ;SOURCE ADDR = ATA  
14698 030600 012703 067554 MOV #ATA+10,R3 ;[R3] = ADDR OF DEST ADDR  
14699 030604 005012 CLR (R2) ;[DEST] = 0  
14700 030606 000257 CCC ;SCOPE SYNC  
14701  
14702 030610 062533 I0407: ADD (R5)+,@(R3)+ ;TEST THE ADD - SM2,DM3  
14703  
14704 030612 020412 CMP R4,(R2) ;RESULT = #DWTA?  
14705 030614 001407 BEQ A0407 ;BR IF YES  
14706  
14707 030616 010337 067564 MOV R3,@#MBUF1 ;SAVE R3  
14708 030622 011203 MOV (R2),R3 ;GET WAS DATA  
14709 030624 104000 E10407: ERROR ;ADD DELIVERED WRONG RESULT  
14710 030626 030574 R0407 ;ERROR LOOP RETURN  
14711  
14712 030630 013703 067564 MOV @#MBUF1,R3 ;RESTORE R3  
14713 030634 022703 067556 A0407: CMP #ATA+12,R3 ;DID ADD INCREMENT DEST REG  
14714 030640 001402 BEQ 00407 ;BR IF YES  
14715  
14716 030642 104005 E20407: ERROR5 ;ADD FAILED TO UPDATE DEST REG  
14717 030644 030574 R0407 ;ERROR LOOP RETURN  
14718  
14719 030646 000004 00407: SCOPE ;CALL SCOPE LOOP UTILITY  
14720
```

```
14721 ; *****  
14722 ; .SBTTL T0410 ADD SM1,DM4 TEST  
14723 ; *****  
14724  
14725 ;MICROPROGRAMMING / LOGIC INFORMATION  
14726  
14727 ;ROM SEQ: [141,247,250,164,260,267,225,367,375,016] FC 1,2,3,8  
14728  
14729 ;ACT BUTS: 37[004]100,141 / 35[247]120,164 / 33[260]220,225 / 16[367]016,016  
14730  
14731 ;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA  
14732  
14733 ;CODES: [367] SPS=3  
14734  
14735 ;SYNC: B05J2 (-) T = 2.6 USEC  
14736  
14737 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=1L / K3-3 DM=4L  
14738  
14739 030650 012700 000410 T0410: MOV #0410,R0 ;LOAD R0 WITH THE TEST NO.  
14740 030654 013701 030704 MOV @#I0410,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
14741 030660 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
14742 030664 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA  
14743 030670 012705 067544 R0410: MOV #ATA,R5 ;SOURCE ADDR = ATA  
14744 030674 012703 067562 MOV #MBUF0+2,R3 ;R3 POINTS TO DEST ADDR +2  
14745 030700 005012 CLR (R2) ;[DEST] = 0  
14746 030702 000257 CCC ;SCOPE SYNC  
14747  
14748 030704 061543 I0410: ADD (R5),-(R3) ;TEST THE ADD - SM1,DM4  
14749  
14750 030706 020412 CMP R4,(R2) ;RESULT = #DWTA?  
14751 030710 001407 BEQ A0410 ;BR IF YES  
14752  
14753 030712 010337 067564 MOV R3,@#MBUF1 ;SAVE R3  
14754 030716 011203 MOV (R2),R3 ;GET WAS DATA  
14755 030720 104000 E10410: ERROR ;ADD DELIVERED WRONG RESULT  
14756 030722 030670 R0410 ;ERROR LOOP RETURN  
14757  
14758 030724 013703 067564 A0410: MOV @#MBUF1,R3 ;RESTORE R3  
14759 030730 020302 CMP R3,R2 ;DID ADD INCREMENT DEST REG?  
14760 030732 001402 BEQ 00410 ;BR IF YES  
14761  
14762 030734 104005 E20410: ERROR5 ;ADD FAILED TO UPDATE DEST REG.  
14763 030736 030670 R0410 ;ERROR LOOP RETURN  
14764  
14765 030740 000004 00410: SCOPE ;CALL SCOPE LOOP UTILITY  
14766
```

```
14767 ; *****  
14768 ; .SBTTL T0411 ADD SM2,DM4 TEST  
14769 ; *****  
14770  
14771 ;MICROPROGRAMMING / LOGIC INFORMATION  
14772  
14773 ;ROM SEQ: [142,240,250,164,260,267,225,367,375,016] FC 1,2,3,8  
14774  
14775 ;ACT BUTS: 37[004]100,142 / 35[240]120,164 / 33[260]220,225 / 16[367]016,016  
14776  
14777 ;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA  
14778  
14779 ;CODES: [367] SPS=3  
14780  
14781 ;SYNC: B05J2 (-) T = 2.6 USEC  
14782  
14783 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM0=2L / K3-3 DM=4L  
14784  
14785 030742 012700 000411 T0411: MOV #0411,R0 ;LOAD R0 WITH THE TEST NO.  
14786 030746 013701 030776 MOV @#I0411,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
14787 030752 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
14788 030756 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA  
14789 030762 012705 067544 R0411: MOV #ATA,R5 ;SOURCE ADDR = ATA  
14790 030766 012703 067562 MOV #MBUF0+2,R3 ;R3 POINTS TO DEST ADDR +2  
14791 030772 005012 CLR (R2) ;[DEST] = 0  
14792 030774 000257 CCC ;SCOPE SYNC  
14793  
14794 030776 061543 I0411: ADD (R5),-(R3) ;TEST THE ADD - SM2,DM4  
14795  
14796 031000 020412 CMP R4,(R2) ;RESULT = #DWTA?  
14797 031002 001407 BEQ A0411 ;BR IF YES  
14798  
14799 031004 010337 067564 MOV R3,@#MBUF1 ;SAVE R3  
14800 031010 011203 MOV (R2),R3 ;GET WAS DATA  
14801 031012 104000 E10411: ERROR ;ADD DELIVERED WRONG RESULT  
14802 031014 030762 R0411 ;ERROR LOOP RETURN  
14803  
14804 031016 013703 067564 MOV @#MBUF1,R3 ;RESTORE R3  
14805 031022 020302 A0411: CMP R3,R2 ;DID ADD INCREMENT DEST REG?  
14806 031024 001402 BEQ 00411 ;BR IF YES  
14807  
14808 031026 104005 E20411: ERROR5 ;ADD FAILED TO UPDATE DEST REG.  
14809 031030 030762 R0411 ;ERROR LOOP RETURN  
14810  
14811 031032 000004 00411: SCOPE ;CALL SCOPE LOOP UTILITY  
14812
```

```
14813 ; *****  
14814 ; .SBTTL T0412 ADD SM1,DM5 TEST  
14815 ; *****  
14816  
14817 ;MICROPROGRAMMING / LOGIC INFORMATION  
14818  
14819 ;ROM SEQ: [141,247,250,165,264,265,266,267,225,367,375,016] FC 1,2,3,8  
14820  
14821 ;ACT BUTS: 37[004]100,141 / 35[247]120,165 / 33[266]220,225 / 16[367]016,016  
14822  
14823 ;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA  
14824  
14825 ;CODES: [367] SPS=3  
14826  
14827 ;SYNC: B05J2 (-) T = 3.6 USEC  
14828  
14829 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=1L / K3-3 DM=5L  
14830  
14831 031034 012700 000412 T0412: MOV #0412,R0 ;LOAD R0 WITH THE TEST NO.  
14832 031040 013701 031070 MOV @#I0412,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
14833 031044 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
14834 031050 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA  
14835 031054 012705 067544 R0412: MOV #ATA,R5 ;SOURCE ADDR = ATA  
14836 031060 012703 067556 MOV #ATA+12,R3 ;R3 CONTAINS ADDR OF DEST ADDR PLUS 2  
14837 031064 005012 CLR (R2) ;[DEST] = 0  
14838 031066 000257 CCC ;SCOPE SYNC  
14839  
14840 031070 061553 I0412: ADD (R5),@-(R3) ;TEST THE ADD - SM1,DM5  
14841  
14842 031072 020412 CMP R4,(R2) ;RESULT = #DWTA?  
14843 031074 001407 BEQ A0412 ;BR IF YES  
14844  
14845 031076 010337 067564 MOV R3,@#MBUF1 ;SAVE R3  
14846 031102 011203 MOV (R2),R3 ;GET WAS DATA  
14847 031104 104000 E10412: ERROR ;ADD DELIVERED WRONG RESULT  
14848 031106 031054 R0412 ;ERROR LOOP RETURN  
14849  
14850 031110 013703 067564 A0412: MOV @#MBUF1,R3 ;RESTORE R3  
14851 031114 022703 067554 CMP #ATA+10,R3 ;DID ADD DECREMENT DEST REG?  
14852 031120 001402 BEQ 00412 ;BR IF YES  
14853  
14854 031122 104005 E20412: ERROR5 ;ADD FAILED TO UPDATE DEST REG.  
14855 031124 031054 R0412 ;ERROR LOOP RETURN  
14856  
14857 031126 000004 00412: SCOPE ;CALL SCOPE LOOP UTILITY  
14858
```

```
14859 ; *****
14860 ; .SBTTL T0413 ADD SM2,DM5 TEST
14861 ; *****
14862
14863 ;MICROPROGRAMMING / LOGIC INFORMATION
14864
14865 ;ROM SEQ: [142,240,250,165,264,265,266,267,225,367,375,016] FC 1,2,3,8
14866
14867 ;ACT BUTS: 37[004]100,142 / 35[240]120,165 / 33[266]220,225 / 16[367]016,016
14868
14869 ;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA
14870
14871 ;CODES: [367] SPS = 3
14872
14873 ;SYNC: B05J2 (-) T = 3.6 USEC
14874
14875 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=2L / K3-3 DM=5L
14876
14877 031130 012700 000413 T0413: MOV #0413,R0 ;LOAD R0 WITH THE TEST NO.
14878 031134 013701 031164 MOV @#I0413,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
14879 031140 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
14880 031144 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA
14881 031150 012705 067544 R0413: MOV #ATA,R5 ;SOURCE ADDR = ATA
14882 031154 012703 067556 MOV #ATA+12,R3 ;R3 CONTAINS ADDR OF DEST ADDR PLUS 2
14883 031160 005012 CLR (R2) ;[DEST] = 0
14884 031162 000257 CCC ;SCOPE SYNC
14885
14886 031164 062553 I0413: ADD (R5)+,@-(R3) ;TEST THE ADD - SM2,DM5
14887
14888 031166 020412 CMP R4,(R2) ;RESULT = #DWTA?
14889 031170 001407 BEQ A0413 ;BR IF YES
14890
14891 031172 010337 067564 MOV R3,@#MBUF1 ;SAVE R3
14892 031176 011203 MOV (R2),R3 ;GET WAS DATA
14893 031200 104000 E10413: ERROR ;ADD DELIVERED WRONG RESULT
14894 031202 031150 R0413 ;ERROR LOOP RETURN
14895
14896 031204 013703 067564 MOV @#MBUF1,R3 ;RESTORE R3
14897 031210 022703 067554 A0413: CMP #ATA+10,R3 ;DID ADD DECREMENT DEST REG?
14898 031214 001402 BEQ 00413 ;BR IF YES
14899
14900 031216 104005 E20413: ERROR5 ;ADD FAILED TO UPDATE DEST REG
14901 031220 031150 R0413 ;ERROR LOOP RETURN
14902
14903 031222 000004 00413: SCOPE ;CALL SCOPE LOOP UTILITY
14904
```

```
14905 ; *****
14906 ; .SBTTL T0414 ADD SM1,DM6 TEST
14907 ; *****
14908
14909 ;MICROPROGRAMMING / LOGIC INFORMATION
14910
14911 ;ROM SEQ: [141,247,250,167,261,262,266,267,225,367,375,016] FC 1,2,3,8
14912
14913 ;ACT BUTS: 37[004]100,141 / 35[247]120,167 / 17[167]262,262 / 33[266]220,225
14914 ; / 16[367]016,016
14915
14916 ;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA
14917
14918 ;CODES: [367] SPS=3
14919
14920 ;SYNC: B05J2 (-) T = 3.25 USEC
14921
14922 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=1L / K3-3 DM=6L
14923
14924 031224 012700 000414 T0414: MOV #0414,R0 ;LOAD R0 WITH THE TEST NO.
14925 031230 013701 031260 MOV @#I0414,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
14926 031234 012702 067564 MOV #MBUF0+4,R2 ;DEST ADDR = MBUF0+4
14927 031240 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA
14928 031244 012705 067544 R0414: MOV #ATA,R5 ;SOURCE ADDR = ATA
14929 031250 012703 067560 MOV #MBUF0,R3 ;[R3] = BASE DEST ADDR
14930 031254 005012 CLR (R2) ;[DEST] = 0
14931 031256 000257 CCC ;SCOPE SYNC
14932
14933 031260 061563 000004 I0414: ADD (R5),4(R3) ;TEST THE ADD - SM1,DM6
14934
14935 031264 020412 CMP R4,(R2) ;RESULT = #DWTA?
14936 031266 001403 BEQ 00414 ;BR IF YES
14937
14938 031270 011203 MOV (R2),R3 ;GET WAS DATA
14939 031272 104000 E0414: ERROR ;ADD DELIVERED WRONG RESULT
14940 031274 031244 R0414 ;ERROR LOOP RETURN
14941
14942 031276 000004 00414: SCOPE ;CALL SCOPE LOOP UTILITY
14943
```

```
14944 ; *****
14945 ; .SBTTL T0415 ADD SM2,DM6 TEST
14946 ; *****
14947 ;MICROPROGRAMMING / LOGIC INFORMATION
14948 ;ROM SEQ: [142,240,250,167,261,262,266,267,225,367,375,016] FC 1,2,3,8
14949 ;ACT BUTS: 37[004]100,142 / 35[240]120,167 / 17[167]262,262 / 33[266]220,225
14950 ; / 16[367]016,016
14951 ;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA
14952 ;CODES: [367] SPS=3
14953 ;SYNC: B05J2 (-) T = 3.25 USEC
14954 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=2L / K3-3 DM=6L
14955
14956 T0415: MOV #0415,R0 ;LOAD R0 WITH THE TEST NO.
14957 MOV @#I0415,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
14958 MOV #MBUF0+4,R2 ;DEST ADDR = MBUF0+4
14959 MOV #DWTA,R4 ;RESULT S / B = #DWTA
14960 R0415: MOV #ATA,R5 ;SOURCE ADDR = ATA
14961 MOV #MBUF0,R3 ;[R3] = BASE DEST ADDR
14962 CLR (R2) ;[DEST] = 0
14963 CCC ;SCOPE SYNC
14964
14965 I0415: ADD (R5)+,4(R3) ;TEST THE ADD - SM2,DM6
14966
14967 CMP R4,(R2) ;RESULT = #DWTA?
14968 BEQ 00415 ;BR IF YES
14969
14970 E0415: MOV (R2),R3 ;GET WAS DATA
14971 ERROR ;ADD DELIVERED WRONG RESULT
14972 R0415 ;ERROR LOOP RETURN
14973
14974 00415: SCOPE ;CALL SCOPE LOOP UTILITY
14975
14976
14977
14978
14979
14980
14981
14982
```

Address	Op1	Op2	Op3	Op4	Comment
14963	031300	012700	000415		
14964	031304	013701	031334		
14965	031310	012702	067564		
14966	031314	012704	067570		
14967	031320	012705	067544		
14968	031324	012703	067560		
14969	031330	005012			
14970	031332	000257			
14972	031334	062563	000004		
14974	031340	020412			
14975	031342	001403			
14977	031344	011203			
14978	031346	104000			
14979	031350	031320			
14981	031352	000004			

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

031354 012700 000416
031360 013701 031406
031364 012702 067560
031370 012704 067570
031374 012705 067544
031400 010503
031402 005012
031404 000257
031406 061573 000010
031412 020412
031414 001403
031416 011203
031420 104000
031422 031374
031424 000004

```
; *****  
; .SBTTL T0416 ADD SM1,DM7 TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,167,261,263,264,265,266,267,225,367,375,016] FC 1,2,3,8  
;ACT BUTS: 37[004]100,141 / 35[247]120,167 / 17[167]262,263 / 33[266]220,225  
; / 16[367]016,016  
;EXEC: [225]ALUC=LHLLH :[367] D = #DWTA  
;CODES: [367] SPS=3  
;SYNC: B05J2 (-) T = 4.25 USEC  
;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=1L / K3-3 DM=7L  
T0416: MOV #0416,R0 ;:LOAD R0 WITH THE TEST NO.  
MOV @#I0416,R1 ;:LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUFO,R2 ;:DEST ADDR = MBUFO  
MOV #DWTA,R4 ;:RESULT S / B = #DWTA  
R0416: MOV #ATA,R5 ;:SOURCE ADDR = ATA  
MOV R5,R3 ;:BASE DEST ADDR = ATA  
CLR (R2) ;:[DEST] = 0  
CCC ;:SCOPE SYNC  
I0416: ADD (R5),@10(R3) ;:TEST THE ADD - SM1,DM7  
CMP R4,(R2) ;:RESULT = #DWTA?  
BEQ 00416 ;:BR IF YES  
E0416: MOV (R2),R3 ;:GET WAS DATA  
ERROR ;:ADD DELIVERED WRONG RESULT  
R0416 ;:ERROR LOOP RETURN  
00416: SCOPE ;:CALL SCOPE LOOP UTILITY
```

```
15022 : *****
15023 : .SBTTL T0417 ADD SM2,DM7 TEST
15024 : *****
15025 :
15026 :MICROPROGRAMMING / LOGIC INFORMATION
15027 :
15028 :ROM SEQ: [142,240,250,167,261,263,264,265,266,267,225,367,375,016] FC 1,2,3,8
15029 :
15030 :ACT BUTS: 37[004]100,142 / 35[240]120,167 / 17[167]262,263 / 33[266]220,225
15031 : / 16[367]016,016
15032 :
15033 :EXEC: [225]ALUC=LHLLH :[367] D = #DWTA
15034 :
15035 :CODES: [367] SPS=3
15036 :
15037 :SYNC: B05J2 (-) T = 4.25 USEC
15038 :
15039 :KEY SIG: K3-3 ADD+SUB L / K3-3 SM=2L / K3-3 DM=7L
15040 :
15041 031426 012700 000417 T0417: MOV #0417,R0 ;LOAD R0 WITH THE TEST NO.
15042 031432 013701 031460 MOV @#I0417,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15043 031436 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
15044 031442 012704 067570 MOV #DWTA,R4 ;RESULT S / B = #DWTA
15045 031446 012705 067544 R0417: MOV #ATA,R5 ;SOURCE ADDR = ATA
15046 031452 010503 MOV R5,R3 ;BASE DEST ADDR = ATA
15047 031454 005012 CLR (R2) ;[DEST] = 0
15048 031456 000257 CCC ;SCOPE SYNC
15049 :
15050 031460 062573 000010 I0417: ADD (R5)+,@10(R3) ;TEST THE ADD - SM2,DM7
15051 :
15052 031464 020412 CMP R4,(R2) ;RESULT = #DWTA?
15053 031466 001403 BEQ 00417 ;BR IF YES
15054 :
15055 031470 011203 MOV (R2),R3 ;GET WAS DATA
15056 031472 104000 E0417: ERROR ;ADD DELIVERED WRONG RESULT
15057 031474 031446 R0417 ;ERROR LOOP RETURN
15058 :
15059 031476 000004 00417: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
15060 ; *****  
15061 ; .SBTTL T0420 'XOR RA,RB' TEST - A=B=000000 N:C=1010  
15062 ; *****  
15063 ;  
15064 ;MICROPROGRAMMING / LOGIC INFORMATION  
15065 ;  
15066 ;ROM SEQ: [102,364,360,001] FC 1,7,8  
15067 ;  
15068 ;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001  
15069 ;  
15070 ;EXEC: [364]ALUC=HLHHL :[360]D=000000  
15071 ;  
15072 ;CODES: [360] SPS=3 / N:C=0100  
15073 ;  
15074 ;SYNC: B05J2 (-) / T=1 USEC  
15075 ;  
15076 ;KEY SIG: K3-5 XOR L / K3-3 DM=0 L / K3-4 OVLAP INSTR H  
15077 ;  
15078 031500 012700 000420 T0420: MOV #0420,R0 ;LOAD R0 WITH TEST NO.  
15079 031504 013701 031524 MOV @#I0420,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
15080 031510 005004 CLR R4 ;RESULT AND MASK = 000000  
15081 031512 012702 177703 MOV #177703,R2 ;DEST ADDR = R3  
15082 031516 005003 R0420: CLR R3 ;[DEST] = 000000  
15083 031520 000257 CCC ;SCOPE SYNC  
15084 031522 000272 272 ;MAKE N:C=1010  
15085 ;  
15086 031524 074403 I0420: XOR R4,R3 ;TEST THE XOR  
15087 ;  
15088 031526 100403 BMI E10420 ;N:C=0100 ??  
15089 031530 001002 BNE E10420  
15090 031532 102401 BVS E10420  
15091 031534 103002 BCC A0420  
15092 ;  
15093 031536 104000 E10420: ERROR ;XOR FAILED TO SET FLAGS PROPERLY  
15094 031540 031516 R0420 ;ERROR LOOP RETURN ADDRESS  
15095 ;  
15096 031542 020403 A0420: CMP R4,R3 ;RESULT CORRECT?  
15097 031544 001402 BEQ 00420 ;BR IF YES  
15098 ;  
15099 031546 104000 E20420: ERROR ;XOR DELIVERED THE WRONG RESULT  
15100 031550 031516 R0420 ;ERROR LOOP RETURN ADDRESS  
15101 ;  
15102 031552 000004 00420: SCOPE ;CALL THE SCOPE LOOP UTILITY  
15103 ;
```

```
15104 ; *****
15105 ; .SBTTL T0421 'XOR RA,RB' TEST - A=B=177777 N:C=0101
15106 ; *****
15107
15108 ;MICROPROGRAMMING / LOGIC INFORMATION
15109
15110 ;ROM SEQ: [102,364,360,001] FC 1,7,8
15111
15112 ;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001
15113
15114 ;EXEC: [364]ALUC=HLHHL :[360]D=000000
15115
15116 ;CODES: [360] SPS=3 / N:C=0101
15117
15118 ;SYNC: B05J2 (-) / T=1 USEC
15119
15120 ;KEY SIG: K3-5 XOR L / K3-3 DM=0 L / K3-4 OVLAP INSTR H
15121
15122 031554 012700 000421 T0421: MOV #0421,R0 ;LOAD R0 WITH TEST NO.
15123 031560 013701 031604 MOV @#I0421,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15124 031564 005004 CLR R4 ;RESULT = 000000
15125 031566 012705 177777 MOV #-1,R5 ;MASK = 177777
15126 031572 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
15127 031576 010503 R0421: MOV R5,R3 ;[DEST]=177777
15128 031600 000257 CCC ;SCOPE SYNC
15129 031602 000265 265 ;MAKE N:C=0101
15130
15131 031604 074503 I0421: XOR R5,R3 ;TEST THE XOR
15132
15133 031606 100403 BMI E10421 ;N:C=0101 ??
15134 031610 001002 BNE E10421
15135 031612 102401 BVS E10421
15136 031614 103402 BCS A0421
15137
15138 031616 104000 E10421: ERROR ;XOR FAILED TO SET FLAGS PROPERLY
15139 031620 031576 R0421 ;ERROR LOOP RETURN ADDRESS
15140
15141 031622 020403 A0421: CMP R4,R3 ;RESULT CORRECT?
15142 031624 001402 BEQ 00421 ;BR IF YES
15143
15144 031626 104000 E20421: ERROR ;XOR DELIVERED THE WRONG RESULT
15145 031630 031576 R0421 ;ERROR LOOP RETURN ADDRESS
15146
15147 031632 000004 00421: SCOPE ;CALL THE SCOPE LOOP UTILITY
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

031634 012700 000422
031640 013701 031670
031644 012704 177777
031650 012702 177703
031654 012705 125252
031660 012703 052525
031664 000257
031666 000266
031670 074503
031672 100003
031674 001402
031676 102401
031700 103002
031702 104000
031704 031660
031706 020403
031710 001402
031712 104000
031714 031660
031716 000004

```
; *****  
; .SBTTL T0422 'XOR RA,RB' TEST - A=125252,B=052525 N:C=0110  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [102,364,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001  
;EXEC: [364]ALUC=HLHHL :[360]D=177777  
;CODES: [360] SPS=3 / N:C=1000  
;SYNC: B05J2 (-) / T=1 USEC  
;KEY SIG: K3-5 XOR L / K3-3 DM=0 L / K3-4 OVLAP INSTR H  
T0422: MOV #0422,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0422,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #-1,R4 ;RESULT S/B = 177777  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #125252,R5 ;MASK=125252  
R0422: MOV #052525,R3 ;[DEST] = 052525  
CCC ;SCOPE SYNC  
266 ;MAKE N:C=0110  
I0422: XOR R5,R3 ;TEST THE XOR  
BPL E10422 ;N:C=1000 ??  
BEQ E10422  
BVS E10422  
BCC A0422  
E10422: ERROR ;XOR FAILED TO SET FLAGS PROPERLY  
R0422 ;ERROR LOOP RETURN ADDRESS  
A0422: CMP R4,R3 ;RESULT CORRECT?  
BEQ 00422 ;BR IF YES  
E20422: ERROR ;XOR DELIVERED THE WRONG RESULT  
R0422 ;ERROR LOOP RETURN ADDRESS  
00422: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

```

15195 ; *****
15196 ; .SBTTL T0423 'XOR RA,RB' TEST - A=052525,B=125252 N:C=1001
15197 ; *****
15198
15199 ;MICROPROGRAMMING / LOGIC INFORMATION
15200
15201 ;ROM SEQ: [102,364,360,001] FC 1,7,8
15202
15203 ;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001
15204
15205 ;EXEC: [364]ALUC=HLHHL :[360]D=000000
15206
15207 ;CODES: [360] SPS=3 / N:C=1001
15208
15209 ;SYNC: B05J2 (-) / T=1 USEC
15210
15211 ;KEY SIG: K3-5 XOR L / K3-3 DM=0 L / K3-4 OVLAP INSTR H
15212
15213 031720 012700 000423 T0423: MOV #0423,R0 ;LOAD R0 WITH TEST NO.
15214 031724 013701 031754 MOV @#10423,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15215 031730 012704 177777 MOV #-1,R4 ;RESULT S/B = 177777
15216 031734 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
15217 031740 012705 052525 MOV #52525,R5 ;MASK=052525
15218 031744 012703 125252 R0423: MOV #125252,R3 ;[DEST] = 125252
15219 031750 000257 CCC ;SCOPE SYNC
15220 031752 000271 271 ;MAKE N:C=1001
15221
15222 031754 074503 I0423: XOR R5,R3 ;TEST THE XOR
15223
15224 031756 100003 BPL E10423 ;N:C=1001 ??
15225 031760 001402 BEQ E10423
15226 031762 102401 BVS E10423
15227 031764 103402 BCS A0423
15228
15229 031766 104000 E10423: ERROR ;XOR FAILED TO SET FLAGS PROPERLY
15230 031770 031744 R0423 ;ERROR LOOP RETURN ADDRESS
15231
15232 031772 020403 A0423: CMP R4,R3 ;RESULT CORRECT?
15233 031774 001402 BEQ 00423 ;BR IF YES
15234
15235 031776 104000 E20423: ERROR ;XOR DELIVERED THE WRONG RESULT
15236 032000 031744 R0423 ;ERROR LOOP RETURN ADDRESS
15237
15238 032002 000004 00423: SCOPE ;CALL THE SCOPE LOOP UTILITY
15239

```

```

15240 ; *****
15241 ; .SBTTL T0424 'XOR RA,(RB)' TEST - A=B=000000 N:C=1010
15242 ; *****
15243
15244 ;MICROPROGRAMMING / LOGIC INFORMATION
15245
15246 ;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8
15247
15248 ;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016
15249
15250 ;EXEC: [224]JALUC=HLHHL :[367]D=000000
15251
15252 ;CODES: [367] SPS=3 / N:C=0100
15253
15254 ;SYNC: B05J2 (-) / T=2.6 USEC
15255
15256 ;KEY SIG: K3-5 XOR L / K3-3 DM=1L
15257
15258 032004 012700 000424 T0424: MOV #0424,R0 ;LOAD R0 WITH TEST NO.
15259 032010 013701 032032 MOV @#I0424,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15260 032014 005004 CLR R4 ;RESULT S / B = 000000
15261 032016 005005 CLR R5 ;MASK = 000000
15262 032020 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
15263 032024 005012 R0424: CLR (R2) ;[DEST] = 000000
15264 032026 000257 CCC ;SCOPE SYNC
15265 032030 000272 272 ;MAKE N:C=1010
15266
15267 032032 074512 I0424: XOR R5,(R2) ;TEST THE XOR
15268
15269 032034 100403 BMI E10424 ;N:C = 0100 ??
15270 032036 001002 BNE E10424
15271 032040 102401 BVS E10424
15272 032042 103002 BCC A0424
15273
15274 032044 104000 E10424: ERROR ;XOR FAILED TO ALTER CODES PROPERLY
15275 032046 032024 R0424 ;ERROR LOOP RETURN ADDRESS
15276
15277 032050 020412 A0424: CMP R4,(R2) ;RESULT CORRECT?
15278 032052 001403 BEQ 00424 ;BR IF YES
15279
15280 032054 011203 MOV (R2),R3 ;GET THE WAS DATA
15281 032056 104000 E20424: ERROR ;XOR DELIVERED THE WRONG RESULT
15282 032060 032024 R0424 ;ERROR LOOP RETURN ADDRESS
15283
15284 032062 000004 00424: SCOPE ;CALL THE SCOPE LOOP UTILITY
15285

```

```

15286 ; *****
15287 ; .SBTTL T0425 'XOR RA,(RB)'' TEST - A=B=177777 N:C=0101
15288 ; *****
15289
15290 ;MICROPROGRAMMING / LOGIC INFORMATION
15291
15292 ;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8
15293
15294 ;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016
15295
15296 ;EXEC: [224]ALUC=HLHHL :[367]D=000000
15297
15298 ;CODES: [367] SPS=3 / N:C=0101
15299
15300 ;SYNC: B05J2 (-) / T=2.6 USEC
15301
15302 ;KEY SIG: K3-5 XOR L / K3-3 DM=1L
15303
15304 032064 012700 000425 T0425: MOV #0425,R0 ;LOAD R0 WITH TEST NO.
15305 032070 013701 032116 MOV @#I0425,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15306 032074 005004 CLR R4 ;RESULT S / B = 000000
15307 032076 012705 177777 MOV #-1,R5 ;MASK = 177777
15308 032102 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
15309 032106 012712 177777 R0425: MOV #-1,(R2) ;[DEST] = 177777
15310 032112 000257 CCC ;SCOPE SYNC
15311 032114 000265 265 ;MAKE N:C=0101
15312
15313 032116 074512 I0425: XOR R5,(R2) ;TEST THE XOR
15314
15315 032120 100403 BMI E10425 ;N:C = 0101 ??
15316 032122 001002 BNE E10425
15317 032124 102401 BVS E10425
15318 032126 103402 BCS A0425
15319
15320 032130 104000 E10425: ERROR ;XOR FAILED TO ALTER CODES PROPERLY
15321 032132 032106 R0425 ;ERROR LOOP RETURN ADDRESS
15322
15323 032134 020412 A0425: CMP R4,(R2) ;RESULT CORRECT?
15324 032136 001403 BEQ 00425 ;BR IF YES
15325
15326 032140 011203 E20425: MOV (R2),R3 ;GET THE WAS DATA
15327 032142 104000 ERROR ;XOR DELIVERED THE WRONG RESULT
15328 032144 032106 R0425 ;ERROR LOOP RETURN ADDRESS
15329
15330 032146 000004 00425: SCOPE ;CALL THE SCOPE LOOP UTILITY
15331

```

```
15332 ; *****  
15333 ; .SBTTL T0426 'XOR RA,(RB)'' TEST - A=125252,B=052525 N:C=0110  
15334 ; *****  
15335  
15336 ;MICROPROGRAMMING / LOGIC INFORMATION  
15337  
15338 ;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8  
15339  
15340 ;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016  
15341  
15342 ;EXEC: [224]ALUC=HLHHL :[367]D=00177777  
15343  
15344 ;CODES: [367] SPS=3 / N:C=1000  
15345  
15346 ;SYNC: B05J2 (-) / T=2.6 USEC  
15347  
15348 ;KEY SIG: K3-5 XOR L / K3-3 DM=1L  
15349  
15350 032150 012700 000426 T0426: MOV #0426,R0 ;LOAD R0 WITH TEST NO.  
15351 032154 013701 032204 MOV @#10426,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
15352 032160 012704 177777 MOV #-1,R4 ;RESULT S/B = 177777  
15353 032164 012705 125252 MOV #125252,R5 ;MASK = 125252  
15354 032170 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
15355 032174 012712 052525 R0426: MOV #052525,(R2) ;[DEST] = 052525  
15356 032200 000257 CCC ;SCOPE SYNC  
15357 032202 000266 266 ;MAKE N:C=0110  
15358  
15359 032204 074512 I0426: XOR R5,(R2) ;TEST THE XOR  
15360  
15361 032206 100003 BPL E10426 ;N:C = 1000 ??  
15362 032210 001402 BEQ E10426  
15363 032212 102401 BVS E10426  
15364 032214 103002 BCC A0426  
15365  
15366 032216 104000 E10426: ERROR ;XOR FAILED TO ALTER CODES PROPERLY  
15367 032220 032174 R0426 ;ERROR LOOP RETURN ADDRESS  
15368  
15369 032222 020412 A0426: CMP R4,(R2) ;RESULT CORRECT?  
15370 032224 001403 BEQ 00426 ;BR IF YES  
15371  
15372 032226 011203 MOV (R2),R3 ;GET THE WAS DATA  
15373 032230 104000 E20426: ERROR ;XOR DELIVERED THE WRONG RESULT  
15374 032232 032174 R0426 ;ERROR LOOP RETURN ADDRESS  
15375  
15376 032234 000004 00426: SCOPE ;CALL THE SCOPE LOOP UTILITY  
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

: *****
.SBTTL T0427 'XOR RA,(RB)'' TEST - A=052525,B=125252 N:C=1001
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8
:ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016
:EXEC: [224]ALUC=HLHHL :[367]D=177777
:CODES: [367] SPS=3 / N:C=1001
:SYNC: B05J2 (-) / T=2.6 USEC
:KEY SIG: K3-5 XOR L / K3-3 DM=1L

T0427: MOV #0427,R0 ;LOAD R0 WITH TEST NO.
MOV @#10427,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #-1,R4 ;RESULT S/B = 177777
MOV #52525,R5 ;MASK = 052525
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
R0427: MOV #125252,(R2) ;[DEST] = 125252
CCC ;SCOPE SYNC
271 ;MAKE N:C=1001

I0427: XOR R5,(R2) ;TEST THE XOR

BPL E10427 ;N:C = 1001 ??
BEQ E10427
BVS E10427
BCS A0427

E10427: ERROR ;XOR FAILED TO ALTER CODES PROPERLY
R0427 ;ERROR LOOP RETURN ADDRESS

A0427: CMP R4,(R2) ;RESULT CORRECT?
BEQ 00427 ;BR IF YES

E20427: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;XOR DELIVERED THE WRONG RESULT
R0427 ;ERROR LOOP RETURN ADDRESS

00427: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

15425 ; *****
15426 ; .SBTTL T0430 SUB TEST SMO,DMO - (SRC) = (DEST) = +,+
15427 ; *****
15428
15429 ;MICROPROGRAMMING / LOGIC INFORMATION
15430
15431 ;ROM SEQ: [103,363,360,001] FC 1,7,8
15432
15433 ;ACT BUTS: 37[004]100,103 / 27[363]000,001
15434
15435 ;EXEC: [363]ALUC=LLHHL :[360] D = 000000
15436
15437 ;CODES: [360] SPS=3 / N:C = 0100
15438
15439 ;SYNC: B05J2 (-) T = 1 USEC
15440
15441 ;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=0L / K3-4 OVLAP INSTR H
15442 ; K3-8 CIN00 L
15443
15444 032324 012700 000430 T0430: MOV #0430,R0 ;LOAD R0 WITH TEST NO.
15445 032330 013701 032354 MOV @#10430,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15446
15447 032334 012702 177703 MOV #177703,R2 ;R2 CONTAINS DEST ADDR
15448 032340 005004 CLR R4 ;RESULT S / B = 0
15449 032342 012703 052525 R0430: MOV #052525,R3 ;[R3] = DEST OP = 52525
15450 032346 010305 MOV R3,R5 ;[R5] = SRC OP = 52525
15451 032350 000257 CCC ;CLEAR FLAGS
15452 032352 000273 273 ;MAKE N:C = 1011
15453
15454 032354 160503 I0430: SUB R5,R3 ;TEST THE SUB
15455
15456 032356 100403 BMI E10430
15457 032360 001002 BNE E10430 ;DID N:C = 0100
15458 032362 102401 BVS E10430
15459 032364 103002 BCC A0430
15460
15461 032366 104000 E10430: ERROR ;SUB FAILED TO ALTER CODES PROPERLY
15462 032370 032342 R0430 ;ERROR LOOP RETURN
15463
15464 032372 020304 A0430: CMP R3,R4 ;WAS RESULT = 0?
15465 032374 001402 BEQ 00430 ;BR IF YES
15466
15467 032376 104000 E20430: ERROR ;SUB DELIVERED WRONG RESULT
15468 032400 032342 R0430 ;ERROR LOOP RETURN
15469
15470 032402 000004 00430: SCOPE ;CALL SCOPE LOOP UTILITY
15471
15472

```

T0430 SUB TEST SMO,DMO - (SRC) = (DEST) = +,+

SEQ 0400

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

; *****
; .SBTTL T0431 SUB TEST SMO,DMO - (SRC) = (DEST) = -,-
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [103,363,360,001] FC 1,7,8
;ACT BUTS: 37[004]100,103 / 27[363]000,001
;EXEC: [363]ALUC=LLHHL :[360] D = 000000
;CODES: [360] SPS=3 / N:C = 0100
;SYNC: B05J2 (-) T = 1 USEC
;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=0L / K3-4 OVLAP INSTR H
; K3-8 CIN00 L

T0431: MOV #0431,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0431,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD

MOV #177703,R2 ;DEST ADDR = R3
CLR R4 ;RESULT S / B = 0
R0431: MOV #125252,R3 ;[R3] = DEST OP = 125252
MOV R3,R5 ;[R5] = SOURCE OP = 125252
CCC ;CLEAR FLAGS
273 ;MAKE N:C = 1011

I0431: SUB R5,R3 ;TEST THE SUB

BMI E10431
BNE E10431 ;N:C = 0100?
BVS E10431
BCC A0431

E10431: ERROR ;SUB FAILED TO ALTER CODES PROPERLY
R0431 ;ERROR LOOP RETURN

A0431: CMP R3,R4 ;RESULT = 0?
BEQ 00431 ;BR IF YES

E20431: ERROR ;SUB DELIVERED WRONG RESULT
R0431 ;ERROR LOOP RETURN

00431: SCOPE ;CALL SCOPE LOOP UTILITY

```

15521 ; *****
15522 ; .SBTTL T0432 SUB TEST SMO,DMO - (SRC) = (DEST) = -,+
15523 ; *****
15524
15525 ;MICROPROGRAMMING / LOGIC INFORMATION
15526
15527 ;ROM SEQ: [103,363,360,001] FC 1,7,8
15528
15529 ;ACT BUTS: 37[004]100,103 / 27[363]000,001
15530
15531 ;EXEC: [363]ALUC=LLHHL :[360] D = 000002
15532
15533 ;CODES: [360] SPS=3 / N:C = 0001
15534
15535 ;SYNC: B05J2 (-) T = 1 USEC
15536
15537 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=0L / K3-3 DM=0L / K3-4 OVLAP INSTR H
15538 ; K3-8 CIN00 L
15539
15540 032464 012700 000432 T0432: MOV #0432,R0 ;LOAD R0 WITH TEST NO.
15541 032470 013701 032520 MOV @#10432,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15542
15543 032474 012702 177703 MOV #177703,R2 ;R2 CONTAINS DEST ADDR
15544 032500 012704 000002 MOV #2,R4 ;RESULT S / B = 2
15545 032504 012703 000001 R0432: MOV #1,R3 ;[R3] = DEST OP = 1
15546 032510 012705 177777 MOV #-1,R5 ;[R5] = SRC OP = -1
15547 032514 000257 CCC ;CLEAR FLAGS
15548 032516 000276 276 ;MAKE N:C = 1110
15549
15550 032520 160503 I0432: SUB R5,R3 ;TEST THE SUB
15551
15552 032522 100403 BMI E10432
15553 032524 001402 BEQ E10432 ;N:C = 0001
15554 032526 102401 BVS E10432
15555 032530 103402 BCS A0432
15556
15557 032532 104000 E10432: ERROR ;SUB FAILED TO ALTER CODES PROPERLY
15558 032534 032504 R0432 ;ERROR LOOP RETURN
15559
15560 032536 020304 A0432: CMP R3,R4 ;RESULT = +2?
15561 032540 001402 BEQ 00432 ;BR IF YES
15562
15563 032542 104000 E20432: ERROR ;SUB DELIVERED WRONG RESULT
15564 032544 032504 R0432 ;ERROR LOOP RETURN
15565
15566 032546 000004 00432: SCOPE ;CALL SCOPE LOOP UTILITY
15567
15568

```

```
15569 ; *****
15570 ; .SBTTL T0433 SUB TEST SMO,DMO (SRC) = -(DEST) = +,-
15571 ; *****
15572
15573 ;MICROPROGRAMMING / LOGIC INFORMATION
15574
15575 ;ROM SEQ: [103,363,360,001] FC 1,7,8
15576
15577 ;ACT BUTS: 37[004]100,103 / 27[363]000,001
15578
15579 ;EXEC: [363]ALUC=LLHHL :[360] D = 177776
15580
15581 ;CODES: [360] SPS=3 / N:C = 1000
15582
15583 ;SYNC: B05J2 (-) T = 1 USEC
15584
15585 ;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=0L / K3-4 OVLAP INSTR H
15586 ; K3-8 CIN00 L
15587
15588 032550 012700 000433 T0433: MOV #0433,R0 ;LOAD R0 WITH TEST NO.
15589 032554 013701 032604 MOV @#I0433,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15590
15591 032560 012702 177703 MOV #177703,R2 ;R2 CONTAINS DEST ADDR
15592 032564 012704 177776 MOV #-2,R4 ;RESULT S / B = -2
15593 032570 012703 177777 R0433: MOV #-1,R3 ;[R3] = [DEST] = -1
15594 032574 012705 000001 MOV #1,R5 ;[R5] = [SOURCE] = +1
15595 032600 000257 CCC ;CLEAR FLAGS
15596 032602 000267 267 ;MAKE N:C = 0111
15597
15598 032604 160503 I0433: SUB R5,R3 ;TEST THE SUB
15599
15600 032606 100003 BPL E10433
15601 032610 001402 BEQ E10433 ;N:C = 1000
15602 032612 102401 BVS E10433
15603 032614 103002 BCC A0433
15604
15605 032616 104000 E10433: ERROR ;SUB DID NOT ALTER CODES PROPERLY
15606 032620 032570 R0433 ;ERROR LOOP RETURN
15607
15608 032622 020403 A0433: CMP R4,R3 ;RESULT = -2?
15609 032624 001402 BEQ 00433 ;BR IF YES
15610
15611 032626 104000 E20433: ERROR ;SUB DELIVERED WRONG RESULT
15612 032630 032570 R0433 ;ERROR LOOP RETURN
15613
15614 032632 000004 00433: SCOPE ;CALL SCOPE LOOP UTILITY
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

032634 012700 000434
032640 013701 032670
032644 012702 177703
032650 012704 077777
032654 012703 100000
032660 012705 000001
032664 000257
032666 000274
032670 160503
032672 100403
032674 001402
032676 102001
032700 103002
032702 104000
032704 032654
032706 020304
032710 001402
032712 104000
032714 032654
032716 000004

```
; *****  
; .SBTTL T0434 SUB TEST SMO,DMO - 'V' BIT SETS  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [103,363,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,103 / 27[363]000,001  
;EXEC: [363]ALUC=LLHHL :[360] D = 77777  
;CODES: [360] SPS=3 / N:C = 0011  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-3 ADD+SUB L / K3-3 DM=0L / K3-3 SM=0L / K3-4 OVLAP INSTR H  
; K3-8 CIN00 L  
T0434: MOV #0434,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0434,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = 177703  
MOV #77777,R4 ;RESULT = 77777  
R0434: MOV #100000,R3 ;[R3] = DEST OP = 100000  
MOV #1,R5 ;[R5] = SRC OP = 1  
CCC ;CLEAR FLAGS  
274 ;MAKE N:C = 1100  
I0434: SUB R5,R3 ;TEST THE SUB  
BMI E10434  
BEQ E10434 ;N:C = 0011 ('V' BIT SHOULD SET)  
BVC E10434  
BCC A0434  
E10434: ERROR ;SUB FAILED TO ALTER CODES PROPERLY  
R0434 ;ERROR LOOP RETURN  
A0434: CMP R3,R4 ;RESULT = 77777?  
BEQ 00434 ;BR IF YES  
E20434: ERROR ;SUB DELIVERED WRONG RESULT  
R0434 ;ERROR LOOP RETURN  
00434: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

15665 ; *****
15666 ; .SBTTL T0435 SUB TEST - SMO,DM1 - <N:C> = 0110
15667 ; *****
15668
15669 ;MICROPROGRAMMING / LOGIC INFORMATION
15670
15671 ;ROM SEQ: [161,266,267,226,365,367,375,016] FC 1,3,8
15672
15673 ;ACT BUTS: 37[004]100,161 / 33[266]220,226 / 16[367]016,016
15674
15675 ;EXEC: [365]ALUC=LLHHL :[367] D = 177777
15676
15677 ;CODES: [367] SPS=3 / N:C = 1001
15678
15679 ;SYNC: B05J2 (-) T = 2.7 USEC
15680
15681 ;KEY SIG: K3-8 CIN00 L / K3-3 ADD+SUB L / K3-3 SM=0L / K3-3 DM=1L
15682
15683 032720 012700 000435 T0435: MOV #0435,R0 ;LOAD R0 WITH TEST NO.
15684 032724 013701 032752 MOV @#I0435,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15685 032730 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
15686 032734 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777
15687 032740 012705 000001 MOV #+1,R5 ;SRC OPR = +1
15688 032744 005012 R0435: CLR (R2) ;[DEST] = 000000
15689 032746 000257 CCC ;CLEAR FLAGS
15690 032750 000266 266 ;N:C = 0110
15691
15692 032752 160512 I0435: SUB R5,(R2) ;TEST THE SUB
15693
15694 032754 100003 BPL E10435 ;N:C = 1001
15695 032756 001402 BEQ E10435
15696 032760 102401 BVS E10435
15697 032762 103402 BCS A0435
15698
15699 032764 104005 E10435: ERROR5 ;SUB FAILED TO ALTER CODES PROPERLY
15700 032766 032744 R0435 ;ERROR LOOP RETURN ADDRESS
15701
15702 032770 020412 A0435: CMP R4,(R2) ;CORRECT RESULT ?
15703 032772 001403 BEQ 00435 ;BR IF YES
15704
15705 032774 011203 MOV (R2),R3 ;GET THE WAS DATA
15706 032776 104000 E20435: ERROR ;SUB DELIVERED THE WRONG RESULT
15707 033000 032744 R0435 ;ERROR LOOP RETURN ADDRESS
15708
15709 033002 000004 00435: SCOPE ;CALL SCOPE LOOP UTILITY

```

```
15710 ; *****
15711 ; .SBTTL T0436 SUB TEST - SMO,DM1 - <N:C> = 1010
15712 ; *****
15713 ;MICROPROGRAMMING / LOGIC INFORMATION
15714 ;ROM SEQ: 1616,266,267,226,365,367,375,016] FC 1,3,8
15715 ;ACT BUTS: 37[004]100,161 / 33[266]220,226 / 16[367]016,016
15716 ;EXEC: [365]ALUC=LLHHL :[367] D = 000000
15717 ;CODES: [367] SPS=3 / N:C = 0100
15718 ;SYNC: B05J2 (-) T = 2.7 USEC
15719 ;KEY SIG: K3-8 CIN00 L / K3-3 ADD+SUB L / K3-3 SM=0L / K3-3 DM=1L
15720
15721
15722
15723
15724
15725
15726
15727
15728 033004 012700 000436 T0436: MOV #0436,R0 ;LOAD R0 WITH TEST NO.
15729 033010 013701 033036 MOV @#I0436,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15730 033014 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
15731 033020 005004 CLR R4 ;RESULT S / B = 000000
15732 033022 012705 177777 MOV #-1,R5 ;SRC OPR = 177777
15733 033026 012712 177777 R0436: MOV #-1,(R2) ;[DEST] = 177777
15734 033032 000257 CCC ;CLEAR FLAGS
15735 033034 000272 272 ;N:C = 1010
15736
15737 033036 160512 I0436: SUB R5,(R2) ;TEST THE SUB
15738
15739 033040 100403 BMI E10436 ;N:C = 0100
15740 033042 001002 BNE E10436
15741 033044 102401 BVS E10436
15742 033046 103002 BCC A0436
15743
15744 033050 104005 E10436: ERROR5 ;SUB FAILED TO ALTER CODES PROPERLY
15745 033052 033026 R0436 ;ERROR LOOP RETURN ADDRESS
15746
15747 033054 020412 A0436: CMP R4,(R2) ;CORRECT RESULT ?
15748 033056 001403 BEQ 00436 ;BR IF YES
15749
15750 033060 011203 MOV (R2),R3 ;GET THE WAS DATA
15751 033062 104000 E20436: ERROR ;SUB DELIVERED THE WRONG RESULT
15752 033064 033026 R0436 ;ERROR LOOP RETURN ADDRESS
15753
15754 033066 000004 00436: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

15755 ; *****
15756 ; .SBTTL T0437 SUB TEST - SMO,DM1 - <N:C> = 0000
15757 ; *****
15758
15759 ;MICROPROGRAMMING / LOGIC INFORMATION
15760
15761 ;ROM SEQ: [161,266,267,226,365,367,375,016] FC 1,3,8
15762
15763 ;ACT BUTS: 37[004]100,161 / 33[266]220,226 / 16[367]016,016
15764
15765 ;EXEC: [365]ALUC=LLHHL :[367] D = 77777
15766
15767 ;CODES: [367] SPS=3 / N:C = 0010
15768
15769 ;SYNC: B05J2 (-) T = 2.7 USEC
15770
15771 ;KEY SIG: K3-8 CIN00 L / K3-3 ADD+SUB L / K3-3 SM=0L / K3-3 DM=1L
15772
15773 033070 012700 000437 T0437: MOV #0437,R0 ;LOAD R0 WITH TEST NO.
15774 033074 013701 033122 MOV @#I0437,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15775 033100 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
15776 033104 012704 077777 MOV #77777,R4 ;RESULT S / B = 77777
15777 033110 012705 000001 MOV #+1,R5 ;SRC OPR = +1
15778 033114 012712 100000 R0437: MOV #100000,(R2) ;[DEST] = 100000
15779 033120 000257 CCC ;CLEAR FLAGS
15780
15781 033122 160512 I0437: SUB R5,(R2) ;TEST THE SUB
15782
15783 033124 100403 BMI E10437 ;N:C = 0010
15784 033126 001402 BEQ E10437
15785 033130 102001 BVC E10437
15786 033132 103002 BCC A0437
15787
15788 033134 104005 E10437: ERROR5 ;SUB FAILED TO ALTER CODES PROPERLY
15789 033136 033114 R0437 ;ERROR LOOP RETURN ADDRESS
15790
15791 033140 020412 A0437: CMP R4,(R2) ;CORRECT RESULT ?
15792 033142 001403 BEQ 00437 ;BR IF YES
15793
15794 033144 011203 MOV (R2),R3 ;GET THE WAS DATA
15795 033146 104000 E20437: ERROR ;SUB DELIVERED THE WRONG RESULT
15796 033150 033114 R0437 ;ERROR LOOP RETURN ADDRESS
15797
15798 033152 000004 00437: SCOPE ;CALL SCOPE LOOP UTILITY

```

15799
15800
15801
15802
15803
15804
15805
15806
15807
15808
15809
15810
15811
15812
15813
15814
15815
15816
15817 033154 012700 000440
15818 033160 013701 033206
15819 033164 012702 177703
15820 033170 012704 177777
15821 033174 012705 070116
15822 033200 005003
15823 033202 000257
15824 033204 000266
15825
15826 033206 161503
15827
15828 033210 100003
15829 033212 001402
15830 033214 102401
15831 033216 103402
15832
15833 033220 104005
15834 033222 033200
15835
15836 033224 020403
15837 033226 001402
15838
15839 033230 104000
15840 033232 033200
15841
15842 033234 000004

```
; *****  
; .SBTTL T0440 SUB TEST - SM1,DM0 - <N:C> = 0110  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,121,370,360,000] FC 1,2,8  
;ACT BUTS: 37[004]100,141 / 35[247]120,121 / 27[370]000,000  
;EXEC: [370]ALUC=LLHHL :[360] D = 177777  
;CODES: [360] SPS=3 / N:C = 1001  
;SYNC: B05J2 (-) T = 2.7 USEC  
;KEY SIG: K3-8 CIN00 L / K3-3 ADD+SUB L / K3-3 SM=1L* / K3-3 DM=0L  
T0440: .MOV #0440,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0440,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
MOV #-1,R4 ;RESULT S / B = 177777  
MOV #DWTB+2,R5 ;SRC ADDR = DWTB+2  
R0440: CLR R3 ;[DEST] = 000000  
CCC ;CLEAR FLAGS  
266 ;N:C = 0110  
I0440: SUB (R5),R3 ;TEST THE SUB  
BPL E10440 ;N:C = 1001  
BEQ E10440  
BVS E10440  
BCS A0440  
E10440: ERROR5 ;SUB FAILED TO ALTER CODES PROPERLY  
R0440 ;ERROR LOOP RETURN ADDRESS  
A0440: CMP R4,R3 ;CORRECT RESULT ?  
BEQ 00440 ;BR IF YES  
E20440: ERROR ;SUB DELIVERED THE WRONG RESULT  
R0440 ;ERROR LOOP RETURN ADDRESS  
00440: SCOPE ;CALL SCOPE LOOP UTILITY
```

15843
15844
15845
15846
15847
15848
15849
15850
15851
15852
15853
15854
15855
15856
15857
15858
15859
15860
15861 033236 012700 000441
15862 033242 013701 033266
15863 033246 012702 177703
15864 033252 005004
15865 033254 012705 067572
15866 033260 011503
15867 033262 000257
15868 033264 000272
15869
15870 033266 161503
15871
15872 033270 100403
15873 033272 001002
15874 033274 102401
15875 033276 103002
15876
15877 033300 104005
15878 033302 033260
15879
15880 033304 020403
15881 033306 001402
15882
15883 033310 104000
15884 033312 033260
15885
15886 033314 000004

```
; *****  
; .SBTTL T0441 SUB TEST - SM1,DMO - <N:C> = 1010  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,121,370,360,000] FC 1,2,8  
;ACT BUTS: 37[004]100,141 / 35[247]120,121 / 27[370]000,000  
;EXEC: [370]ALUC=LLHHL :[360] D = 000000  
;CODES: [360] SPS=3 / N:C = 0100  
;SYNC: B05J2 (-) T = 2.7 USEC  
;KEY SIG: K3-8 CIN00 L / K3-3 ADD+SUB L / K3-3 SM=1L / K3-3 DM=0L  
T0441: MOV #0441,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0441,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
CLR R4 ;RESULT S / B = 000000  
R0441: MOV #DWTA+2,R5 ;SRC ADDR = DWTA+2  
MOV (R5),R3 ;[DEST] = 177777  
CCC ;CLEAR FLAGS  
272 ;N:C = 1010  
I0441: SUB (R5),R3 ;TEST THE SUB  
BMI E10441 ;N:C = 0100  
BNE E10441  
BVS E10441  
BCC A0441  
E10441: ERROR5 ;SUB FAILED TO ALTER CODES PROPERLY  
R0441 ;ERROR LOOP RETURN ADDRESS  
A0441: CMP R4,R3 ;CORRECT RESULT ?  
BEQ 00441 ;BR IF YES  
E20441: ERROR ;SUB DELIVERED THE WRONG RESULT  
R0441 ;ERROR LOOP RETURN ADDRESS  
00441: SCOPE ;CALL SCOPE LOOP UTILITY
```

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

033316 012700 000442
033322 013701 033354
033326 012702 067560
033332 012704 077777
033336 012705 067564
033342 012703 100000
033346 012715 000001
033352 000257
033354 161503
033356 100403
033360 001402
033362 102001
033364 103002
033366 104005
033370 033342
033372 020403
033374 001402
033376 104000
033400 033342
033402 000004

```
; *****  
; .SBTTL T0442 SUB TEST - SM1,DMO - <N:C> = 0000  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,121,370,360,000] FC 1,2,8  
;ACT BUTS: 37[004]100,141 / 35[247]120,121 / 27[370]000,000  
;EXEC: [370]ALUC=LLHHL :[360] D = 077777  
;CODES: [360] SPS=3 / N:C = 0010  
;SYNC: B05J2 (-) T = 2.7 USEC  
;KEY SIG: K3-8 CIN00 L / K3-3 ADD+SUB L / K3-3 SM=1L / K3-3 DM=0L  
T0442: MOV #0442,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0442,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #77777,R4 ;RESULT S / B = 77777  
MOV #MBUF1,R5 ;SRC ADDR =MBUF1  
R0442: MOV #100000,R3 ;[DEST] = 100000  
MOV #+1,(R5) ;SRC OPR = +1  
CCC ;CLEAR FLAGS  
I0442: SUB (R5),R3 ;TEST THE SUB  
BMI E10442 ;N:C = 0010  
BEQ E10442  
BVC E10442  
BCC A0442  
E10442: ERROR5 ;SUB FAILED TO ALTER CODES PROPERLY  
R0442 ;ERROR LOOP RETURN ADDRESS  
A0442: CMP R4,R3 ;CORRECT RESULT ?  
BEQ 00442 ;BR IF YES  
E20442: ERROR ;SUB DELIVERED THE WRONG RESULT  
R0442 ;ERROR LOOP RETURN ADDRESS  
00442: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
15931 ; *****
15932 ; .SBTTL T0443 SUB SM1,DM1 TEST - <N:C> = 0110
15933 ; *****
15934
15935 ;MICROPROGRAMMING / LOGIC INFORMATION
15936
15937 ;ROM SEQ: [141,247,250,161,266,267,227,365,367,375,016] FC 1,2,3,8
15938
15939 ;ACT BUTS: 37[004]100,141 / 35[247]120,161 / 33[266]220,227 / 16[367]016,016
15940
15941 ;EXEC: [365]ALUC=LLHHL :[367]D=177777
15942
15943 ;CODES: [367]SPS=3 / N:C=1001
15944
15945 ;SYNC: B05J2 (-) T=3.4 USEC
15946
15947 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=1 L / K3-3 DM=1 L / K3-8 CIN00 L
15948
15949 033404 012700 000443 T0443: MOV #0443,R0 ;LOAD R0 WITH TEST NO.
15950 033410 013701 033444 MOV @#I0443,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15951 033414 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
15952 033420 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777
15953 033424 012705 067564 MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
15954 033430 012715 000001 R0443: MOV #+1,(R5) ;[SOURCE] = 000001
15955 033434 012712 000000 MOV #0,(R2) ;[DEST] = 000000
15956 033440 000257 CCC ;CLEAR FLAGS
15957 033442 000266 266 ;N:C = 0110
15958
15959 033444 161512 I0443: SUB (R5),(R2) ;TEST THE SUB
15960
15961 033446 100003 BPL E10443 ;N:C = 1001 ?
15962 033450 001402 BEQ E10443
15963 033452 102401 BVS E10443
15964 033454 103402 BCS A0443
15965
15966 033456 104005 E10443: ERROR5 ;SUB FAILED TO ALTER CODES PROPERLY
15967 033460 033430 R0443 ;ERROR LOOP RETURN ADDRESS
15968
15969 033462 020412 A0443: CMP R4,(R2) ;CORRECT RESULT ?
15970 033464 001403 BEQ 00443 ;BR IF YES
15971
15972 033466 011203 MOV (R2),R3 ;GET THE WAS DATA
15973 033470 104000 E20443: ERROR ;SUB DELIVERED THE WRONG RESULT
15974 033472 033430 R0443 ;ERROR LOOP RETURN ADDRESS
15975
15976 033474 000004 00443: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

```
15977 ; *****
15978 ; .SBTTL T0444 SUB SM1,DM2 TEST - <N:C> = 0110
15979 ; *****
15980 ;MICROPROGRAMMING / LOGIC INFORMATION
15981 ;ROM SEQ: [141,247,250,162,260,267,227,365,367,375,016] FC 1,2,3,8
15982 ;ACT BUTS: 37[004]100,141 / 35[247]120,162 / 33[260]220,227 / 16[367]016,016
15983 ;EXEC: [365]ALUC=LLHHL :[367]D=177777
15984 ;CODES: [367]SPS=3 / N:C=1001
15985 ;SYNC: B05J2 (-) T=3.4 USEC
15986 ;KEY SIG: K3-3 ADD+SUB L / K3-3 SM=1 L / K3-3 DM=2 / K3-8 CIN00 L
15987 ; K5-5 BCON(1+2) H
15988
15989
15990
15991
15992
15993
15994
15995
15996 033476 012700 000444 T0444: MOV #0444,R0 ;LOAD R0 WITH TEST NO.
15997 033502 013701 033552 MOV @#I0444,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
15998 033506 032737 000400 066642 BIT #400,@#BPTLOC ;BREAKPOINT HALT SET ??
15999 033514 001401 BEQ .+4 ;BR IF NOT
16000 033516 000000 HALT ;BREAK-DEPRESS CONTINUE TO RESTART
16001 033520 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
16002 033524 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777
16003 033530 012705 067564 MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
16004 033534 012715 000001 R0444: MOV #+1,(R5) ;[SOURCE] = 000001
16005 033540 012712 000000 MOV #0,(R2) ;[DEST] = 000000
16006 033544 010203 MOV R2,R3 ;R3 GETS DEST ADDR
16007 033546 000257 CCC ;CLEAR FLAGS
16008 033550 000266 266 ;N:C = 0110
16009
16010 033552 161523 I0444: SUB (R5),(R3)+ ;TEST THE SUB
16011
16012 033554 100003 BPL E10444 ;N:C = 1001 ?
16013 033556 001402 BEQ E10444
16014 033560 102401 BVS E10444
16015 033562 103402 BCS A0444
16016
16017 033564 104005 E10444: ERROR5 ;SUB FAILED TO ALTER CODES PROPERLY
16018 033566 033534 R0444 ;ERROR LOOP RETURN ADDRESS
16019
16020 033570 020412 A0444: CMP R4,(R2) ;CORRECT RESULT ?
16021 033572 001403 BEQ 00444 ;BR IF YES
16022
16023 033574 011203 MOV (R2),R3 ;GET THE WAS DATA
16024 033576 104000 E20444: ERROR ;SUB DELIVERED THE WRONG RESULT
16025 033600 033534 R0444 ;ERROR LOOP RETURN ADDRESS
16026
16027 033602 000004 00444: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

```
16028 ; *****  
16029 ; .SBTTL T0445 NEG DM2 TEST  
16030 ; *****  
16031  
16032 ;MICROPROGRAMMING / LOGIC INFORMATION  
16033  
16034 ;ROM SEQ: [162,260,267,221,367,375,016] FC 1,3,9,8  
16035  
16036 ;ACT BUTS: 37[004]100,162 / 33[260]220,221 / 16[367]016,016  
16037  
16038 ;EXEC: [221]ALUC=LLHHL :[367] D = 125252  
16039  
16040 ;CODES: [367] SPS=3 / N:C = 1001  
16041  
16042 ;SYNC: B05J2 (-) T = 2 USEC  
16043  
16044 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=2L / K3-4 NEG L / K5-5 BCON(1+2)H  
16045  
16046 033604 012700 000445 T0445: MOV #0445,R0 ;LOAD R0 WITH THE TEST NO.  
16047 033610 013701 033634 MOV @#I0445,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
16048 033614 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
16049 033620 012704 125252 MOV #125252,R4 ;RESULT S / B = 125252  
16050 033624 010205 R0445: MOV R2,R5 ;[R5] = DEST ADDR  
16051 033626 012712 052526 MOV #52526,(R2) ;[DEST] = 52526  
16052 033632 000257 CCC ;SCOPE SYNC  
16053  
16054 033634 005425 I0445: NEG (R5)+ ;TEST THE NEG - MODE 2  
16055  
16056 033636 020412 CMP R4,(R2) ;RESULT = 125252?  
16057 033640 001403 BEQ A0445 ;BR IF YES  
16058  
16059 033642 011203 MOV (R2),R3 ;GET THE WAS DATA  
16060 033644 104000 E10445: ERROR ;NEG DELIVERED WRONG RESULT  
16061 033646 033624 R0445 ;ERROR LOOP RETURN  
16062  
16063 033650 022705 067562 A0445: CMP #MBUF0+2,R5 ;DID REG. GET AUTO INCREMENTED?  
16064 033654 001402 BEQ 00445 ;BR IF YES  
16065  
16066 033656 104005 E20445: ERROR5 ;NEG FAILED TO UPDATE REG.  
16067 033660 033624 R0445 ;ERROR LOOP RETURN  
16068  
16069 033662 000004 00445: SCOPE ;CALL SCOPE LOOP UTILITY  
16070
```

```
16071 : *****
16072 : .SBTTL T0446 NEG DM3 TEST
16073 : *****
16074
16075 ;MICROPROGRAMMING / LOGIC INFORMATION
16076
16077 ;ROM SEQ: [163,264,265,266,267,221,367,375,016] FC 1,3,9,8
16078
16079 ;ACT BUTS: 37[004]100,163 / 33[266]220,221 / 16[367]016,016
16080
16081 ;EXEC: [221]ALUC=LLHHL :[367] D = 125252
16082
16083 ;CODES: [367] SPS=3 / N:C = 1001
16084
16085 ;SYNC: B05J2 (-) T = 2.75 USEC
16086
16087 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=3L / K3-4 NEG L / K5-5 BC01 H
16088
16089 033664 012700 000446 T0446: MOV #0446,R0 ;LOAD R0 WITH THE TEST NO.
16090 033670 013701 033716 MOV @#I0446,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16091 033674 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
16092 033700 012704 125252 MOV #125252,R4 ;RESULT S / B = 125252
16093 033704 012705 067554 R0446: MOV #ATA+10,R5 ;[ATA+10] = MBUFO
16094 033710 012712 052526 MOV #52526,(R2) ;[DEST] = 52526
16095 033714 000257 CCC ;SCOPE SYNC
16096
16097 033716 005435 I0446: NEG @ (R5)+ ;TEST THE NEG - MODE 3
16098
16099 033720 020412 CMP R4,(R2) ;RESULT = 125252?
16100 033722 001403 BEQ A0446 ;BR IF YES
16101
16102 033724 011203 MOV (R2),R3 ;GET WAS DATA
16103 033726 104000 E10446: ERROR ;NEG DELIVERED WRONG RESULT
16104 033730 033704 R0446 ;ERROR LOOP RETURN
16105
16106 033732 022705 067556 A0446: CMP #ATA+12,R5 ;DID REG GET AUTO INCREMENTED?
16107 033736 001402 BEQ 00446 ;BR IF YES
16108
16109 033740 104005 E20446: ERROR5 ;NEG FAILED TO UPDATE REG.
16110 033742 033704 R0446 ;ERROR LOOP RETURN
16111
16112 033744 000004 00446: SCOPE ;CALL SCOPE LOOP UTILITY
16113
```

```
16114 ; *****
16115 ; .SBTTL T0447 NEG DM4 TEST
16116 ; *****
16117
16118 ;MICROPROGRAMMING / LOGIC INFORMATION
16119
16120 ;ROM SEQ: [164,260,267,221,367,375,061] FC 1,3,9,8
16121
16122 ;ACT BUTS: 37[004]100,164 / 33[260]220,221 / 16[367]016,016
16123
16124 ;EXEC: [221]ALUC=LLHHL :[367] D = 125252
16125
16126 ;CODES: [367] SPS=3 / N:C = 1001
16127
16128 ;SYNC: B05J2 (-) T = 2 USEC
16129
16130 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=4L / K3-4 NEG L / K5-5 BCON(1+2) H
16131
16132 033746 012700 000447 T0447: MOV #0447,R0 ;LOAD R0 WITH THE TEST NO.
16133 033752 013701 034000 MOV @#I0447,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16134 033756 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
16135 033762 012704 125252 MOV #125252,R4 ;RESULT S / B = 125252
16136 033766 012705 067562 R0447: MOV #MBUF0+2,R5 ;[R5] = DEST ADDR + 2
16137 033772 012712 052526 MOV #52526,(R2) ;[DEST] = 52526
16138 033776 000257 CCC ;SCOPE SYNC
16139
16140 034000 005445 I0447: NEG -(R5) ;TEST THE NEG - MODE 4
16141
16142 034002 020412 CMP R4,(R2) ;RESULT = 125252?
16143 034004 001403 BEQ A0447 ;BR IF YES
16144
16145 034006 011203 MOV (R2),R3 ;GET WAS DATA
16146 034010 104000 E10447: ERROR ;NEG DELIVERED WRONG RESULT
16147 034012 033766 R0447 ;ERROR LOOP RETURN
16148
16149 034014 020502 A0447: CMP R5,R2 ;DID REG GET AUTO INCREMENTED?
16150 034016 001402 BEQ 00447 ;BR IF YES
16151
16152 034020 104005 E20447: ERROR5 ;NEG FAILED TO UPDATE REG
16153 034022 033766 R0447 ;ERROR LOOP RETURN
16154
16155 034024 000004 00447: SCOPE ;CALL SCOPE LOOP UTILITY
16156
```

```
16157 ; *****
16158 ; .SBTTL T0450 NEG DM5 TEST
16159 ; *****
16160
16161 ;MICROPROGRAMMING / LOGIC INFORMATION
16162
16163 ;ROM SEQ: [165,264,265,266,267,221,367,375,016] FC 1,3,9,8
16164
16165 ;ACT BUTS: 37[004]100,165 / 33[266]220,221 / 16[367]016,016
16166
16167 ;EXEC: [221]ALUC=LLHHL :[367] D = 125252
16168
16169 ;CODES: [367] SPS=3 / N:C = 1001
16170
16171
16172 ;SYNC: B05J2 (-) T = 2 USEC
16173
16174 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=5L / K3-4 NEG L / K5-5 BC01 H
16175
16176 034026 012700 000450 T0450: MOV #0450,R0 ;LOAD R0 WITH THE TEST NO.
16177 034032 013701 034060 MOV @#I0450,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16178 034036 012702 067560 MOV #M0450,R2 ;DEST ADDR = M0450
16179 034042 012704 125252 MOV #125252,R4 ;RESULT S / B = 125252
16180 034046 012705 067556 R0450: MOV #ATA+12,R5 ;[R5] = (ADR OF M0450) +2
16181 034052 012712 052526 MOV #52526,(R2) ;[DEST] = 52526
16182 034056 000257 CCC ;SCOPE SYNC
16183
16184 034060 005455 I0450: NEG @-(R5) ;TEST THE NEG - MODE 5
16185
16186 034062 020412 CMP R4,(R2) ;RESULT = 125252?
16187 034064 001403 BEQ A0450 ;BR IF YES
16188
16189 034066 011203 MOV (R2),R3 ;GET WAS DATA
16190 034070 104000 E10450: ERROR R0450 ;NEG DELIVERED WRONG RESULT
16191 034072 034046 ;ERROR LOOP RETURN
16192
16193 034074 022705 067554 A0450: CMP #ATA+10,R5 ;DID NEG UPDATE REG
16194 034100 001402 BEQ 00450 ;BR IF YES
16195
16196 034102 104005 E20450: ERROR5 ;NEG FAILED TO UPDATE REG
16197 034104 034046 R0450 ;ERROR LOOP RETURN
16198
16199 034106 000004 00450: SCOPE ;CALL SCOPE LOOP UTILITY
16200
```

```

16201 ; *****
16202 ; .SBTTL T0451 NEG DM6 TEST
16203 ; *****
16204
16205 ;MICROPROGRAMMING / LOGIC INFORMATION
16206
16207 ;ROM SEQ: [166,261,262,266,267,221,367,375,016] FC 1,3,9,8
16208
16209 ;ACT BUTS: 37[004]100,166 / 33[266]220,221 / 16[367]016,016
16210
16211 ;EXEC: [221]ALUC=LLHHL :[367] D = 125252
16212
16213 ;CODES: [367] SPS=3 / N:C = 1001
16214
16215 ;SYNC: B05J2 (-) T = 2.5 USEC
16216
16217 ;KEY SIG: K3-8 CIN00 L / K3-3 DM=6L / K3-4 NEG L / K5-5 BC01 H
16218 ; K3-4 OVLAP CYCLE L
16219
16220 034110 012700 000451 T0451: MOV #0451,R0 ;LOAD R0 WITH THE TEST NO.
16221 034114 013701 034142 MOV @#I0451,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16222 034120 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
16223 034124 012704 125252 MOV #125252,R4 ;RESULT S / B = 125252
16224 034130 012705 067556 R0451: MOV #MBUF0-2,R5 ;[R5] = BASE ADDR
16225 034134 012712 052526 MOV #52526,(R2) ;[DEST] = 52526
16226 034140 000257 CCC ;SCOPE SYNC
16227
16228 034142 005465 000002 I0451: NEG 2(R5) ;TEST THE NEG - MODE 6
16229
16230 034146 020412 CMP R4,(R2) ;RESULT = 125252?
16231 034150 001403 BEQ 00451 ;BR IF YES
16232
16233 034152 011203 MOV (R2),R3 ;GET WAS DATA
16234 034154 104000 E0451: ERROR ;NEG DELIVERED WRONG RESULT
16235 034156 034130 R0451 ;ERROR LOOP RETURN
16236
16237 034160 000004 00451: SCOPE ;CALL SCOPE LOOP UTILITY
16238

```

16239
16240
16241
16242
16243
16244
16245
16246
16247
16248
16249
16250
16251
16252
16253
16254
16255
16256
16257
16258
16259
16260
16261
16262
16263
16264
16265
16266
16267
16268
16269
16270
16271
16272
16273
16274
16275
16276

034162 012700 000452
034166 013701 034214
034172 012702 067560
034176 012704 125252
034202 012705 067544
034206 012712 052526
034212 000257
034214 005475 000010
034220 020412
034222 001403
034224 011203
034226 104000
034230 034202
034232 000004

; *****
; .SBTTL T0452 NEG DM7 TEST
; *****
;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ: [166,261,263,264,265,266,267,221,367,375,016] FC 1,3,9,8
;ACT BUTS: 37[004]100,166 / 33[266]220,221 / 16[367]016,016
;EXEC: [221]ALUC=LLHHL :[367] D = 125252
;CODES: [367] SPS=3 / N:C = 1001
;SYNC: B05J2 (-) T = 3.5 USEC
;KEY SIG: K3-8 CIN00 L / K3-3 DM=7L / K3-4 NEG L / K5-5 BC01 H
; K3-4 OVLAP CYCLE L

T0452: MOV #0452,R0 ;LOAD R0 WITH THE TEST NO.
MOV @#I0452,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #M0452,R2 ;DEST ADDR = M0452
MOV #125252,R4 ;RESULT S / B = 125252
R0452: MOV #ATA,R5 ;[R5] = BASE ADDR
MOV #52526,(R2) ;[DEST] = 52526
CCC ;SCOPE SYNC
I0452: NEG @10(R5) ;TEST THE NEG - MODE 7
CMP R4,(R2) ;RESULT = 125252?
BEQ 00452 ;BR IF YES
E0452: MOV (R2),R3 ;GET WAS DATA
ERROR ;NEG DELIVERED WRONG RESULT
R0452 ;ERROR LOOP RETURN
00452: SCOPE ;CALL SCOPE LOOP UTILITY

```
16277 ; *****  
16278 ; .SBTTL T0453 MOV SM1,DM1 TEST - <N:C> = 0100  
16279 ; *****  
16280  
16281 ;MICROPROGRAMMING / LOGIC INFORMATION  
16282  
16283 ;ROM SEQ: [141,247,250,171,257,200,125,375,016] FC 1,2,4,8  
16284  
16285 ;ACT BUTS: 37[004]100,141 / 35[247]120,171 / 22[171]200,200 / 16[125]016,016  
16286  
16287 ;EXEC: [200]ALUC=LLLLL :[125] D = 177777  
16288  
16289 ;CODES: [125] SPS=3 / N:C = 1000  
16290  
16291 ;SYNC: B05J2 (-) T = 3.25 USEC  
16292  
16293 ;KEY SIG: K3-3 SM=1L / K3-3 DM=1L / K3-3 MOV L  
16294  
16295 034234 012700 000453 T0453: MOV #0453,R0 ;LOAD R0 WITH TEST NO.  
16296 034240 013701 034270 MOV @#10453,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
16297 034244 005004 CLR R4 ;RESULT S / B = 177777  
16298 034246 005104 COM R4  
16299 034250 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
16300 034254 012705 067572 MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2  
16301 034260 010203 R0453: MOV R2,R3 ;BASE DEST ADDR = MBUFO  
16302 034262 005012 CLR (R2) ;MAKE [DEST] = 000000  
16303 034264 000257 CCC ;CLEAR FLAGS  
16304 034266 000264 264 ;N:C = 0100  
16305  
16306 034270 011513 I0453: MOV (R5),(R3) ;TEST THE MOV - SM1,DM1  
16307  
16308 034272 100003 BPL E10453 ;N:C = 1000 ?  
16309 034274 001402 BEQ E10453  
16310 034276 102401 BVS E10453  
16311 034300 103002 BCC A0453  
16312  
16313 034302 104005 E10453: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY  
16314 034304 034260 R0453 ;ERROR LOOP RETURN  
16315  
16316 034306 020412 A0453: CMP R4,(R2) ;RESULT CORRECT ??  
16317 034310 001404 BEQ 00453 ;BR IF YES  
16318  
16319 034312 005003 CLR R3 ;GET THE WAS DATA  
16320 034314 051203 BIS (R2),R3  
16321 034316 104000 E20453: ERROR ;MOV DELIVERED THE WRONG RESULT  
16322 034320 034260 R0453 ;ERROR LOOP RETURN  
16323  
16324 034322 000004 00453: SCOPE ;CALL SCOPE LOOP UTILITY  
16325
```

```
16326 ; *****
16327 ; .SBTTL T0454 MOV SM2,DM1 TEST - <N:C> = 0100
16328 ; *****
16329
16330 ;MICROPROGRAMMING / LOGIC INFORMATION
16331
16332 ;ROM SEQ: [142,240,250,171,257,200,125,375,016] FC 1,2,4,8
16333
16334 ;ACT BUTS: 37[004]100,142 / 35[240]120,171 / 22[171]200,200 / 16[125]016,016
16335
16336 ;EXEC: [200]ALUC=LLLLL :[125] D = 177777
16337
16338 ;CODES: [125] SPS=3 / N:C = 1000
16339
16340 ;SYNC: B05J2 (-) T = 3.25 USEC
16341
16342 ;KEY SIG: K3-3 SM=2L / K3-3 DM=1L / K3-3 MOV L / K5-5 BCON (1+2) H
16343
16344 034324 012700 000454 T0454: MOV #0454,R0 ;LOAD R0 WITH TEST NO.
16345 034330 013701 034360 MOV @#10454,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16346 034334 005004 CLR R4 ;RESULT S / B = 177777
16347 034336 005104 COM R4
16348 034340 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
16349 034344 012705 067572 MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
16350 034350 010203 R0454: MOV R2,R3 ;BASE DEST ADDR = MBUFO
16351 034352 005012 CLR (R2) ;MAKE [DEST] = 000000
16352 034354 000257 CCC ;CLEAR FLAGS
16353 034356 000264 264 ;N:C = 0100
16354
16355 034360 012513 I0454: MOV (R5)+,(R3) ;TEST THE MOV - SM2,DM1
16356
16357 034362 100003 BPL E10454 ;N:C = 1000 ?
16358 034364 001402 BEQ E10454
16359 034366 102401 BVS E10454
16360 034370 103002 BCC A0454
16361
16362 034372 104005 E10454: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
16363 034374 034350 R0454 ;ERROR LOOP RETURN
16364
16365 034376 020412 A0454: CMP R4,(R2) ;RESULT CORRECT ??
16366 034400 001404 BEQ 00454 ;BR IF YES
16367
16368 034402 005003 CLR R3 ;GET THE WAS DATA
16369 034404 051203 BIS (R2),R3
16370 034406 104000 E20454: ERROR ;MOV DELIVERED THE WRONG RESULT
16371 034410 034350 R0454 ;ERROR LOOP RETURN
16372
16373 034412 000004 00454: SCOPE ;CALL SCOPE LOOP UTILITY
```

16374
16375
16376
16377
16378
16379
16380
16381
16382
16383
16384
16385
16386
16387
16388
16389
16390
16391
16392
16393
16394
16395
16396
16397
16398
16399
16400
16401
16402
16403
16404
16405
16406
16407
16408
16409
16410
16411
16412
16413
16414
16415
16416
16417
16418
16419
16420
16421
16422

034414 012700 000455
034420 013701 034450
034424 005004
034426 012702 067560
034432 012705 067570
034436 010203
034440 005012
034442 005112
034444 000257
034446 000273
034450 011513
034452 100403
034454 001002
034456 102401
034460 103402
034462 104005
034464 034436
034466 020412
034470 001404
034472 005003
034474 051203
034476 104000
034500 034436
034502 000004

```
; *****  
; .SBTTL T0455 MOV SM1,DM1 TEST - <N:C> = 1011  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [142,247,250,171,257,200,125,375,016] FC 1,2,4,8  
;ACT BUTS: 37[004]100,142 / 35[247]120,171 / 22[171]200,200 / 16[125]016,016  
;EXEC: [200]ALUC=LLLLL :[125] D = 000000  
;CODES: [125] SPS=3 / N:C = 0101  
;SYNC: B05J2 (-) T = 3.25 USEC  
;KEY SIG: K3-3 SM=1L / K3-3 DM=1L / K3-3 MOV L  
T0455: MOV #0455,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0455,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  
R0455: MOV R2,R3 ;BASE DEST ADDR = MBUFO  
CLR (R2) ;MAKE [DEST] = 177777  
COM (R2)  
CCC ;CLEAR FLAGS  
273 ;N:C = 1011  
I0455: MOV (R5),(R3) ;TEST THE MOV - SM1,DM1  
BMI E10455 ;N:C = 0101 ?  
BNE E10455  
BVS E10455  
BCS A0455  
E10455: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY  
R0455 ;ERROR LOOP RETURN  
A0455: CMP R4,(R2) ;RESULT CORRECT ??  
BEQ 00455 ;BR IF YES  
CLR R3 ;GET THE WAS DATA  
BIS (R2),R3  
E20455: ERROR ;MOV DELIVERED THE WRONG RESULT  
R0455 ;ERROR LOOP RETURN  
00455: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
16423 ; *****
16424 ; .SBTTL T0456 MOV SM2,DM1 TEST - <N:C> = 1011
16425 ; *****
16426
16427 ;MICROPROGRAMMING / LOGIC INFORMATION
16428
16429 ;ROM SEQ: [142,240,250,171,257,200,125,375,016] FC 1,2,4,8
16430
16431 ;ACT BUTS: 37[004]100,142 / 35[240]120,171 / 22[171]200,200 / 16[125]016,016
16432
16433 ;EXEC: [200]ALUC=LLLLL :[125] D = 000000
16434
16435 ;CODES: [125] SP=3 / N:C = 0101
16436
16437 ;SYNC: B05J2 (-) T = 3.25 USEC
16438
16439 ;KEY SIG: K3-3 SM=2L / K3-3 DM=1L / K3-3 MOV L / K5-5 BCON (1+2) H
16440
16441 034504 012700 000456 T0456: MOV #0456,R0 ;LOAD R0 WITH TEST NO.
16442 034510 013701 034540 MOV @#I0456,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16443 034514 005004 CLR R4 ;RESULT S / B = 000000
16444 034516 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
16445 034522 012705 067570 MOV #DWTA,R5 ;SOURCE ADDR = DWTA
16446 034526 010203 R0456: MOV R2,R3 ;BASE DEST ADDR = MBUFO
16447 034530 005012 CLR (R2) ;MAKE [DEST] = 177777
16448 034532 005112 COM (R2)
16449 034534 000257 CCC ;CLEAR FLAGS
16450 034536 000273 273 ;N:C = 1011
16451
16452 034540 012513 I0456: MOV (R5)+,(R3) ;TEST THE MOV - SM2,DM1
16453
16454 034542 100403 BMI E10456 ;N:C = 0101 ?
16455 034544 001002 BNE E10456
16456 034546 102401 BVS E10456
16457 034550 103402 BCS A0456
16458
16459 034552 104005 E10456: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
16460 034554 034526 R0456 ;ERROR LOOP RETURN
16461
16462 034556 020412 A0456: CMP R4,(R2) ;RESULT CORRECT ??
16463 034560 001404 BEQ 00456 ;BR IF YES
16464
16465 034562 005003 CLR R3 ;GET THE WAS DATA
16466 034564 051203 BIS (R2),R3
16467 034566 104000 E20456: ERROR ;MOV DELIVERED THE WRONG RESULT
16468 034570 034526 R0456 ;ERROR LOOP RETURN
16469
16470 034572 000004 00456: SCOPE ;CALL SCOPE LOOP UTILITY
16471
```

```
16472 ; *****
16473 ; .SBTTL T0457 MOV SM1,DM2 TEST - <N:C> = 0100
16474 ; *****
16475 ;MICROPROGRAMMING / LOGIC INFORMATION
16476 ;ROM SEQ: [141,247,250,172,257,200,125,375,016] FC 1,2,4,8
16477 ;ACT BUTS: 37[004]100,141 / 35[247]120,172 / 22[172]200,200 / 16[125]016,016
16478 ;EXEC: [200]ALUC=LLLLL :[125] D = 177777
16479 ;CODES: [125] SPS=3 / N:C = 1000
16480 ;SYNC: B05J2 (-) T = 3.25 USEC
16481 ;KEY SIG: K3-3 SM=1L / K3-3 DM=2L / K3-3 MOV L / K5-5 BCON (1+2) H
16482
16483
16484
16485
16486
16487
16488
16489
16490 034574 012700 000457 T0457: MOV #0457,R0 ;LOAD R0 WITH TEST NO.
16491 034600 013701 034630 MOV @#10457,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16492 034604 005004 CLR R4 ;RESULT S / B = 177777
16493 034606 005104 COM R4
16494 034610 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
16495 034614 012705 067572 MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA
16496 034620 010203 R0457: MOV R2,R3 ;BASE DEST ADDR = MBUFO
16497 034622 005012 CLR (R2) ;MAKE [DEST] = 000000
16498 034624 000257 CCC ;CLEAR FLAGS
16499 034626 000264 264 ;N:C = 0100
16500
16501 034630 011523 I0457: MOV (R5),(R3)+ ;TEST THE MOV - SM1,DM2
16502
16503 034632 100003 BPL E10457 ;N:C = 1000 ?
16504 034634 001402 BEQ E10457
16505 034636 102401 BVS E10457
16506 034640 103002 BCC A0457
16507
16508 034642 104005 E10457: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
16509 034644 034620 R0457 ;ERROR LOOP RETURN
16510
16511 034646 022703 067562 A0457: CMP #MBUFO+2,R3 ;DID MOV INCREMENT DEST REG ?
16512 034652 001402 BEQ B0457 ;BR IF YES
16513
16514 034654 104005 E20457: ERROR5 ;MOV FAILED TO UPDATE DEST REG
16515 034656 034620 R0457 ;ERROR LOOP RETURN
16516
16517 034660 020412 B0457: CMP R4,(R2) ;RESULT CORRECT ??
16518 034662 001404 BEQ 00457 ;BR IF YES
16519
16520 034664 005003 CLR R3 ;GET THE WAS DATA
16521 034666 051203 BIS (R2),R3
16522 034670 104000 E30457: ERROR ;MOV DELIVERED THE WRONG RESULT
16523 034672 034620 R0457 ;ERROR LOOP RETURN
16524
16525 034674 000004 00457: SCOPE ;CALL SCOPE LOOP UTILITY
16526
```

```
16527 ; *****
16528 ; .SBTTL T0460 MOV SM2,DM2 TEST - <N:C> = 0100
16529 ; *****
16530
16531 ;MICROPROGRAMMING / LOGIC INFORMATION
16532
16533 ;ROM SEQ: [142,240,250,172,257,200,125,375,016] FC 1,2,4,8
16534
16535 ;ACT BUTS: 37[004]100,142 / 35[240]120,172 / 22[172]200,200 / 16[125]016,016
16536
16537 ;EXEC: [200]ALUC=LLLLL :[125] D = 177777
16538
16539 ;CODES: [125] SPS=3 / N:C = 1000
16540
16541 ;SYNC: B05J2 (-) T = 3.25 USEC
16542
16543 ;KEY SIG: K3-3 SM=2L / K3-3 DM=2L / K3-3 MOV L / K5-5 BCON (1+2) H
16544
16545 034676 012700 000460 T0460: MOV #0460,R0 ;LOAD R0 WITH TEST NO.
16546 034702 013701 034732 MOV @#I0460,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16547 034706 005004 CLR R4 ;RESULT S / B = 177777
16548 034710 005104 COM R4
16549 034712 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
16550 034716 012705 067572 MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA
16551 034722 010203 R0460: MOV R2,R3 ;BASE DEST ADDR = MBUFO
16552 034724 005012 CLR (R2) ;MAKE [DEST] = 000000
16553 034726 000257 CCC ;CLEAR FLAGS
16554 034730 000264 264 ;N:C = 0100
16555
16556 034732 012523 I0460: MOV (R5)+,(R3)+ ;TEST THE MOV - SM2,DM2
16557
16558 034734 100003 BPL E10460 ;N:C = 1000 ?
16559 034736 001402 BEQ E10460
16560 034740 102401 BVS E10460
16561 034742 103002 BCC A0460
16562
16563 034744 104005 E10460: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
16564 034746 034722 R0460 ;ERROR LOOP RETURN
16565
16566 034750 022703 067562 A0460: CMP #MBUFO+2,R3 ;DID MOV INCREMENT DEST REG ?
16567 034754 001402 BEQ B0460 ;BR IF YES
16568
16569 034756 104005 E20460: ERROR5 ;MOV FAILED TO UPDATE DEST REG
16570 034760 034722 R0460 ;ERROR LOOP RETURN
16571
16572 034762 020412 B0460: CMP R4,(R2) ;RESULT CORRECT ??
16573 034764 001404 BEQ 00460 ;BR IF YES
16574
16575 034766 005003 CLR R3 ;GET THE WAS DATA
16576 034770 051203 BIS (R2),R3
16577 034772 104000 E30460: ERROR ;MOV DELIVERED THE WRONG RESULT
16578 034774 034722 R0460 ;ERROR LOOP RETURN
16579
16580 034776 000004 00460: SCOPE ;CALL SCOPE LOOP UTILITY
16581
```

```
16582 ; *****
16583 ; .SBTTL T0461 MOV SM1,DM2 TEST - <N:C> = 1011
16584 ; *****
16585
16586 ;MICROPROGRAMMING / LOGIC INFORMATION
16587
16588 ;ROM SEQ: [141,247,250,172,257,200,125,375,016] FC 1,2,4,8
16589
16590 ;ACT BUTS: 37[004]100,141 / 35[247]120,172 / 22[172]200,200 / 16[125]016,016
16591
16592 ;EXEC: [200]ALUC=LLLLL :[125] D = 000000
16593
16594 ;CODES: [125] SPS=3 / N:C = 0101
16595
16596 ;SYNC: B05J2 (-) T = 3.25 USEC
16597
16598 ;KEY SIG: K3-3 SM=1L / K3-3 DM=2L / K3-3 MOV L / K5-5 BCON (1+2) H
16599
16600 035000 012700 000461 T0461: MOV #0461,R0 ;LOAD R0 WITH TEST NO.
16601 035004 013701 035034 MOV @#I0461,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16602 035010 005004 CLR R4 ;RESULT S / B = 000000
16603 035012 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
16604 035016 012705 067570 MOV #DWTA,R5 ;SOURCE ADDR = DWTA
16605 035022 010203 R0461: MOV R2,R3 ;BASE DEST ADDR = MBUFO
16606 035024 005012 CLR (R2) ;MAKE [DEST] = 177777
16607 035026 005112 COM (R2)
16608 035030 000257 CCC ;CLEAR FLAGS
16609 035032 000273 273 ;N:C = 1011
16610
16611 035034 011523 I0461: MOV (R5),(R3)+ ;TEST THE MOV - SM1,DM2
16612
16613 035036 100403 BMI E10461 ;N:C = 0101 ?
16614 035040 001002 BNE E10461
16615 035042 102401 BVS E10461
16616 035044 103402 BCS A0461
16617
16618 035046 104005 E10461: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
16619 035050 035022 R0461 ;ERROR LOOP RETURN
16620
16621 035052 022703 067562 A0461: CMP #MBUFO+2,R3 ;DID MOV INCREMENT DEST REG ?
16622 035056 001402 BEQ B0461 ;BR IF YES
16623
16624 035060 104005 E20461: ERROR5 ;MOV FAILED TO UPDATE DEST REG
16625 035062 035022 R0461 ;ERROR LOOP RETURN
16626
16627 035064 020412 B0461: CMP R4,(R2) ;RESULT CORRECT ??
16628 035066 001404 BEQ 00461 ;BR IF YES
16629
16630 035070 005003 CLR R3 ;GET THE WAS DATA
16631 035072 051203 BIS (R2),R3
16632 035074 104000 E30461: ERROR ;MOV DELIVERED THE WRONG RESULT
16633 035076 035022 R0461 ;ERROR LOOP RETURN
16634
16635 035100 000004 00461: SCOPE ;CALL SCOPE LOOP UTILITY
16636
```

```

16637 ; *****
16638 ; .SBTTL T0462 MOV SM2,DM2 TEST - <N:C> = 1011
16639 ; *****
16640
16641 ;MICROPROGRAMMING / LOGIC INFORMATION
16642
16643 ;ROM SEQ: [142,240,250,172,257,200,125,375,016] FC 1,2,4,8
16644
16645 ;ACT BUTS: 37[004]100,142 / 35[240]120,172 / 22[172]200,200 / 16[125]016,016
16646
16647 ;EXEC: [200]ALUC=LLLLL :[125] D = 000000
16648
16649 ;CODES: [125] SPS=3 / N:C = 0101
16650
16651 ;SYNC: B05J2 (-) T = 3.25 USEC
16652
16653 ;KEY SIG: K3-3 SM=2L / K3-3 DM=2L / K3-3 MOV L / K5-5 BCON (1+2) H
16654
16655 035102 012700 000462 T0462: MOV #0462,R0 ;LOAD R0 WITH TEST NO.
16656 035106 013701 035136 MOV @#10462,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16657 035112 005004 CLR R4 ;RESULT S / B = 000000
16658 035114 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
16659 035120 012705 067570 MOV #DWTA,R5 ;SOURCE ADDR = DWTA
16660 035124 010203 R0462: MOV R2,R3 ;BASE DEST ADDR = MBUF0
16661 035126 005012 CLR (R2) ;MAKE [DEST] = 177777
16662 035130 005112 COM (R2)
16663 035132 000257 CCC ;CLEAR FLAGS
16664 035134 000273 273 ;N:C = 1011
16665
16666 035136 012523 I0462: MOV (R5)+,(R3)+ ;TEST THE MOV - SM2,DM2
16667
16668 035140 100403 BMI E10462 ;N:C = 0101 ?
16669 035142 001002 BNE E10462
16670 035144 102401 BVS E10462
16671 035146 103402 BCS A0462
16672
16673 035150 104005 E10462: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
16674 035152 035124 R0462 ;ERROR LOOP RETURN
16675
16676 035154 022703 067562 A0462: CMP #MBUF0+2,R3 ;DID MOV INCREMENT DEST REG ?
16677 035160 001402 BEQ B0462 ;BR IF YES
16678
16679 035162 104005 E20462: ERROR5 ;MOV FAILED TO UPDATE DEST REG
16680 035164 035124 R0462 ;ERROR LOOP RETURN
16681
16682 035166 020412 B0462: CMP R4,(R2) ;RESULT CORRECT ??
16683 035170 001404 BEQ 00462 ;BR IF YES
16684
16685 035172 005003 CLR R3 ;GET THE WAS DATA
16686 035174 051203 BIS (R2),R3
16687 035176 104000 E30462: ERROR ;MOV DELIVERED THE WRONG RESULT
16688 035200 035124 R0462 ;ERROR LOOP RETURN
16689
16690 035202 000004 00462: SCOPE ;CALL SCOPE LOOP UTILITY
16691
    
```

16692
16693
16694
16695
16696
16697
16698
16699
16700
16701
16702
16703
16704
16705
16706
16707
16708
16709
16710
16711
16712
16713
16714
16715
16716
16717
16718
16719
16720
16721
16722
16723
16724
16725
16726
16727
16728
16729
16730
16731
16732
16733
16734
16735
16736
16737
16738
16739
16740
16741
16742
16743
16744
16745
16746

035204 012700 000463
035210 013701 035242
035214 005004
035216 005104
035220 012702 067560
035224 012705 067572
035230 012703 067554
035234 005012
035236 000257
035240 000264

035242 011533

035244 100003
035246 001402
035250 102401
035252 103002

035254 104005
035256 035230

035260 022703 067556
035264 001402

035266 104005
035270 035230

035272 020412
035274 001404

035276 005003
035300 051203
035302 104000
035304 035230

035306 000004

; *****
.SBTTL T0463 MOV SM1,DM3 TEST - <N:C> = 0100
; *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [141,247,250,173,207,2010,200,125,375,016] FC 1,2,4,8
:ACT BUTS: 37[004]100,141 / 35[247]120,173 / 22[207]200,200 / 16[125]016,016
:EXEC: [200]ALUC=L L L L L :[125] D = 177777
:CODES: [125] SPS=3 / N:C = 1000
:SYNC: B05J2 (-) T = 4 USEC
:KEY SIG: K3-3 SM=1L / K3-3 DM=3L / K3-3 MOV L / K5-5 BC01 H

T0463: MOV #0463,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0463,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
R0463: MOV #ATA+10,R3 ;BASE DEST ADDR = ATA+10
CLR (R2) ;MAKE [DEST] = 000000
CCC ;CLEAR FLAGS
264 ;N:C = 0100

I0463: MOV (R5),a(R3)+ ;TEST THE MOV - SM1,DM3

BPL E10463 ;N:C = 1000 ?
BEQ E10463
BVS E10463
BCC A0463

E10463: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
R0463 ;ERROR LOOP RETURN

A0463: CMP #ATA+12,R3 ;DID MOV INCREMENT DEST REG ?
BEQ B0463 ;BR IF YES

E20463: ERROR5 ;MOV FAILED TO UPDATE DEST REG
R0463 ;ERROR LOOP RETURN

B0463: CMP R4,(R2) ;RESULT CORRECT ??
BEQ 00463 ;BR IF YES

CLR R3 ;GET THE WAS DATA
BIS (R2),R3
E30463: ERROR ;MOV DELIVERED THE WRONG RESULT
R0463 ;ERROR LOOP RETURN

00463: SCOPE ;CALL SCOPE LOOP UTILITY

16747
16748
16749
16750
16751
16752
16753
16754
16755
16756
16757
16758
16759
16760
16761
16762
16763
16764
16765
16766
16767
16768
16769
16770
16771
16772
16773
16774
16775
16776
16777
16778
16779
16780
16781
16782
16783
16784
16785
16786
16787
16788
16789
16790
16791
16792
16793
16794
16795
16796
16797
16798
16799
16800
16801
16802

; *****
.SBTTL T0464 MOV SM2,DM3 TEST - <N:C> = 0100
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,173,207,210,200,125,375,016] FC 1,2,4,8
;ACT BUTS: 37[004]100,142 / 35[240]120,173 / 22[207]200,200 / 16[125]016,016
;EXEC: [200]ALUC=LLLLL :[125] D = 177777
;CODES: [125] SPS=3 / N:C = 1000
;SYNC: B05J2 (-) T = 4 USEC
;KEY SIG: K3-3 SM=2L / K3-3 DM=3L / K3-3 MOV L / K5-5 BC01 H
; K5-5 BCON (1+2) H

035310 012700 000464
035314 013701 035346
035320 005004
035322 005104
035324 012702 067560
035330 012705 067572
035334 012703 067554
035340 005012
035342 000257
035344 000264
035346 012533
035350 100003
035352 001402
035354 102401
035356 103002
035360 104005
035362 035334
035364 022703 067556
035370 001402
035372 104005
035374 035334
035376 020412
035400 001404
035402 005003
035404 051203
035406 104000
035410 035334
035412 000004

T0464: MOV #0464,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0464,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
R0464: MOV #ATA+10,R3 ;BASE DEST ADDR = ATA+10
CLR (R2) ;MAKE [DEST] = 000000
CCC ;CLEAR FLAGS
264 ;N:C = 0100
I0464: MOV (R5)+,@(R3)+ ;TEST THE MOV - SM2,DM3
BPL E10464 ;N:C = 1000 ?
BEQ E10464
BVS E10464
BCC A0464
E10464: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
R0464 ;ERROR LOOP RETURN
A0464: CMP #ATA+12,R3 ;DID MOV INCREMENT DEST REG ?
BEQ B0464 ;BR IF YES
E20464: ERROR5 ;MOV FAILED TO UPDATE DEST REG
R0464 ;ERROR LOOP RETURN
B0464: CMP R4,(R2) ;RESULT CORRECT ??
BEQ 00464 ;BR IF YES
E30464: CLR R3 ;GET THE WAS DATA
BIS (R2),R3
ERROR R0464 ;MOV DELIVERED THE WRONG RESULT
R0464 ;ERROR LOOP RETURN
00464: SCOPE ;CALL SCOPE LOOP UTILITY

16803
16804
16805
16806
16807
16808
16809
16810
16811
16812
16813
16814
16815
16816
16817
16818
16819
16820
16821 035414 012700 000465
16822 035420 013701 035452
16823 035424 005004
16824 035426 012702 067560
16825 035432 012705 067570
16826 035436 012703 067554
16827 035442 005012
16828 035444 005112
16829 035446 000257
16830 035450 000273
16831
16832 035452 011533
16833
16834 035454 100403
16835 035456 001002
16836 035460 102401
16837 035462 103402
16838
16839 035464 104005
16840 035466 035436
16841
16842 035470 022703 067556
16843 035474 001402
16844
16845 035476 104005
16846 035500 035436
16847
16848 035502 020412
16849 035504 001404
16850
16851 035506 005003
16852 035510 051203
16853 035512 104000
16854 035514 035436
16855
16856 035516 000004
16857

```

; *****
; .SBTTL T0465 MOV SM1,DM3 TEST - <N:C> = 1011
; *****

```

;MICROPROGRAMMING / LOGIC INFORMATION

```

;ROM SEQ:      [141,247,250,173,207,210,200,125,375,016] FC 1,2,4,8
;ACT BUTS:     37[004]100,141 / 35[247]120,173 / 22[207]200,200 / 16[125]016,016
;EXEC:         [200]ALUC-LLLLL :[125] D = 000000
;CODES:        [125] SPS=3 / N:C = 0101
;SYNC:         B05J2 (-) T = 4 USEC
;KEY SIG:      K3-3 SM=1L / K3-3 DM=3L / K3-3 MOV L / K5-5 BC01 H

```

```

T0465:  MOV #0465,R0          ;LOAD R0 WITH TEST NO.
        MOV @#I0465,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        CLR R4              ;RESULT S / B = 000000
        MOV #MBUF0,R2       ;DEST ADDR = MBUF0
        MOV #DWTA,R5        ;SOURCE ADDR = DWTA
R0465:  MOV #ATA+10,R3      ;BASE DEST ADDR = ATA+10
        CLR (R2)            ;MAKE [DEST] = 177777
        COM (R2)
        CCC                 ;CLEAR FLAGS
        273                 ;N:C = 1011

I0465:  MOV (R5),@(R3)+     ;TEST THE MOV - SM1,DM3
        BMI E10465         ;N:C = 0101 ?
        BNE E10465
        BVS E10465
        BCS A0465

E10465: ERROR5             ;MOV FAILED TO ALTER CODES PROPERLY
        R0465              ;ERROR LOOP RETURN

A0465:  CMP #ATA+12,R3     ;DID MOV INCREMENT DEST REG ?
        BEQ B0465         ;BR IF YES

E20465: ERROR5             ;MOV FAILED TO UPDATE DEST REG
        R0465              ;ERROR LOOP RETURN

B0465:  CMP R4,(R2)        ;RESULT CORRECT ??
        BEQ 00465         ;BR IF YES

E30465: ERROR             ;GET THE WAS DATA
        R3                 ;MOV DELIVERED THE WRONG RESULT
        BIS (R2),R3       ;ERROR LOOP RETURN
        R0465

00465:  SCOPE              ;CALL SCOPE LOOP UTILITY

```

16858
16859
16860
16861
16862
16863
16864
16865
16866
16867
16868
16869
16870
16871
16872
16873
16874
16875
16876
16877
16878
16879
16880
16881
16882
16883
16884
16885
16886
16887
16888
16889
16890
16891
16892
16893
16894
16895
16896
16897
16898
16899
16900
16901
16902
16903
16904
16905
16906
16907
16908
16909
16910
16911
16912
16913

035520 012700 000466
035524 013701 035556
035530 005004
035532 012702 067560
035536 012705 067570
035542 012703 067554
035546 005012
035550 005112
035552 000257
035554 000273

035556 012533

035560 100403
035562 001002
035564 102401
035566 103402

035570 104005
035572 035542

035574 022703 067556
035600 001402

035602 104005
035604 035542

035606 020412
035610 001404

035612 005003
035614 051203
035616 104000
035620 035542

035622 000004

; *****
; .SBTTL T0466 MOV SM2,DM3 TEST - <N:C> = 1011
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,173,207,210,200,125,375,016] FC 1,2,4,8
;ACT BUTS: 37[004]100,142 / 35[240]120,173 / 22[207]200,200 / 16[125]016,016
;EXEC: [200]ALUC=LLLLL :[125] D = 000000
;CODES: [125] SPS=3 / N:C = 0101
;SYNC: B05J2 (-) T = 4 USEC
;KEY SIG: K3-3 SM=2L / K3-3 DM=3L / K3-3 MOV L / K5-5 BC01 H
; K5-5 BCON (1+2) H

T0466: MOV #0466,R0 ;LOAD R0 WITH TEST NO.
MOV @#10466,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
R0466: MOV #ATA+10,R3 ;BASE DEST ADDR = ATA+10
CLR (R2) ;MAKE [DEST] = 177777
COM (R2)
CCC ;CLEAR FLAGS
273 ;N:C = 1011

I0466: MOV (R5)+,@(R3)+ ;TEST THE MOV - SM2,DM3

BMI E10466 ;N:C = 0101 ?
BNE E10466
BVS E10466
BCS A0466

E10466: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
R0466 ;ERROR LOOP RETURN

A0466: CMP #ATA+12,R3 ;DID MOV INCREMENT DEST REG ?
BEQ B0466 ;BR IF YES

E20466: ERROR5 ;MOV FAILED TO UPDATE DEST REG
R0466 ;ERROR LOOP RETURN

B0466: CMP R4,(R2) ;RESULT CORRECT ??
BEQ 00466 ;BR IF YES

E30466: CLR R3 ;GET THE WAS DATA
BIS (R2),R3
ERROR ;MOV DELIVERED THE WRONG RESULT
R0466 ;ERROR LOOP RETURN

00466: SCOPE ;CALL SCOPE LOOP UTILITY

```
16914 ; *****
16915 ; .SBTTL T0467 MOV SM1,DM4 TEST - <N:C> = 0100
16916 ; *****
16917
16918 ;MICROPROGRAMMING / LOGIC INFORMATION
16919
16920 ;ROM SEQ: [141,247,250,174,257,200,125,375,016] FC 1,2,4,8
16921
16922 ;ACT BUTS: 37[004]100,141 / 35[247]120,174 / 22[174]200,200 / 16[125]016,016
16923
16924 ;EXEC: [200]ALUC=LLLLL :[125] D = 177777
16925
16926 ;CODES: [125] SPS=3 / N:C = 1000
16927
16928 ;SYNC: B05J2 (-) T = 3.2 USEC
16929
16930 ;KEY SIG: K3-3 SM=1L / K3-3 DM=4L / K3-3 MOV L / K5-5 BCON (1+2) H
16931
16932 035624 012700 000467 T0467: MOV #0467,R0 ;LOAD R0 WITH TEST NO.
16933 035630 013701 035662 MOV @#I0467,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
16934 035634 005004 CLR R4 ;RESULT S / B = 177777
16935 035636 005104 COM R4
16936 035640 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
16937 035644 012705 067572 MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
16938 035650 012703 067562 R0467: MOV #MBUF0+2,R3 ;BASE DEST ADDR = MBUF0+2
16939 035654 005012 CLR (R2) ;MAKE [DEST] = 000000
16940 035656 000257 CCC ;CLEAR FLAGS
16941 035660 000264 264 ;N:C = 0100
16942
16943 035662 011543 I0467: MOV (R5),-(R3) ;TEST THE MOV - SM1,DM4
16944
16945 035664 100003 BPL E10467 ;N:C = 1000 ?
16946 035666 001402 BEQ E10467
16947 035670 102401 EVS E10467
16948 035672 103002 BCC A0467
16949
16950 035674 104005 E10467: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
16951 035676 035650 R0467 ;ERROR LOOP RETURN
16952
16953 035700 020203 A0467: CMP R2,R3 ;DID MOV DECREMENT DEST REG ?
16954 035702 001402 BEQ B0467 ;BR IF YES
16955
16956 035704 104005 E20467: ERROR5 ;MOV FAILED TO UPDATE DEST REG
16957 035706 035650 R0467 ;ERROR LOOP RETURN
16958
16959 035710 020412 B0467: CMP R4,(R2) ;RESULT CORRECT ??
16960 035712 001404 BEQ 00467 ;BR IF YES
16961
16962 035714 005003 CLR R3 ;GET THE WAS DATA
16963 035716 051203 BIS (R2),R3
16964 035720 104000 E30467: ERROR ;MOV DELIVERED THE WRONG RESULT
16965 035722 035650 R0467 ;ERROR LOOP RETURN
16966
16967 035724 000004 00467: SCOPE ;CALL SCOPE LOOP UTILITY
16968
```

16969
16970
16971
16972
16973
16974
16975
16976
16977
16978
16979
16980
16981
16982
16983
16984
16985
16986
16987 035726 012700 000470
16988 035732 013701 035764
16989 035736 005004
16990 035740 005104
16991 035742 012702 067560
16992 035746 012705 067572
16993 035752 012703 067562
16994 035756 005012
16995 035760 000257
16996 035762 000264
16997
16998 035764 012543
16999
17000 035766 100003
17001 035770 001402
17002 035772 102401
17003 035774 103002
17004
17005 035776 104005
17006 036000 035752
17007
17008 036002 020203
17009 036004 001402
17010
17011 036006 104005
17012 036010 035752
17013
17014 036012 020412
17015 036014 001404
17016
17017 036016 005003
17018 036020 051203
17019 036022 104000
17020 036024 035752
17021
17022 036026 000004
17023

; *****
.SBTTL T0470 MOV SM2,DM4 TEST - <N:C> = 0100
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,174,257,200,125,375,016] FC 1,2,4,8
;ACT BUTS: 37[004]100,142 / 35[240]140,174 / 22[174]200,200 / 16[125]016,016
;EXEC: [200]ALUC=LLLLL :[125] D = 177777
;CODES: [125] SPS=3 / N:C = 1000
;SYNC: B05J2 (-) T = 3.2 USEC
;KEY SIG: K3-3 SM=2L / K3-3 DM=4L / K3-3 MOV L / K5-5 BCON (1+2) H

T0470: MOV #0470,R0 ;LOAD R0 WITH TEST NO.
MOV @#10470,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
R0470: MOV #MBUFO+2,R3 ;BASE DEST ADDR = MBUFO+2
CLR (R2) ;MAKE [DEST] = 000000
CCC ;CLEAR FLAGS
264 ;N:C = 0100

I0470: MOV (R5)+,-(R3) ;TEST THE MOV - SM2,DM4

BPL E10470 ;N:C = 1000 ?
BEQ E10470
BVS E10470
BCC A0470

E10470: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
R0470 ;ERROR LOOP RETURN

A0470: CMP R2,R3 ;DID MOV INCREMENT DEST REG ?
BEQ B0470 ;BR IF YES

E20470: ERROR5 ;MOV FAILED TO UPDATE DEST REG
R0470 ;ERROR LOOP RETURN

B0470: CMP R4,(R2) ;RESULT CORRECT ??
BEQ 00470 ;BR IF YES

E30470: CLR R3 ;GET THE WAS DATA
BIS (R2),R3
ERROR ;MOV DELIVERED THE WRONG RESULT
R0470 ;ERROR LOOP RETURN

00470: SCOPE ;CALL SCOPE LOOP UTILITY

```
17024 ; *****
17025 ; .SBTTL T0471 MOV SM1,DM4 TEST - <N:C> = 1011
17026 ; *****
17027 ;MICROPROGRAMMING / LOGIC INFORMATION
17028 ;ROM SEQ: [141,247,250,174,257,200,125,375,016] FC 1,2,4,8
17029 ;ACT BUTS: 37[004]100,141 / 35[247]120,174 / 22[174]200,200 / 16[125]016,016
17030 ;EXEC: [200]ALUC=LLLLL :[125] D = 000000
17031 ;CODES: [125] SPS=3 / N:C = 0101
17032 ;SYNC: B05J2 (-) T = 3.2 USEC
17033 ;KEY SIG: K3-3 SM=1L / K3-3 DM=4L / K3-3 MOV L / K5-5 BCON (1+2) H
17034
17035
17036
17037
17038
17039
17040
17041
17042 036030 012700 000471 T0471: MOV #0471,R0 ;LOAD R0 WITH TEST NO.
17043 036034 013701 036066 MOV @#I0471,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
17044 036040 005004 CLR R4 ;RESULT S / B = 000000
17045 036042 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
17046 036046 012705 067570 MOV #DWTA,R5 ;SOURCE ADDR = DWTA
17047 036052 012703 067562 R0471: MOV #MBUFO+2,R3 ;BASE DEST ADDR = MBUFO+2
17048 036056 005012 CLR (R2) ;MAKE [DEST] = 177777
17049 036060 005112 COM (R2)
17050 036062 000257 CCC ;CLEAR FLAGS
17051 036064 000273 273 ;N:C = 1011
17052
17053 036066 011543 I0471: MOV (R5),-(R3) ;TEST THE MOV - SM1,DM4
17054
17055 036070 100403 BMI E10471 ;N:C = 0101 ?
17056 036072 001002 BNE E10471
17057 036074 102401 BVS E10471
17058 036076 103402 BCS A0471
17059
17060 036100 104005 E10471: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
17061 036102 036052 R0471 ;ERROR LOOP RETURN
17062
17063 036104 020203 A0471: CMP R2,R3 ;DID MOV INCREMENT DEST REG ?
17064 036106 001402 BEQ B0471 ;BR IF YES
17065
17066 036110 104005 E20471: ERROR5 ;MOV FAILED TO UPDATE DEST REG
17067 036112 036052 R0471 ;ERROR LOOP RETURN
17068
17069 036114 020412 B0471: CMP R4,(R2) ;RESULT CORRECT ??
17070 036116 001404 BEQ 00471 ;BR IF YES
17071
17072 036120 005003 CLR R3 ;GET THE WAS DATA
17073 036122 051203 BIS (R2),R3
17074 036124 104000 E30471: ERROR ;MOV DELIVERED THE WRONG RESULT
17075 036126 036052 R0471 ;ERROR LOOP RETURN
17076
17077 036130 000004 00471: SCOPE ;CALL SCOPE LOOP UTILITY
17078
```

17079
17080
17081
17082
17083
17084
17085
17086
17087
17088
17089
17090
17091
17092
17093
17094
17095
17096
17097
17098
17099
17100
17101
17102
17103
17104
17105
17106
17107
17108
17109
17110
17111
17112
17113
17114
17115
17116
17117
17118
17119
17120
17121
17122
17123
17124
17125
17126
17127
17128
17129
17130
17131
17132
17133

036132 012700 000472
036136 013701 036170
036142 005004
036144 012702 067560
036150 012705 067570
036154 012703 067562
036160 005012
036162 005112
036164 000257
036166 000273
036170 012543
036172 100403
036174 001002
036176 102401
036200 103402
036202 104005
036204 036154
036206 020203
036210 001402
036212 104005
036214 036154
036216 020412
036220 001404
036222 005003
036224 051203
036226 104000
036230 036154
036232 000004

```
; *****  
; .SBTTL T0472 MOV SM2,DM4 TEST - <N:C> = 1011  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [142,240,250,174,257,200,125,375,016] FC 1,2,4,8  
;ACT BUTS: 37[004]100,142 / 35[240]120,174 / 22[174]200,200 / 16[125]016,016  
;EXEC: [200]ALUC=LLLLL :[125] D = 000000  
;CODES: [125] SPS=3 / N:C = 0101  
;SYNC: B05J2 (-) T = 3.2 USEC  
;KEY SIG: K3-3 SM=2L / K3-3 DM=4L / K3-3 MOV L / K5-5 BCON (1+2) H  
T0472: MOV #0472,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0472,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
CLR R4 ;RESULT S / B = 000000  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #DWTA,R5 ;SOURCE ADDR = DWTA  
R0472: MOV #MBUF0+2,R3 ;BASE DEST ADDR = MBUF0+2  
CLR (R2) ;MAKE [DEST] = 177777  
COM (R2)  
CCC ;CLEAR FLAGS  
273 ;N:C = 1011  
I0472: MOV (R5)+,-(R3) ;TEST THE MOV - SM2,DM4  
BMI E10472 ;N:C = 0101 ?  
BNE E10472  
BVS E10472  
BCS A0472  
E10472: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY  
R0472 ;ERROR LOOP RETURN  
A0472: CMP R2,R3 ;DID MOV INCREMENT DEST REG ?  
BEQ B0472 ;BR IF YES  
E20472: ERROR5 ;MOV FAILED TO UPDATE DEST REG  
R0472 ;ERROR LOOP RETURN  
B0472: CMP R4,(R2) ;RESULT CORRECT ??  
BEQ 00472 ;BR IF YES  
CLR R3 ;GET THE WAS DATA  
BIS (R2),R3  
E30472: ERROR ;MOV DELIVERED THE WRONG RESULT  
R0472 ;ERROR LOOP RETURN  
00472: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
17134 ; *****  
17135 ; .SBTTL T0473 MOV SM1,DM5 TEST - <N:C> = 0100  
17136 ; *****  
17137  
17138 ;MICROPROGRAMMING / LOGIC INFORMATION  
17139  
17140 ;ROM SEQ: [141,247,250,175,207,210,200,125,375,016] FC 1,2,4,8  
17141  
17142 ;ACT BUTS: 37[004]100,141 / 35[247]120,175 / 22[207]200,200 / 16[125]016,016  
17143  
17144 ;EXEC: [200]ALUC=LLLLL :[125] D = 177777  
17145  
17146 ;CODES: [125] SPS=3 / N:C = 1000  
17147  
17148 ;SYNC: B05J2 (-) T = 4 USEC  
17149  
17150 ;KEY SIG: K3-3 SM=1L / K3-3 DM=5L / K3-3 MOV L / K5-5 BC01 H  
17151  
17152  
17153 036234 012700 000473 T0473: MOV #0473,R0 ;LOAD R0 WITH TEST NO.  
17154 036240 013701 036272 MOV @#I0473,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
17155 036244 005004 CLR R4 ;RESULT S / B = 177777  
17156 036246 005104 COM R4  
17157 036250 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
17158 036254 012705 067572 MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2  
17159 036260 012703 067556 R0473: MOV #ATA+12,R3 ;BASE DEST ADDR = ATA+12  
17160 036264 005012 CLR (R2) ;MAKE [DEST] = 000000  
17161 036266 000257 CCC ;CLEAR FLAGS  
17162 036270 000264 264 ;N:C = 0100  
17163  
17164 036272 011553 I0473: MOV (R5),@-(R3) ;TEST THE MOV - SM1,DM5  
17165  
17166 036274 100003 BPL E10473 ;N:C = 0100 ?  
17167 036276 001402 BEQ E10473  
17168 036300 102401 BVS E10473  
17169 036302 103002 BCC A0473  
17170  
17171 036304 104005 E10473: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY  
17172 036306 036260 R0473 ;ERROR LOOP RETURN  
17173  
17174 036310 022703 067554 A0473: CMP #ATA+10,R3 ;DID MOV DECREMENT DEST REG ?  
17175 036314 001402 BEQ B0473 ;BR IF YES  
17176  
17177 036316 104005 E20473: ERROR5 ;MOV FAILED TO UPDATE DEST REG  
17178 036320 036260 R0473 ;ERROR LOOP RETURN  
17179  
17180 036322 020412 B0473: CMP R4,(R2) ;RESULT CORRECT ??  
17181 036324 001404 BEQ 00473 ;BR IF YES  
17182  
17183 036326 005003 CLR R3 ;GET THE WAS DATA  
17184 036330 051203 BIS (R2),R3  
17185 036332 104000 E30473: ERROR ;MOV DELIVERED THE WRONG RESULT  
17186 036334 036260 R0473 ;ERROR LOOP RETURN  
17187  
17188 036336 000004 00473: SCOPE ;CALL SCOPE LOOP UTILITY  
17189
```

```
17190 ; *****
17191 ; .SBTTL T0474 MOV SM2,DM5 TEST - <N:C> = 0100
17192 ; *****
17193
17194 ;MICROPROGRAMMING / LOGIC INFORMATION
17195
17196 ;ROM SEQ: [142,240,250,175,207,210,200,125,375,016] FC 1,2,4,8
17197
17198 ;ACT BUTS: 37[004]100,142 / 35[240]120,175 / 22[207]200,200 / 16[125]016,016
17199
17200 ;EXEC: [200]ALUC=LLLLL :[125] D = 177777
17201
17202 ;CODES: [125] SPS=3 / N:C = 1000
17203
17204 ;SYNC: B05J2 (-) T = 4 USEC
17205
17206 ;KEY SIG: K3-3 SM=2L / K3-3 DM=5L / K3-3 MOV L / K5-5 BC01 H
17207 ; K5-5 BCON (1+2) H
17208
17209 036340 012700 000474 T0474: MOV #0474,R0 ;LOAD R0 WITH TEST NO.
17210 036344 013701 036376 MOV @#I0474,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
17211 036350 005004 CLR R4 ;RESULT S / B = 177777
17212 036352 005104 COM R4
17213 036354 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO
17214 036360 012705 067572 MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
17215 036364 012703 067556 R0474: MOV #ATA+12,R3 ;BASE DEST ADDR = ATA+12
17216 036370 005012 CLR (R2) ;MAKE [DEST] = 00000
17217 036372 000257 CCC ;CLEAR FLAGS
17218 036374 000264 264 ;N:C = 1000
17219
17220 036376 012553 I0474: MOV (R5)+,@-(R3) ;TEST THE MOV - SM2,DM5
17221
17222 036400 100003 BPL E10474 ;N:C = 1000 ?
17223 036402 001402 BEQ E10474
17224 036404 102401 BVS E10474
17225 036406 103002 BCC A0474
17226
17227 036410 104005 E10474: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
17228 036412 036364 R0474 ;ERROR LOOP RETURN
17229
17230 036414 022703 067554 A0474: CMP #ATA+10,R3 ;DID MOV DECREMENT DEST REG ?
17231 036420 001402 BEQ B0474 ;BR IF YES
17232
17233 036422 104005 E20474: ERROR5 ;MOV FAILED TO UPDATE DEST REG
17234 036424 036364 R0474 ;ERROR LOOP RETURN
17235
17236 036426 020412 B0474: CMP R4,(R2) ;RESULT CORRECT ??
17237 036430 001404 BEQ 00474 ;BR IF YES
17238
17239 036432 005003 CLR R3 ;GET THE WAS DATA
17240 036434 051203 BIS (R2),R3
17241 036436 104000 E30474: ERROR ;MOV DELIVERED THE WRONG RESULT
17242 036440 036364 R0474 ;ERROR LOOP RETURN
17243
17244 036442 000004 00474: SCOPE ;CALL SCOPE LOOP UTILITY
17245
```

```

17246 ; *****
17247 ; .SBTTL T0475 MOV SM1,DM5 TEST - <N:C> = 1011
17248 ; *****
17249
17250 ;MICROPROGRAMMING / LOGIC INFORMATION
17251
17252 ;ROM SEQ: [141,247,250,175,207,210,200,125,375,016] FC 1,2,4,8
17253
17254 ;ACT BUTS: 37[004]100,141 / 35[247]120,175 / 22[207]200,200 / 16[125]016,016
17255
17256 ;EXEC: [200]JALUC=LLLLL :[125] D = 000000
17257
17258 ;CODES: [125] SPS=3 / N:C = 0101
17259
17260 ;SYNC: B05J2 (-) T = 4 USEC
17261
17262 ;KEY SIG: K3-3 SM=1L / K3-3 DM=5L / K3-3 MOV L / K5-5 BC01 H
17263
17264 036444 012700 000475 T0475: MOV #0475,R0 ;LOAD R0 WITH TEST NO.
17265 036450 013701 036502 MOV @#I0475,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
17266 036454 005004 CLR R4 ;RESULT S / B = 000000
17267 036456 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
17268 036462 012705 067570 MOV #DWTA,R5 ;SOURCE ADDR = DWTA
17269 036466 012703 067556 R0475: MOV #ATA+12,R3 ;BASE DEST ADDR = ATA+12
17270 036472 005012 CLR (R2) ;MAKE [DEST] = 177777
17271 036474 005112 COM (R2)
17272 036476 000257 CCC ;CLEAR FLAGS
17273 036500 000273 273 ;N:C = 1011
17274
17275 036502 011553 I0475: MOV (R5),@-(R3) ;TEST THE MOV - SM1,DM5
17276
17277 036504 100403 BMI E10475 ;N:C = 0101 ?
17278 036506 001002 BNE E10475
17279 036510 102401 BVS E10475
17280 036512 103402 BCS A0475
17281
17282 036514 104005 E10475: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
17283 036516 036466 R0475 ;ERROR LOOP RETURN
17284
17285 036520 022703 067554 A0475: CMP #ATA+10,R3 ;DID MOV DECREMENT DEST REG ?
17286 036524 001402 BEQ B0475 ;BR IF YES
17287
17288 036526 104005 E20475: ERROR5 ;MOV FAILED TO UPDATE DEST REG
17289 036530 036466 R0475 ;ERROR LOOP RETURN
17290
17291 036532 020412 B0475: CMP R4,(R2) ;RESULT CORRECT ??
17292 036534 001404 BEQ 00475 ;BR IF YES
17293
17294 036536 005003 CLR R3 ;GET THE WAS DATA
17295 036540 051203 BIS (R2),R3
17296 036542 104000 E30475: ERROR ;MOV DELIVERED THE WRONG RESULT
17297 036544 036466 R0475 ;ERROR LOOP RETURN
17298
17299 036546 000004 00475: SCOPE ;CALL SCOPE LOOP UTILITY
17300

```

```
17301 ; *****  
17302 ; .SBTTL T0476 MOV SM2,DM5 TEST - <N:C> = 1011  
17303 ; *****  
17304  
17305 ;MICROPROGRAMMING / LOGIC INFORMATION  
17306  
17307 ;ROM SEQ: [142,240,250,175,207,210,200,125,375,016] FC 1,2,4,8  
17308  
17309 ;ACT BUTS: 37[004]100,142 / 35[240]120,175 / 22[207]200,200 / 16[125]016,016  
17310  
17311 ;EXEC: [200]ALUC=LLLLL :[125] D = 000000  
17312  
17313 ;CODES: [125] SPS=3 / N:C = 0101  
17314  
17315 ;SYNC: B05J2 (-) T = 4 USEC  
17316  
17317 ;KEY SIG: K3-3 SM=2L / K3-3 DM=5L / K3-3 MOV L / K5-5 BC01 H  
17318 ; K5-5 BCON (1+2) H  
17319  
17320 036550 012700 000476 T0476: MOV #0476,R0 ;LOAD R0 WITH TEST NO.  
17321 036554 013701 036606 MOV @#I0476,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
17322 036560 005004 CLR R4 ;RESULT S / B = 000000  
17323 036562 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
17324 036566 012705 067570 MOV #DWTA,R5 ;SOURCE ADDR = DWTA  
17325 036572 012703 067556 R0476: MOV #ATA+12,R3 ;BASE DEST ADDR = ATA+12  
17326 036576 005012 CLR (R2) ;MAKE [DEST] = 177777  
17327 036600 005112 COM (R2)  
17328 036602 000257 CCC ;CLEAR FLAGS  
17329 036604 000273 273 ;N:C = 1011  
17330  
17331 036606 011553 I0476: MOV (R5),@-(R3) ;TEST THE MOV - SM2,DM5  
17332  
17333 036610 100403 BMI E10476 ;N:C = 0101 ?  
17334 036612 001002 BNE E10476  
17335 036614 102401 BVS E10476  
17336 036616 103402 BCS A0476  
17337  
17338 036620 104005 E10476: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY  
17339 036622 036572 R0476 ;ERROR LOOP RETURN  
17340  
17341 036624 022703 067554 A0476: CMP #ATA+10,R3 ;DID MOV DECREMENT DEST REG ?  
17342 036630 001402 BEQ B0476 ;BR IF YES  
17343  
17344 036632 104005 E20476: ERROR5 ;MOV FAILED TO UPDATE DEST REG  
17345 036634 036572 R0476 ;ERROR LOOP RETURN  
17346  
17347 036636 020412 B0476: CMP R4,(R2) ;RESULT CORRECT ??  
17348 036640 001404 BEQ 00476 ;BR IF YES  
17349  
17350 036642 005003 CLR R3 ;GET THE WAS DATA  
17351 036644 051203 BIS (R2),R3  
17352 036646 104000 E30476: ERROR ;MOV DELIVERED THE WRONG RESULT  
17353 036650 036572 R0476 ;ERROR LOOP RETURN  
17354  
17355 036652 000004 00476: SCOPE ;CALL SCOPE LOOP UTILITY  
17356
```

```

17357 ; *****
17358 ; .SBTTL T0477 MOV SM1,DM6 TEST - <N:C> = 0100
17359 ; *****
17360 ;MICROPROGRAMMING / LOGIC INFORMATION
17361 ;ROM SEQ: [141,247,250,177,206,212,200,125,375,016] FC 1,2,4,8
17362 ;ACT BUTS: 37[004]100,141 / 35[247]120,177 / 17[177]212,212 / 21[206]200,200
17363 ; / 16[125]016,016
17364 ;EXEC: [200]ALUC=LLLLL :[125] D = 177777
17365 ;CODES: [125] SPS=3 / N:C = 1000
17366 ;SYNC: B05J2 (-) T = 4 USEC
17367 ;KEY SIG: K3-3 SM=1L / K3-3 DM=6L K3-3 MOV L
17368
17369
17370
17371
17372
17373
17374
17375
17376 036654 012700 000477 T0477: MOV #0477,R0 ;LOAD R0 WITH TEST NO.
17377 036660 013701 036712 MOV @#I0477,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
17378 036664 005004 CLR R4 ;RESULT S / B = 177777
17379 036666 005104 COM R4
17380 036670 012702 067566 MOV #MBUF0+6,R2 ;DEST ADDR = MBUF0+6
17381 036674 012705 067572 MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2
17382 036700 012703 067560 R0477: MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0
17383 036704 005012 CLR (R2) ;MAKE [DEST] = 000000
17384 036706 000257 CCC ;CLEAR FLAGS
17385 036710 000264 264 ;N:C = 0100
17386
17387 036712 011563 000006 I0477: MOV (R5),6(R3) ;TEST THE MOV - SM1,DM6
17388
17389 036716 100003 BPL E10477 ;N:C = 1000 ?
17390 036720 001402 BEQ E10477
17391 036722 102401 BVS E10477
17392 036724 103002 BCC A0477
17393
17394 036726 104005 E10477: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
17395 036730 036700 R0477 ;ERROR LOOP RETURN
17396
17397 036732 020412 A0477: CMP R4,(R2) ;RESULT CORRECT ??
17398 036734 001404 BEQ 00477 ;BR IF YES
17399
17400 036736 005003 CLR R3 ;GET THE WAS DATA
17401 036740 051203 BIS (R2),R3
17402 036742 104000 E20477: ERROR ;MOV DELIVERED THE WRONG RESULT
17403 036744 036700 R0477 ;ERROR LOOP RETURN
17404
17405 036746 000004 00477: SCOPE ;CALL SCOPE LOOP UTILITY
17406

```

17407
17408
17409
17410
17411
17412
17413
17414
17415
17416
17417
17418
17419
17420
17421
17422
17423
17424
17425
17426 036750 012700 000500
17427 036754 013701 037006
17428 036760 005004
17429 036762 005104
17430 036764 012702 067566
17431 036770 012705 067572
17432 036774 012703 067560
17433 037000 005012
17434 037002 000257
17435 037004 000264
17436
17437 037006 012563 000006
17438
17439 037012 100003
17440 037014 001402
17441 037016 102401
17442 037020 103002
17443
17444 037022 104005
17445 037024 036774
17446
17447 037026 020412
17448 037030 001404
17449
17450 037032 005003
17451 037034 051203
17452 037036 104000
17453 037040 036774
17454
17455 037042 000004
17456

```

; *****
; .SBTTL T0500 MOV SM2,DM6 TEST - <N:C> = 0100
; *****

```

;MICROPROGRAMMING / LOGIC INFORMATION

```

;ROM SEQ:      [142,240,250,177,206,212,200,125,375,016] FC 1,2,4,8
;ACT BUTS:     37[004]100,142 / 35[240]120,177 / 17[177]212,212 / 21[206]200,200
;              : / 16[125]016,016
;EXEC:         [200]ALUC=LLLLL :[125] D = 177777
;CODES:        [125] SPS=3 / N:C = 1000
;SYNC:         B05J2 (-) T = 4 USEC
;KEY SIG:      K3-3 SM=1L / K3-3 DM=6L / K3-3 MOV L / K5-5 BCON (1+2) H

```

```

T0500:  MOV    #0500,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0500,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        CLR    R4              ;RESULT S / B = 177777
        COM    R4
        MOV    #MBUF0+6,R2     ;DEST ADDR = MBUF0+6
        MOV    #DWTA+2,R5     ;SOURCE ADDR = DWTA+2
R0500:  MOV    #MBUF0,R3       ;BASE DEST ADDR = MBUF0
        CLR    (R2)           ;MAKE [DEST] = 000000
        CCC
        264                   ;CLEAR FLAGS
        ;N:C = 0100

I0500:  MOV    (R5)+,6(R3)     ;TEST THE MOV - SM2,DM6
        BPL    E10500         ;N:C = 1000 ?
        BEQ    E10500
        BVS    E10500
        BCC    A0500

E10500:  ERROR5
        R0500                 ;MOV FAILED TO ALTER CODES PROPERLY
        ;ERROR LOOP RETURN

A0500:  CMP    R4,(R2)         ;RESULT CORRECT ??
        BEQ    00500          ;BR IF YES

        CLR    R3             ;GET THE WAS DATA
        BIS    (R2),R3

E20500:  ERROR
        R0500                 ;MOV DELIVERED THE WRONG RESULT
        ;ERROR LOOP RETURN

00500:  SCOPE                  ;CALL SCOPE LOOP UTILITY

```

17457
17458
17459
17460
17461
17462
17463
17464
17465
17466
17467
17468
17469
17470
17471
17472
17473
17474
17475
17476 037044 012700 000501
17477 037050 013701 037102
17478 037054 005004
17479 037056 012702 067566
17480 037062 012705 067570
17481 037066 012703 067560
17482 037072 005012
17483 037074 005112
17484 037076 000257
17485 037100 000273
17486
17487 037102 011563 000006
17488
17489 037106 100403
17490 037110 001002
17491 037112 102401
17492 037114 103402
17493
17494 037116 104005
17495 037120 037066
17496
17497 037122 020412
17498 037124 001404
17499
17500 037126 005003
17501 037130 051203
17502 037132 104000
17503 037134 037066
17504
17505 037136 000004
17506

; *****
; .SBTTL T0501 MOV SM1,DM6 TEST - <N:C> = 1011
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,177,206,212,200,125,375,016] FC 1,2,4,8
;ACT BUTS: 37[004]100,141 / 35[247]120,177 / 17[177]212,212 / 21[206]200,200
; / 16[125]016,016
;EXEC: [200]ALUC=L L L L L :[125] D = 000000
;CODES: [125] SPS=3 / N:C = 0101
;SYNC: B05J2 (-) T = 4 USEC
;KEY SIG: K3-3 SM=1L / K3-3 DM=6L / K3-3 MOV L

T0501: MOV #0501,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0501,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
CLR R4 ;RESULT S / B = 000000
MOV #MBUF0+6,R2 ;DEST ADDR = MBUF0+6
MOV #DWTA,R5 ;SOURCE ADDR = DWTA
R0501: MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0
CLR (R2) ;MAKE [DEST] = 177777
COM (R2)
CCC ;CLEAR FLAGS
273 ;N:C = 1011

I0501: MOV (R5),6(R3) ;TEST THE MOV - SM1,DM6
BMI E10501 ;N:C = 0101 ?
BNE E10501
BVS E10501
BCS A0501

E10501: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
R0501 ;ERROR LOOP RETURN

A0501: CMP R4,(R2) ;RESULT CORRECT ??
BEQ 00501 ;BR IF YES

E20501: CLR R3 ;GET THE WAS DATA
BIS (R2),R3
ERROR R0501 ;MOV DELIVERED THE WRONG RESULT
R0501 ;ERROR LOOP RETURN

00501: SCOPE ;CALL SCOPE LOOP UTILITY

```
17507 ; *****
17508 ; .SBTTL T0502 MOV SM2,DM6 TEST - <N:C> = 1011
17509 ; *****
17510
17511 ;MICROPROGRAMMING / LOGIC INFORMATION
17512
17513 ;ROM SEQ: [142,240,250,177,206,212,200,125,375,016] FC 1,2,4,8
17514
17515 ;ACT BUTS: 37[004]100,142 / 35[240]120,177 / 17[177]212,212 / 21[206]200,200
17516 ; / 16[125]016,016
17517
17518 ;EXEC: [200]ALUC=LLLLL :[125] D = 000000
17519
17520 ;CODES: [125] SPS=3 / N:C = 0101
17521
17522 ;SYNC: B05J2 (-) T = 4 USEC
17523
17524 ;KEY SIG: K3-3 SM=2L / K3-3 DM=6L / K3-3 MOV L / K5-5 BCON (1+2) H
17525
17526 037140 012700 000502 T0502: MOV #0502,R0 ;LOAD R0 WITH TEST NO.
17527 037144 013701 037176 MOV @#I0502,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
17528 037150 005004 CLR R4 ;RESULT S / B = 000000
17529 037152 012702 067566 MOV #MBUF0+6,R2 ;DEST ADDR = MBUF0+6
17530 037156 012705 067570 MOV #DWTA,R5 ;SOURCE ADDR = DWTA
17531 037162 012703 067560 R0502: MOV #MBUF0,R3 ;BASE DEST ADDR = MBUF0
17532 037166 005012 CLR (R2) ;MAKE [DEST] = 177777
17533 037170 005112 COM (R2)
17534 037172 000257 CCC ;CLEAR FLAGS
17535 037174 000273 273 ;N:C = 1011
17536
17537 037176 012563 000006 I0502: MOV (R5)+,6(R3) ;TEST THE MOV - SM2,DM6
17538
17539 037202 100403 BMI E10502 ;N:C = 0101 ?
17540 037204 001002 BNE E10502
17541 037206 102401 BVS E10502
17542 037210 103402 BCS A0502
17543
17544 037212 104005 E10502: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
17545 037214 037162 R0502 ;ERROR LOOP RETURN
17546
17547 037216 020412 A0502: CMP R4,(R2) ;RESULT CORRECT ??
17548 037220 001404 BEQ 00502 ;BR IF YES
17549
17550 037222 005003 CLR R3 ;GET THE WAS DATA
17551 037224 051203 BIS (R2),R3
17552 037226 104000 E20502: ERROR ;MOV DELIVERED THE WRONG RESULT
17553 037230 037162 R0502 ;ERROR LOOP RETURN
17554
17555 037232 000004 00502: SCOPE ;CALL SCOPE LOOP UTILITY
17556
```

```
17557 ; *****  
17558 ; .SBTTL T0503 MOV SM1,DM7 TEST - <N:C> = 0100  
17559 ; *****  
17560  
17561 ;MICROPROGRAMMING / LOGIC INFORMATION  
17562  
17563 ;ROM SEQ: [141,247,250,177,206,213,207,210,200,125,375,016] FC 1,2,4,8  
17564  
17565 ;ACT BUTS: 37[004]100,141 / 35[247]120,177 / 17[177]212,213 / 22[207]200,200  
17566 ; / 16[125]016,016  
17567  
17568 ;EXEC: [200]JALUC=LLLLL :[125] D = 177777  
17569  
17570 ;CODES: [125] SPS=3 / N:C = 1000  
17571  
17572 ;SYNC: B05J2 (-) T = 5 USEC  
17573  
17574 ;KEY SIG: K3-3 SM=1L / K3-3 DM=7L / K3-3 MOV L  
17575  
17576 037234 012700 000503 T0503: MOV #0503,R0 ;LOAD R0 WITH TEST NO.  
17577 037240 013701 037272 MOV @#I0503,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
17578 037244 005004 CLR R4 ;RESULT S / B = 177777  
17579 037246 005104 COM R4  
17580 037250 012702 067560 MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
17581 037254 012705 067572 MOV #DWTA+2,R5 ;SOURCE ADDR = DWTA+2  
17582 037260 012703 067544 R0503: MOV #ATA,R3 ;BASE DEST ADDR = ATA  
17583 037264 005012 CLR (R2) ;MAKE [DEST] = 000000  
17584 037266 000257 CCC ;CLEAR FLAGS  
17585 037270 000264 264 ;N:C = 0100  
17586  
17587 037272 011573 000010 I0503: MOV (R5),@10(R3) ;TEST THE MOV - SM1,DM7  
17588  
17589 037276 100003 BPL E10503 ;N:C = 1000 ?  
17590 037300 001402 BEQ E10503  
17591 037302 102401 BVS E10503  
17592 037304 103002 BCC A0503  
17593  
17594 037306 104005 E10503: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY  
17595 037310 037260 R0503 ;ERROR LOOP RETURN  
17596  
17597 037312 020412 A0503: CMP R4,(R2) ;RESULT CORRECT ??  
17598 037314 001404 BEQ 00503 ;BR IF YES  
17599  
17600 037316 005003 CLR R3 ;GET THE WAS DATA  
17601 037320 051203 BIS (R2),R3  
17602 037322 104000 E20503: ERROR ;MOV DELIVERED THE WRONG RESULT  
17603 037324 037260 R0503 ;ERROR LOOP RETURN  
17604  
17605 037326 000004 00503: SCOPE ;CALL SCOPE LOOP UTILITY  
17606
```

T0503 MOV SM1,DM7 TEST - <N:C> = 0100

SEQ 0443

17607
17608
17609
17610
17611
17612
17613
17614
17615
17616
17617
17618
17619
17620
17621
17622
17623
17624
17625
17626
17627
17628
17629
17630
17631
17632
17633
17634
17635
17636
17637
17638
17639
17640
17641
17642
17643
17644
17645
17646
17647
17648
17649
17650
17651
17652
17653
17654
17655
17656
17657
17658
17659

037330 012700 000504
037334 013701 037400
037340 032737 001000 066642
037346 001401
037350 000000
037352 005004
037354 005104
037356 012702 067560
037362 012705 067572
037366 012703 067544
037372 005012
037374 000257
037376 000264
037400 011573 000010
037404 100003
037406 001402
037410 102401
037412 103002
037414 104005
037416 037366
037420 020412
037422 001404
037424 005003
037426 051203
037430 104000
037432 037366
037434 000004

; *****
.SBTTL T0504 MOV SM2,DM7 TEST - <N:C> = 0100
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [142,240,250,177,206,213,207,210,200,125,375,016] FC 1,2,4,8
;ACT BUTS: 37[004]100,142 / 35[240]120,177 / 17[177]212,213 / 21[207]200,200
; / 16[125]016,016
;EXEC: [200]ALUC=LLLLL :[125] D = 177777
;CODES: [125] SPS=3 / N:C = 1000
;SYNC: B05J2 (-) T = 5 USEC
;KEY SIG: K3-3 SM=2L / K3-3 DM=7L / K3-3 MOV L / K5-5 BCON (1+2) H

T0504: MOV #0504,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0504,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
BIT #1000,@#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
R0504: MOV #ATA,R3 ;BASE DEST ADDR = ATA
CLR (R2) ;MAKE [DEST] = 000000
CCC ;CLEAR FLAGS
264 ;N:C = 0100
I0504: MOV (R5),@10(R3) ;TEST THE MOV - SM2,DM7
BPL E10504 ;N:C = 1000 ?
BEQ E10504
BVS E10504
BCC A0504
E10504: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
R0504 ;ERROR LOOP RETURN
A0504: CMP R4,(R2) ;RESULT CORRECT ??
BEQ 00504 ;BR IF YES
CLR R3 ;GET THE WAS DATA
BIS (R2),R3
E20504: ERROR ;MOV DELIVERED THE WRONG RESULT
R0504 ;ERROR LOOP RETURN
00504: SCOPE ;CALL SCOPE LOOP UTILITY

17660
17661
17662
17663
17664
17665
17666
17667
17668
17669
17670
17671
17672
17673
17674
17675
17676
17677
17678
17679
17680
17681
17682
17683
17684
17685
17686
17687
17688
17689
17690
17691
17692
17693
17694
17695
17696
17697
17698
17699
17700
17701
17702
17703
17704
17705
17706
17707
17708
17709

037436 012700 000505
037442 013701 037474
037446 005004
037450 012702 067560
037454 012705 067570
037460 012703 067544
037464 005012
037466 005112
037470 000257
037472 000273
037474 011573 000010
037500 100403
037502 001002
037504 102401
037506 103402
037510 104005
037512 037460
037514 020412
037516 001404
037520 005003
037522 051203
037524 104000
037526 037460
037530 000004

: *****
.SBTTL T0505 MOV SM1,DM7 TEST - <N:C> = 1011
: *****

;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ: [141,247,250,177,206,213,207,210,200,125,375,016] FC 1,2,4,8
;ACT BUTS: 37[004]100,141 / 35[247]120,177 / 17[177]212,213 / 22[207]200,200
: / 16[125]016,016
;EXEC: [200]ALUC=LLLLL :[125] D = 000000
;CODES: [125] SPS= 3 / N:C = 0101
;SYNC: B05J2 (-) T = 5 USEC
;KEY SIG: K3-3 SM=1L / K3-3 DM=7L / K3-3 MOV L

T0505: MOV #0505,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0505,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
R0505: MOV #ATA,R3 ;BASE DEST ADDR = ATA
CLR (R2) ;MAKE [DEST] = 177777
COM (R2)
CCC ;CLEAR FLAGS
273 ;N:C = 1011
I0505: MOV (R5),@10(R3) ;TEST THE MOV - SM1,DM7
BMI E10505 ;N:C = 0101 ?
BNE E10505
BVS E10505
BCS A0505
E10505: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
R0505 ;ERROR LOOP RETURN
A0505: CMP R4,(R2) ;RESULT CORRECT ??
BEQ 00505 ;BR IF YES
CLR R3 ;GET THE WAS DATA
BIS (R2),R3
E20505: ERROR ;MOV DELIVERED THE WRONG RESULT
R0505 ;ERROR LOOP RETURN
00505: SCOPE ;CALL SCOPE LOOP UTILITY

```
17710 ; *****
17711 ; .SBTTL T0506 MOV SM2,DM7 TEST - <N:C> = 1011
17712 ; *****
17713 ;MICROPROGRAMMING / LOGIC INFORMATION
17714 ;ROM SEQ: [142,240,250,177,206,213,207,210,200,125,375,016] FC 1,2,4,8
17715 ;ACT BUTS: 37[004]100,142 / 35[240]120,177 / 17[177]212,213 / 22[207]200,200
17716 ; / 16[125]016,016
17717 ;EXEC: [200]ALUC=LLLLL :[125] D = 000000
17718 ;CODES: [125] SPS=3 / N:C = 0101
17719 ;SYNC: B05J2 (-) T = 5 USEC
17720 ;KEY SIG: K3-3 SM=2L / K3-3 DM=7L / K3-3 MOV L / K5-5 BCON (1+2) H
17721
17722
17723
17724
17725
17726
17727
17728
17729 037532 012700 000506 T0506: MOV #0506,R0 ;LOAD R0 WITH TEST NO.
17730 037536 013701 037570 MOV @#I0506,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
17731 037542 005004 CLR R4 ;RESULT S / B = 000000
17732 037544 012702 067560 MOV #M0UF0,R2 ;DEST ADDR = M0UF0
17733 037550 012705 067570 MOV #DWTA,R5 ;SOURCE ADDR = DWTA
17734 037554 012703 067544 R0506: MOV #ATA,R3 ;BASE DEST ADDR = ATA
17735 037560 005012 CLR (R2) ;MAKE [DEST] = 177777
17736 037562 005112 COM (R2)
17737 037564 000257 CCC ;CLEAR FLAGS
17738 037566 000273 273 ;N:C = 1011
17739
17740 037570 011573 000010 I0506: MOV (R5),@10(R3) ;TEST THE MOV - SM2,DM7
17741
17742 037574 100403 BMI E10506 ;N:C = 0101 ?
17743 037576 001002 BNE E10506
17744 037600 102401 BVS E10506
17745 037602 103402 BCS A0506
17746
17747 037604 104005 E10506: ERROR5 ;MOV FAILED TO ALTER CODES PROPERLY
17748 037606 037554 R0506 ;ERROR LOOP RETURN
17749
17750 037610 020412 A0506: CMP R4,(R2) ;RESULT CORRECT ??
17751 037612 001404 BEQ 00506 ;BR IF YES
17752
17753 037614 005003 CLR R3 ;GET THE WAS DATA
17754 037616 051203 BIS (R2),R3
17755 037620 104000 E20506: ERROR ;MOV DELIVERED THE WRONG RESULT
17756 037622 037554 R0506 ;ERROR LOOP RETURN
17757
17758 037624 000004 00506: SCOPE ;CALL SCOPE LOOP UTILITY
17759
```

```

17760
17761
17762
17763
17764
17765
17766
17767
17768
17769
17770
17771
17772
17773
17774
17775
17776
17777
17778 037626 012700 000507
17779 037632 013701 037652
17780 037636 012702 067560
17781 037642 010004
17782 037644 010205
17783 037646 005012
17784 037650 000257
17785
17786 037652 010015
17787
17788 037654 020412
17789 037656 001403
17790
17791 037660 011203
17792 037662 104000
17793 037664 037644
17794
17795 037666 000004

```

```

; *****
; .SBTTL T0507 MOV SMO,DM1 TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ:      [171,257,201,125,375,016] FC 1,4,8

;ACT BUTS:     37[004]100,171 / 22[171]200,201 / 16[125]016,016

;EXEC:         [201]ALUC=LLLLL :[125]D=TEST NO.

;CODES:        [125] SPS=3      /      N:C=X000

;SYNC:         B05J2  (-)      T=2.42 USEC

;KEY SIG:      K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=1 L

T0507:  MOV      #0507,R0          ;LOAD R0 WITH TEST NO.
        MOV      @#I0507,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV      #MBUFO,R2      ;DEST ADDR = MBUFO
        MOV      R0,R4          ;RESULT S / B = #T0507
R0507:  MOV      R2,R5          ;R5 GETS DEST ADDR
        CLR      (R2)           ;[DEST] = 000000
        CCC                     ;SCOPE SYNC

I0507:  MOV      R0,(R5)         ;TEST THE MOV

        CMP      R4,(R2)        ;RESULT CORRECT ?
        BEQ      00507          ;BR IF YES

E0507:  MOV      (R2),R3        ;GET THE WAS DATA
        ERROR   R0507          ;MOV DELIVERED THE WRONG RESULT
        R0507          ;ERROR LOOP RETURN ADDRESS

00507:  SCOPE                   ;CALL THE SCOPE LOOP UTILITY

```

```
17796 ; *****  
17797 ; .SBTTL T0510 MOV SMO,DM2 TEST  
17798 ; *****  
17799  
17800 ;MICROPROGRAMMING / LOGIC INFORMATION  
17801  
17802 ;ROM SEQ: [172,257,201,125,375,016] FC 1,4,8  
17803  
17804 ;ACT BUTS: 37[004]100,172 / 22[172]200,201 / 16[125]016,016  
17805  
17806 ;EXEC: [201]ALUC=LLLLL :[125]D= TEST NO.  
17807  
17808 ;CODES: [125]SPS=3 / N:C=X000  
17809  
17810 ;SYNC: B05J2 (-) T=2.5 USEC  
17811  
17812 ;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=2 L / K5-5 BCON(1+2) H  
17813 ;K3-8 CIN00 L  
17814  
17815 037670 012700 000510 T0510: MOV #0510,R0 ;LOAD R0 WITH TEST NO.  
17816 037674 013701 037714 MOV @#I0510,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
17817 037700 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
17818 037704 010004 MOV R0,R4 ;RESULT S / B = #T0510  
17819 037706 010205 R0510: MOV R2,R5 ;R5 GETS DEST ADDR  
17820 037710 005012 CLR (R2) ;[DEST] = 000000  
17821 037712 000257 CCC ;SCOPE SYNC  
17822  
17823 037714 010025 I0510: MOV R0,(R5)+ ;TEST THE MOV  
17824  
17825 037716 020412 CMP R4,(R2) ;RESULT CORRECT ?  
17826 037720 001403 BEQ 00510 ;BR IF YES  
17827  
17828 037722 011203 MOV (R2),R3 ;GET THE WAS DATA  
17829 037724 104000 E0510: ERROR ;MOV DELIVERED THE WRONG RESULT  
17830 037726 037706 R0510 ;ERROR LOOP RETURN ADDRESS  
17831  
17832 037730 000004 00510: SCOPE ;CALL THE SCOPE LOOP UTILITY  
17833
```

```
17834 ; *****  
17835 ; .SBTTL T0511 MOV SMO,DM3 TEST  
17836 ; *****  
17837  
17838 ;MICROPROGRAMMING / LOGIC INFORMATION  
17839  
17840 ;ROM SEQ: [173,207,210,201,125,375,016] FC 1,4,8  
17841  
17842 ;ACT BUTS: 37[004]100,173 / 22[207]200,201 / 16[125]016,016  
17843  
17844 ;EXEC: [201]ALUC=LLLLL :[125]D= TEST NO.  
17845  
17846 ;CODES: [125]SPS=3 / N:C=X000  
17847  
17848 ;SYNC: B05J2 (-) T=3.2 USEC  
17849  
17850 ;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=3 L / K5-5 BC01 H  
17851  
17852 037732 012700 000511 T0511: MOV #0511,R0 ;LOAD R0 WITH TEST NO.  
17853 037736 013701 037760 MOV @#I0511,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
17854 037742 012702 067560 MOV #M0511,R2 ;DEST ADDR = M0511  
17855 037746 010004 MOV R0,R4 ;RESULT S / B = #T0511  
17856 037750 012705 067554 R0511: MOV #ATA+10,R5 ;BASE DEST ADDR = ATA+10  
17857 037754 005012 CLR (R2) ;[DEST] = 000000  
17858 037756 000257 CCC ;SCOPE SYNC  
17859  
17860 037760 010035 I0511: MOV R0,@(R5)+ ;TEST THE MOV  
17861  
17862 037762 020412 CMP R4,(R2) ;CORRECT RESULT  
17863 037764 001403 BEQ 00511 ;BR IF YES  
17864  
17865 037766 011203 MOV (R2),R3 ;GET THE WAS DATA  
17866 037770 104000 E0511: ERROR ;MOV DELIVERED THE WRONG RESULT  
17867 037772 037750 R0511 ;ERROR LOOP RETURN ADDRESS  
17868  
17869 037774 000004 00511: SCOPE ;CALL THE SCOPE LOOP UTILITY  
17870
```

```
17871 ; *****  
17872 ; .SBTTL T0512 MOV SMO,DM4 TEST  
17873 ; *****  
17874 ;MICROPROGRAMMING / LOGIC INFORMATION  
17875 ;ROM SEQ: [174,257,201,125,375,016] FC 1,4,8  
17876 ;ACT BUTS: 37[004]100,174 / 22[174]200,201 / 16[125]016,016  
17877 ;EXEC: [201]ALUC=LLLLL :[125]D= TEST NO.  
17878 ;CODES: [125]SPS=3 / N:C=X000  
17879 ;SYNC: B05J2 (-) T=2.5 USEC  
17880 ;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=4 L / K5-5 BCON(1+2) H  
17881 ;K3-8 CIN00 L  
17882  
17883  
17884  
17885  
17886  
17887  
17888  
17889  
17890 037776 012700 000512 T0512: MOV #0512,R0 ;LOAD R0 WITH TEST NO.  
17891 040002 013701 040024 MOV @#I0512,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
17892 040006 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
17893 040012 010004 MOV R0,R4 ;RESULT S / B = #T0512  
17894 040014 012705 067562 R0512: MOV #MBUF0+2,R5 ;R5 CONTAINS BASE DEST ADDR  
17895 040020 005012 CLR (R2) ;[DEST] = 000000  
17896 040022 000257 CCC ;SCOPE SYNC  
17897  
17898 040024 010045 I0512: MOV R0,-(R5) ;TEST THE MOV  
17899  
17900 040026 020412 CMP R4,(R2) ;CORRECT RESULT ?  
17901 040030 001403 BEQ 00512 ;BR IF YES  
17902  
17903 040032 011203 MOV (R2),R3 ;GET THE WAS DATA  
17904 040034 104000 E0512: ERROR ;MOV DELIVERED THE WRONG RESULT  
17905 040036 040014 R0512 ;ERROR LOOP RETURN ADDRESS  
17906  
17907 040040 000004 00512: SCOPE ;CALL THE SCOPE LOOP UTILITY  
17908
```

17909
17910
17911
17912
17913
17914
17915
17916
17917
17918
17919
17920
17921
17922
17923
17924
17925
17926
17927
17928
17929
17930
17931
17932
17933
17934
17935
17936
17937
17938
17939
17940
17941
17942
17943
17944
17945

040042 012700 000513
040046 013701 040070
040052 012702 067560
040056 010004
040060 012705 067556
040064 005012
040066 000257
040070 010055
040072 020412
040074 001403
040076 011203
040100 104000
040102 040060
040104 000004

```
; *****  
; .SBTTL T0513 MOV SMO,DM5 TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [175,207,210,201,125,375,016] FC 1,4,8  
;ACT BUTS: 37[004]100,175 / 22[207]200,201 / 16[125]016,016  
;EXEC: [201]ALUC=LLLLL :[125]D= TEST NO.  
;CODES: [125]SPS=3 / N:C=X000  
;SYNC: B05J2 (-) T= 3.2 USEC  
;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=5 L / K5-5 BC01 H  
T0513: MOV #0513,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0513,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV R0,R4 ;RESULT S / B = #T0513  
R0513: MOV #ATA+12,R5 ;R5 CONTAINS BASE DEST ADDR  
CLR (R2) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
I0513: MOV R0,@-(R5) ;TEST THE MOV  
CMP R4,(R2) ;CORRECT RESULT ?  
BEQ 00513 ;BR IF YES  
E0513: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;MOV DELIVERED THE WRONG RESULT  
R0513 ;ERROR LOOP RETURN ADDRESS  
00513: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

17946
17947
17948
17949
17950
17951
17952
17953
17954
17955
17956
17957
17958
17959
17960
17961
17962
17963
17964
17965
17966
17967
17968
17969
17970
17971
17972
17973
17974
17975
17976
17977
17978
17979
17980
17981
17982
17983

040106 012700 000514
040112 013701 040134
040116 012702 067564
040122 010004
040124 012705 067560
040130 005012
040132 000257
040134 010065 000004
040140 020412
040142 001403
040144 011203
040146 104000
040150 040130
040152 000004

```
; *****  
; .SBTTL T0514 MOV SMO,DM6 TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [176,206,212,201,125,375,016] FC 1,4,8  
;ACT BUTS: 37[004]100,176 / 17[176]212,212 / 21[206]200,201  
; / 16[125]016,016  
;EXEC: [201]ALUC=LLLLL :[125]D= TEST NO.  
;CODES: [125]SPS=3 / N:C=X000  
;SYNC: B05J2 (-) T= 2.84 USEC  
;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=6 L / K5-5 BC01 H  
;K4-4 OVLAP CYCLE L  
T0514: MOV #0514,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0514,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF1,R2 ;DEST ADDR = MBUF1  
MOV R0,R4 ;RESULT S / B = #T0514  
MOV #MBUF0,R5 ;BASE DEST ADDR = MBUF0  
R0514: CLR (R2) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
I0514: MOV R0,4(R5) ;TEST THE MOV  
CMP R4,(R2) ;RESULT CORRECT ?  
BEQ 00514 ;BR IF YES  
E0514: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;MOV DELIVERED THE WRONG RESULT  
R0514 ;ERROR LOOP RETURN ADDRESS  
00514: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

```
17984 ; *****  
17985 ; .SBTTL T0515 MOV B TEST - SMO,DM0 - EXTEND 1'S  
17986 ; *****  
17987  
17988 ;MICROPROGRAMMING / LOGIC INFORMATION  
17989  
17990 ;ROM SEQ: [170,204,003,204,001] FC 1,4  
17991  
17992 ;ACT BUTS: 37[004]100,170 / 20[170]000,003 / 27[003]000,001  
17993  
17994 ;EXEC: [170]ALUC=LLLLL :[204] AND TIME D = 177652  
17995  
17996 ;CODES: [204] SPS=3 / N:C = 1000  
17997  
17998 ;SYNC: B05J2 (-) T = 1.8 USEC  
17999  
18000 ;KEY SIG: K3-6 BYTE INSTR H / K3-3 MOV L / K3-3 SM=0L / K3-3 DM=0L  
18001  
18002 040154 012700 000515 T0515: MOV #0515,R0 ;LOAD R0 WITH TEST NO.  
18003 040160 013701 040206 MOV @#10515,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
18004 040164 012702 177703 MOV #177703,R2 ;DEST ADDR = R3  
18005 040170 012704 177652 MOV #177652,R4 ;RESULT S / B = 177652  
18006 040174 012705 000252 MOV #252,R5 ;SOURCE OP = 252  
18007 040200 005003 R0515: CLR R3 ;[DEST] = 000000  
18008 040202 000257 CCC ;CLEAR FLAGS  
18009 040204 000266 266 ;N:C = 0110  
18010  
18011 040206 110503 I0515: MOV B R5,R3 ;TEST THE MOV B  
18012  
18013 040210 100003 BPL E10515 ;N:C = 1000 ?  
18014 040212 001402 BEQ E10515  
18015 040214 102401 BVS E10515  
18016 040216 103002 BCC A0515  
18017  
18018 040220 104005 E10515: ERROR5 ;MOV B FAILED TO ALTER CODES PROPERLY  
18019 040222 040200 R0515 ;ERROR LOOP RETURN ADDRESS  
18020  
18021 040224 020403 A0515: CMP R4,R3 ;RESULT CORRECT ?  
18022 040226 001402 BEQ 00515 ;BR IF YES  
18023  
18024 040230 104000 E20515: ERROR ;MOV B DELIVERED THE WRONG RESULT  
18025 040232 040200 R0515 ;ERROR LOOP RETURN ADDRESS  
18026  
18027 040234 000004 00515: SCOPE ;CALL SCOPE LOOP UTILITY  
18028  
18029
```

18030
18031
18032
18033
18034
18035
18036
18037
18038
18039
18040
18041
18042
18043
18044
18045
18046
18047
18048 040236 012700 000516
18049 040242 013701 040270
18050 040246 012702 177703
18051 040252 005004
18052 040254 012705 177400
18053 040260 005003
18054 040262 005103
18055 040264 000257
18056 040266 000271
18057
18058 040270 110503
18059
18060 040272 100403
18061 040274 001002
18062 040276 102401
18063 040300 103402
18064
18065 040302 104005
18066 040304 040260
18067
18068 040306 020403
18069 040310 001402
18070
18071 040312 104000
18072 040314 040260
18073
18074 040316 000004
18075
18076

```

; *****
; .SBTTL T0516 MOV B TEST - SMO,DMO - EXTEND 0'S
; *****

```

;MICROPROGRAMMING / LOGIC INFORMATION

```

;ROM SEQ:      [170,204,003,204,001] FC 1,4
;ACT BUTS:     37[004]100,170 / 20[170]000,003 / 27[003]000,001
;EXEC:         [170]ALUC=LLLLL :[204] D = 000000
;CODES:        [204] SPS=3 / N:C = 0101
;SYNC:         B05J2 (-) T = 1.8 USEC
;KEY SIG:      K3-6 BYTE INSTR H / K3-3 MOV L

```

```

T0516:  MOV    #0516,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0516,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV    #177703,R2     ;DEST ADDR = R3
        CLR    R4              ;RESULT S / B = 000000
        MOV    #177400,R5     ;SOURCE OP = 177400
R0516:  CLR    R3              ;[DEST] = 177777
        COM    R3
        CCC
        271
        ;CLEAR FLAGS
        ;N:C = 1001

```

```

I0516:  MOV B  R5,R3          ;TEST THE MOV B
        BMI   E10516         ;N:C = 0101 ?
        BNE   E10516
        BVS   E10516
        BCS   A0516

```

```

E10516: ERROR5
        R0516
        ;MOV B FAILED TO ALTER CODES PROPERLY
        ;ERROR LOOP RETURN ADDRESS

```

```

A0516:  CMP    R4,R3          ;RESULT CORRECT ?
        BEQ   00516         ;BR IF YES

```

```

E20516: ERROR
        R0516
        ;MOV B DELIVERED THE WRONG RESULT
        ;ERROR LOOP RETURN ADDRESS

```

```

00516:  SCOPE
        ;CALL SCOPE LOOP UTILITY

```

```

18077
18078
18079
18080
18081
18082
18083
18084
18085
18086
18087
18088
18089
18090
18091
18092
18093
18094
18095 040320 012700 000517
18096 040324 013701 040350
18097 040330 012702 177703
18098 040334 005004
18099 040336 012705 070130
18100 040342 005003
18101 040344 005103
18102 040346 000257
18103
18104 040350 111503
18105
18106 040352 020403
18107 040354 001402
18108
18109 040356 104000
18110 040360 040342
18111
18112 040362 000004

```

```

; *****
; .SBTTL T0517 MOV B TEST - SM1,DMO - SOURCE ADDR EVEN
; *****
;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ:      [141,247,250,160,204,003,204,000] FC 1,2,4
;ACT BUTS:     37[004]140,141 / 35[247]120,160 / 20[160]000,003 / 27[003]000,000
;EXEC:        [160]ALUC=LLLLL :[204] 2ND TIME D = 000000
;CODES:       [204] SPS=3 / N:C = 0100
;SYNC:       B05J2 (-) T = 2 USEC
;KEY SIG:     K3-6 BYTE INSTR H / K3-3 MOV L / K3-3 SM=1L / K3-3 DM=0L
T0517:  MOV    #0517,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0517,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV    #177703,R2     ;DEST ADDR = R3
        CLR    R4              ;RESULT S / B = 000000
        MOV    #DBTA,R5       ;SOURCE ADDR = DBTA
R0517:  CLR    R3              ;[DEST] = 177777
        COM   R3
        CCC
        ;SCOPE SYNC
I0517:  MOV B   (R5),R3        ;TEST THE MOV B
        CMP   R4,R3          ;RESULT CORRECT ?
        BEQ  00517          ;BR IF YES
E0517:  ERROR
        R0517
        ;MOV B DELIVERED THE WRONG RESULT
        ;ERROR LOOP RETURN ADDRESS
00517:  SCOPE
        ;CALL SCOPE LOOP UTILITY

```

```
18113 ; *****  
18114 ; .SBTTL T0520 MOV B TEST - SM1,DMO - SOURCE ADDR ODD  
18115 ; *****  
18116  
18117 ;MICROPROGRAMMING / LOGIC INFORMATION  
18118  
18119 ;ROM SEQ: [141,247,250,137,251,160,204,003,204,000] FC 1,2,4  
18120  
18121 ;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 20[160]000,003 / 27[003]000,000  
18122  
18123 ;EXEC: [160]ALUC=LLLLL :[204](2ND TIME) D = 000125  
18124  
18125 ;CODES: [204] SPS=3 / N:C = 0000  
18126  
18127 ;SYNC: B05J2 (-) T = 2 USEC  
18128  
18129 ;KEY SIG: K3-6 BYTE INSTR H / K3-3 MOV L / K3-7 ODD BYTE H / K1-6 BA00 (1) H  
18130  
18131  
18132 040364 012700 000520 T0520: MOV #0520,R0 ;LOAD R0 WITH TEST NO.  
18133 040370 013701 040416 MOV @#I0520,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
18134 040374 012702 177703 MOV #177703,R2 ;DEST ADDR = R3  
18135 040400 012704 000125 MOV #125,R4 ;RESULT S / B = 125  
18136 040404 012705 070133 MOV #DBTA+3,R5 ;SOURCE ADDR = DBTA+3  
18137 040410 012703 177400 R0520: MOV #177400,R3 ;[DEST] = 177400  
18138 040414 000257 CCC ;SCOPE SYNC  
18139  
18140 040416 111503 I0520: MOV B (R5),R3 ;TEST THE MOV B  
18141  
18142 040420 020403 CMP R4,R3 ;RESULT CORRECT ?  
18143 040422 001402 BEQ 00520 ;BR IF YES  
18144  
18145 040424 104000 E0520: ERROR ;MOV B DELIVERED THE WRONG RESULT  
18146 040426 040410 R0520 ;ERROR LOOP RETURN ADDRESS  
18147  
18148 040430 000004 00520: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
18149 ; *****
18150 ; .SBTTL T0521 MOV B TEST - SM2,DMO - SOURCE ADDR ODD
18151 ; *****
18152
18153 ;MICROPROGRAMMING / LOGIC INFORMATION
18154
18155 ;ROM SEQ: [142,240,250,137,251,160,204,003,204,000] FC 1,2,4
18156
18157 ;ACT BUTS: 37[004]100,142 / 35[240]120,137 / 36[137]120,160 / 20[160]000,003
18158 ; / 27[003]000,000
18159
18160 ;EXEC: [160]ALUC=LLLLL :[204] 2ND TIME D = 177777
18161
18162 ;CODES: [204] SPS=3 / N:C = 1000
18163
18164 ;SYNC: B05J2 (-) T = 2.1 USEC
18165
18166 ;KEY SIG: K3-6 BYTE INSTR H / K3-3 MOV L / K5-5 BCON (1+2) H / K3-7 ODD BYTE
18167 ; K1-6 BA00 (1) H
18168
18169 040432 012700 000521 T0521: MOV #0521,R0 ;LOAD R0 WITH TEST NO.
18170 040436 013701 040462 MOV @#I0521,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
18171 040442 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
18172 040446 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777
18173 040452 012705 070131 R0521: MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
18174 040456 005003 CLR R3 ;[DEST] = 000000
18175 040460 000257 CCC ;SCOPE SYNC
18176
18177 040462 112503 I0521: MOV B (R5)+,R3 ;TEST THE MOV B
18178
18179 040464 020403 CMP R4,R3 ;RESULT CORRECT ?
18180 040466 001402 BEQ A0521 ;BR IF YES
18181
18182 040470 104000 E10521: ERROR ;MOV B DELIVERED THE WRONG RESULT
18183 040472 040452 R0521 ;ERROR LOOP RETURN ADDRESS
18184
18185 040474 022705 070132 A0521: CMP #DBTA+2,R5 ;DID MOV B INCREMENT SRC REG ?
18186 040500 001402 BEQ 00521 ;BR IF YES
18187
18188 040502 104005 E20521: ERROR5 ;MOV B FAILED TO UPDATE SRC REG
18189 040504 040452 R0521 ;ERROR LOOP RETURN ADDRESS
18190
18191 040506 000004 00521: SCOPE ;CALL SCOPE LOOP UTILITY
```

18192
18193
18194
18195
18196
18197
18198
18199
18200
18201
18202
18203
18204
18205
18206
18207
18208
18209
18210
18211
18212
18213
18214
18215
18216
18217
18218
18219
18220
18221
18222
18223
18224
18225
18226
18227
18228
18229
18230
18231
18232

040510 012700 000522
040514 013701 040540
040520 012702 177703
040524 005004
040526 012705 070130
040532 012703 177400
040536 000257
040540 112503
040542 020403
040544 001402
040546 104000
040550 040526
040552 022705 070131
040556 001402
040560 104005
040562 040526
040564 000004

```
; *****  
; .SBTTL T0522 MOV B TEST - SM2,DMO - SOURCE ADDR EVEN  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [142,240,250,160,204,003,204,000] FC 1,2,4  
;ACT BUTS: 37[004]100,142 / 35[240]120,160 / 20[160]000,003 / 27[003]000,000  
;EXEC: [160]ALUC=LLLLL :[204]2ND TIME D = 000000  
;CODES: [204] SPS=3 / N:C - 0100  
;SYNC: B05J2 (-) T = 2.1 USEC  
;KEY SIG: K3-6 BYTE INSTR H / K3-3 MOV L / K5-5 BCON (1+2) H  
T0522: MOV #0522,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0522,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = R3  
CLR R4 ;RESULT S / B = 000000  
R0522: MOV #DBTA,R5 ;SOURCE ADDR = DBTA  
MOV #177400,R3 ;[DEST] = 177400  
CCC ;SCOPE SYNC  
I0522: MOV B (R5)+,R3 ;TEST THE MOV B  
CMP R4,R3 ;RESULT CORRECT ?  
BEQ A0522 ;BR IF YES  
E10522: ERROR ;MOV B DELIVERED THE WRONG RESULT  
R0522 ;ERROR LOOP RETURN ADDRESS  
A0522: CMP #DBTA+1,R5 ;DID MOV B INCREMENT SRC REG ?  
BEQ 00522 ;BR IF YES  
E20522: ERROR5 ;MOV B FAILED TO UPDATE SOURCE REG  
R0522 ;ERROR LOOP RETURN ADDRESS  
00522: SCOPE ;CALL SCOPE LOOP UTILITY
```

18233
18234
18235
18236
18237
18238
18239
18240
18241
18242
18243
18244
18245
18246
18247
18248
18249
18250
18251
18252
18253
18254
18255
18256
18257
18258
18259
18260
18261
18262
18263
18264
18265
18266
18267
18268
18269
18270

040566 012700 000523
040572 013701 040616
040576 012702 067560
040602 012704 000377
040606 012705 070131
040612 005012
040614 000257
040616 111512
040620 020412
040622 001403
040624 011203
040626 104000
040630 040612
040632 000004

```
; *****  
; .SBTTL T0523 MOV B TEST - SM1,DM1 - SRC ADR ODD / DST ADR EVEN  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,137,251,171,257,202,205,125,375,016] FC 1,2,4,8  
;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,171 / 22[171]200,202  
; / 16[125]016,016  
;EXEC: [202]ALUC=LLLLL :[125] D = 177777  
;CODES: [125] SPS=3 / N:C = 1000  
;SYNC: B05J2 (-) T = 3.2 USEC  
;KEY SIG: K3-6 BYTE INSTR H / K3-3 MOV L / K3-3 SM=1L / K3-3 DM=1L  
; K1-6 BA00(1) H / K3-7 ODD BYTE H  
T0523: MOV #0523,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0523,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  
R0523: CLR (R2) ;[DEST] = 000000  
CCC ;CLEAR FLAGS - SCOPE SYNC  
I0523: MOV B (R5),(R2) ;TEST THE MOV B  
CMP R4,(R2) ;CORRECT RESULT ?  
BEQ 00523 ;BR IF YES  
E0523: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;MOV B DELIVERED WRONG RESULT  
R0523 ;ERROR LOOP RETURN ADDRESS  
00523: SCOPE ;CALL SCOPE LOOP UTILITY
```

18271
18272
18273
18274
18275
18276
18277
18278
18279
18280
18281
18282
18283
18284
18285
18286
18287
18288
18289
18290
18291
18292
18293
18294
18295
18296
18297
18298
18299
18300
18301
18302
18303
18304
18305
18306
18307
18308
18309
18310
18311
18312
18313
18314
18315

040634 012700 000524
040640 013701 040666
040644 012702 067560
040650 012704 000377
040654 012705 070131
040660 005012
040662 010203
040664 000257
040666 111523
040670 020412
040672 001403
040674 011203
040676 104000
040700 040660
040702 022703 067561
040706 001402
040710 104005
040712 040660
040714 000004

```
; *****  
; .SBTTL T0524 MOV B TEST - SM1,DM2 - SRC ADR ODD / DST ADR EVEN  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,137,251,172,257,202,205,125,375,016] FC 1,2,4,8  
;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,172 / 22[172]200,202  
; / 16[125]016,016  
;EXEC: [202]ALUC=LLLLL :[125] D = 177777  
;CODES: [125] SPS=3 / N:C = 1000  
;SYNC: B05J2 (-) T = 3.7 USEC  
;KEY SIG: K3-6 BYTE INSTR H / K3-3 MOV L / K5-5 BCON (1+2) H / K3-7 ODD BYTE  
; K1-6 BA00(1) H  
T0524: MOV #0524,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0524,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  
R0524: CLR (R2) ;[DEST] = 000000  
MOV R2,R3 ;[R3] = DEST ADDR  
CCC ;CLEAR FLAGS - SCOPE SYNC  
I0524: MOV B (R5),(R3)+ ;TEST THE MOV B  
CMP R4,(R2) ;CORRECT RESULT ?  
BEQ A0524 ;BR IF YES  
E10524: MOV (R2),R3 ;GET THE WAS DATA  
ERROR R0524 ;MOV B DELIVERED WRONG RESULT  
;ERROR LOOP RETURN ADDRESS  
A0524: CMP #MBUF0+1,R3 ;DID MOV B INCREMENT THE DEST REG ?  
BEQ 00524 ;BR IF YES  
E20524: ERROR5 ;MOV B FAILED TO UPDATE DEST REG  
R0524 ;ERROR LOOP RETURN ADDRESS  
00524: SCOPE ;CALL SCOPE LOOP UTILITY
```

18316
18317
18318
18319
18320
18321
18322
18323
18324
18325
18326
18327
18328
18329
18330
18331
18332
18333
18334
18335
18336
18337
18338
18339
18340
18341
18342
18343
18344
18345
18346
18347
18348
18349
18350
18351
18352
18353
18354
18355
18356
18357
18358
18359
18360
18361

040716 012700 000525
040722 013701 040752
040726 012702 067560
040732 012704 000377
040736 012705 070131
040742 005012
040744 012703 067554
040750 000257

040752 111533

040754 022703 067556
040760 001402

040762 104005
040764 040742

040766 020412
040770 001403

040772 011203
040774 104000
040776 040742

041000 000004

```
; *****  
; .SBTTL T0525 MOV B TEST - SM1,DM3 - SRC ADR ODD / DST ADR EVEN  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,137,251,173,207,210,202,205,125,375,016] FC 1,2,4,8  
;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,173 / 22[207]200,202  
; / 16[125]016,016  
;EXEC: [202]ALUC=LLLLL :[125] D = 177777  
;CODES: [125] SPS=3 / N:C = 1000  
;SYNC: B05J2 (-) T = 4 USEC  
;KEY SIG: K3-6 BYTE INSTR H / K3-3 MOV L / K5-5 BC01 H / K3-7 ODD BYTE H  
; K1-6 BA00 (1) H  
  
T0525: MOV #0525,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0525,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  
R0525: CLR (R2) ;[DEST] = 000000  
MOV #ATA+10,R3 ;BASE DEST ADDR = ATA +10  
CCC ;CLEAR FLAGS - SCOPE SYNC  
  
I0525: MOV B (R5),@ (R3)+ ;TEST THE MOV B  
  
CMP #ATA+12,R3 ;DID DEST REG GET INCREMENTED ?  
BEQ A0525 ;BR IF YES  
  
E10525: ERROR5 ;MOV B FAILED TO UPDATE DEST REG  
R0525 ;ERROR LOOP RETURN ADDRESS  
  
A0525: CMP R4,(R2) ;CORRECT RESULT ?  
BEQ 00525 ;BR IF YES  
  
E20525: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;MOV B DELIVERED WRONG RESULT  
R0525 ;ERROR LOOP RETURN ADDRESS  
  
00525: SCOPE ;CALL SCOPE LOOP UTILITY
```

18362
18363
18364
18365
18366
18367
18368
18369
18370
18371
18372
18373
18374
18375
18376
18377
18378
18379
18380
18381
18382
18383
18384
18385
18386
18387
18388
18389
18390
18391
18392
18393
18394
18395
18396
18397
18398
18399
18400
18401
18402
18403
18404
18405
18406

041002 012700 000526
041006 013701 041036
041012 012702 067560
041016 012704 000377
041022 012705 070131
041026 005012
041030 012703 067561
041034 000257

041036 111543

041040 020302
041042 001402

041044 104005
041046 041026

041050 020412
041052 001403

041054 011203
041056 104000
041060 041026

041062 000004

; *****
.SBTTL T0526 MOV B TEST - SM1,DM4 - SRC ADR ODD / DST ADR EVEN
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,137,251,174,257,202,205,125,375,016] FC 1,2,4,8
;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,174 / 22[174]200,202
; / 16[125]016,016
;EXEC: [202]ALUC=LLLLL :[125] D = 177777
;CODES: [125] SPS=3 / N:C = 1000
;SYNC: B05J2 (-) T = 3.1 USEC
;KEY SIG: K3-6 BYTE INSTR H / K3-3 MOV L / K5-5 BCON (1+2) H / K3-7 ODD BYTE
; K1-6 BA00(1) H

T0526: MOV #0526,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0526,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
R0526: CLR (R2) ;[DEST] = 000000
MOV #MBUF0+1,R3 ;INITIAL DEST ADDR = MBUF0+1
CCC ;CLEAR FLAGS - SCOPE SYNC

I0526: MOV B (R5),-(R3) ;TEST THE MOV B

CMP R3,R2 ;DID MOV B DECREMENT DEST REG ?
BEQ A0526 ;BR IF YES

E10526: ERROR5 ;MOV B FAILED TO UPDATE DEST REG
R0526 ;ERROR LOOP RETURN ADDRESS

A0526: CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00526 ;BR IF YES

E20526: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;MOV B DELIVERED WRONG RESULT
R0526 ;ERROR LOOP RETURN ADDRESS

00526: SCOPE ;CALL SCOPE LOOP UTILITY

18407
18408
18409
18410
18411
18412
18413
18414
18415
18416
18417
18418
18419
18420
18421
18422
18423
18424
18425
18426
18427
18428
18429
18430
18431
18432
18433
18434
18435
18436
18437
18438
18439
18440
18441
18442
18443
18444
18445
18446
18447
18448
18449
18450
18451

041064 012700 000527
041070 013701 041120
041074 012702 067560
041100 012704 000377
041104 012705 070131
041110 005012
041112 012703 067556
041116 000257
041120 111553
041122 022703 067554
041126 001402
041130 104005
041132 041110
041134 020412
041136 001403
041140 011203
041142 104000
041144 041110
041146 000004

; *****
; .SBTTL T0527 MOV B TEST - SM1,DM5 - SRC ADR ODD / DST ADR EVEN
; *****
; MICROPROGRAMMING / LOGIC INFORMATION
; ROM SEQ: [141,247,250,137,251,175,207,210,202,205,125,375,016] FC 1,2,4,8
; ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,175 / 22[207]200,202
; / 16[125]016,016
; EXEC: [202]ALUC=LLLLL :[125] D = 177777
; CODES: [125] SPS=3 / N:C = 1000
; SYNC: B05J2 (-) T = 4 USEC
; KEY SIG: K3-6 BYTE INSTR H / K3-3 MOV L / K5-5 BC01 H / K3-7 ODD BYTE H
; K1-6 BA00(1) H

T0527: MOV #0527,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0527,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
R0527: CLR (R2) ;[DEST] = 000000
MOV #ATA+12,R3 ;INITIAL DEST ADDR = ATA +12
CCC ;CLEAR FLAGS - SCOPE SYNC
I0527: MOV B (R5),@-(R3) ;TEST THE MOV B
CMP #ATA+10,R3 ;DID MOV B DECREMENT DEST REG ?
BEQ A0527 ;BR IF YES
E10527: ERROR5 ;MOV B FAILED TO UPDATE DEST REG
R0527 ;ERROR LOOP RETURN ADDRESS
A0527: CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00527 ;BR IF YES
E20527: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;MOV B DELIVERED WRONG RESULT
R0527 ;ERROR LOOP RETURN ADDRESS
00527: SCOPE ;CALL SCOPE LOOP UTILITY

18452
18453
18454
18455
18456
18457
18458
18459
18460
18461
18462
18463
18464
18465
18466
18467
18468
18469
18470
18471
18472 041150 012700 000530
18473 041154 013701 041204
18474 041160 012702 067560
18475 041164 012704 000377
18476 041170 012705 070131
18477 041174 005012
18478 041176 012703 067566
18479 041202 000257
18480
18481 041204 111563 177772
18482
18483 041210 020412
18484 041212 001403
18485
18486 041214 011203
18487 041216 104000
18488 041220 041174
18489
18490 041222 000004

```

; *****
; .SBTTL T0530 MOV B TEST - SM1,DM6 - SRC ADR ODD / DST ADR EVEN
; *****
;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ:      [141,247,250,137,251,177,206,212,202,205,125,375,016] FC 1,2,4,8
;ACT BUTS:     37[004]100,141 / 35[247]120,137 / 36[137]120,177 / 17[177]212,212
;              ; / 21[206]200,202 / 16[125]016,016
;EXEC:         [202]ALUC=LLLLL :[125] D = 177777
;CODES:        [125] SPS=3 / N:C = 1000
;SYNC:         B05J2 (-) T = 4.4 USEC
;KEY SIG:      K3-6 BYTE INSTR H / K3-3 MOV L / K5-5 BC01 H / K3-7 ODD BYTE H
;              ; K1-6 BA00(1) H

T0530:  MOV    #0530,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0530,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
R0530:  CLR    (R2)            ;[DEST] = 000000
        MOV    #MBUF0+6,R3    ;BASE DEST ADDR = MBUF0+6
        CCC                   ;CLEAR FLAGS - SCOPE SYNC

I0530:  MOV B  (R5),-6(R3)     ;TEST THE MOV B

        CMP    R4,(R2)        ;CORRECT RESULT ?
        BEQ   00530           ;BR IF YES

E0530:  MOV    (R2),R3        ;GET THE WAS DATA
        ERROR R0530          ;MOV B DELIVERED WRONG RESULT
        ERROR LOOP RETURN ADDRESS

00530:  SCOPE                 ;CALL THE SCOPE LOOP UTILITY

```

18491
18492
18493
18494
18495
18496
18497
18498
18499
18500
18501
18502
18503
18504
18505
18506
18507
18508
18509
18510
18511
18512
18513
18514
18515
18516
18517
18518
18519
18520
18521
18522
18523
18524
18525
18526
18527
18528
18529
18530
18531
18532

041224 012700 000531
041230 013701 041260
041234 012702 067560
041240 012704 000377
041244 012705 070131
041250 005012
041252 012703 067544
041256 000257
041260 111573 000010
041264 020412
041266 001403
041270 011203
041272 104000
041274 041250
041276 000004

; *****
.SBTTL T0531 MOV B TEST - SM1,DM7 - SRC ADR ODD / DST ADR EVEN
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,137,251,177,206,213,207,210,202,205,125,375,016] FC 1,2,4,8
;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,177 / 17[177]212,213
; / 21[206]XNU / 22[207]200,202 / 16[125]016,016
;EXEC: [202]ALUC=LLLLL :[125] D = 177777
;CODES: [125] SPS=3 / N:C - 1000
;SYNC: B05J2 (-) T = 4.8 USEC
;KEY SIG: K3-6 BYTE INSTR H / K3-3 MOV L / K5-5 BC01 H / K3-7 ODD BYTE H
; K1-6 BA00 (1) H

T0531: MOV #0531,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0531,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #M0531,R2 ;DEST ADDR = M0531
MOV #377,R4 ;RESULT S / B = 377
MOV #DBTA+1,R5 ;SRC ADDR = DBTA +1
R0531: CLR (R2) ;[DEST] = 000000
MOV #ATA,R3 ;BASE DEST ADDR = ATA
CCC ;CLEAR FLAGS - SCOPE SYNC
I0531: MOV B (R5),@10(R3) ;TEST THE MOV B
CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00531 ;BR IF YES
E0531: MOV (R2),R3 ;GET THE WAS DATA
ERROR R0531 ;MOV B DELIVERED WRONG RESULT
R0531 ;ERROR LOOP RETURN ADDRESS
00531: SCOPE ;CALL SCOPE LOOP UTILITY

18533
18534
18535
18536
18537
18538
18539
18540
18541
18542
18543
18544
18545
18546
18547
18548
18549
18550
18551 041300 012700 000532
18552 041304 013701 041332
18553 041310 012702 067560
18554 041314 012704 000377
18555 041320 012703 177777
18556 041324 010205
18557 041326 005012
18558 041330 000257
18559
18560 041332 110315
18561
18562 041334 020412
18563 041336 001403
18564
18565 041340 011203
18566 041342 104000
18567 041344 041324
18568
18569 041346 000004

```

; *****
; .SBTTL T0532 MOV B SMO,DM1 TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ:      [171,257,203,205,125,375,016] FC 1,4,8

;ACT BUTS:     37[004]100,171 / 22[171]200,203 / 16[125]016,016

;EXEC:         [203]ALUC=LLLLL :[125]D=177777

;CODES:        [125]SPS=3      /      N:C=1000

;SYNC:         B05J2 (-)      T=2.6 USEC

;KEY SIG:      K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=1 L / K3-6 BYTE INSTR H

T0532:  MOV      #0532,R0          ;LOAD R0 WITH TEST NO.
        MOV      @#I0532,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
R0532:  MOV      R2,R5          ;R5 CONTAINS DEST ADDR
        CLR      (R2)          ;[DEST] = 000000
        CCC

I0532:  MOV B     R3,(R5)        ;TEST THE MOV B
        CMP      R4,(R2)        ;RESULT CORRECT ?
        BEQ      00532          ;BR IF YES

E0532:  MOV      (R2),R3        ;GET THE WAS DATA
        ERROR   R0532          ;MOV B DELIVERED THE WRONG RESULT
        ERROR   R0532          ;ERROR LOOP RETURN ADDRESS

00532:  SCOPE

;CALL THE SCOPE LOOP UTILITY

```

18570
18571
18572
18573
18574
18575
18576
18577
18578
18579
18580
18581
18582
18583
18584
18585
18586
18587
18588
18589
18590
18591
18592
18593
18594
18595
18596
18597
18598
18599
18600
18601
18602
18603
18604
18605
18606
18607
18608

041350 012700 000533
041354 013701 041402
041360 012702 067560
041364 012704 000377
041370 012703 177777
041374 010205
041376 005012
041400 000257
041402 110325
041404 020412
041406 001403
041410 011203
041412 104000
041414 041374
041416 000004

```
; *****  
; .SBTTL T0533 MOV B SMO,DM2 TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [172,257,203,205,125,375,016] FC 1,4,8  
;ACT BUTS: 37[004]100,172 / 22[172]200,203 / 16[125]016,016  
;EXEC: [203]ALUC=LLLLL :[125]D=177777  
;CODES: [125]SPS=3 / N:C=1000  
;SYNC: B05J2 (-) T=2.6 USEC  
;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=1 L / K3-6 BYTE INSTR H  
; K5-5 BCON(1+2) H  
T0533: MOV #0533,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0533,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  
R0533: MOV R2,R5 ;R5 CONTAINS DEST ADDR  
CLR (R2) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
I0533: MOV B R3,(R5)+ ;TEST THE MOV B  
CMP R4,(R2) ;RESULT CORRECT ?  
BEQ 00533 ;BR IF YES  
E0533: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;MOV B DELIVERED THE WRONG RESULT  
R0533 ;ERROR LOOP RETURN ADDRESS  
00533: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

```

18609 ; *****
18610 ; .SBTTL T0534 MOV B SMO,DM3 TEST
18611 ; *****
18612 ;MICROPROGRAMMING / LOGIC INFORMATION
18613 ;ROM SEQ: [173,207,210,203,205,125,375,016] FC 1,4,8
18614 ;ACT BUTS: 37[004]100,173 / 22[207]200,203 / 16[125]016,016
18615 ;EXEC: [203]ALUC=LLLLL :[125]D=177777
18616 ;CODES: [125]SPS=3 / N:C=1000
18617 ;SYNC: B05J2 (-) T=3.4 USEC
18618 ;KEY SIG: K3-3 MOV L / K3-6 BYTE INSTR H / K3-3 SM=0 L / K3-3 DM=3 L
18619
18620
18621
18622
18623
18624
18625
18626
18627 041420 012700 000534 T0534: MOV #0534,R0 ;LOAD R0 WITH TEST NO.
18628 041424 013701 041454 MOV @#I0534,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
18629 041430 012702 067560 MOV #M0534,R2 ;DEST ADDR = M0534
18630 041434 012704 000377 MOV #377,R4 ;RESULT S / B = 377
18631 041440 012703 177777 MOV #-1,R3 ;SOURCE OP IN R3
18632 041444 012705 067554 R0534: MOV #ATA+10,R5 ;BASE DEST ADDR = ATA+10
18633 041450 005012 CLR (R2) ;[DEST] = 000000
18634 041452 000257 CCC ;SCOPE SYNC
18635
18636 041454 110335 I0534: MOV B R3,@(R5)+ ;TEST THE MOV B
18637
18638 041456 020412 CMP R4,(R2) ;RESULT CORRECT ?
18639 041460 001403 BEQ 00534 ;BR IF YES
18640
18641 041462 011203 MOV (R2),R3 ;GET THE WAS DATA
18642 041464 104000 E0534: ERROR ;MOV B DELIVERED THE WRONG RESULT
18643 041466 041444 R0534 ;ERROR LOOP RETURN ADDRESS
18644
18645 041470 000004 O0534: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

18646
18647
18648
18649
18650
18651
18652
18653
18654
18655
18656
18657
18658
18659
18660
18661
18662
18663
18664
18665
18666
18667
18668
18669
18670
18671
18672
18673
18674
18675
18676
18677
18678
18679
18680
18681
18682
18683
18684

```
; *****  
; .SBTTL T0535 MOV B SMO,DM4 TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [174,257,203,205,125,375,016] FC 1,4,8  
;ACT BUTS: 37[004]100,174 / 22[174]200,203 / 16[125]016,016  
;EXEC: [203]ALUC=LLLLL :[125]D=177777  
;CODES: [125]SPS=3 / N:C=1000  
;SYNC: B05J2 (-) T=2.6 USEC  
;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=4 L / K3-6 BYTE INSTR H  
; K5-5 BCON(1+2) H
```

```
041472 012700 000535  
041476 013701 041526  
041502 012702 067560  
041506 012704 177400  
041512 012703 177777  
041516 012705 067562  
041522 005012  
041524 000257  
  
041526 110345  
  
041530 020412  
041532 001403  
  
041534 011203  
041536 104000  
041540 041516  
  
041542 000004
```

```
T0535: MOV #0535,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0535,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
MOV #177400,R4 ;RESULT S / B = 177400  
MOV #-1,R3 ;R3 CONTAINS SOURCE OP  
R0535: MOV #MBUF0+2,R5 ;BASE DEST ADDR = MBUF0+2  
CLR (R2) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
  
I0535: MOV B R3,-(R5) ;TEST THE MOV B  
  
CMP R4,(R2) ;RESULT CORRECT ?  
BEQ 00535 ;BR IF YES  
  
E0535: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;MOV B DELIVERED THE WRONG RESULT  
R0535 ;ERROR LOOP RETURN ADDRESS  
  
00535: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

18685
18686
18687
18688
18689
18690
18691
18692
18693
18694
18695
18696
18697
18698
18699
18700
18701
18702
18703
18704
18705
18706
18707
18708
18709
18710
18711
18712
18713
18714
18715
18716
18717
18718
18719
18720
18721
18722

041544 012700 000536
041550 013701 041600
041554 012702 067560
041560 012704 000377
041564 012703 177777
041570 012705 067562
041574 005012
041576 000257

041600 110365 177776

041604 020412
041606 001403

041610 011203
041612 104000
041614 041570

041616 000004

```
; *****  
; .SBTTL T0536 MOV B SMO,DM6 TEST  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [176,206,212,203,205,125,375,016] FC 1,4,8  
;ACT BUTS: 37[004]100,176 / 17[176]212,212 / 21[206]200,203 / 16[125]016,016  
;EXEC: [203]ALUC=LLLLL :[125]D=177777  
;CODES: [125]SPS=3 / N:C=1000  
;SYNC: B05J2 (-) T=3 USEC  
;KEY SIG: K3-3 MOV L / K3-3 SM=0 L / K3-3 DM=6 L / K3-6 BYTE INSTR H  
; K3-4 OVLAP CYCLE L
```

```
T0536: MOV #0536,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0536,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  
R0536: MOV #MBUF0+2,R5 ;BASE DEST ADDR = MBUF0+2  
CLR (R2) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
  
I0536: MOV B R3,-2(R5) ;TEST THE MOV B  
  
CMP R4,(R2) ;RESULT CORRECT ?  
BEQ 00536 ;BR IF YES  
  
MOV (R2),R3 ;GET THE WAS DATA  
E0536: ERROR ;MOV B DELIVERED THE WRONG RESULT  
R0536 ;ERROR LOOP RETURN ADDRESS  
  
00536: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

18723
18724
18725
18726
18727
18728
18729
18730
18731
18732
18733
18734
18735
18736
18737
18738
18739
18740
18741
18742
18743
18744
18745
18746
18747
18748
18749
18750
18751
18752
18753
18754
18755
18756
18757
18758
18759
18760
18761
18762
18763
18764
18765
18766

041620 012700 000537
041624 013701 041654
041630 012702 177703
041634 012704 177777
041640 012705 125252
041644 012703 052525
041650 000257
041652 000267
041654 050503
041656 100003
041660 001402
041662 102401
041664 103402
041666 104005
041670 041644
041672 020403
041674 001402
041676 104000
041700 041644
041702 000004

```
; *****  
; .SBTTL T0537 BIS TEST - SMO,DMO - <N:C> = 0111  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [102,364,360,001] FC 1,7,8  
;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001  
;EXEC: [364]ALUC=L L L L H : [360] D = 177777  
;CODES: [360] SPS=3 / N:C = 1001  
;SYNC: B05J2 (-) T = 1 USEC  
;KEY SIG: K3-4 OVLAP INSTR H / K3-3 BIS L / K3-3 SM=0L / K3-3 DM=0L  
T0537: MOV #0537,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0537,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = 177703  
MOV #-1,R4 ;RESULT S / B = 177777  
MOV #125252,R5 ;SRC OPR = 125252  
R0537: MOV #52525,R3 ;[DEST] = 52525  
CCC ;CLEAR FLAGS  
267 ;N:C = 0111  
I0537: BIS R5,R3 ;TEST THE BIS  
BPL E10537 ;N:C = 1001 ?  
BEQ E10537  
BVS E10537  
BCS A0537  
E10537: ERROR5 ;BIS FAILED TO ALTER CODES PROPERLY  
R0537 ;ERROR LOOP RETURN ADDRESS  
A0537: CMP R4,R3 ;CORRECT RESULT ?  
BEQ 00537 ;BR IF YES  
E20537: ERROR ;BIS DELIVERED THE WRONG RESULT  
R0537 ;ERROR LOOP RETURN ADDRESS  
00537: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
18767 ; *****  
18768 ; .SBTTL T0540 BIS TEST - SMO,DMO - <N:C> = 1000  
18769 ; *****  
18770  
18771 ;MICROPROGRAMMING / LOGIC INFORMATION  
18772  
18773 ;ROM SEQ: [102,364,360,001] FC 1,7,8  
18774  
18775 ;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001  
18776  
18777 ;EXEC: [364]ALUC=LLLLH :[360] D = 000000  
18778  
18779 ;CODES: [360] SPS=3 / N:C = 0100  
18780  
18781 ;SYNC: B05J2 (-) T = 1 USEC  
18782  
18783 ;KEY SIG: K3-4 OVLAP INSTR H / K3-3 BIS L / K3-3 SM=0L / K3-3 DM=0L  
18784  
18785 041704 012700 000540 T0540: MOV #0540,R0 ;LOAD R0 WITH TEST NO.  
18786 041710 013701 041732 MOV @#I0540,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
18787 041714 012702 177703 MOV #177703,R2 ;DEST ADDR = 177703  
18788 041720 005004 CLR R4 ;RESULT S / B = 000000  
18789 041722 005005 CLR R5 ;SRC OPR = 000000  
18790 041724 005003 R0540: CLR R3 ;[DEST] = 000000  
18791 041726 000257 CCC ;CLEAR FLAGS  
18792 041730 000270 SEN ;N:C = 1000  
18793  
18794 041732 050503 I0540: BIS R5,R3 ;TEST THE BIS  
18795  
18796 041734 100403 BMI E10540 ;N:C = 0100  
18797 041736 001002 BNE E10540  
18798 041740 102401 BVS E10540  
18799 041742 103002 BCC A0540  
18800  
18801 041744 104005 E10540: ERROR5 ;BIS FAILED TO ALTER CODES PROPERLY  
18802 041746 041724 R0540 ;ERROR LOOP RETURN ADDRESS  
18803  
18804 041750 020403 A0540: CMP R4,R3 ;CORRECT RESULT ?  
18805 041752 001402 BEQ 00540 ;BR IF YES  
18806  
18807 041754 104000 E20540: ERROR ;BIS DELIVERED THE WRONG RESULT  
18808 041756 041724 R0540 ;ERROR LOOP RETURN ADDRESS  
18809  
18810 041760 000004 00540: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
18811 ; *****  
18812 ; .SBTTL T0541 BIC TEST - SMO,DMO - <N:C> = 0111  
18813 ; *****  
18814 ;MICROPROGRAMMING / LOGIC INFORMATION  
18815 ;ROM SEQ: [102,364,360,001] FC 1,7,8  
18816 ;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001  
18817 ;EXEC: [364]ALUC=HLLHL :[360] D = 100000  
18818 ;CODES: [360] SPS=3 / N:C = 1001  
18819 ;SYNC: B05J2 (-) T = 1 USEC  
18820 ;KEY SIG: K3-4 OVLAP INSTR H / K3-3 BIC L / K3-3 SM=0L / K3-3 DM=0L  
18821  
18822  
18823  
18824  
18825  
18826  
18827  
18828  
18829 041762 012700 000541 T0541: MOV #0541,R0 ;LOAD R0 WITH TEST NO.  
18830 041766 013701 042016 MOV @#I0541,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
18831 041772 012702 177703 MOV #177703,R2 ;DEST ADDR = 177703  
18832 041776 012704 100000 MOV #100000,R4 ;RESULT S / B = 100000  
18833 042002 012705 077777 MOV #77777,R5 ;SRC OPR = 77777  
18834 042006 012703 177777 R0541: MOV #-1,R3 ;[DEST] = 177777  
18835 042012 000257 CCC ;CLEAR FLAGS  
18836 042014 000267 267 ;N:C = 0111  
18837  
18838 042016 040503 I0541: BIC R5,R3 ;TEST THE BIC  
18839  
18840 042020 100003 BPL E10541 ;N:C = 1001 ?  
18841 042022 001402 BEQ E10541  
18842 042024 102401 BVS E10541  
18843 042026 103402 BCS A0541  
18844  
18845 042030 104005 E10541: ERROR5 ;BIC FAILED TO ALTER CODES PROPERLY  
18846 042032 042006 R0541 ;ERROR LOOP RETURN ADDRESS  
18847  
18848 042034 020403 A0541: CMP R4,R3 ;CORRECT RESULT ?  
18849 042036 001402 BEQ 00541 ;BR IF YES  
18850  
18851 042040 104000 E20541: ERROR ;BIC DELIVERED THE WRONG RESULT  
18852 042042 042006 R0541 ;ERROR LOOP RETURN ADDRESS  
18853  
18854 042044 000004 00541: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
18855 ; *****  
18856 ; .SBTTL T0542 BIC TEST - SMO,DMO - <N:C> = 1000  
18857 ; *****  
18858  
18859 :MICROPROGRAMMING / LOGIC INFORMATION  
18860  
18861 ;ROM SEQ: [102,364,360,001] FC 1,7,8  
18862  
18863 ;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,001  
18864  
18865 ;EXEC: [364]ALUC=HLLHL :[360] D = 000000  
18866  
18867 ;CODES: [360] SPS=3 / N:C = 0100  
18868  
18869 ;SYNC: B05J2 (-) T = 1 USEC  
18870  
18871 ;KEY SIG: K3-4 OVLAP INSTR H / K3-3 BIC L / K3-3 SM=0L / K3-3 DM=0L  
18872  
18873 042046 012700 000542 T0542: MOV #0542,R0 ;LOAD R0 WITH TEST NO.  
18874 042052 013701 042074 MOV @#10542,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
18875 042056 012702 177703 MOV #177703,R2 ;DEST ADDR = 177703  
18876 042062 005004 CLR R4 ;RESULT S / B = 000000  
18877 042064 005005 CLR R5 ;SRC OPR = 000000  
18878 042066 005003 R0542: CLR R3 ;[DEST] = 000000  
18879 042070 000257 CCC ;CLEAR FLAGS  
18880 042072 000270 SEN ;N:C = 1000  
18881  
18882 042074 040503 I0542: BIC R5,R3 ;TEST THE BIC  
18883  
18884 042076 100403 BMI E10542 ;N:C = 0100  
18885 042100 001002 BNE E10542  
18886 042102 102401 BVS E10542  
18887 042104 103002 BCC A0542  
18888  
18889 042106 104005 E10542: ERROR5 ;BIC FAILED TO ALTER CODES PROPERLY  
18890 042110 042066 R0542 ;ERROR LOOP RETURN ADDRESS  
18891  
18892 042112 020403 A0542: CMP R4,R3 ;CORRECT RESULT ?  
18893 042114 001402 BEQ 00542 ;BR IF YES  
18894  
18895 042116 104000 E20542: ERROR ;BIC DELIVERED THE WRONG RESULT  
18896 042120 042066 R0542 ;ERROR LOOP RETURN ADDRESS  
18897  
18898 042122 000004 00542: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

18899 ; *****
18900 ; .SBTTL T0543 BIT TEST - SMO,DMO - <N:C> = 0111
18901 ; *****
18902
18903 ;MICROPROGRAMMING / LOGIC INFORMATION
18904
18905 ;ROM SEQ: [102,364,362,001] FC 1,7,8
18906
18907 ;ACT BUTS: 37[004]100,102 / 31[102]360,362 / 27[364]000,001
18908
18909 ;EXEC: [364]ALUC=HHLHH :[362] D = 100000
18910
18911 ;CODES: [362] SPS=3 / N:C = 1001
18912
18913 ;SYNC: B05J2 (-) T = 1 USEC
18914
18915 ;KEY SIG: K3-4 OVLAP INSTR H / K3-3 BIT L / K3-3 SM=0L / K3-3 DM=0L
18916 ; K4-4 ALLOW CLK L
18917
18918 042124 012700 000543 T0543: MOV #0543,R0 ;LOAD R0 WITH TEST NO.
18919 042130 013701 042160 MOV @#I0543,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
18920 042134 012702 177703 MOV #177703,R2 ;DEST ADDR = 177703
18921 042140 012704 100000 MOV #100000,R4 ;RESULT S / B = 100000
18922 042144 012705 100000 MOV #100000,R5 ;SRC OPR = 100000
18923 042150 012703 100000 R0543: MOV #100000,R3 ;[DEST] = 100000
18924 042154 000257 CCC ;CLEAR FLAGS
18925 042156 000267 267 ;N:C = 0111
18926
18927 042160 030503 I0543: BIT R5,R3 ;TEST THE BIT
18928
18929 042162 100003 BPL E10543 ;N:C = 1001
18930 042164 001402 BEQ E10543
18931 042166 102401 BVS E10543
18932 042170 103402 BCS A0543
18933
18934 042172 104005 E10543: ERROR5 ;BIT FAILED TO ALTER CODES PROPERLY
18935 042174 042150 R0543 ;ERROR LOOP RETURN ADDRESS
18936
18937 042176 020403 A0543: CMP R4,R3 ;CORRECT RESULT ?
18938 042200 001403 BEQ 00543 ;BR IF YES
18939
18940 042202 011203 MOV (R2),R3 ;GET THE WAS DATA
18941 042204 104000 E20543: ERROR ;BIT DELIVERED A RESULT
18942 042206 042150 R0543 ;ERROR LOOP RETURN ADDRESS
18943
18944 042210 000004 00543: SCOPE ;CALL SCOPE LOOP UTILITY

```


18993
18994
18995
18996
18997
18998
18999
19000
19001
19002
19003
19004
19005
19006
19007
19008
19009
19010
19011
19012
19013
19014
19015
19016
19017
19018
19019
19020
19021
19022
19023
19024
19025
19026
19027
19028
19029
19030
19031
19032
19033
19034
19035
19036
19037

042306 012700 000545
042312 013701 042340
042316 012702 177703
042322 012704 000001
042326 005005
042330 012703 000001
042334 000257
042336 000266

042340 020503

042342 100003
042344 001402
042346 102401
042350 103402

042352 104005
042354 042330

042356 020403
042360 001402

042362 104000
042364 042330

042366 000004

```
; *****  
      .SBTTL T0545 CMP TEST - SMO,DMO - <N:C> = 0110  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ:      [102,364,362,001]FC 1,7,8  
  
;ACT BUTS:     37[004]100,102 / 31[102]360,362 / 27[364]000,001  
  
;EXEC:         [364]ALUC=LLHHL :[362] D = 177777  
  
;CODES:        [362] SPS=3 / N:C = 1001  
  
;SYNC:         B05J2 (-) T = 1 USEC  
  
;KEY SIG:      K3-8 CIN00 L / K3-3 CMP L / K3-3 SM=0L / K3-3 DM=0L  
; K3-4 OVLAP INSTR H / K4-4 ALLOW CLK L  
  
T0545:  MOV    #0545,R0          ;LOAD R0 WITH TEST NO.  
        MOV    @#I0545,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
        MOV    #177703,R2     ;DEST ADDR = R3  
        MOV    #+1,R4         ;RESULT S / B = +1  
        CLR    R5             ;SRC OPR = 000000  
R0545:  MOV    #+1,R3         ;[DEST0546  
        CCC                    ;CLEAR FLAGS  
        266                   ;N:C = 0110  
  
I0545:  CMP    R5,R3          ;TEST THE CMP  
  
        BPL    E10545         ;N:C = 1001  
        BEQ    E10545  
        BVS    E10545  
        BCS    A0545  
  
E10545: ERROR5  
        R0545                 ;CMP FAILED TO ALTER CODES PROPERLY  
                                ;ERROR LOOP RETURN ADDRESS  
  
A0545:  CMP    R4,R3          ;CORRECT RESULT ?  
        BEQ    00545         ;BR IF YES  
  
E20545: ERROR  
        R0545                 ;CMP DELIVERED A RESULT  
                                ;ERROR LOOP RETURN ADDRESS  
  
00545:  SCOPE                 ;CALL SCOPE LOOP UTILITY
```

19038
19039
19040
19041
19042
19043
19044
19045
19046
19047
19048
19049
19050
19051
19052
19053
19054
19055
19056
19057 042370 012700 000546
19058 042374 013701 042422
19059 042400 012702 177703
19060 042404 012704 177777
19061 042410 012705 177777
19062 042414 010403
19063 042416 000257
19064 042420 000272
19065
19066 042422 020503
19067
19068 042424 100403
19069 042426 001002
19070 042430 102401
19071 042432 103002
19072
19073 042434 104005
19074 042436 042414
19075
19076 042440 020403
19077 042442 001402
19078
19079 042444 104000
19080 042446 042414
19081
19082 042450 000004

```

: *****
: .SBTTL T0546 CMP TEST - SMO,DMO - <N:C> = 1010
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ:      [102,364,362,001] FC 1,7,8
:ACT BUTS:     37[004]100,102 / 31[102]360,362 / 27[364]000,001
:EXEC:         [364]ALUC=LLHHL :[362] D = 000000
:CODES:        [362] SPS=3 / N:C = 0100
:SYNC:         B05J2 (-) T = 1 USEC
:KEY SIG:      K3-8 CIN00 L / K3-3 CMP L / K3-3 SM=0L / K3-3 DM=0L
:              ; K3-4 OVLAP INSTR H / K4-4 ALLOW CLK L

T0546:  MOV    #0546,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0546,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV    #177703,R2     ;DEST ADDR = R3
        MOV    #-1,R4         ;RESULT S / B = 177777
        MOV    #-1,R5         ;SRC OPR = 177777
R0546:  MOV    R4,R3           ;[DEST] = 177777
        CCC
        272                   ;CLEAR FLAGS
                                   ;N:C = 1010

I0546:  CMP    R5,R3           ;TEST THE CMP
                                   ;N:C = 0100
        BMI    E10546
        BNE    E10546
        BVS    E10546
        BCC    A0546

E10546:  ERROR5
        R0546                 ;CMP FAILED TO ALTER CODES PROPERLY
                                   ;ERROR LOOP RETURN ADDRESS

A0546:  CMP    R4,R3           ;CORRECT RESULT ?
        BEQ    00546          ;BR IF YES

E20546:  ERROR
        R0546                 ;CMP DELIVERED A RESULT
                                   ;ERROR LOOP RETURN ADDRESS

00546:  SCOPE                  ;CALL SCOPE LOOP UTILITY

```

19083
19084
19085
19086
19087
19088
19089
19090
19091
19092
19093
19094
19095
19096
19097
19098
19099
19100
19101
19102 042452 012700 000547
19103 042456 013701 042504
19104 042462 012702 177703
19105 042466 012704 000001
19106 042472 012705 100000
19107 042476 012703 000001
19108
19109 042502 000257
19110
19111 042504 020503
19112
19113 042506 100403
19114 042510 001402
19115 042512 102001
19116 042514 103002
19117
19118 042516 104005
19119 042520 042476
19120
19121 042522 020403
19122 042524 001402
19123
19124 042526 104000
19125 042530 042476
19126
19127 042532 000004

```

; *****
; .SBTTL T0547 CMP TEST - SMO,DMO - <N:C> = 0000
; *****

```

;MICROPROGRAMMING / LOGIC INFORMATION

```

;ROM SEQ:      [102,364,362,001] FC 1,7,8
;ACT BUTS:     37[004]100,102 / 31[102]360,362 / 27[364]000,001
;EXEC:        [364]ALUC=LLHHL :[362] D = 077777
;CODES:       [362] SPS=3 / N:C = 0010
;SYNC:        B05J2 (-) T = 1 USEC
;KEY SIG:     K3-8 CIN00 L / K3-3 CMP L / K3-3 SM=0L / K3-3 DM=0L
;             ; K3-4 OVLAP INSTR H / K34-4 ALLOW CLK L

```

```

T0547:  MOV #0547,R0           ;LOAD R0 WITH TEST NO.
        MOV @#I0547,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV #177703,R2      ;DEST ADDR = R3
        MOV #+1,R4          ;RESULT S / B = +1
        MOV #100000,R5      ;SRC OPR = 100000
R0547:  MOV #+1,R3           ;[DEST0550
        CCC                 ;CLEAR FLAGS
I0547:  CMP R5,R3           ;TEST THE CMP
        BMI E10547         ;N:C = 0010
        BEQ E10547
        BVC E10547
        BCC A0547
E10547: ERROR5             ;CMP FAILED TO ALTER CODES PROPERLY
        R0547             ;ERROR LOOP RETURN ADDRESS
A0547:  CMP R4,R3          ;CORRECT RESULT ?
        BEQ 00547         ;BR IF YES
E20547: ERROR             ;CMP DELIVERED A RESULT
        R0547             ;ERROR LOOP RETURN ADDRESS
00547:  SCOPE             ;CALL SCOPE LOOP UTILITY

```

19128
19129
19130
19131
19132
19133
19134
19135
19136
19137
19138
19139
19140
19141
19142
19143
19144
19145
19146 042534 012700 000550
19147 042540 013701 042570
19148 042544 012702 067560
19149 042550 012704 177777
19150 042554 012705 125252
19151 042560 012712 052525
19152 042564 000257
19153 042566 000267
19154
19155 042570 050512
19156
19157 042572 100003
19158 042574 001402
19159 042576 102401
19160 042600 103402
19161
19162 042602 104005
19163 042604 042560
19164
19165 042606 020412
19166 042610 001403
19167
19168 042612 011203
19169 042614 104000
19170 042616 042560
19171
19172 042620 000004

; *****
; .SBTTL T0550 BIS TEST - SMO,DM1 - <N:C> = 0111
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,7,8
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016
;EXEC: [224]ALUC=L L L L H :[367] D = 177777
;CODES: [367] SPS=3 / N:C = 1001
;SYNC: B05J2 (-) T = 2.8 USEC
;KEY SIG: K3-3 BIS L / K3-3 SM=0L / K3-3 DM=1L

T0550: MOV #0550,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0550,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
R0550: MOV #52525,(R2) ;[DEST] = 52525
CCC ;CLEAR FLAGS
267 ;N:C = 0111

I0550: BIS R5,(R2) ;TEST THE BIS

BPL E10550 ;N:C = 1001
BEQ E10550
BVS E10550
BCS A0550

E10550: ERROR5 ;BIS FAILED TO ALTER CODES PROPERLY
R0550 ;ERROR LOOP RETURN ADDRESS

A0550: CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00550 ;BR IF YES

E20550: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;BIS DELIVERED THE WRONG RESULT
R0550 ;ERROR LOOP RETURN ADDRESS

00550: SCOPE ;CALL SCOPE LOOP UTILITY

19173
19174
19175
19176
19177
19178
19179
19180
19181
19182
19183
19184
19185
19186
19187
19188
19189
19190
19191 042622 012700 000551
19192 042626 013701 042650
19193 042632 012702 067560
19194 042636 005004
19195 042640 005005
19196 042642 005012
19197 042644 000257
19198 042646 000270
19199
19200 042650 050512
19201
19202 042652 100403
19203 042654 001002
19204 042656 102401
19205 042660 103002
19206
19207 042662 104005
19208 042664 042642
19209
19210 042666 020412
19211 042670 001403
19212
19213 042672 011203
19214 042674 104000
19215 042676 042642
19216
19217 042700 000004

; *****
; .SBTTL T0551 BIS TEST - SMO,DM1 - <N:C> = 1000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,7,8
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016
;EXEC: [224]ALUC=LLLLH :[367] D = 000000
;CODES: [367] SPS=3 / N:C = 0100
;SYNC: B05J2 (-) T = 2.8 USEC
;KEY SIG: K3-3 BIS L / K3-3 SM=0L / K3-3 DM=1L

T0551: MOV #0551,R0 ;LOAD R0 WITH TEST NO.
MOV @#10551,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
R4 ;RESULT S / B = 000000
R5 ;SRC OPR = 000000
R0551: CLR (R2) ;[DEST] = 000000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

I0551: BIS R5,(R2) ;TEST THE BIS

BMI E10551 ;N:C = 0100
BNE E10551
BVS E10551
BCC A0551

E10551: ERROR5 ;BIS FAILED TO ALTER CODES PROPERLY
R0551 ;ERROR LOOP RETURN ADDRESS

A0551: CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00551 ;BR IF YES

E20551: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;BIS DELIVERED THE WRONG RESULT
R0551 ;ERROR LOOP RETURN ADDRESS

00551: SCOPE ;CALL SCOPE LOOP UTILITY

19263
19264
19265
19266
19267
19268
19269
19270
19271
19272
19273
19274
19275
19276
19277
19278
19279
19280
19281 042770 012700 000553
19282 042774 013701 043016
19283 043000 012702 067560
19284 043004 005004
19285 043006 005005
19286 043010 005012
19287 043012 000257
19288 043014 000270
19289
19290 043016 040512
19291
19292 043020 100403
19293 043022 001002
19294 043024 102401
19295 043026 103002
19296
19297 043030 104005
19298 043032 043010
19299
19300 043034 020412
19301 043036 001403
19302
19303 043040 011203
19304 043042 104000
19305 043044 043010
19306
19307 043046 000004

; *****
; .SBTTL T0553 BIC TEST - SMO,DM1 - <N:C> = 1000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,7,8
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016
;EXEC: [224]ALUC=HLLHL :[367] D = 000000
;CODES: [367] SPS=3 / N:C = 0100
;SYNC: B05J2 (-) T = 2.7 USEC
;KEY SIG: K3-3 BIC L / K3-3 SM=0L / K3-3 DM=1L

T0553: MOV #0553,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0553,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
CLR R4 ;RESULT S / B = 000000
CLR R5 ;SRC OPR = 000000
R0553: CLR (R2) ;[DEST] = 000000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

I0553: BIC R5,(R2) ;TEST THE BIC

BMI E10553 ;N:C = 0100
BNE E10553
BVS E10553
BCC A0553

E10553: ERROR5 ;BIC FAILED TO ALTER CODES PROPERLY
R0553 ;ERROR LOOP RETURN ADDRESS

A0553: CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00553 ;BR IF YES

E20553: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;BIC DELIVERED THE WRONG RESULT
R0553 ;ERROR LOOP RETURN ADDRESS

00553: SCOPE ;CALL SCOPE LOOP UTILITY

19308
19309
19310
19311
19312
19313
19314
19315
19316
19317
19318
19319
19320
19321
19322
19323
19324
19325
19326
19327
19328
19329
19330
19331
19332
19333
19334
19335
19336
19337
19338
19339
19340
19341
19342
19343
19344
19345
19346
19347
19348
19349
19350
19351
19352

043050 012700 000554
043054 013701 043104
043060 012702 067560
043064 012704 100000
043070 012705 100000
043074 012712 100000
043100 000257
043102 000267

043104 030512

043106 100003
043110 001402
043112 102401
043114 103402

043116 104005
043120 043074

043122 020412
043124 001403

043126 011203
043130 104000
043132 043074

043134 000004

; *****
; .SBTTL T0554 BIT TEST - SMO,DM1 - <N:C> = 0111
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,7,8
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016
;EXEC: [224]ALUC=HHLHH :[367] D = 100000
;CODES: [367] SPS=3 / N:C = 1001
;SYNC: B05J2 (-) T = 2.7 USEC
;KEY SIG: K3-3 BIT L / K3-3 SM=0L / K3-3 DM=0L / K4-4 ALLOW CLK L

T0554: MOV #0554,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0554,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #M0554,R2 ;DEST ADDR = M0554
MOV #100000,R4 ;RESULT S / B = 100000
MOV #100000,R5 ;SRC OPR = 100000
R0554: MOV #100000,(R2) ;[DEST] = 100000
CCC ;CLEAR FLAGS
267 ;N:C = 0111

I0554: BIT R5,(R2) ;TEST THE BIT

BPL E10554 ;N:C = 1001
BEQ E10554
BVS E10554
BCS A0554

E10554: ERROR5 ;BIT FAILED TO ALTER CODES PROPERLY
R0554 ;ERROR LOOP RETURN ADDRESS

A0554: CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00554 ;BR IF YES

E20554: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;BIT DELIVERED A RESULT
R0554 ;ERROR LOOP RETURN ADDRESS

00554: SCOPE ;CALL SCOPE LOOP UTILITY

19353
19354
19355
19356
19357
19358
19359
19360
19361
19362
19363
19364
19365
19366
19367
19368
19369
19370
19371 043136 012700 000555
19372 043142 013701 043172
19373 043146 012702 067560
19374 043152 012704 052525
19375 043156 012705 125252
19376 043162 012712 052525
19377 043166 000257
19378 043170 000270
19379
19380 043172 030512
19381
19382 043174 100403
19383 043176 001002
19384 043200 102401
19385 043202 103002
19386
19387 043204 104005
19388 043206 043162
19389
19390 043210 020412
19391 043212 001403
19392
19393 043214 011203
19394 043216 104000
19395 043220 043162
19396 043222 000004

```
; *****  
      .SBTTL T0555 BIT TEST - SMO,DM1 - <N:C> = 1000  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ:      [161,266,267,224,367,375,016] FC 1,3,7,8  
;ACT BUTS:     37[004]100,161 / 33[266]220,224 / 16[367]016,016  
;EXEC:         [224]ALUC=HHLHH :[367] D = 000000  
;CODES:        [367] SPS=3 / N:C = 0100  
;SYNC:         B05J2 (-) T = 2.7 USEC  
;KEY SIG:      K3-3 BIT L / K3-3 SM=0L / K3-3 DM=1L / K4-4 ALLOW CLK L  
  
T0555:  MOV      #0555,R0          ;LOAD R0 WITH TEST NO.  
        MOV      @#10555,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD  
        MOV      #M0555,R2       ;DEST ADDR = M0555  
        MOV      #52525,R4        ;RESULT S / B = 52525  
        MOV      #125252,R5       ;SRC OPR = 125252  
R0555:  MOV      #52525,(R2)      ;[DEST] = 52525  
        CCC  
        SEN  
        ;CLEAR FLAGS  
        ;N:C = 1000  
  
I0555:  BIT      R5,(R2)         ;TEST THE BIT  
        BMI      E10555          ;N:C = 0100  
        BNE      E10555  
        BVS      E10555  
        BCC      A0555  
  
E10555:  ERROR5  
        R0555                   ;BIT FAILED TO ALTER CODES PROPERLY  
        ;ERROR LOOP RETURN ADDRESS  
  
A0555:  CMP      R4,(R2)         ;CORRECT RESULT ?  
        BEQ      00555          ;BR IF YES  
  
E20555:  MOV      (R2),R3        ;GET THE WAS DATA  
        ERROR  
        R0555                   ;BIT DELIVERED A RESULT  
        R0555                   ;ERROR LOOP RETURN ADDRESS  
00555:  SCOPE                   ;CALL SCOPE LOOP UTILITY
```

19397
19398
19399
19400
19401
19402
19403
19404
19405
19406
19407
19408
19409
19410
19411
19412
19413
19414
19415
19416 043224 012700 000556
19417 043230 013701 043260
19418 043234 012702 067560
19419 043240 012704 177777
19420 043244 012705 177777
19421 043250 012712 177777
19422 043254 000257
19423 043256 000272
19424
19425 043260 020512
19426
19427 043262 100403
19428 043264 001002
19429 043266 102401
19430 043270 103002
19431
19432 043272 104005
19433 043274 043250
19434
19435 043276 020412
19436 043300 001403
19437
19438 043302 011203
19439 043304 104000
19440 043306 043250
19441
19442 043310 000004

```
; *****  
; .SBTTL T0556 CMP TEST - SMO,DM1 - <N:C> = 1010  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016  
;EXEC: [224]ALUC=LLHHL :[367] D = 000000  
;CODES: [367] SPS=3 / N:C = 0100  
;SYNC: B05J2 (-) T = 2.7 USEC  
;KEY SIG: K3-8 CIN00 L / K3-3 CMP L / K3-3 SM=0L / K3-3 DM=1L  
; K4-4 ALLOW CLK L  
T0556: MOV #0556,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0556,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #M0556,R2 ;DEST ADDR = M0556  
MOV #-1,R4 ;RESULT S / B = -1  
MOV #-1,R5 ;SRC OPR = 177777  
R0556: MOV #-1,(R2) ;[DEST] = 177777  
CCC ;CLEAR FLAGS  
272 ;N:C = 1010  
I0556: CMP R5,(R2) ;TEST THE CMP  
BMI E10556 ;N:C = 0100  
BNE E10556  
BVS E10556  
BCC A0556  
E10556: ERROR5 ;CMP FAILED TO ALTER CODES PROPERLY  
R0556 ;ERROR LOOP RETURN ADDRESS  
A0556: CMP R4,(R2) ;CORRECT RESULT ?  
BEQ 00556 ;BR IF YES  
E20556: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;CMP DELIVERED A RESULT  
R0556 ;ERROR LOOP RETURN ADDRESS  
00556: SCOPE ;CALL SCOPE LOOP UTILITY
```

19443
19444
19445
19446
19447
19448
19449
19450
19451
19452
19453
19454
19455
19456
19457
19458
19459
19460
19461
19462
19463
19464
19465
19466
19467
19468
19469
19470
19471
19472
19473
19474
19475
19476
19477
19478
19479
19480
19481
19482
19483
19484
19485
19486
19487
19488

043312 012700 000557
043316 013701 043344
043322 012702 067560
043326 012704 000001
043332 005005
043334 012712 000001
043340 000257
043342 000266
043344 020512
043346 100003
043350 001402
043352 102401
043354 103402
043356 104005
043360 043334
043362 020412
043364 001403
043366 011203
043370 104000
043372 043334
043374 000004

```
; *****  
; .SBTTL T0557 CMP TEST - SMO,DM1 - <N:C> = 0110  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016  
;EXEC: [224]ALUC=LLHHL :[367] D = 177777  
;CODES: [367] SPS=3 / N:C = 1001  
;SYNC: B05J2 (-) T = 2.7 USEC  
;KEY SIG: K3-8 CIN00 L / K3- CMP L / K3-3 SM=0L / K3-3 DM=1L  
; K4-4 ALLOW CLK L  
  
T0557: MOV #0557,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0557,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #MBUFO,R2 ;DEST ADDR = MBUFO  
MOV #+1,R4 ;RESULT S / B = +1  
CLR R5 ;SRC OPR = 000000  
R0557: MOV #+1,(R2) ;[DEST0560  
CCC ;CLEAR FLAGS  
266 ;N:C = 0110  
  
I0557: CMP R5,(R2) ;TEST THE CMP  
BPL E10557 ;N:C = 1001  
BEQ E10557  
BVS E10557  
BCS A0557  
  
E10557: ERROR5 ;CMP FAILED TO ALTER CODES PROPERLY  
R0557 ;ERROR LOOP RETURN ADDRESS  
  
A0557: CMP R4,(R2) ;CORRECT RESULT ?  
BEQ 00557 ;BR IF YES  
  
E20557: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;CMP DELIVERED A RESULT  
R0557 ;ERROR LOOP RETURN ADDRESS  
  
00557: SCOPE ;CALL SCOPE LOOP UTILITY
```

19489
19490
19491
19492
19493
19494
19495
19496
19497
19498
19499
19500
19501
19502
19503
19504
19505
19506
19507
19508
19509
19510
19511
19512
19513
19514
19515
19516
19517
19518
19519
19520
19521
19522
19523
19524
19525
19526
19527
19528
19529
19530
19531
19532
19533

043376 012700 000560
043402 013701 043430
043406 012702 067560
043412 012704 000001
043416 012705 100000
043422 012712 000001
043426 000257
043430 020512
043432 100403
043434 001402
043436 102001
043440 103002
043442 104005
043444 043422
043446 020412
043450 001403
043452 011203
043454 104000
043456 043422
043460 000004

; *****
; .SBTTL T0560 CMP TEST - SMO,DM1 - <N:C> = 0000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016
;EXEC: [224]ALUC=LLHHL :[367] D = 077777
;CODES: [367] SPS=3 / N:C = 0010
;SYNC: B05J2 (-) T = 2.7 USEC
;KEY SIG: K3-8 CIN00 L / K3-3 CMP L / K3-3 SM=0L / K3-3 DM=1L
; K4-4 ALLOW CLK L

T0560: MOV #0560,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0560,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
R0560: MOV #+1,(R2) ;[DEST0561
CCC ;CLEAR FLAGS
I0560: CMP R5,(R2) ;TEST THE CMP
BMI E10560 ;N:C = 0010
BEQ E10560
BVC E10560
BCC A0560
E10560: ERROR5 ;CMP FAILED TO ALTER CODES PROPERLY
R0560 ;ERROR LOOP RETURN ADDRESS
A0560: CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00560 ;BR IF YES
E20560: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;CMP DELIVERED A RESULT
R0560 ;ERROR LOOP RETURN ADDRESS
00560: SCOPE ;CALL SCOPE LOOP UTILITY

19578
19579
19580
19581
19582
19583
19584
19585
19586
19587
19588
19589
19590
19591
19592
19593
19594
19595
19596
19597
19598
19599
19600
19601
19602
19603
19604
19605
19606
19607
19608
19609
19610
19611
19612
19613
19614
19615
19616
19617
19618
19619
19620
19621

043546 012700 000562
043552 013701 043576
043556 012702 177703
043562 005004
043564 012705 067570
043570 005003
043572 000257
043574 000270
043576 051503
043600 100403
043602 001002
043604 102401
043606 103002
043610 104005
043612 043570
043614 020403
043616 001402
043620 104000
043622 043570
043624 000004

```
; *****  
; .SBTTL T0562 BIS TEST - SM1,DMO - <N:C> = 1000  
; *****  
:MICROPROGRAMMING / LOGIC INFORMATION  
:ROM SEQ: [141,247,250,120,371,360,000] FC 1,2,8  
:ACT BUTS: 37[004]100,141 / 35[247]120,120 / 31[120]360,360 / 27[371]000,000  
:EXEC: [371]ALUC=LLLLH :[360] D = 000000  
:CODES: [360] SPS=3 / N:C =0100  
:SYNC: B05J2 (-) T = 2.8 USEC  
:KEY SIG: K3-3 BIS L / K3-3 SM=1L / K3-3 DM=0L  
T0562: MOV #0562,R0 ;LOAD R0 WITH TEST NO.  
MOV @#10562,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #177703,R2 ;DEST ADDR = 177703  
CLR R4 ;RESULT S / B =000000  
MOV #DWTA,R5 ;SRC ADDR = DWTA  
R0562: CLR R3 ;[DEST] = 000000  
CCC ;CLEAR FLAGS  
SEN ;N:C = 1000  
I0562: BIS (R5),R3 ;TEST THE BIS  
BMI E10562 ;N:C = 0100  
BNE E10562  
BVS E10562  
BCC A0562  
E10562: ERROR5 ;BIS FAILED TO ALTER CODES PROPERLY  
R0562 ;ERROR LOOP RETURN ADDRESS  
A0562: CMP R4,R3 ;CORRECT RESULT ?  
BEQ 00562 ;BR IF YES  
E20562: ERROR ;BIS DELIVERED THE WRONG RESULT  
R0562 ;ERROR LOOP RETURN ADDRESS  
00562: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

19622
19623
19624
19625
19626
19627
19628
19629
19630
19631
19632
19633
19634
19635
19636
19637
19638
19639
19640 043626 012700 000563
19641 043632 013701 043666
19642 043636 012702 177703
19643 043642 012704 100000
19644 043646 012705 067564
19645 043652 012703 177777
19646 043656 012715 077777
19647 043662 000257
19648 043664 000267
19649
19650 043666 041503
19651
19652 043670 100003
19653 043672 001402
19654 043674 102401
19655 043676 103402
19656
19657 043700 104005
19658 043702 043652
19659
19660 043704 020403
19661 043706 001402
19662
19663 043710 104000
19664 043712 043652
19665
19666 043714 000004

```

```

; *****
; .SBTTL T0563 BIC TEST - SM1,DMO - <N:C> = 0111
; *****
;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ: [141,247,250,120,371,360,000] FC 1,2,8
;ACT BUTS: 37[004]100,141 / 35[247]120,120 / 31[120]360,360 / 27[371]000,000
;EXEC: [371]ALUC=HLLHL :[360] D = 100000
;CODES: [360] SPS=3 / N:C = 1001
;SYNC: B05J2 (-) T = 2.5 USEC
;KEY SIG: K3-3 BIC L / K3-3 SM=1L / K3-3 DM=0L
T0563: MOV #0563,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0563,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = 177703
MOV #100000,R4 ;RESULT S / B = 100000
MOV #MBUF1,R5 ;SRC ADDR = MBUF1
R0563: MOV #-1,R3 ;[DEST] = 177777
MOV #77777,(R5) ;SRC OPR = 77777
CCC ;CLEAR FLAGS
267 ;N:C = 0111
I0563: BIC (R5),R3 ;TEST THE BIC
BPL E10563 ;N:C = 1001 ?
BEQ E10563
BVS E10563
BCS A0563
E10563: ERROR5 ;BIC FAILED TO ALTER CODES PROPERLY
R0563 ;ERROR LOOP RETURN ADDRESS
A0563: CMP R4,R3 ;CORRECT RESULT ?
BEQ 00563 ;BR IF YES
E20563: ERROR ;BIC DELIVERED THE WRONG RESULT
R0563 ;ERROR LOOP RETURN ADDRESS
00563: SCOPE ;CALL SCOPE LOOP UTILITY

```

```
19667 ; *****  
19668 ; .SBTTL T0564 BIC TEST - SM1,DMO - <N:C> = 1000  
19669 ; *****  
19670 ;MICROPROGRAMMING / LOGIC INFORMATION  
19671 ;ROM SEQ: [141,247,250,120,371,360,000] FC 1,2,8  
19672 ;ACT BUTS: 37[004]100,141 / 35[247]120,120 / 31[120]360,360 / 27[371]000,000  
19673 ;EXEC: [371]ALUC=HLLHL :[360] D = 000000  
19674 ;CODES: [360] SPS=3 / N:C = 0100  
19675 ;SYNC: B05J2 (-) T = 2.5 USEC  
19676 ;KEY SIG: K3-3 BIC L / K3-3 SM=1L / K3-3 DM=0L  
19677  
19678  
19679  
19680  
19681  
19682  
19683  
19684  
19685 043716 012700 000564 T0564: MOV #0564,R0 ;LOAD R0 WITH TEST NO.  
19686 043722 013701 043746 MOV @#I0564,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
19687 043726 012702 177703 MOV #177703,R2 ;DEST ADDR = 177703  
19688 043732 005004 CLR R4 ;RESULT S / B = 000000  
19689 043734 012705 067570 MOV #DWTA,R5 ;SRC ADDR = DWTA  
19690 043740 005003 R0564: CLR R3 ;[DEST] =000000  
19691 043742 000257 CCC ;CLEAR FLAGS  
19692 043744 000270 SEN ;N:C = 1000  
19693  
19694 043746 041503 I0564: BIC (R5),R3 ;TEST THE BIC  
19695  
19696 043750 100403 BMI E10564 ;N:C = 0100  
19697 043752 001002 BNE E10564  
19698 043754 102401 BVS E10564  
19699 043756 103002 BCC A0564  
19700  
19701 043760 104005 E10564: ERROR5 ;BIC FAILED TO ALTER CODES PROPERLY  
19702 043762 043740 R0564 ;ERROR LOOP RETURN ADDRESS  
19703  
19704 043764 020403 A0564: CMP R4,R3 ;CORRECT RESULT ?  
19705 043766 001402 BEQ 00564 ;BR IF YES  
19706  
19707 043770 104000 E20564: ERROR ;BIC DELIVERED THE WRONG RESULT  
19708 043772 043740 R0564 ;ERROR LOOP RETURN ADDRESS  
19709  
19710 043774 000004 00564: SCOPE ;CALL SCOPE LOOP UTILITY
```

19711
19712
19713
19714
19715
19716
19717
19718
19719
19720
19721
19722
19723
19724
19725
19726
19727
19728
19729
19730
19731
19732
19733
19734
19735
19736
19737
19738
19739
19740
19741
19742
19743
19744
19745
19746
19747
19748
19749
19750
19751
19752
19753
19754

043776 012700 000565
044002 013701 044030
044006 012702 177703
044012 012704 100000
044016 012705 067572
044022 010403
044024 000257
044026 000267
044030 031503
044032 100003
044034 001402
044036 102401
044040 103402
044042 104005
044044 044022
044046 020403
044050 001402
044052 104000
044054 044022
044056 000004

; *****
.SBTTL T0565 BIT TEST - SM1,DMO - <N:C> = 0111
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,120,371,362,000] FC 1,2,8
;ACT BUTS: 37[004]100,141 / 35[247]120,120 / 31[120]360,362 / 27[371]000,000
;EXEC: [371]ALUC=HHLHH :[362] D = 100000
;CODES: [362] SPS=3 / N:C = 1001
;SYNC: B05J2 (-) T = 2.5 USEC
;KEY SIG: K3-3 BIT L / K3-3 SM=1L / K3-3 DM=0L / K4-4 ALLOW CLK L

T0565: MOV #0565,R0 ;LOAD R0 WITH TEST NO.
MOV @#10565,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = 177703
MOV #100000,R4 ;RESULT S / B = 100000
MOV #DWTA+2,R5 ;SRC ADDR = DWTA+2
R0565: MOV R4,R3 ;[DEST] = 100000
CCC ;CLEAR FLAGS
267 ;N:C = 0111
I0565: BIT (R5),R3 ;TEST THE BIT
BPL E10565 ;N:C = 1001 ?
BEQ E10565
BVS E10565
BCS A0565
E10565: ERROR5 ;BIT FAILED TO ALTER CODES PROPERLY
R0565 ;ERROR LOOP RETURN ADDRESS
A0565: CMP R4,R3 ;CORRECT RESULT ?
BEQ 00565 ;BR IF YES
E20565: ERROR ;BIT DELIVERED A RESULT
R0565 ;ERROR LOOP RETURN ADDRESS
00565: SCOPE ;CALL SCOPE LOOP UTILITY

```

19755 ; *****
19756 ; .SBTTL T0566 BIT TEST - SM1,DMO - <N:C> = 1000
19757 ; *****
19758
19759 ;MICROPROGRAMMING / LOGIC INFORMATION
19760
19761 ;ROM SEQ: [141,247,250,120,371,362,000] FC 1,2,8
19762
19763 ;ACT BUTS: 37[004]100,141 / 35[247]120,120 / 31[120]360,362 / 27[371]000,000
19764
19765 ;EXEC: [371]ALUC=HHLHH :[362] D = 000000
19766
19767 ;CODES: [362] SPS=3 / N:C = 0100
19768
19769 ;SYNC: B05J2 (-) T = 2.5 USEC
19770
19771 ;KEY SIG: K3-3 BIT L / K3-3 SM=1L / K3-3 DM=0L / K4-4 ALLOW CLK L
19772
19773 044060 012700 000566 T0566: MOV #0566,R0 ;LOAD R0 WITH TEST NO.
19774 044064 013701 044112 MOV @#I0566,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
19775 044070 012702 177703 MOV #177703,R2 ;DEST ADDR = R3
19776 044074 012704 052525 MOV #52525,R4 ;RESULT S / B = 52525
19777 044100 012705 067600 MOV #DWTA+10,R5 ;SRC ADDR = DWTA+10
19778 044104 010403 R0566: MOV R4,R3 ;[DEST] = 52525
19779 044106 000257 CCC ;CLEAR FLAGS
19780 044110 000270 SEN ;N:C = 1000
19781
19782 044112 031503 I0566: BIT (R5),R3 ;TEST THE BIT
19783
19784 044114 100403 BMI E10566 ;N:C = 0100
19785 044116 001002 BNE E10566
19786 044120 102401 BVS E10566
19787 044122 103002 BCC A0566
19788
19789 044124 104005 E10566: ERROR5 ;BIT FAILED TO ALTER CODES PROPERLY
19790 044126 044104 R0566 ;ERROR LOOP RETURN ADDRESS
19791
19792 044130 020403 A0566: CMP R4,R3 ;CORRECT RESULT ?
19793 044132 001402 BEQ 00566 ;BR IF YES
19794
19795 044134 104000 E20566: ERROR ;BIT DELIVERED A RESULT
19796 044136 044104 R0566 ;ERROR LOOP RETURN ADDRESS
19797 044140 000004 00566: SCOPE ;CALL SCOPE LOOP UTILITY

```

19798
19799
19800
19801
19802
19803
19804
19805
19806
19807
19808
19809
19810
19811
19812
19813
19814
19815
19816
19817
19818
19819
19820
19821
19822
19823
19824
19825
19826
19827
19828
19829
19830
19831
19832
19833
19834
19835
19836
19837
19838
19839
19840
19841
19842

044142 012700 000567
044146 013701 044174
044152 012702 177703
044156 012704 000001
044162 012705 067570
044166 010403
044170 000257
044172 000266
044174 021503
044176 100003
044200 001402
044202 102401
044204 103402
044206 104005
044210 044166
044212 020403
044214 001402
044216 104000
044220 044166
044222 000004

; *****
; .SBTTL T0567 CMP TEST - SM1,DMO - <N:C> = 0110
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,120,371,362,000] FC 1,2,8
;ACT BUTS: 37[004]100,141 / 35[247]120,120 / 31[120]360,362 / 27[371]000,000
;EXEC: [371]ALUC=LLHHL :[362] D = 177777
;CODES: [362] SPS=3 / N:C = 1001
;SYNC: B05J2 (-) T = 2.7 USEC
;KEY SIG: K3-8 CIN00 L / K3-3 CMP L / K3-3 SM=1L / K3-3 DM=0L
; K4-4 ALLOW CLK L

T0567: MOV #0567,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0567,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
MOV #+1,R4 ;RESULT S / B = +1
MOV #DWTA,R5 ;SRC ADDR = DWTA
R0567: MOV R4,R3 ;[DEST0570
CCC ;CLEAR FLAGS
266 ;N:C = 0110
I0567: CMP (R5),R3 ;TEST THE CMP
BPL E10567 ;N:C = 1001
BEQ E10567
BVS E10567
BCS A0567
E10567: ERROR5 ;CMP FAILED TO ALTER CODES PROPERLY
R0567 ;ERROR LOOP RETURN ADDRESS
A0567: CMP R4,R3 ;CORRECT RESULT ?
BEQ 00567 ;BR IF YES
E20567: ERROR ;CMP DELIVERED A RESULT
R0567 ;ERROR LOOP RETURN ADDRESS
00567: SCOPE ;CALL SCOPE LOOP UTILITY

19843
19844
19845
19846
19847
19848
19849
19850
19851
19852
19853
19854
19855
19856
19857
19858
19859
19860
19861
19862 044224 012700 000570
19863 044230 013701 044256
19864 044234 012702 177703
19865 044240 012704 177777
19866 044244 012705 067572
19867 044250 010403
19868 044252 000257
19869 044254 000272
19870
19871 044256 021503
19872
19873 044260 100403
19874 044262 001002
19875 044264 102401
19876 044266 103002
19877
19878 044270 104005
19879 044272 044250
19880
19881 044274 020403
19882 044276 001402
19883
19884 044300 104000
19885 044302 044250
19886
19887 044304 000004

; *****
; .SBTTL T0570 CMP TEST - SM1,DMO - <N:C> = 1010
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,120,371,362,000] FC 1,2,8
;ACT BUTS: 37[004]100,141 / 35[247]120,120 / 31[120]360,362 / 27[371]000,000
;EXEC: [371]ALUC=LLHHL :[362] D = 000000
;CODES: [362] SPS=3 / N:C = 0100
;SYNC: B05J2 (-) T = 2.7 USEC
;KEY SIG: K3-8 CIN00 L / K3-3 CMP L / K3-3 SM=1L / K3-3 DM=0L
; K4-4 ALLOW CLK L

T0570: MOV #0570,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0570,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
MOV #-1,R4 ;RESULT S / B = 177777
MOV #DWTA+2,R5 ;SRC ADDR = DWTA+2
R0570: MOV R4,R3 ;[DEST] = 177777
CCC ;CLEAR FLAGS
272 ;N:C = 1010

I0570: CMP (R5),R3 ;TEST THE CMP

BMI E10570 ;N:C = 0100
BNE E10570
BVS E10570
BCC A0570

E10570: ERROR5 ;CMP FAILED TO ALTER CODES PROPERLY
R0570 ;ERROR LOOP RETURN ADDRESS

A0570: CMP R4,R3 ;CORRECT RESULT ?
BEQ 00570 ;BR IF YES

E20570: ERROR ;CMP DELIVERED A RESULT
R0570 ;ERROR LOOP RETURN ADDRESS

00570: SCOPE ;CALL SCOPE LOOP UTILITY

19888
19889
19890
19891
19892
19893
19894
19895
19896
19897
19898
19899
19900
19901
19902
19903
19904
19905
19906
19907 044306 012700 000571
19908 044312 013701 044344
19909 044316 012702 177703
19910 044322 012704 000001
19911 044326 012705 067564
19912 044332 012703 000001
19913 044336 012715 100000
19914 044342 000257
19915
19916 044344 021503
19917
19918 044346 100403
19919 044350 001402
19920 044352 102001
19921 044354 103002
19922
19923 044356 104005
19924 044360 044332
19925
19926 044362 020403
19927 044364 001402
19928
19929 044366 104000
19930 044370 044332
19931
19932 044372 000004

```

; *****
; .SBTTL T0571 CMP TEST - SM1,DMO - <N:C> = 0000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ:      [141,247,250,120,371,362,000] FC 1,2,8

;ACT BUTS:     37[004]100,141 / 35[247]120,120 / 31[120]360,362 / 27[371]000,000

;EXEC:         [371]ALUC=LLHHL :[362] D = 077777

;CODES:        [362] SPS=3 / N:C = 0010

;SYNC:         B05J2 (-) T = 2.7 USEC

;KEY SIG:      K3-8 CIN00 L / K3-3 CMP L / K3-3 SM=1L / K3-3 DM=0L
;              ; K4-4 ALLOW CLK L

T0571:  MOV    #0571,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0571,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV    #177703,R2     ;DEST ADDR = R3
        MOV    #+1,R4         ;RESULT S / B = +1
        MOV    #MBUF1,R5      ;SRC ADDR = MBUF1
R0571:  MOV    #+1,R3          ;[DEST0572
        MOV    #100000,(R5)    ;SRC OPR = 100000
        CCC                   ;CLEAR FLAGS

I0571:  CMP    (R5),R3         ;TEST THE CMP
        BMI    E10571         ;N:C = 0010
        BEQ    E10571
        BVC    E10571
        BCC    A0571

E10571: ERROR5                ;CMP FAILED TO ALTER CODES PROPERLY
        R0571                 ;ERROR LOOP RETURN ADDRESS

A0571:  CMP    R4,R3          ;CORRECT RESULT ?
        BEQ    00571         ;BR IF YES

E20571: ERROR                 ;CMP DELIVERED A RESULT
        R0571                 ;ERROR LOOP RETURN ADDRESS

00571:  SCOPE                 ;CALL SCOPE LOOP UTILITY

```

19933
19934
19935
19936
19937
19938
19939
19940
19941
19942
19943
19944
19945
19946
19947
19948
19949
19950
19951 044374 012700 000572
19952 044400 013701 044430
19953 044404 012702 067560
19954 044410 012704 177777
19955 044414 012705 067600
19956 044420 012712 052525
19957 044424 000257
19958 044426 000267
19959
19960 044430 051512
19961
19962 044432 100003
19963 044434 001402
19964 044436 102401
19965 044440 103402
19966
19967 044442 104005
19968 044444 044420
19969
19970 044446 020412
19971 044450 001403
19972
19973 044452 011203
19974 044454 104000
19975 044456 044420
19976
19977 044460 000004

```

; *****
; .SBTTL T0572 BIS SM1,DM1 TEST - <N:C> = 0111
; *****

```

:MICROPROGRAMMING / LOGIC INFORMATION

```

:ROM SEQ:      [141,247,250,161,266,267,225,367,375,016] FC 1,2,3,8
:ACT BUTS:     37[004]100,141 / 35[247]120,161 / 33[266]220,225 / 16[367]016,016
:EXEC:         [225]ALUC=L L L L H :[367]D=177777
:CODES:        [367]SPS=3      /      N:C=1001
:SYNC:         B05J2 (-)      T= 3.4 USEC
:KEY SIG:      K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=1 L

```

```

T0572: MOV #0572,R0 ;LOAD R0 WITH TEST NO.
        MOV @#10572,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV #MBUF0,R2 ;DEST ADDR = MBUF0
        MOV #-1,R4 ;RESULT S / B = 1777777
        MOV #DWTA+10,R5 ;SOURCE ADDR = DWTA+10
R0572: MOV #52525,(R2) ;[DEST] = 052525
        CCC ;CLEAR FLAGS
        267 ;N:C = 0111

I0572: BIS (R5),(R2) ;TEST THE BIS
        BPL E10572 ;N:C = 1001?
        BEQ E10572
        BVS E10572
        BCS A0572

E10572: ERROR5 ;BIS FAILED TO ALTER CODES PROPERLY
        R0572 ;ERROR LOOP RETURN ADDRESS

A0572: CMP R4,(R2) ;CORRECT RESULT ?
        BEQ 00572 ;BR IF YES

E20572: MOV (R2),R3 ;GET THE WAS DATA
        ERROR ;BIS DELIVERED THE WRONG RESULT
        R0572 ;ERROR LOOP RETURN ADDRESS

00572: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

19978
19979
19980
19981
19982
19983
19984
19985
19986
19987
19988
19989
19990
19991
19992
19993
19994
19995
19996 044462 012700 000573
19997 044466 013701 044516
19998 044472 012702 067560
19999 044476 012704 000000
20000 044502 012705 067570
20001 044506 012712 000000
20002 044512 000257
20003 044514 000270
20004
20005 044516 051512
20006
20007 044520 100403
20008 044522 001002
20009 044524 102401
20010 044526 103002
20011
20012 044530 104005
20013 044532 044506
20014
20015 044534 020412
20016 044536 001403
20017
20018 044540 011203
20019 044542 104000
20020 044544 044506
20021
20022 044546 000004

```

; *****
; .SBTTL T0573 BIS SM1,DM1 TEST - <N:C> = 1000
; *****

```

;MICROPROGRAMMING / LOGIC INFORMATION

```

;ROM SEQ:      [141,247,250,161,266,267,225,367,375,016] FC 1,2,3,8
;ACT BUTS:     37[004]100,141 / 35[247]120,161 / 33[266]220,225 / 16[367]016,016
;EXEC:         [225]ALUC=LLLLH :[367]D=000000
;CODES:        [367]SPS=3      /      N:C=0100
;SYNC:         B05J2 (-)      T=3.4 USEC
;KEY SIG:      K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=1 L

```

```

T0573: MOV #0573,R0 ;LOAD R0 WITH TEST NO.
        MOV @#I0573,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV #M0573,R2 ;DEST ADDR = M0573
        MOV #0,R4 ;RESULT S / B = 000000
        MOV #DWTA,R5 ;SOURCE ADDR = DWTA
R0573: MOV #0,(R2) ;[DEST] = 000000
        CCC ;CLEAR FLAGS
        SEN ;N:C = 1000

I0573: BIS (R5),(R2) ;TEST THE BIS

        BMI E10573 ;N:C = 0100 ?
        BNE E10573
        BVS E10573
        BCC A0573

E10573: ERROR5 ;BIS FAILED TO ALTER CODES PROPERLY
        R0573 ;ERROR LOOP RETURN ADDRESS

A0573: CMP R4,(R2) ;CORRECT RESULT ?
        BEQ 00573 ;BR IF YES

E20573: MOV (R2),R3 ;GET THE WAS DATA
        ERROR R0573 ;BIS DELIVERED THE WRONG RESULT
        R0573 ;ERROR LOOP RETURN ADDRESS

00573: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

20023 ; *****
20024 ; .SBTTL T0574 BIC SM1,DM1 TEST - <N:C> = 0111
20025 ; *****
20026
20027 ;MICROPROGRAMMING / LOGIC INFORMATION
20028
20029 ;ROM SEQ: [141,247,250,161,266,267,225,367,375,016] FC 1,2,3,8
20030
20031 ;ACT BUTS: 37[004]100,141 / 35[247]120,161 / 33[266]220,225 / 16[367]016,016
20032
20033 ;EXEC: [225]ALUC=HLLHL :[367]D=100000
20034
20035 ;CODES: [367]SPS=3 / N:C=1001
20036
20037 ;SYNC: B05J2 (-) T= 3.4 USEC
20038
20039 ;KEY SIG: K3-3 BIC L / K3-3 SM=1 L / K3-3 DM=1 L
20040
20041 044550 012700 000574 T0574: MOV #0574,R0 ;LOAD R0 WITH TEST NO.
20042 044554 013701 044610 MOV @#I0574,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20043 044560 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
20044 044564 012704 100000 MOV #100000,R4 ;RESULT S / B = 100000
20045 044570 012705 067564 MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
20046 044574 012715 077777 R0574: MOV #77777,(R5) ;[SOURCE] = 77777
20047 044600 012712 177777 MOV #-1,(R2) ;[DEST] = 177777
20048 044604 000257 CCC ;CLEAR FLAGS
20049 044606 000267 267 ;N:C = 0111
20050
20051 044610 041512 I0574: BIC (R5),(R2) ;TEST THE BIC
20052
20053 044612 100003 BPL E10574 ;N:C = 1001 ?
20054 044614 001402 BEQ E10574
20055 044616 102401 BVS E10574
20056 044620 103402 BCS A0574
20057
20058 044622 104005 E10574: ERROR5 ;BIC FAILED TO ALTER CODES PROPERLY
20059 044624 044574 R0574 ;ERROR LOOP RETURN ADDRESS
20060
20061 044626 020412 A0574: CMP R4,(R2) ;CORRECT RESULT ?
20062 044630 001403 BEQ 00574 ;BR IF YES
20063
20064 044632 011203 MOV (R2),R3 ;GET THE WAS DATA
20065 044634 104000 E20574: ERROR ;BIC DELIVERED THE WRONG RESULT
20066 044636 044574 R0574 ;ERROR LOOP RETURN ADDRESS
20067
20068 044640 000004 00574: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

20069
20070
20071
20072
20073
20074
20075
20076
20077
20078
20079
20080
20081
20082
20083
20084
20085
20086
20087
20088
20089
20090
20091
20092
20093
20094
20095
20096
20097
20098
20099
20100
20101
20102
20103
20104
20105
20106
20107
20108
20109
20110
20111
20112
20113
20114
20115

044642 012700 000575
044646 013701 044702
044652 012702 067560
044656 012704 000000
044662 012705 067564
044666 012715 000000
044672 012712 000000
044676 000257
044700 000270

044702 041512

044704 100403
044706 001002
044710 102401
044712 103002

044714 104005
044716 044666

044720 020412
044722 001403

044724 011203
044726 104000
044730 044666

044732 000004

; *****
; .SBTTL T0575 BIC SM1,DM1 TEST - <N:C> = 1000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,162,266,267,225,367,375,016] FC 1,2,3,8
;ACT BUTS: 37[004]100,141 / 35[247]120,161 / 33[266]220,225 / 16[367]016,016
;EXEC: [225]ALUC=HLLHL :[367]D=000000
;CODES: [367]SPS=3 / N:C=0100
;SYNC: B05J2 (-) T= 3.4 USEC
;KEY SIG: K3-3 BIC L / K3-3 SM=1 L / K3-3 DM=1 L

T0575: MOV #0575,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0575,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #0,R4 ;RESULT S / B = 000000
MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
R0575: MOV #0,(R5) ;[SOURCE] = 000000
MOV #0,(R2) ;[DEST] = 000000
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

I0575: BIC (R5),(R2) ;TEST THE BIC

BMI E10575 ;N:C = 0100 ?
BNE E10575
BVS E10575
BCC A0575

E10575: ERROR5 ;BIC FAILED TO ALTER CODES PROPERLY
R0575 ;ERROR LOOP RETURN ADDRESS

A0575: CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00575 ;BR IF YES

E20575: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;BIC DELIVERED THE WRONG RESULT
R0575 ;ERROR LOOP RETURN ADDRESS

00575: SCOPE ;CALL THE SCOPE LOOP UTILITY

20116
20117
20118
20119
20120
20121
20122
20123
20124
20125
20126
20127
20128
20129
20130
20131
20132
20133
20134
20135
20136
20137
20138
20139
20140
20141
20142
20143
20144
20145
20146
20147
20148
20149
20150
20151
20152
20153
20154
20155
20156
20157
20158
20159
20160
20161

044734 012700 000576
044740 013701 044774
044744 012702 067560
044750 012704 125252
044754 012705 067564
044760 012715 052525
044764 012712 125252
044770 000257
044772 000270
044774 031512
044776 100403
045000 001002
045002 102401
045004 103002
045006 104005
045010 044760
045012 020412
045014 001403
045016 011203
045020 104000
045022 044760
045024 000004

; *****
; .SBTTL T0576 BIT SM1,DM1 TEST - <N:C> = 1000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,161,266,267,225,367,375,016] FC 1,2,3,8

;ACT BUTS: 37[004]100,141 / 35[247]120,161 / 33[266]220,225 / 16[367]016,016

;EXEC: [225]ALUC=HHLHH :[367]D=000000

;CODES: [367]SPS=3 / N:C=0100

;SYNC: B05J2 (-) T=3.4 USEC

;KEY SIG: K3-3 BIT L / K3-3 SM=1 L / K3-3 DM=1 L / K4-4 ALLOW CLK L

T0576: MOV #0576,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0576,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #125252,R4 ;RESULT S / B = 125252
MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
R0576: MOV #52525,(R5) ;[SOURCE] = 052525
MOV #125252,(R2) ;[DEST] = 125252
CCC ;CLEAR FLAGS
SEN ;N:C = 1000

I0576: BIT (R5),(R2) ;TEST THE BIT

BMI E10576 ;N:C = 0100 ?
BNE E10576
BVS E10576
BCC A0576

E10576: ERROR5 ;BIT FAILED TO ALTER CODES PROPERLY
R0576 ;ERROR LOOP RETURN ADDRESS

A0576: CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00576 ;BR IF YES

E20576: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;BIT DELIVERED A RESULT
R0576 ;ERROR LOOP RETURN ADDRESS

00576: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

20162 ; *****
20163 ; .SBTTL T0577 BIT SM1,DM1 TEST - <N:C> = 0111
20164 ; *****
20165 ;MICROPROGRAMMING / LOGIC INFORMATION
20166 ;ROM SEQ: [141,247,250,161,266,267,225,367,375,016] FC 1,2,3,8
20167 ;ACT BUTS: 37[004]100,141 / 35[247]120,161 / 33[266]220,225 / 16[367]016,016
20168 ;EXEC: [225]ALUC=HHLHH :[367]D=100000
20169 ;CODES: [367]SPS=3 / N:C=1001
20170 ;SYNC: B05J2 (-) T=3.4 USEC
20171 ;KEY SIG: K3-3 BIT L / K3-3 SM=1 L / K3-3 DM=1 L / K4-4 ALLOW CLK L
20172
20173
20174
20175
20176
20177
20178
20179
20180 045026 012700 000577 T0577: MOV #0577,R0 ;LOAD R0 WITH TEST NO.
20181 045032 013701 045066 MOV @#I0577,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20182 045036 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
20183 045042 012704 100000 MOV #100000,R4 ;RESULT S / B = 100000
20184 045046 012705 067564 MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
20185 045052 012715 100000 R0577: MOV #100000,(R5) ;[SOURCE] = 100000
20186 045056 012712 100000 MOV #100000,(R2) ;[DEST] = 100000
20187 045062 000257 CCC ;CLEAR FLAGS
20188 045064 000267 267 ;N:C = 0111
20189
20190 045066 031512 I0577: BIT (R5),(R2) ;TEST THE BIT
20191
20192 045070 100003 BPL E10577 ;N:C = 1001 ?
20193 045072 001402 BEQ E10577
20194 045074 102401 BVS E10577
20195 045076 103402 BCS A0577
20196
20197 045100 104005 E10577: ERROR5 ;BIT FAILED TO ALTER CODES PROPERLY
20198 045102 045052 R0577 ;ERROR LOOP RETURN ADDRESS
20199
20200 045104 020412 A0577: CMP R4,(R2) ;CORRECT RESULT ?
20201 045106 001403 BEQ 00577 ;BR IF YES
20202
20203 045110 011203 MOV (R2),R3 ;GET THE WAS DATA
20204 045112 104000 E20577: ERROR ;BIT DELIVERED A RESULT
20205 045114 045052 R0577 ;ERROR LOOP RETURN ADDRESS
20206
20207 045116 000004 O0577: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

20208 ; *****
20209 ; .SBTTL T0600 CMP SM1,DM1 TEST - <N:C> = 1010
20210 ; *****
20211 ;MICROPROGRAMMING / LOGIC INFORMATION
20212
20213 ;ROM SEQ: [141,247,250,161,266,267,225,367,375,016] FC 1,2,3,8
20214
20215 ;ACT BUTS: 37[004]100,141 / 35[247]120,161 / 33[266]220,225 / 16[367]016,016
20216
20217 ;EXEC: [225]ALUC=LLHHL :[367]D=000000
20218
20219 ;CODES: [367]SPS=3 / N:C=0100
20220
20221 ;SYNC: B05J2 (-) T=3.4 USEC
20222
20223 ;KEY SIG: K3-3 CMP L / K3-3 SM=1 L / K3-3 DM=1 L / K3-8 CIN00 L
20224 ; K4-4 ALLOW CLK L
20225
20226
20227 045120 012700 000600 T0600: MOV #0600,R0 ;LOAD R0 WITH TEST NO.
20228 045124 013701 045156 MOV @#I0600,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20229 045130 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
20230 045134 012704 177777 MOV #-1,R4 ;RESULT S / B = 177777
20231 045140 012705 067564 MOV #MBUF1,R5 ;SOURCE ADDR =
20232 045144 012715 177777 R0600: MOV #-1,(R5) ;[SOURCE] = 177777
20233 045150 010412 MOV R4,(R2) ;[DEST] = 177777
20234 045152 000257 CCC ;CLEAR FLAGS
20235 045154 000272 272 ;N:C = 1010
20236
20237 045156 021512 I0600: CMP (R5),(R2) ;TEST THE CMP
20238
20239 045160 100403 BMI E10600 ;N:C = 0100 ?
20240 045162 001002 BNE E10600
20241 045164 102401 BVS E10600
20242 045166 103002 BCC A0600
20243
20244 045170 104005 E10600: ERROR5 ;CMP FAILED TO ALTER CODES PROPERLY
20245 045172 045144 R0600 ;ERROR LOOP RETURN ADDRESS
20246
20247 045174 020412 A0600: CMP R4,(R2) ;CORRECT RESULT ?
20248 045176 001403 BEQ 00600 ;BR IF YES
20249
20250 045200 011203 MOV (R2),R3 ;GET THE WAS DATA
20251 045202 104000 E20600: ERROR ;CMP DELIVERED A RESULT
20252 045204 045144 R0600 ;ERROR LOOP RETURN ADDRESS
20253
20254 045206 000004 O0600: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```
20255 ; *****
20256 ; .SBTTL T0601 CMP SM1,DM1 TEST - <N:C> = 0110
20257 ; *****
20258
20259 ;MICROPROGRAMMING / LOGIC INFORMATION
20260
20261 ;ROM SEQ: [141,247,250,161,266,267,225,367,375,016] FC 1,2,3,8
20262
20263 ;ACT BUTS: 37[004]100,141 / 35[247]120,161 / 33[266]220,225 / 16[367]016,016
20264
20265 ;EXEC: [225]ALUC=LLHHL :[367]D=177777
20266
20267 ;CODES: [367]SPS=3 / N:C=1001
20268
20269 ;SYNC: B05J2 (-) T=3.4 USEC
20270
20271 ;KEY SIG: K3-3 CMP L / K3-3 SM=1 L / K3-3 DM=1 L / K3-8 CIN00 L
20272 ; K4-4 ALLOW CLK L
20273
20274 045210 012700 000601 T0601: MOV #0601,R0 ;LOAD R0 WITH TEST NO.
20275 045214 013701 045250 MOV @#I0601,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20276 045220 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
20277 045224 012704 000001 MOV #+1,R4 ;RESULT S / B = 000001
20278 045230 012705 067564 MOV #MBUF1,R5 ;SOURCE ADDR = MBUF1
20279 045234 012715 000000 R0601: MOV #0,(R5) ;[SOURCE] = 000000
20280 045240 012712 000001 MOV #+1,(R2) ;[DEST] = 000001
20281 045244 000257 CCC ;CLEAR FLAGS
20282 045246 000266 266 ;N:C = 0110
20283
20284 045250 021512 I0601: CMP (R5),(R2) ;TEST THE CMP
20285
20286 045252 100003 BPL E10601 ;N:C = 1001 ?
20287 045254 001402 BEQ E10601
20288 045256 102401 BVS E10601
20289 045260 103402 BCS A0601
20290
20291 045262 104005 E10601: ERROR5 ;CMP FAILED TO ALTER CODES PROPERLY
20292 045264 045234 R0601 ;ERROR LOOP RETURN ADDRESS
20293
20294 045266 020412 A0601: CMP R4,(R2) ;CORRECT RESULT ?
20295 045270 001403 BEQ 00601 ;BR IF YES
20296
20297 045272 011203 MOV (R2),R3 ;GET THE WAS DATA
20298 045274 104000 E20601: ERROR ;CMP DELIVERED A RESULT
20299 045276 045234 R0601 ;ERROR LOOP RETURN ADDRESS
20300
20301 045300 000004 00601: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

20302
20303
20304
20305
20306
20307
20308
20309
20310
20311
20312
20313
20314
20315
20316
20317
20318
20319
20320
20321
20322
20323
20324
20325
20326
20327
20328
20329
20330
20331
20332
20333
20334
20335
20336
20337
20338
20339
20340
20341
20342
20343
20344
20345
20346
20347
20348
20349

; *****
; .SBTTL T0602 CMP SM1,DM1 TEST - <N:C> = 0000
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,161,266,267,225,367,375,016] FC 1,2,3,8
;ACT BUTS: 37[004]100,141 / 35[247]120,161 / 33[266]220,225 / 16[367]016,016
;EXEC: [225]ALUC=LLHHL :[367]D=077777
;CODES: [367]SPS=3 / N:C=0010
;SYNC: B05J2 (-) T=3.4 USEC
;KEY SIG: K3-3 CMP L / K3-3 SM=1 L / K3-3 DM=1 L / K3-8 CIN00 L
; K4-4 ALLOW CLK L

T0602: MOV #0602,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0602,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
R0602: MOV #100000,(R5) ;[SOURCE] = 100000
MOV #+1,(R2) ;[DEST] = 000001
CCC ;N:C = 0000
I0602: CMP (R5),(R2) ;TEST THE CMP
BMI E10602 ;N:C = 0010 ?
BEQ E10602
BVC E10602
BCC A0602
E10602: ERROR5 ;CMP FAILED TO ALTER CODES PROPERLY
R0602 ;ERROR LOOP RETURN ADDRESS
A0602: CMP R4,(R2) ;CORRECT RESULT ?
BEQ 00602 ;BR IF YES
E20602: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;CMP DELIVERED A RESULT
R0602 ;ERROR LOOP RETURN ADDRESS
00602: SCOPE ;CALL THE SCOPE LOOP UTILITY

20350
20351
20352
20353
20354
20355
20356
20357
20358
20359
20360
20361
20362
20363
20364
20365
20366
20367
20368
20369
20370
20371
20372
20373
20374
20375
20376
20377
20378
20379
20380
20381
20382
20383
20384
20385
20386
20387

045372 012700 000603
045376 013701 045422
045402 012702 177703
045406 012704 000377
045412 012705 070131
045422 151503
045424 020403
045426 001402
045430 104000
045432 045412
045434 000004

: *****
.SBTTL T0603 BISB SM1,DM0 TEST - SOURCE ADDR ODD
: *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,137,251,120,371,360,000] FC 1,2,8
;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,120 / 31[120]360,360
; / 27[371]000,000
;EXEC: [371]ALUC=LLLLH :[360]D=177777
;CODES: [360]SPS=3 / N:C=1000
;SYNC: B05J2 (-) T=2.8 USEC
;KEY SIG: K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=0 L / K1-6 BA00(1) H
; K3-7 ODD BYTE H / K3-6 BYTE INSTR H

T0603: MOV #0603,R0 ;LOAD R0 WITH TEST NO.
MOV @#10603,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #177703,R2 ;DEST ADDR = R3
MOV #377,R4 ;RESULT S / B = 377
R0603: MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
CLR R3 ;[DEST] = 000000
CCC ;SCOPE SYNC
I0603: BISB (R5),R3 ;TEST THE BISB
CMP R4,R3 ;RESULT CORRECT ?
BEQ 00603 ;BR IF YES
E0603: ERROR ;BISB DELIVERED THE WRONG RESULT
R0603 ;ERROR LOOP RETURN ADDRESS
00603: SCOPE ;CALL THE SCOPE LOOP UTILITY

20388
20389
20390
20391
20392
20393
20394
20395
20396
20397
20398
20399
20400
20401
20402
20403
20404
20405
20406
20407
20408
20409
20410
20411
20412
20413
20414
20415
20416
20417
20418
20419
20420
20421
20422
20423
20424
20425
20426
20427
20428

045436 012700 000604
045442 013701 045500
045446 032737 004000 066642
045454 001401
045456 000000
045460 012702 067560
045464 012704 000377
045470 012705 070131
045474 005012
045476 000257
045500 151512
045502 020412
045504 001403
045506 011203
045510 104000
045512 045470
045514 000004

; *****
; .SBTTL T0604 BISB SM1,DM1 TEST - SOURCE ADDR ODD
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,137,251,161,266,267,225,367,375,016] FC 1,2,3,8

;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,161 / 33[266]220,225
; / 16[367]016,016

;EXEC: [225]ALUC=LLLLH :[367]D=177777

;CODES: [367]SPS=3 / N:C=1000

;SYNC: B05J2 (-) T=3.8 USEC

;KEY SIG: K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=1 L / K3-6 BYTE INSTR H
; K3-7 ODD BYTE H / K1-6 BA00(1) H

T0604: MOV #0604,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0604,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
BIT #4000,@#BPTLOC ;BREAKPOINT HALT SET ??
BEQ .+4 ;BR IF NOT
HALT ;BREAK-DEPRESS CONTINUE TO RESTART
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
R0604: MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
I0604: BISB (R5),(R2) ;TEST THE BISB
CMP R4,(R2) ;CORRECT RESULT
BEQ 00604 ;BR IF YES
E0604: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;BISB DELIVERED THE WRONG RESULT
R0604 ;ERROR LOOP RETURN ADDRESS
00604: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

20429 ; *****
20430 ; .SBTTL T0605 BISB SM1,DM2 TEST - SOURCE ADDR ODD
20431 ; *****
20432
20433 ;MICROPROGRAMMING / LOGIC INFORMATION
20434
20435 ;ROM SEQ: [141,247,250,137,251,162,260,267,225,367,375,016] FC 1,2,3,8
20436
20437 ;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,162 / 33[260]220,225
20438 ; / 16[367]016,016
20439
20440 ;EXEC: [225]ALUC=LLLLH :[367]D=177777
20441
20442 ;CODES: [367]SPS=3 / N:C=1000
20443
20444 ;SYNC: B05J2 (-) T=3.8 USEC
20445
20446 ;KEY SIG: K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=2 L / K3-6 BYTE INSTR H
20447 ; / K1-6 BA00(1) H / K3-7 ODD BYTE H / K5-5 BCON(1+2) H
20448
20449 045516 012700 000605 T0605: MOV #0605,R0 ;LOAD R0 WITH TEST NO.
20450 045522 013701 045550 MOV @#I0605,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20451 045526 012702 067560 MOV #M0605,R2 ;DEST ADDR = M0605
20452 045532 012704 000377 MOV #377,R4 ;RESULT S / B = 377
20453 045536 012705 070131 R0605: MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
20454 045542 005012 CLR (R2) ;[DEST] = 000000
20455 045544 010203 MOV R2,R3 ;DEST ADDR IN R3
20456 045546 000257 CCC ;SCOPE SYNC
20457
20458 045550 151523 I0605: BISB (R5),(R3)+ ;TEST THE BISB
20459
20460 045552 020412 CMP R4,(R2) ;CORRECT RESULT
20461 045554 001403 BEQ 00605 ;BR IF YES
20462
20463 045556 011203 MOV (R2),R3 ;GET THE WAS DATA
20464 045560 104000 E0605: ERROR ;BISB DELIVERED THE WRONG RESULT
20465 045562 045536 R0605 ;ERROR LOOP RETURN ADDRESS
20466
20467 045564 000004 00605: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

20468
20469
20470
20471
20472
20473
20474
20475
20476
20477
20478
20479
20480
20481
20482
20483
20484
20485
20486
20487
20488
20489
20490
20491
20492
20493
20494
20495
20496
20497
20498
20499
20500
20501
20502
20503
20504
20505
20506

045566 012700 000606
045572 013701 045622
045576 012702 067560
045602 012704 000377
045606 012705 070131
045612 005012
045614 012703 067554
045620 000257
045622 151533
045624 020412
045626 001403
045630 011203
045632 104000
045634 045606
045636 000004

; *****
; .SBTTL T0606 BISB SM1,DM3 TEST - SOURCE ADDR ODD
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,137,251,163,264,265,266,267,225,367,375,016] FC 1,2,3,8

;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,163 / 33[266]220,225
; / 16[367]016,016

;EXEC: [225]ALUC=LLLLH :[367]D=177777

;CODES: [367]SPS=3 / N:C=1000

;SYNC: B05J2 (-) T=4.5 USEC

;KEY SIG: K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=3 L / K3-6 BYTE INSTR H
; / K3-7 ODD BYTE H / K1-6 BA00(1) H / K5-5 BC01 H

T0606: MOV #0606,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0606,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
R0606: MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
CLR (R2) ;[DEST] = 000000
MOV #ATA+10,R3 ;BASE DEST ADDR = ATA+10
CCC ;SCOPE SYNC
I0606: BISB (R5),@(R3)+ ;TEST THE BISB
CMP R4,(R2) ;CORRECT RESULT
BEQ 00606 ;BR IF YES
E0606: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;BISB DELIVERED THE WRONG RESULT
R0606 ;ERROR LOOP RETURN ADDRESS
00606: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

20507 ; *****
20508 ; .SBTTL T0607 BISB SM1,DM4 TEST - SOURCE ADDR ODD
20509 ; *****
20510 ;MICROPROGRAMMING / LOGIC INFORMATION
20511
20512 ;ROM SEQ: [141,247,250,137,251,164,260,267,225,367,375,016] FC 1,2,3,8
20513
20514 ;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,164 / 33[260]220,225
20515 ; / 16[367]016,016
20516
20517 ;EXEC: [225]ALUC=L L L L H :[367]D=177777
20518
20519 ;CODES: [367]SPS=3 / N:C=1000
20520
20521 ;SYNC: B05J2 (-) T=3.8 USEC
20522
20523 ;KEY SIG: K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=4 L / K3-6 BYTE INSTR H
20524 ; / K1-6 BA00(1) H / K3-7 ODD BYTE H / K5-5 BCON(1+2) H
20525
20526
20527 045640 012700 000607 T0607: MOV #0607,R0 ;LOAD R0 WITH TEST NO.
20528 045644 013701 045674 MOV @#10607,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20529 045650 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
20530 045654 012704 177400 MOV #177400,R4 ;RESULT S / B = 177400
20531 045660 012705 070131 R0607: MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
20532 045664 012703 067562 MOV #MBUF0+2,R3 ;BASE DEST ADDR = MBUF0+2
20533 045670 005012 CLR (R2) ;[DEST] = 000000
20534 045672 000257 CCC ;SCOPE SYNC
20535
20536 045674 151543 I0607: BISB (R5),-(R3) ;TEST THE BISB
20537
20538 045676 020412 CMP R4,(R2) ;CORRECT RESULT
20539 045700 001403 BEQ 00607 ;BR IF YES
20540
20541 045702 011203 MOV (R2),R3 ;GET THE WAS DATA
20542 045704 104000 E0607: ERROR ;BISB DELIVERED THE WRONG RESULT
20543 045706 045660 R0607 ;ERROR LOOP RETURN ADDRESS
20544
20545 045710 000004 00607: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

20546 ; *****
20547 ; .SBTTL T0610 BISB SM1,DM5 TEST - SOURCE ADDR ODD
20548 ; *****
20549
20550 ;MICROPROGRAMMING / LOGIC INFORMATION
20551
20552 ;ROM SEQ: [141,247,250,137,251,165,264,265,266,267,225,367,375,016] FC 1,2,3,8
20553
20554 ;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,165 / 33[266]220,225
20555 ; / 16[367]016,016
20556
20557 ;EXEC: [225]ALUC=L L L L H :[367]D=177777
20558
20559 ;CODES: [367]SPS=3 / N:C=1000
20560
20561 ;SYNC: B05J2 (-) T= 4.5 USEC
20562
20563 ;KEY SIG: K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=5 L / K3-6 BYTE INSTR H
20564 ; K1-6 BA00(1) H / K3-7 ODD BYTE H / K5-5 BC01 H
20565
20566 045712 012700 000610 T0610: MOV #0610,R0 ;LOAD R0 WITH TEST NO.
20567 045716 013701 045746 MOV @#10610,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20568 045722 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
20569 045726 012704 000377 MOV #377,R4 ;RESULT S / B = 377
20570 045732 012705 070131 R0610: MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
20571 045736 012703 067556 MOV #ATA+12,R3 ;BASE DEST ADDR = ATA+12
20572 045742 005012 CLR (R2) ;[DEST] = 000000
20573 045744 000257 CCC ;SCOPE SYNC
20574
20575 045746 151553 I0610: BISB (R5),@-(R3) ;TEST THE BISB
20576
20577 045750 020412 CMP R4,(R2) ;CORRECT RESULT
20578 045752 001403 BEQ 00610 ;BR IF YES
20579
20580 045754 011203 MOV (R2),R3 ;GET THE WAS DATA
20581 045756 104000 E0610: ERROR ;BISB DELIVERED THE WRONG RESULT
20582 045760 045732 R0610 ;ERROR LOOP RETURN ADDRESS
20583
20584 045762 000004 00610: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

20585 ; *****
20586 ; .SBTTL T0611 BISB SM1,DM6 TEST - SOURCE ADDR ODD
20587 ; *****
20588
20589 ;MICROPROGRAMMING / LOGIC INFORMATION
20590
20591 ;ROM SEQ: [141,247,250,137,251,167,261,262,266,267,225,367,375,016] FC 1,2,3,8
20592
20593 ;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,167 / 17[167]262,262
20594 ; / 33[266]220,225 / 16[367]016,016
20595
20596 ;EXEC: [225]ALUC=LLLLH :[367]D=177777
20597
20598 ;CODES: [367]SPS=3 / N:C=1000
20599
20600 ;SYNC: B05J2 (-) T=4.6 USEC
20601
20602 ;KEY SIG: K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=6 L / K3-6 BYTE INSTR H
20603 ; K3-7 ODD BYTE H / K1-6 BA00(1) H
20604
20605 045764 012700 000611 T0611: MOV #0611,R0 ;LOAD R0 WITH TEST NO.
20606 045770 013701 046020 MOV @#I0611,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20607 045774 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0
20608 046000 012704 000377 MOV #377,R4 ;RESULT S / B = 377
20609 046004 012705 070131 R0611: MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
20610 046010 012703 067566 MOV #MBUF0+6,R3 ;BASE DEST ADDR = MBUF0+6
20611 046014 005012 CLR (R2) ;[DEST] = 000000
20612 046016 000257 CCC ;SCOPE SYNC
20613
20614 046020 151563 177772 I0611: BISB (R5),-6(R3) ;TEST THE BISB
20615
20616 046024 020412 CMP R4,(R2) ;CORRECT RESULT
20617 046026 001403 BEQ 00611 ;BR IF YES
20618
20619 046030 011203 MOV (R2),R3 ;GET THE WAS DATA
20620 046032 104000 E0611: ERROR ;BISB DELIVERED THE WRONG RESULT
20621 046034 046004 R0611 ;ERROR LOOP RETURN ADDRESS
20622
20623 046036 000004 O0611: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

20624
20625
20626
20627
20628
20629
20630
20631
20632
20633
20634
20635
20636
20637
20638
20639
20640
20641
20642
20643
20644
20645
20646
20647
20648
20649
20650
20651
20652
20653
20654
20655
20656
20657
20658
20659
20660
20661
20662

046040 012700 000612
046044 013701 046074
046050 012702 067560
046054 012704 000377
046060 012705 070131
046064 012703 067544
046070 005012
046072 000257
046074 151573 000010
046100 020412
046102 001403
046104 011203
046106 104000
046110 046060
046112 000004

; *****
.SBTTL T0612 BISB SM1,DM7 TEST - SOURCE ADDR ODD
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,137,251,167,261,263,264,265,266,267,225,367,375,016] FC 1,2

;ACT BUTS: 37[004]100,141 / 35[247]120,137 / 36[137]120,167 / 17[167]262,263
; / 33[266]220,225 / 16[367]016,016

;EXEC: [225]ALUC=LLLLH :[367]D=177777

;CODES: [367]SPS=3 / N:C=1000

;SYNC: B05J2 (-) T=5.2 USEC

;KEY SIG: K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=7 L / K3-6 BYTE INSTR H
; K3-7 ODD BYTE H / K1-6 BA00(1) H

T0612: MOV #0612,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0612,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV #MBUF0,R2 ;DEST ADDR = MBUF0
MOV #377,R4 ;RESULT S / B = 377
R0612: MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
MOV #ATA,R3 ;BASE DEST ADDR = ATA
CLR (R2) ;[DEST] = 000000
CCC ;SCOPE SYNC
I0612: BISB (R5),@10(R3) ;TEST THE BISB
CMP R4,(R2) ;CORRECT RESULT
BEQ 00612 ;BR IF YES
E0612: MOV (R2),R3 ;GET THE WAS DATA
ERROR ;BISB DELIVERED THE WRONG RESULT
R0612 ;ERROR LOOP RETURN ADDRESS
00612: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

20663 ; *****
20664 ; .SBTTL T0613 BISB SMO,DM2 TEST - DEST ADDR EVEN
20665 ; *****
20666
20667 ;MICROPROGRAMMING / LOGIC INFORMATION
20668
20669 ;ROM SEQ: [162,260,267,224,367,375,016] FC 1,3,8
20670
20671 ;ACT BUTS: 37[004]100,162 / 33[260]220,224 / 16[367]016,016
20672
20673 ;EXEC: [224]ALUC=LLLLH :[367]D=000377
20674
20675 ;CODES: [367]SPS=3 / N:C=1000
20676
20677 ;SYNC: B05J2 (-) T=2.7 USEC
20678
20679 ;KEY SIG: K3-3 BIS L / K3-3 SM=0 L / K3-3 DM=1 L / K3-6 BYTE INSTR H
20680 ; K5-5 BCON(1+2) H
20681
20682 046114 012700 000613 T0613: MOV #0613,R0 ;LOAD R0 WITH TEST NO.
20683 046120 013701 046142 MOV @#I0613,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20684 046124 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF:0
20685 046130 012704 000377 MOV #377,R4 ;RESULT S / B = 377
20686 046134 010203 R0613: MOV R2,R3 ;DEST ADDR IN R3
20687 046136 005012 CLR (R2) ;[DEST] = 000000
20688 046140 000257 CCC ;SCOPE SYNC
20689
20690 046142 150423 I0613: BISB R4,(R3)+ ;TEST THE BISB
20691
20692 046144 020412 CMP R4,(R2) ;CORRECT RESULT
20693 046146 001403 BEQ 00613 ;BR IF YES
20694
20695 046150 011203 MOV (R2),R3 ;GET THE WAS DATA
20696 046152 104000 E0613: ERROR ;BISB DELIVERED THE WRONG RESULT
20697 046154 046134 R0613 ;ERROR LOOP RETURN ADDRESS
20698
20699 046156 000004 O0613: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```
20700 ; *****  
20701 ; .SBTTL T0614 BISB SMO,DM1 TEST - DEST ADDR ODD  
20702 ; *****  
20703  
20704 ;MICROPROGRAMMING / LOGIC INFORMATION  
20705  
20706 ;ROM SEQ: [161,266,267,237,270,230,254,074,366,375,016] FC 1,3,8  
20707  
20708 ;ACT BUTS: 37[004]100,161 / 33[266]220,237 / 34[237]220,230 / 16[366]016,016  
20709  
20710 ;EXEC: [230]ALUC=LLLLH :[375]D=177777  
20711  
20712 ;CODES: [254]SPS=1,[074]SPS=3 / N:C=1000  
20713  
20714 ;SYNC: B05J2 (-) T=3 USEC  
20715  
20716 ;KEY SIG: K3-3 BIS L / K3-3 SM=0 L / K3-3 DM=1 L / K3-6 BYTE INSTR H  
20717 ; K3-7 ODD BYTE H / K1-6 BA00(1) H  
20718  
20719 046160 012700 000614 T0614: MOV #0614,R0 ;LOAD R0 WITH TEST NO.  
20720 046164 013701 046214 MOV @#I0614,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
20721 046170 012702 067560 MOV #MBUF0,R2 ;DEST ADDR = MBUF0  
20722 046174 012704 177400 MOV #177400,R4 ;RESULT S / B = 177400  
20723 046200 012705 000377 MOV #377,R5 ;[R5]=SOURCE OPR = 377  
20724 046204 012703 067561 R0614: MOV #MBUF0+1,R3 ;ODD DEST ADDR IN R3  
20725 046210 005012 CLR (R2) ;[DEST] = 000000  
20726 046212 000257 CCC ;SCOPE SYNC  
20727  
20728 046214 150513 I0614: BISB R5,(R3) ;TEST THE BISB  
20729  
20730 046216 020412 CMP R4,(R2) ;CORRECT RESULT  
20731 046220 001403 BEQ 00614 ;BR IF YES  
20732  
20733 046222 011203 MOV (R2),R3 ;GET THE WAS DATA  
20734 046224 104000 E0614: ERROR ;BISB DELIVERED THE WRONG RESULT  
20735 046226 046204 R0614 ;ERROR LOOP RETURN ADDRESS  
20736  
20737 046230 000004 00614: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

20738
20739
20740
20741
20742
20743
20744
20745
20746
20747
20748
20749
20750
20751
20752
20753
20754
20755
20756
20757
20758
20759
20760
20761
20762
20763
20764
20765
20766
20767
20768
20769
20770
20771
20772
20773

046232 012700 000615
046236 013701 046260
046242 012702 067560
046246 012704 000377

046260 150413

046262 020412
046264 001403

046266 011203
046270 104000
046272 046252

046274 000004

```
; *****  
; .SETTL T0615 BISB SMO,DM1 TEST - DEST ADDR EVEN  
; *****  
  
;MICROPROGRAMMING / LOGIC INFORMATION  
  
;ROM SEQ: [161,266,267,224,367,375,016] FC 1,3,8  
;ACT BUTS: 37[004]100,161 / 33[266]220,224 / 16[367]016,016  
;EXEC: [224]ALUC=LLLLH :[367]D=000377  
;CODES: [367]SPS=3 / N:C=1000  
;SYNC: B05J2 (-) T=2.6 USEC  
;KEY SIG: K3-3 BIS L / K3-3 SM=0 L / K3-3 DM=1 L / K3-6 BYTE INSTR H  
  
T0615: MOV #0615,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0615,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV #M0615,R2 ;DEST ADDR = M0615  
MOV #377,R4 ;RESULT S / B = 377  
R0615: MOV R2,R3 ;DEST ADDR IN R3  
CLR (R2) ;[DEST] = 000000  
CCC ;SCOPE SYNC  
  
I0615: BISB R4,(R3) ;TEST THE BISB  
  
CMP R4,(R2) ;CORRECT RESULT  
BEQ 00615 ;BR IF YES  
  
E0615: MOV (R2),R3 ;GET THE WAS DATA  
ERROR ;BISB DELIVERED THE WRONG RESULT  
R0615 ;ERROR LOOP RETURN ADDRESS  
  
00615: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

20774
20775
20776
20777
20778
20779
20780
20781
20782
20783
20784
20785
20786
20787
20788
20789
20790
20791
20792
20793
20794 046276 012700 000616
20795 046302 013701 046332
20796 046306 012702 067560
20797 046312 012704 177400
20798 046316 012705 070131
20799 046322 012703 067561
20800 046326 005012
20801 046330 000257
20802
20803 046332 151513
20804
20805 046334 020412
20806 046336 001403
20807
20808 046340 011203
20809 046342 104000
20810 046344 046316
20811
20812 046346 000004

```

; *****
; .SBTTL T0616 BISB SM1,DM1 TEST - DEST ADDR ODD
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ:      [141,247,250,137,251,161,266,267,237,270,231,254,074,366,375,016] FC 1,2
;ACT BUTS:     37[004]100,141 / 35[247]120,137 / 36[137]120,161 / 33[266]220,237
;              ; / 34[237]220,231 / 16[366]016,016

;EXEC:         [231]ALUC=LLLLH :[375]D=177777

;CODES:        [254]SPS=1,[074]SPS=3 / N:C=1000

;SYNC:         B05J2 (-) T=4.2 USEC

;KEY SIG:      K3-3 BIS L / K3-3 SM=1 L / K3-3 DM=1 L / K3-6 BYTE INSTR H
;              ; K3-7 ODD BYTE H / K1-6 BA00(1) H

T0616: MOV #0616,R0 ;LOAD R0 WITH TEST NO.
        MOV @#10616,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV #MBUF0,R2 ;DEST ADDR = MBUF0
        MOV #177400,R4 ;RESULT S / B = 177400
R0616: MOV #DBTA+1,R5 ;SOURCE ADDR = DBTA+1
        MOV #MBUF0+1,R3 ;ODD DEST ADDR IN R3
        CLR (R2) ;[DEST] = 000000
        CCC ;SCOPE SYNC

I0616: BISB (R5),(R3) ;TEST THE BISB

        CMP R4,(R2) ;CORRECT RESULT
        BEQ 00616 ;BR IF YES

E0616: MOV (R2),R3 ;GET THE WAS DATA
        ERROR ;BISB DELIVERED THE WRONG RESULT
        R0616 ;ERROR LOOP RETURN ADDRESS

00616: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

20813
20814
20815
20816
20817
20818
20819
20820
20821
20822
20823
20824
20825
20826
20827
20828
20829
20830
20831 046350 012700 000617
20832 046354 013701 046366
20833
20834 046360 012702 046376
20835 046364 000277
20836
20837 046366 000112
20838
20839 046370 104006
20840 046372 046360
20841 046374 000406
20842
20843 046376 103003
20844 046400 102002
20845 046402 001001
20846 046404 100402
20847
20848 046406 104006
20849 046410 046360
20850
20851 046412 000004
20852
20853

```

; *****
; .SBTTL T0617 JMP MODE 1 TEST, FLAGS = 1111
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ:      [151,300,306,313,016] FC 1,5

;ACT BUTS:     37[004]100,151 / 15[151]306,306 / 16[306]016,016

;EXEC:         [306]ALUC=LLLLL :[313] D = #A0617

;CODES:        N:C = 1111 (NO CHANGE)

;SYNC:         B05J2 (-) T = 1.8 USEC

;KEY SIG:      K3-3 DM=1L / K3-5 JMP L / K3-5 JMP+JSR H

T0617:  MOV    #0617,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0617,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD

R0617:  MOV    #A0617,R2        ;R2 CONTAINS JUMP ADDRESS
        SCC                    ;MAKE N:C = 1111

I0617:  JMP    (R2)            ;TEST THE JMP - GO TO A0617

E10617: ERROR6                ;JMP FAILED TO LOAD PC
        R0617                ;ERROR LOOP RETURN
        BR     00617          ;GO CALL SCOPE

A0617:  BCC    E20617          ;BR IF JMP CLEARED 'C'
        BVC    E20617          ;BR IF JMP CLEARED 'V'
        BNE    E20617          ;BR IF JMP CLEARED 'Z'
        BMI    00617          ;BR IF 'N' STILL SET

E20617: ERROR6                ;JMP ALTERED CODES - CLEARED ONE
        R0617                ;ERROR LOOP RETURN

00617:  SCOPE                  ;CALL SCOPE LOOP UTILITY

```

```

20854 ; *****
20855 ; .SBTTL T0620 JMP MODE 1 TEST, FLAGS = 0000
20856 ; *****
20857
20858 ;MICROPROGRAMMING / LOGIC INFORMATION
20859
20860 ;ROM SEQ: [151,300,306,313,016] FC 1,5
20861
20862 ;ACT BUTS: 37[004]100,151 / 15[151]306,306 / 16[306]016,016
20863
20864 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0620
20865
20866 ;CODES: N:C = 0000 (NO CHANGE)
20867
20868 ;SYNC: B05J2 (-) T = 1.8 USEC
20869
20870 ;KEY SIG: K3-3 DM=1L / K3-5 JMP L / K3-5 JMP+JSR H
20871
20872 046414 012700 000620 T0620: MOV #0620,R0 ;LOAD R0 WITH TEST NO.
20873 046420 013701 046432 MOV @#I0620,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20874
20875 046424 012702 046442 R0620: MOV #A0620,R2 ;R2 CONTAINS JMP ADDRESS
20876 046430 000257 CCC ;MAKE N:C = 0000
20877
20878 046432 000112 I0620: JMP (R2) ;TEST THE JMP - GO TO A0620
20879
20880 046434 104006 E10620: ERROR6 ;JMP FAILED TO LOAD PC
20881 046436 046424 R0620 ;ERROR LOOP RETURN
20882 046440 000406 BR 00620 ;GO CALL SCOPE
20883
20884 046442 103403 A0620: BCS E20620 ;BR IF JMP SET 'C'
20885 046444 102402 BVS E20620 ;BR IF JMP SET 'V'
20886 046446 001401 BEQ E20620 ;BR IF JMP SET 'Z'
20887 046450 100002 BPL 00620 ;BR IF 'N' STILL CLEAR
20888
20889 046452 104006 E20620: ERROR6 ;JMP ALTERED CODES - SET ONE
20890 046454 046424 R0620 ;ERROR LOOP RETURN
20891
20892 046456 000004 00620: SCOPE ;CALL SCOPE LOOP UTILITY
20893
20894

```

```

20895 ; *****
20896 ; .SBTTL T0621 JMP MODE 2 TEST; FLAGS = 1111
20897 ; *****
20898 046460 012700 000621 T0621: MOV #0621,R0 ;LOAD R0 WITH TEST NO.
20899 ;MICROPROGRAMMING / LOGIC INFORMATION
20900 ;ROM SEQ: [152,235,300,306,313,016] FC 1,5
20901 ;ACT BUTS: 37[004]100,152 / 15[235]306,306 / 16[306]016,016
20902 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0621
20903 ;CODES: N:C = 1111 (NO CHANGE)
20904 ;SYNC: B05J2 (-) T = 2.1 USEC
20905 ;KEY SIG: K3-3 DM=2L / K3-5 JMP L / K3-5 JMP+JSR H / K5-5 BC01 H
20906
20907
20908
20909
20910
20911
20912
20913 046464 013701 046476 MOV @#I0621,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20914
20915 046470 012702 046506 R0621: MOV #A0621,R2 ;R2 CONTAINS JUMP ADDRESS
20916 046474 000277 SCC ;SET N:C = 1111
20917
20918 046476 000122 I0621: JMP (R2)+ ;TEST THE JMP - GO TO A0621
20919
20920 046500 104006 E10621: ERROR6 ;JMP FAILED TO LOAD PC
20921 046502 046470 R0621 ;ERROR LOOP RETURN
20922 046504 000413 BR 00621 ;GO TO SCOPE EXIT
20923
20924 046506 103003 A0621: BCC E20621 ;BR IF JMP CLEARED 'C'
20925 046510 102002 BVC E20621 ;BR IF JMP CLEARED 'V'
20926 046512 001001 BNE E20621 ;BR IF JMP CLEARED 'Z'
20927 046514 100402 BMI B0621 ;BR IF 'N' STILL SET
20928
20929 046516 104006 E20621: ERROR6 ;JMP ALTERED CODES - CLEARED
20930 046520 046470 R0621 ;ERROR LOOP RETURN
20931
20932 046522 022702 046510 B0621: CMP #A0621+2,R2 ;DID R2 GET AUTO-INCREMENTED?
20933 046526 001402 BEQ 00621 ;BR IF YES
20934
20935 046530 104006 E30621: ERROR6 ;JMP FAILED TO UPDATE REGISTER (R2)
20936 046532 046470 R0621 ;ERROR LOOP RETURN
20937
20938 046534 000004 00621: SCOPE ;CALL SCOPE LOOP UTILITY
20939
20940

```

```

20941 ; *****
20942 ; .SBTTL T0622 JMP MODE 2 TEST; FLAGS = 0000
20943 ; *****
20944
20945 ;MICROPROGRAMMING / LOGIC INFORMATION
20946
20947 ;ROM SEQ: [152,235,300,306,313,016] FC 1,5
20948
20949 ;ACT BUTS: 37[004]100,152 / 15[235]306,306 / 16[306]016,016
20950
20951 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0622
20952
20953 ;CODES: N:C = 0000 (NO CHANGE)
20954
20955 ;SYNC: B05J2 (-) T = 2.1 USEC
20956
20957 ;KEY SIG: K3-3 DM=2L / K3-5 JMP L / K3-5 JMP+JSR H / K5-5 BC01 H
20958
20959 046536 012700 000622 T0622: MOV #0622,R0 ;LOAD R0 WITH TEST NO.
20960 046542 013701 046554 MOV @#I0622,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
20961
20962 046546 012702 046564 R0622: MOV #A0622,R2 ;R2 CONTAINS JUMP ADDRESS
20963 046552 000257 CCC ;MAKE N:C = 0000
20964
20965 046554 000122 I0622: JMP (R2)+ ;TEST THE JMP - GO TO A0622
20966
20967 046556 104006 E10622: ERROR6 ;JMP FAILED TO LOAD PC
20968 046560 046546 R0622 ;ERROR LOOP RETURN
20969 046562 000406 BR 00622 ;GO TO SCOPE EXIT
20970
20971 046564 103403 A0622: BCS E20622 ;BR IF JMP SET 'C'
20972 046566 102402 BVS E20622 ;BR IF JMP SET 'V'
20973 046570 001401 BEQ E20622 ;BR IF JMP SET 'Z'
20974 046572 100002 BPL 00622 ;BR IF 'N' IS CLEAR
20975
20976 046574 104006 E20622: ERROR6 ;JMP ALTERED CODES - SET
20977 046576 046546 R0622 ;ERROR LOOP RETURN
20978
20979 046600 000004 00622: SCOPE ;CALL SCOPE LOOP UTILITY
20980
20981

```

20982
20983
20984
20985
20986
20987
20988
20989
20990
20991
20992
20993
20994
20995
20996
20997
20998
20999
21000
21001
21002
21003
21004
21005
21006
21007
21008
21009
21010
21011
21012
21013
21014
21015
21016
21017
21018
21019
21020
21021
21022
21023
21024
21025
21026
21027
21028
21029
21030
21031
21032
21033
21034

046602 012700 000623
046606 013701 046620
046612 012702 046660
046616 000277
046620 000132
046622 104006
046624 046612
046626 000417
046630 103003
046632 102002
046634 001001
046636 100402
046640 104006
046642 046612
046644 022702 046662
046650 001406
046652 104006
046654 046612
046656 000403
046660 046630
046662 104006
046664 046612
046666 000004

; *****
; .SBTTL T0623 JMP TEST MODE 3; FLAGS = 1111
; *****
;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ: [153,303,306,313,016] FC 1,5
;ACT BUTS: 37[004]100,153 / 15[153]306,306 / 16[306]016,016
;EXEC: [306]ALUC=LLLLL :[313] D = #A0623
;CODES: N:C = 1111 (NO CHANGE)
;SYNC: B05J2 (-) T = 2.3 USEC
;KEY SIG: K3-3 DM=3L / K3-5 JMP L / K3-5 JMP+JSR H / K5-5 BC01 H

T0623: MOV #0623,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0623,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0623: MOV #JMP3,R2 ;R2 CONTAINS ADDRESS OF JUMP ADDRESS
SCC ;SET N:C = 1111
I0623: JMP @ (R2)+ ;TEST THE JMP - GO TO A0623
E10623: ERROR6 ;JMP FAILED TO LOAD PC
R0623 ;ERROR LOOP RETURN
BR 00623 ;GO TO SCOPE EXIT
A0623: BCC E20623 ;BR IF JMP CLEARED 'C'
BVC E20623 ;BR IF JMP CLEARED 'V'
BNE E20623 ;BR IF JMP CLEARED 'Z'
BMI B0623 ;BR IF 'N' STILL SET
E20623: ERROR6 ;JMP ALTERED CODES - CLEAR
R0623 ;ERROR LOOP RETURN
B0623: CMP #JMP3+2,R2 ;DID JMP UPDATE R2?
BEQ 00623 ;BR IF YES
E30623: ERROR6 ;JMP FAILED TO UPDATE REGISTER
R0623 ;ERROR LOOP RETURN
BR 00623 ;GO TO SCOPE EXIT
JMP3: A0623 ;JMP3 CONTAINS JUMP ADDRESS
E40623: ERROR6 ;ERROR CALL OCCURS IF MODE3 HAPPENS
R0623 ;ERROR LOOP RETURN
;TO EXECUTE AS MODE 1 OR 2 AND
;A0623 IS LEGAL INSTRUCTION
00623: SCOPE ;CALL SCOPE LOOP UTILITY

21035
21036
21037
21038
21039
21040
21041
21042
21043
21044
21045
21046
21047
21048
21049
21050
21051
21052
21053
21054
21055
21056
21057
21058
21059
21060
21061
21062
21063
21064
21065
21066
21067
21068
21069
21070
21071
21072
21073
21074
21075
21076
21077
21078
21079
21080
21081

046670 012700 000624
046674 013701 046706
046700 012702 046734
046704 000257
046706 000132
046710 104006
046712 046700
046714 000412
046716 103403
046720 102402
046722 001401
046724 100006
046726 104006
046730 046700
046732 000403
046734 046716
046736 104006
046740 046700
046742 000004

```
; *****  
; .SBTTL T0624 JMP TEST MODE 3; FLAGS = 0000  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [153,303,306,313,016] FC 1,5  
;ACT BUTS: 37[004]100,153 / 15[153]306,306 / 16[306]016,016  
;EXEC: [306]ALUC=LLLLL :[313] D = #A0624  
;CODES: N:C = 0000 (NO CHANGE)  
;SYNC: B05J2 (-) T = 2.3 USEC  
;KEY SIG: K3-3 DM=3L / K3-5 JMP L / K3-5 JMP+JSR H / K5-5 BC01 H  
T0624: MOV #0624,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0624,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0624: MOV #JMP3A,R2 ;R2 CONTAINS ADDRESS OF JUMP ADDRESS  
CCC ;MAKE N:C = 0000  
I0624: JMP @ (R2)+ ;TEST THE JMP - GO TO A0624  
E10624: ERROR6 ;JMP FAILED TO LOAD THE PC  
R0624 ;ERROR LOOP RETURN  
BR 00624 ;GO TO SCOPE EXIT  
A0624: BCS E20624 ;BR IF JMP SET 'C'  
BVS E20624 ;BR IF JMP SET 'V'  
BEQ E20624 ;BR IF JMP SET 'Z'  
BPL 00624 ;BR IF 'N' STILL CLEAR  
E20624: ERROR6 ;JMP ALTERED CODES - SET  
R0624 ;ERROR LOOP RETURN  
BR 00624 ;GO TO SCOPE EXIT  
JMP3A: A0624 ;JUMP ADDRESS IN JMP3A  
E30624: ERROR6 ;JMP MODE 3 EXECUTED LIKE MODE 1 OR 2  
R0624 ;ERROR LOOP RETURN  
00624: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
21082 ; *****  
21083 ; .SBTTL T0625 JMP TEST MODE 4; FLAGS = 1111  
21084 ; *****  
21085  
21086 :MICROPROGRAMMING / LOGIC INFORMATION  
21087  
21088 :ROM SEQ: [154,300,306,313,016] FC 1,5  
21089  
21090 :ACT BUTS: 37[004]100,154 / 15[154]306,306 / 16[306]016,016  
21091  
21092 :EXEC: [306]ALUC=LLLLL :[313] D = #E20625-2  
21093  
21094 :CODES: N:C = 1111 (NO CHANGE)  
21095  
21096 :SYNC: B05J2 (-) T = 1.9 USEC  
21097  
21098 :KEY SIG: K3-3 DM=4L / K3-5 JMP L / K3-5 JMP+JSR H / K5-5 BC01 H  
21099  
21100 046744 012700 000625 T0625: MOV #0625,R0 ;LOAD R0 WITH TEST NO.  
21101 046750 013701 046762 MOV @#10625,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
21102 046754 012702 046774 R0625: MOV #E20625,R2 ;[R2] = JMP ADDRESS PLUS 2  
21103 046760 000277 SCC ;MAKE N:C = 1111  
21104  
21105 046762 000142 I0625: JMP -(R2) ;TEST THE JMP - GO TO E20625 MINUS 2  
21106  
21107 046764 104006 E10625: ERROR6 ;JMP FAILED TO LOAD PC  
21108 046766 046754 R0625 ;ERROR LOOP RETURN  
21109 046770 000417 BR 00625 ;GO TO SCOPE EXIT  
21110  
21111 046772 000403 BR A0625 ;GO TEST FLAGS - JMP LOADED PC OK  
21112 046774 104006 E20625: ERROR6 ;JMP FAILED TO AUTO-DECREMENT R2  
21113 046776 046754 R0625 ;ERROR LOOP RETURN  
21114 047000 000413 BR 00625 ;GO TO SCOPE EXIT  
21115  
21116 047002 103003 A0625: BCC E30625 ;BR IF JMP CLEARED 'C'  
21117 047004 102002 BVC E30625 ;BR IF JMP CLEARED 'V'  
21118 047006 001001 BNE E30625 ;BR IF JMP CLEARED 'Z'  
21119 047010 100402 BMI B0625 ;BR IF 'N' STILL SET  
21120  
21121 047012 104006 E30625: ERROR6 ;JMP ALTERED FLAGS  
21122 047014 046754 R0625 ;ERROR LOOP RETURN  
21123  
21124 047016 022702 046772 B0625: CMP #E20625-2,R2 ;DID JMP UPDATE R2 PROPERLY?  
21125 047022 001402 BEQ 00625 ;BR IF YES  
21126  
21127 047024 104006 E40625: ERROR6 ;JMP FAILED TO UPDATE REGISTER  
21128 047026 046754 R0625 ;ERROR LOOP RETURN  
21129  
21130 047030 000004 00625: SCOPE ;CALL SCOPE LOOP UTILITY  
21131  
21132
```

21133
21134
21135
21136
21137
21138
21139
21140
21141
21142
21143
21144
21145
21146
21147
21148
21149
21150
21151
21152
21153
21154
21155
21156
21157
21158
21159
21160
21161
21162
21163
21164
21165
21166
21167
21168
21169
21170
21171
21172
21173

047032 012700 000626
047036 013701 047050
047042 012702 047062
047046 000257
047050 000142
047052 104006
047054 047042
047056 000406
047060 103403
047062 102402
047064 001401
047066 100002
047070 104006
047072 047042
047074 000004

```
; *****  
      .SBTTL T0626 JMP TEST MODE 4; FLAGS = 0000  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ:      [154,300,306,313,016] FC 1,5  
;ACT BUTS:     37[004]100,154 / 15[154]306,306 / 16[306]016,016  
;EXEC:        [306]ALUC=LLLLL :[313] D = #A0626  
;CODES:       N:C = 0000 (NO CHANGE)  
;SYNC:        B05J2 (-) T = 1.9 USEC  
;KEY SIG:     K3-3 DM=4L / K3-5 JMP L / K3-5 JMP+JSR H / K5-5 BC01 H  
T0626:  MOV    #0626,R0          ;LOAD R0 WITH TEST NO.  
        MOV    @#I0626,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0626:  MOV    #A0626+2,R2     ;[R2] = JUMP ADDRESS PLUS 2  
        CCC                          ;MAKE N:C = 0000  
I0626:  JMP    -(R2)           ;TEST THE JMP - TO TO A0626  
E10626:  ERROR6                ;JMP FAILED TO LOAD PC  
        R0626                ;ERROR LOOP RETURN  
        BR     00626           ;GO TO SCOPE EXIT  
A0626:  BCS    E20626          ;BR IF JMP SET 'C'  
        BVS    E20626          ;BR IF JMP SET 'V'  
        BEQ    E20626          ;BR IF JMP SET 'Z'  
        BPL    00626           ;BR IF 'N' STILL CLEAR  
E20626:  ERROR6                ;JMP ALTERED CODES - SET  
        R0626                ;ERROR LOOP RETURN  
00626:  SCOPE                  ;CALL SCOPE LOOP UTILITY
```

21174
21175
21176
21177
21178
21179
21180
21181
21182
21183
21184
21185
21186
21187
21188
21189
21190
21191
21192
21193
21194
21195
21196
21197
21198
21199
21200
21201
21202
21203
21204
21205
21206
21207
21208
21209
21210
21211
21212
21213
21214
21215
21216
21217
21218
21219
21220
21221
21222
21223
21224

047076 012700 000627
047102 013701 047114
047106 012702 047156
047112 000277
047114 000152
047116 104006
047120 047106
047122 000417
047124 103003
047126 102002
047130 001001
047132 100402
047134 104006
047136 047106
047140 022702 047154
047144 001406
047146 104006
047150 047106
047152 000403
047154 047124
047156 104006
047160 047106
047162 000004

```
; *****  
      .SBTTL T0627 JMP TEST MODE 5; FLAGS = 1111  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ:      [155,303,306,313,016] FC 1,5  
;ACT BUTS:     37[004]100,155 / 15[155]306,306 / 16[306]016,016  
;EXEC:         [306]ALUC=LLLLL :[313] D = #A0627  
;CODES:        N:C = 1111 (NO CHANGE)  
;SYNC:         B05J2 (-) T = 2.3 USEC  
;KEY SIG:      K3-3 DM=5L / K3-5 JMP L / K3-5 JMP+JSR H / K5-5 BC01 H  
T0627:  MOV    #0627,R0          ;LOAD R0 WITH TEST NO.  
        MOV    @#I0627,R1       ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0627:  MOV    #JMP5,R2         ;JMP CONTAINS ADDR+2 OF JUMP ADDRESS  
        SCC  
I0627:  JMP    @-(R2)           ;TEST THE JMP - GO TO A0627  
E10627:  ERROR6 R0627          ;JMP FAILED TO LOAD PC  
        BR    00627            ;ERROR LOOP RETURN  
        ;GO TO SCOPE OXIT  
A0627:  BCC    E20627           ;BR IF JMP CLEARED 'C'  
        BVC    E20627  
        BNE    E20627  
        BMI    B0627  
E20627:  ERROR6 R0627          ;JMP ALTERED CODES - CLEARED  
        ;ERROR LOOP RETURN  
B0627:  CMP    #JMP5-2,R2      ;DID R2 GET AUTO-DECREMENTED  
        BEQ    00627           ;BR IF YES  
E30627:  ERROR6 R0627          ;JMP FAILED TO UPDATE REGISTER  
        BR    00627            ;ERROR LOOP RETURN  
        ;GO TO SCOPE EXIT  
        ;THIS LOCATION CONTAINS JMP ADDRESS  
JMP5:   ERROR6 R0627          ;JMP EXECUTED LIKE A MODE 1 OR 2  
        ;ERROR LOOP RETURN  
00627:  SCOPE                  ;CALL SCOPE LOOP UTILITY
```

```

21225 ; *****
21226 ; .SBTTL T0630 JMP TEST MODE 5; FLAG = 0000
21227 ; *****
21228
21229 ;MICROPROGRAMMING / LOGIC INFORMATION
21230
21231 ;ROM SEQ: [155,303,306,313,016] FC 1,5
21232
21233 ;ACT BUTS: 37[004]100,155 / 15[155]306,306 / 16[306]016,016
21234
21235 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0630
21236
21237 ;CODES: N:C = 0000 (NO CHANGE)
21238
21239 ;SYNC: B05J2 (-) T = 2.3 USEC
21240
21241 ;KEY SIG: K3-3 DM=5L / K3-5 JMP L / K3-5 JMP+JSR H / K5-5 BC01 H
21242
21243 047164 012700 000630 T0630: MOV #0630,R0 ;LOAD R0 WITH TEST NO.
21244 047170 013701 047202 MOV @#I0630,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21245
21246 047174 012702 047232 R0630: MOV #JMP5A,R2 ;[R2] = ADDR +2 OF JUMP ADDRESS
21247 047200 000257 CCC ;SET N:C = 0000
21248
21249 047202 000152 I0630: JMP @-(R2) ;TEST THE JMP - GO TO A0630
21250
21251 047204 104006 E10630: ERROR6 ;JMP FAILED TO LOAD PC
21252 047206 047174 R0630 ;ERROR LOOP RETURN
21253 047210 000412 BR 00630 ;GO TO SCOPE EXIT
21254
21255 047212 103403 A0630: BCS E20630 ;BR IF JMP SET 'C'
21256 047214 102402 BVS E20630 ;BR IF JMP SET 'V'
21257 047216 001401 BEQ E20630 ;BR IF JMP SET 'Z'
21258 047220 100006 BPL 00630 ;BR IF 'N' STILL CLEAR
21259
21260 047222 104006 E20630: ERROR6 ;JMP ALTERED THE CODES - SET
21261 047224 047174 R0630 ;ERROR LOOP RETURN
21262 047226 000403 BR 00630 ;GO TO SCOPE EXIT
21263
21264 047230 047212 JMP5A: A0630 ;THIS LOCATION CONTAINS JUMP ADDRESS
21265 047232 104006 ERROR6 ;JMP EXECUTED LIKE A MODE 1 OR 2
21266 047234 047174 R0630 ;ERROR LOOP RETURN
21267
21268 047236 000004 00630: SCOPE ;CALL SCOPE LOOP UTILITY
21269
21270

```

```

21271 ; *****
21272 ; .SBTTL T0631 JMP TEST MODE 6; FLAGS = 1111
21273 ; *****
21274
21275 ;MICROPROGRAMMING / LOGIC INFORMATION
21276
21277 ;ROM SEQ: [156,304,305,300,306,313,016] FC 1,5
21278
21279 ;ACT BUTS: 37[004]100,156 / 15[305]306,306 / 16[306]016,016
21280
21281 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0631
21282
21283 ;CODES: N:C = 1111 (NO CHANGE)
21284
21285 ;SYNC: B05J2 (-) T = 2.4 USEC
21286
21287 ;KEY SIG: K3-3 DM=6L / K3-5 JMP L / K3-5 JMP+JSR H
21288
21289 047240 012700 000631 T0631: MOV #0631,R0 ;LOAD R0 WITH TEST NO.
21290 047244 013701 047256 MOV @#I0631,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21291
21292 047250 012702 047306 R0631: MOV #E30631,R2 ;[R2] = BASE ADDRESS TO BE INDEXED
21293 047254 000277 SCC ;MAKE N:C = 1111
21294
21295 047256 000162 177762 I0631: JMP A0631-E30631(R2) ;TEST THE JMP - GO TO A0631
21296
21297 047262 104006 E10631: ERROR6 ;JMP FAILED TO LOAD THE PC
21298 047264 047250 R0631 ;ERROR LOOP RETURN
21299 047266 000411 BR 00631 ;GO TO SCOPE EXIT
21300
21301 047270 103003 A0631: BCC E20631 ;BR IF JMP CLEARED 'C'
21302 047272 102002 BVC E20631
21303 047274 001001 BNE E20631
21304 047276 100405 BMI 00631 ;BR IF 'N' STILL SET
21305
21306 047300 104006 E20631: ERROR6 ;JMP ALTERED CODES - CLEARED
21307 047302 047250 R0631 ;ERROR LOOP RETURN
21308 047304 000402 BR 00631 ;GO TO SCOPE EXIT
21309
21310 047306 104006 E30631: ERROR6 ;JMP EXECUTED LIKE A MODE 1 OR 2 OR
21311 047310 047250 R0631 ;ERROR LOOP RETURN
21312 ;FAILED TO INDEX [R2]
21313
21314 047312 000004 00631: SCOPE ;CALL SCOPE LOOP UTILITY
21315
21316

```

```

21317 ; *****
21318 ; .SBTTL T0632 JMP TEST MODE 6; FLAGS = 0000
21319 ; *****
21320
21321 ;MICROPROGRAMMING / LOGIC INFORMATION
21322
21323 ;ROM SEQ: [156,304,305,300,306,313,016] FC 1,5
21324
21325 ;ACT BUTS: 37[004]100,156 / 15[305]306,306 / 16[306]016,016
21326
21327 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0632
21328
21329 ;CODES: N:C = 0000 (NO CHANGE)
21330
21331 ;SYNC: B05J2 (-) T = 2.4 USEC
21332
21333 ;KEY SIG: K3-3 DM=6L / K3-5 JMP L / K3-5 JMP+JSR H
21334
21335 047314 012700 000632 T0632: MOV #0632,R0 ;LOAD R0 WITH TEST NO.
21336 047320 013701 047332 MOV @#I0632,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21337
21338 047324 012702 047362 R0632: MOV #E30632,R2 ;[R2] = BASE ADDRESS FOR JUMP
21339 047330 000257 CCC ;MAKE N:C = 0000
21340
21341 047332 000162 177762 I0632: JMP A0632-E30632(R2) ;TEST THE JMP -- GO TO A0632
21342
21343 047336 104006 E10632: ERROR6 ;JMP FAILED TO LOAD PC
21344 047340 047324 R0632 ;ERROR LOOP RETURN
21345 047342 000411 BR 00632 ;GO TO SCOPE EXIT
21346
21347 047344 103403 A0632: BCS E20632 ;BR IF JMP SET 'C'
21348 047346 102402 BVS E20632 ;BR IF JMP SET 'V'
21349 047350 001401 BFQ E20632 ;BR IF JMP SET 'Z'
21350 047352 100005 BFL 00632 ;BR IF 'N' STILL CLEAR
21351
21352 047354 104006 E20632: ERROR6 ;JMP ALTERED CODES
21353 047356 047324 R0632 ;ERROR LOOP RETURN
21354 047360 000402 BR 00632 ;GO TO SCOPE EXIT
21355
21356 047362 104006 E30632: ERROR6 ;JMP EXECUTED LIKE A MODE 1 OR 2, OR
21357 047364 047324 R0632 ;ERROR LOOP RETURN
21358 ;FAILED TO INDEX [R2]
21359
21360 047366 000004 00632: SCOPE ;CALL SCOPE LOOP UTILITY
21361
21362

```

```

21363 ; *****
21364 ; .SBTTL T0633 JMP TEST MODE 7; FLAGS = 1111
21365 ; *****
21366
21367 ;MICROPROGRAMMING / LOGIC INFORMATION
21368
21369 ;ROM SEQ: [157,301,302,303,306,313,016] FC 1,5
21370
21371 ;ACT BUTS: 37[004]100,157 / 15[302]306,306 / 16[306]016,016
21372
21373 ;EXEC: [360]ALUC=LLLLL :[313] D = #A0633
21374
21375 ;CODES: N:C = 1111 (NO CHANGE)
21376
21377 ;SYNC: B05J2 (-) T = 3 USEC
21378
21379 ;KEY SIG: K3-3 DM=7L / K3-5 JMP L / K3-5 JMP+JSR H
21380
21381 047370 012700 000633 T0633: MOV #0633,R0 ;LOAD R0 WITH TEST NO.
21382 047374 013701 047406 MOV @#I0633,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21383
21384 047400 012702 047420 R0633: MOV #E20633,R2 ;[R2] = BASE ADDRESS
21385 047404 000277 SCC ;MAKE N:C = 1111
21386
21387 047406 000172 000024 I0633: JMP @JMP7-E20633(R2) ;TEST THE JMP - GO TO A0633
21388
21389 047412 104006 E10633: ERROR6 ;JMP FAILED TO LOAD PC
21390 047414 047400 R0633 ;ERROR LOOP RETURN
21391 047416 000415 BR 00633 ;GO TO SCOPE EXIT
21392
21393 047420 104006 E20633: ERROR6 ;JMP FAILED TO INDEX OR ACTED LIKE MODE 1 OR 2
21394 047422 047400 R0633 ;ERROR LOOP RETURN
21395 047424 000412 BR 00633 ;GO TO SCOPE EXIT
21396
21397 047426 103003 A0633: BCC E30633 ;BR IF JMP CLEARED 'C'
21398 047430 102002 BVC E30633 ;BR IF JMP CLEARED 'V'
21399 047432 001001 BNE E30633 ;BR IF JMP CLEARED 'Z'
21400 047434 100406 BMI 00633 ;BR IF 'N' STILL SET
21401
21402 047436 104006 E30633: ERROR6 ;JMP ALTERED CODES - CLEARED
21403 047440 047400 R0633 ;ERROR LOOP RETURN
21404 047442 000403 BR 00633 ;GO TO SCOPE EXIT
21405
21406 047444 047426 JMP7: A0633 ;THIS LOCATION CONTAINS JMP ADDRESS
21407
21408 047446 104006 E40633: ERROR6 ;JMP EXECUTED LIKE MODE 6
21409 047450 047400 R0633 ;ERROR LOOP RETURN
21410
21411 047452 000004 00633: SCOPE ;CALL SCOPE LOOP UTILITY
21412
21413

```

```
21414 ; *****  
21415 ; .SBTTL T0634 JMP TEST MODE 7; FLAGS = 0000  
21416 ; *****  
21417  
21418 ;MICROPROGRAMMING / LOGIC INFORMATION  
21419  
21420 ;ROM SEQ: [157,301,302,303,306,313,016] FC 1,5  
21421  
21422 ;ACT BUTS: 37[004]100,157 / 15[302]306,306 / 16[306]016,016  
21423  
21424 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0634  
21425  
21426 ;CODES: N:C = 0000 (NO CHANGE)  
21427  
21428 ;SYNC: B05J2 (-) T = 3 USEC  
21429  
21430 ;KEY SIG: K3-3 DM=7L / K3-5 JMP L / K3-5 JMP+JSR H  
21431  
21432 047454 012700 000634 T0634: MOV #0634,R0 ;LOAD R0 WITH TEST NO.  
21433 047460 013701 047472 MOV @#I0634,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
21434  
21435 047464 012702 047504 R0634: MOV #E20634,R2 ;[R2] = BASE ADDRESS  
21436 047470 000257 CCC ;MAKE N:C = 0000  
21437  
21438 047472 000172 000024 I0634: JMP @JMP7A-E20634(R2) ;TEST THE JMP - GO TO A0634  
21439  
21440 047476 104006 E10634: ERROR6 ;JMP FAILED TO LOAD PC  
21441 047500 047464 R0634 ;ERROR LOOP RETURN  
21442 047502 000415 BR 00634 ;GO TO SCOPE EXIT  
21443  
21444 047504 104006 E20634: ERROR6 ;JMP FAILED TO INDEX  
21445 047506 047464 R0634 ;ERROR LOOP RETURN  
21446 047510 000412 BR 00634 ;GO TO SCOPE EXIT  
21447  
21448 047512 103403 A0634: BCS E30634 ;BR IF JMP SET 'C'  
21449 047514 102402 BVS E30634 ;BR IF JMP SET 'V'  
21450 047516 001401 BEQ E30634 ;BR IF JMP SET 'Z'  
21451 047520 100006 BPL 00634 ;BR IF 'N' STILL CLEAR  
21452  
21453 047522 104006 E30634: ERROR6 ;JMP ALTERED CODES - SET  
21454 047524 047464 R0634 ;ERROR LOOP RETURN  
21455 047526 000403 BR 00634 ;GO TO SCOPE EXIT  
21456  
21457 047530 047512 JMP7A: A0634 ;THIS LOCATION CONTAINS JUMP ADDRESS  
21458  
21459 047532 104006 E40634: ERROR6 ;JMP EXECUTED LIKE A MODE 6  
21460 047534 047464 R0634 ;ERROR LOOP RETURN  
21461  
21462 047536 000004 00634: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

21463 ; *****
21464 ; .SBTTL T0635 JSR MODE 1 TEST - LOAD PC / PUSH SP
21465 ; *****
21466
21467 ;MICROPROGRAMMING / LOGIC INFORMATION
21468
21469 ;ROM SEQ: [151,300,307,310,311,312,306,313,016] FC 1,5
21470
21471 ;ACT BUTS: 37[004]100,151 / 15[151]306,307 / 16[306]016,016
21472
21473 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0635
21474
21475 ;CODES: N / A
21476
21477 ;SYNC: B05J2 (-) T = 3 USEC
21478
21479 ;KEY SIG: K3-3 DM=1L / K3-5 JMP+JSR H / K3-5 JSR H
21480
21481 047540 012700 000635 T0635: MOV #0635,R0 ;LOAD R0 WITH TEST NO.
21482 047544 013701 047560 MOV @#I0635,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21483 047550 010605 MOV SP,R5 ;SAVE THE SP
21484 047552 012702 047566 R0635: MOV #A0635,R2 ;DEST ADDR = A0635
21485 047556 000257 CCC ;SCOPE SYNC
21486
21487 047560 004412 I0635: JSR R4,(R2) ;TEST THE JSR - GO TO A0635
21488
21489 047562 104005 E10635: ERROR5 ;JSR FAILED TO LOAD THE PC
21490 047564 047552 R0635 ;ERROR LOOP RETURN
21491
21492 047566 005726 A0635: TST (SP)+ ;POP THE SP
21493 047570 020605 CMP SP,R5 ;DID JSR PUSH THE SP ?
21494 047572 001404 BEQ 00635 ;BR IF YES
21495
21496 047574 005746 E20635: TST -(SP) ;RESTORE ERROR SP
21497 047576 104005 ERROR5 ;JSR FAILED TO PUSH THE SP
21498 047600 047552 R0635 ;ERROR LOOP RETURN
21499
21500 047602 010506 MOV R5,SP ;RESTORE SP IN CASE OF ERROR
21501 047604 000004 00635: SCOPE ;CALL SCOPE LOOP UTILITY

```

```

21502 ; *****
21503 ; .SBTTL T0636 JSR MODE 1 TEST - CHECK RN AND OLD PC
21504 ; *****
21505
21506 ;MICROPROGRAMMING / LOGIC INFORMATION
21507
21508 ;ROM SEQ: [151,300,307,310,311,312,306,313,016] FC 1,5
21509
21510 ;ACT BUTS: 37[004]100,151 / 15[151]306,307 / 16[306]016,016
21511
21512 ;EXEC: [310] D = 125252 / [312] D = #E10636 / [313] D = #A0636
21513
21514 ;CODES: N / A
21515
21516 ;SYNC: B05J2 (-) T = 3 USEC
21517
21518 ;KEY SIG: K3-3 DM=1L / K3-5 JSR H / K3-5 JMP+JSR H
21519
21520 047606 012700 000636 T0636: MOV #0636,R0 ;LOAD R0 WITH TEST NO.
21521 047612 013701 047636 MOV @#I0636,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21522 047616 010605 MOV SP,R5 ;SAVE THE SP
21523 047620 012702 047644 R0636: MOV #A0636,R2 ;DEST ADDR = A0636
21524 047624 005066 177776 CLR -2(SP) ;INIT STACK LOC TO GET [R4]
21525 047630 012704 125252 MOV #125252,R4 ;INIT RN = 125252
21526 047634 000257 CCC ;SCOPE SYNC
21527
21528 047636 004412 I0636: JSR R4,(R2) ;TEST THE JSR - GO TO A0636
21529
21530 047640 104005 E10636: ERROR5 ;JSR FAILED TO LOAD THE PC
21531 047642 047620 R0636 ;ERROR LOOP RETURN
21532
21533 047644 022726 125252 A0636: CMP #125252,(SP)+ ;DID JSR SAVE REG ON STACK
21534 047650 001402 BEQ C0636 ;BR IF IT DID
21535
21536 047652 104005 E20636: ERROR5 ;JSR FAILED TO SAVE REG ON STACK
21537 047654 047620 R0636 ;ERROR LOOP RETURN
21538
21539 047656 022704 047640 C0636: CMP #E10636,R4 ;DID OLD PC GET SAVED ?
21540 047662 001402 BEQ B0636 ;BR IF YES
21541
21542 047664 104005 E30636: ERROR5 ;JSR FAILED TO SAVE TH OLD PC
21543 047666 047620 R0636 ;ERROR LOOP RETURN
21544
21545 047670 010506 B0636: MOV R5,SP ;RESTORE SP IN CASE ERROR SCREWED IT UP
21546 047672 000004 00636: SCOPE ;CALL SCOPE LOOP UTILITY
21547
21548

```

```

21549 ; *****
21550 ; .SBTTL T0637 JSR MODE 1 TEST - <N:C> = 0000
21551 ; *****
21552 047674 012700 000637 T0637: MOV #0637,R0 ;LOAD R0 WITH TEST NO.
21553 ;MICROPROGRAMMING / LOGIC INFORMATION
21554
21555 ;ROM SEQ: [151,300,307,310,311,312,306,313,016] FC 1,5
21556
21557 ;ACT BUTS: 37[004]100,151 / 15[151]306,307 / 16[306]016,016
21558
21559 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0637
21560
21561 ;CODES: N:C = 0000 (NO CHANGE)
21562
21563 ;SYNC: B05J2 (-) T = 3 USEC
21564
21565 ;KEY SIG: K3-3 DM=1L / K3-5 JSR H / K3-5 JMP+JSR H
21566
21567 047700 013701 047714 MOV @#I0637,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21568 047704 010605 MOV SP,R5 ;SAVE THE SP
21569 047706 012702 047722 R0637: MOV #A0637,R2 ;DEST ADDR = A0637
21570 047712 000257 CCC ;N:C = 0000
21571
21572 047714 004412 I0637: JSR R4,(R2) ;TEST THE JSR - GO TO A0637
21573
21574 047716 104005 E10637: ERROR5 ;JSR FAILED TO LOAD THE PC
21575 047720 047706 R0637 ;ERROR LOOP RETURN
21576
21577 047722 100403 A0637: BMI E20637 ;N:C = 0000 ?
21578 047724 001402 BEQ E20637
21579 047726 102401 BVS E20637
21580 047730 103002 BCC B0637
21581
21582 047732 104005 E20637: ERROR5 ;JSR FAILED - ALTERED FLAGS
21583 047734 047706 R0637 ;ERROR LOOP RETURN
21584
21585 047736 010506 B0637: MOV R5,SP ;RESET SP IN CASE OF ERROR
21586 047740 000004 00637: SCOPE ;CALL SCOPE LOOP UTILITY

```

```

21587 ; *****
21588 ; .SBTTL T0640 JSR MODE 1 TEST - <N:C> = 1111
21589 ; *****
21590 047742 012700 000640 T0640: MOV #0640,R0 ;LOAD R0 WITH TEST NO.
21591 ;MICROPROGRAMMING / LOGIC INFORMATION
21592
21593 ;ROM SEQ: [151,300,307,311,312,306,313,016] FC 1,5
21594
21595 ;ACT BUTS: 37[004]100,151 / 15[151]306,307 / 16[306]016,016
21596
21597 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0640
21598
21599 ;CODES: N:C = 1111 (NO CHANGE)
21600
21601 ;SYNC: B05J2 (-) T = 3 USEC
21602
21603 ;KEY SIG: K3-3 DM=1L / K3-5 JSR H / K3-5 JMP+JSR H
21604
21605 047746 013701 047762 MOV @#I0640,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21606 047752 010605 MOV SP,R5 ;SAVE THE SP
21607 047754 012702 047770 R0640: MOV #A0640,R2 ;DEST ADDR = A0640
21608 047760 000277 SCC ;N:C = 1111
21609
21610 047762 004412 I0640: JSR R4,(R2) ;TEST THE JSR - GO TO A0640
21611
21612 047764 104005 E10640: ERROR5 ;JSR FAILED TO LOAD THE PC
21613 047766 047754 R0640 ;ERROR LOOP RETURN
21614
21615 047770 100003 A0640: BPL E20640 ;N:C = 1111 ?
21616 047772 001002 BNE E20640
21617 047774 102001 BVC E20640
21618 047776 103402 BCS B0640
21619 050000 104005 E20640: ERROR5 ;JSR ALTERED FLAGS
21620 050002 047754 R0640 ;ERROR LOOP RETURN
21621
21622 050004 010506 B0640: MOV R5,SP ;RESET SP IN CASE OF ERROR
21623 050006 000004 00640: SCOPE ;CALL SCOPE LOOP UTILITY
21624

```

```

21625 ; *****
21626 ; .SBTTL T0641 JSR MODE 2 TEST
21627 ; *****
21628
21629 ;MICROPROGRAMMING / LOGIC INFORMATION
21630
21631 ;ROM SEQ: [152,235,300,307,310,311,312,306,313,016] FC 1,5
21632
21633 ;ACT BUTS: 37[004]100,152 / 15[235]306,307 / 16[306]016,016
21634
21635 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0641
21636
21637 ;CODES: N / A
21638
21639 ;SYNC: B05J2 (-) T = 3.25 USEC
21640
21641 ;KEY SIG: K3-3 DM=2L / K3-5 JSR H / K3-5 JMP+JSR H / K5-5 BC01 H
21642
21643 050010 012700 000641 T0641: MOV #0641,R0 ;LOAD R0 WITH TEST NO.
21644 050014 013701 050032 MOV @#I0641,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21645 050020 010605 MOV SP,R5 ;SAVE THE SP
21646 050022 010506 R0641: MOV R5,SP ;RESET SP FOR ERROR LOOPS
21647 050024 012702 050040 MOV #A0641,R2 ;DEST ADDR = A0641
21648 050030 000257 CCC ;SCOPE SYNC
21649
21650 050032 004422 I0641: JSR R4,(R2)+ ;TEST THE JSR - GO TO A0641
21651
21652 050034 104005 E10641: ERROR5 ;JSR FAILED TO LOAD THE PC
21653 050036 050022 R0641 ;ERROR LOOP RETURN
21654
21655 050040 005726 A0641: TST (SP)+ ;RESET SP
21656 050042 020605 CMP SP,R5 ;DID JSR PUSH STACK ?
21657 050044 001404 BEQ 00641 ;BR IF YES
21658
21659 050046 005746 E20641: TST -(SP) ;RESET SP TO ERROR VALUE
21660 050050 104005 ERROR5 ;JSR FAILED TO PUSH SP
21661 050052 050022 R0641 ;ERROR LOOP RETURN
21662
21663 050054 010506 MOV R5,SP ;RESTORE SP JUST IN CASE
21664
21665 050056 000004 00641: SCOPE ;CALL SCOPE LOOP UTILITY

```

```

21666 ; *****
21667 ; .SBTTL T0642 JSR MODE 3 TEST
21668 ; *****
21669
21670 ;MICROPROGRAMMING / LOGIC INFORMATION
21671
21672 ;ROM SEQ: [153,303,307,310,311,312,306,313,016] FC 1,5
21673
21674 ;ACT BUTS: 37[004]100,153 / 15[153]306,307 / 16[313]016,016
21675
21676 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0642
21677
21678 ;CODES: N / A
21679
21680 ;SYNC: B05J2 (-) T = 3.5 USEC
21681
21682 ;KEY SIG: K3-3 DM=3L / K3-5 JSR H / K3-5 JMP+JSR H / K5-5 BC01 H
21683
21684 050060 012700 000642 T0642: MOV #0642,R0 ;LOAD R0 WITH TEST NO.
21685 050064 013701 050102 MOV @#I0642,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21686 050070 010605 MOV SP,R5 ;SAVE THE SP
21687 050072 010506 R0642: MOV R5,SP ;RESET SP FOR ERROR LOOPS
21688 050074 012702 050126 MOV #JSR3,R2 ;DEST ADDR = [JSR3]
21689 050100 000257 CCC ;SCOPE SYNC
21690
21691 050102 004432 I0642: JSR R4,@(R2)+ ;TEST THE JSR - GO TO A0642 VIA JSR3
21692
21693 050104 104005 E10642: ERROR5 ;JSR FAILED TO LOAD THE PC
21694 050106 050072 R0642 ;ERROR LOOP RETURN
21695
21696 050110 005726 A0642: TST (SP)+ ;RESET SP
21697 050112 020605 CMP SP,R5 ;DID JSR PUSH STACK ?
21698 050114 001410 BEQ 00642 ;BR IF YES
21699
21700 050116 005746 E20642: TST -(SP) ;RESET SP TO ERROR VALUE
21701 050120 104005 ERROR5 ;JSR FAILED
21702 050122 050072 R0642 ;ERROR LOOP RETURN
21703 050124 000403 BR B0642 ;GO EXIT
21704
21705 050126 050110 JSR3: A0642 ;CONTAINS JUMP ADDR
21706 050130 104005 E30642: ERROR5 ;JSR EXECUTED LIKE A MODE 1 OR 2
21707 050132 050072 R0642 ;ERROR LOOP RETURN
21708
21709 050134 010506 B0642: MOV R5,SP ;RESTORE SP JUST IN CASE
21710
21711 050136 000004 00642: SCOPE ;CALL SCOPE LOOP UTILITY

```

```

21712 ; *****
21713 ; .SBTTL T0643 JSR MODE 4 TEST
21714 ; *****
21715
21716 ;MICROPROGRAMMING / LOGIC INFORMATION
21717
21718 ;ROM SEQ: [154,300,307,311,312,306,313,016] FC 1,5
21719
21720 ;ACT BUTS: 37[004]100,154 / 15[154]306,307 / 16[306]016,016
21721
21722 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0643
21723
21724 ;CODES: N / A
21725
21726 ;SYNC: B05J2 (-) T = 3.1 USEC
21727
21728 ;KEY SIG: K3-3 DM=4L / K3-5 JSR H / K3-5 JMP+JSR H / K5-5 BC01 H
21729
21730 050140 012700 000643 T0643: MOV #0643,R0 ;LOAD R0 WITH TEST NO.
21731 050144 013701 050162 MOV @#I0643,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21732 050150 010605 MOV SP,R5 ;SAVE THE SP
21733 050152 010506 R0643: MOV R5,SP ;RESET SP FOR ERROR LOOPS
21734 050154 012702 050172 MOV #E20643,R2 ;DEST ADDR = A0643+2
21735 050160 000257 CCC ;SCOPE SYNC
21736
21737 050162 004442 I0643: JSR R4,-(R2) ;TEST THE JSR - GO TO A0643
21738
21739 050164 104005 E10643: ERROR5 ;JSR FAILED TO LOAD THE PC
21740 050166 050152 R0643 ;ERROR LOOP RETURN
21741
21742 050170 000402 A0643: BR B0643 ;JUMPED OK - GO CHECK SP
21743 050172 104005 E20643: ERROR5 ;JSR FAILED TO DECREMENT DEST REG
21744 050174 050152 R0643 ;ERROR LOOP RETURN
21745
21746 050176 005726 B0643: TST (SP)+ ;RESET SP
21747 050200 020605 CMP SP,R5 ;DID JSR PUSH STACK ?
21748 050202 001404 BEQ 00643 ;BR IF YES
21749
21750 050204 005746 E30643: TST -(SP) ;RESET SP TO ERROR VALUE
21751 050206 104005 ERROR5 ;JSR FAILED TO PUSH SP
21752 050210 050152 R0643 ;ERROR LOOP RETURN
21753
21754 050212 010506 C0643: MOV R5,SP ;RESTORE SP JUST IN CASE
21755
21756 050214 000004 O0643: SCOPE ;CALL SCOPE LOOP UTILITY

```

```

21757 ; *****
21758 ; .SBTTL T0644 JSR MODE 5 TEST
21759 ; *****
21760
21761 ;MICROPROGRAMMING / LOGIC INFORMATION
21762
21763 ;ROM SEQ: [155,303,307,310,311,312,306,313,016] FC 1,5
21764
21765 ;ACT BUTS: 37[004]100,155 / 15[155]306,307 / 16[306]016,016
21766
21767 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0644
21768
21769 ;CODES: N / A
21770
21771 ;SYNC: B05J2 (-) T = 3.5 USEC
21772
21773 ;KEY SIG: K3-3 DM=5L / K3-5 JSR H / K3-5 JMP+JSR H / K5-5 BC01 H
21774
21775 050216 012700 000644 T0644: MOV #0644,R0 ;LOAD R0 WITH TEST NO.
21776 050222 013701 050252 MOV @#I0644,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21777 050226 032737 010000 066642 BIT #10000,@#BPTLOC ;BREAKPOINT HALT SET ??
21778 050234 001401 BEQ .+4 ;BR IF NOT
21779 050236 000000 HALT ;BREAK-DEPRESS CONTINUE TO RESTART
21780 050240 010605 MOV SP,R5 ;SAVE THE SP
21781 050242 010506 R0644: MOV R5,SP ;RESET SP FOR ERROR LOOPS
21782 050244 012702 050300 MOV #E30644,R2 ;DEST ADDR = [E30644 - 2]
21783 050250 000257 CCC ;SCOPE SYNC
21784
21785 050252 004452 I0644: JSR R4,@-(R2) ;TEST THE JSR - GO TO A0644
21786
21787 050254 104005 E10644: ERRORS R0644 ;JSR FAILED TO LOAD THE PC
21788 050256 050242 ;ERROR LOOP RETURN
21789
21790 050260 005726 A0644: TST (SP)+ ;RESET SP
21791 050262 020605 CMP SP,R5 ;DID JSR PUSH STACK ?
21792 050264 001410 BEQ 00644 ;BR IF YES
21793
21794 050266 005746 E20644: TST -(SP) ;RESET SP TO ERROR VALUE
21795 050270 104005 ERRORS R0644 ;JSR FAILED TO PUSH SP
21796 050272 050242 R0644 ;ERROR LOOP RETURN
21797 050274 000403 BR B0644 ;GO EXIT
21798
21799 050276 050260 E30644: A0644 ;CONTAINS JUMP ADDRESS
21800 050300 104005 ERRORS R0644 ;JSR EXECUTED LIKE A MODE 1 OR 2
21801 050302 050242 R0644 ;ERROR LOOP RETURN
21802
21803 050304 010506 B0644: MOV R5,SP ;RESTORE SP JUST IN CASE
21804
21805 050306 000004 00644: SCOPE ;CALL SCOPE LOOP UTILITY

```

```

21806 ; *****
21807 ; .SBTTL T0645 JSR MODE 6 TEST
21808 ; *****
21809
21810 ;MICROPROGRAMMING / LOGIC INFORMATION
21811
21812 ;ROM SEQ: [156,304,305,300,307,310,311,312,306,313,016] FC 1,5
21813
21814 ;ACT BUTS: 37[004]100,156 / 15[305]306,307 / 16[306]016,016
21815
21816 ;EXEC: [306]ALUC=LLLLL :[313] D = #A0645
21817
21818 ;CODES: N / A
21819
21820 ;SYNC: B05J2 (-) T = 3.5 USEC
21821
21822 ;KEY SIG: K3-3 DM=6L / K3-5 JSR H / K3-5 JMP+JSR H
21823
21824 050310 012700 000645 T0645: MOV #0645,R0 ;LOAD R0 WITH TEST NO.
21825 050314 013701 050332 MOV @#I0645,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21826 050320 010605 MOV SP,R5 ;SAVE THE SP
21827 050322 010506 R0645: MOV R5,SP ;RESET SP FOR ERROR LOOPS
21828 050324 012702 050336 MOV #E10645,R2 ;[R2] = BASE DEST ADDR
21829 050330 000257 CCC ;SCOPE SYNC
21830
21831 050332 004462 000004 I0645: JSR R4,A0645-E10645(R2) ;TEST THE JSR - GO TO A0645
21832
21833 050336 104005 E10645: ERRORS ;JSR FAILED TO LOAD THE PC OR INDEX FAILED
21834 050340 050322 R0645 ;ERROR LOOP RETURN
21835
21836 050342 005726 A0645: TST (SP)+ ;RESET SP
21837 050344 020605 CMP SP,R5 ;DID JSR PUSH STACK ?
21838 050346 001404 BEQ 00645 ;BR IF YES
21839
21840
21841 050350 005746 E20645: TST -(SP) ;RESET SP TO ERROR VALUE
21842 050352 104005 ERRORS ;JSR FAILED TO PUSH STACK
21843 050354 050322 R0645 ;ERROR LOOP RETURN
21844 050356 010506 MOV R5,SP ;RESET SP JUST IN CASE
21845
21846 050360 000004 00645: SCOPE ;CALL SCOPE LOOP UTILITY

```

21847
21848
21849
21850
21851
21852
21853
21854
21855
21856
21857
21858
21859
21860
21861
21862
21863
21864
21865
21866
21867
21868
21869
21870
21871
21872
21873
21874
21875
21876
21877
21878
21879
21880
21881
21882
21883
21884
21885
21886
21887
21888
21889
21890
21891
21892
21893
21894
21895

050362 012700 000646
050366 013701 050404
050372 010605
050374 010506
050376 012702 050410
050402 000257

050404 004472 000022
050410 104005
050412 050374
050414 005726
050416 020605
050420 001410
050422 005746
050424 104005
050426 050374
050430 000403
050432 050414
050434 104005
050436 050374
050440 010506
050442 000004

; *****
; .SBTTL T0646 JSR MODE 7 TEST
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [157,301,302,303,307,310,311,312,306,313,016] FC 1,5
;ACT BUTS: 37[004]100,157 / 15[302]306,307 / 16[306]016,016
;EXEC: [306]ALUC=LLLLL :[313] D = #A0646
;CODES: N / A
;SYNC: B05J2 (-) T = 4.1 USEC
;KEY SIG: K3-3 DM=7L / K3-5 JSR H / K3-5 JMP+JSR H

T0646: MOV #0646,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0646,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R0646: MOV R5,SP ;RESET SP FOR ERROR LOOPS
MOV #E10646,R2 ;BASE DEST ADDR = E10646
CCC ;SCOPE SYNC

I0646: JSR R4,@JSR7-E10646(R2);TEST THE JSR - GO TO A0646 VIA JSR7

E10646: ERROR5 ;JSR FAILED TO LOAD THE PC
R0646 ;OR INDEX FAILED
;ERROR LOOP RETURN

A0646: TST (SP)+ ;RESET SP
CMP SP,R5 ;DID JSR PUSH STACK ?
BEQ 00646 ;BR IF YES

E20646: TST -(SP) ;RESET SP TO ERROR VALUE
ERROR5 ;JSR FAILED TO PUSH STACK
R0646 ;ERROR LOOP RETURN
BR B0646 ;SKIP TO EXIT

JSR7: A0646 ;CONTAINS JUMP ADDR
E30646: ERROR5 ;JSR WORKED LIKE A MODE 1 OR 2
R0646 ;ERROR LOOP RETURN

B0646: MOV R5,SP ;RESTORE SP JUST IN CASE

00646: SCOPE ;CALL SCOPE LOOP UTILITY

21896
21897
21898
21899
21900
21901
21902
21903
21904
21905
21906
21907
21908
21909
21910
21911
21912
21913
21914
21915
21916
21917
21918
21919
21920
21921
21922
21923
21924
21925
21926
21927
21928
21929

050444 012700 000647
050450 013701 050472
050454 012702 000001
050462 104006
050464 050454
050466 000402
050470 000257
050472 077205
050474 000004

; *****
; .SBTTL T0647 SOB TEST, [R] = 1, NO BRANCH
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [130,342,343,345,347,016] FC 1,7
;ACT BUTS: 37[004]100,130 / 12[342]344,345 / 16[345]016,016
;EXEC: [130]ALUC=LHHHH :[343] D = 000000
;CODES: N / A
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K3-6 SOB L / K1-7 D(15:00)=0 H

T0647: MOV #0647,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0647,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0647: MOV #1,R2 ;SET SOB COUNTER = 1
BR I0647-2 ;GO DO THE SOB
E0647: ERROR6 ;SOB SHOULDN'T HAVE BRANCHED HERE
R0647 ;ERROR LOOP RETURN
BR 00647 ;GO TO SCOPE CALL
I0647: CCC ;SYNC INSTR.
SOB R2,E0647 ;TEST THE SOB
00647: SCOPE ;CALL SCOPE LOOP UTILITY

```

21930 ; *****
21931 ; .SBTTL T0650 SOB TEST, [R] = 5, BRANCH 4 TIMES
21932 ; *****
21933 ;MICROPROGRAMMING / LOGIC INFORMATION
21934 ;ROM SEQ: [130,342,343,344,346,016] FC 1,7
21935 ;ACT BUTS: 37[004]100,130 / 12[342]344,344 / 16[344]016,016
21936 ;EXEC: [130]ALUC=LHHHH :[343] D = 000004 (1ST TIME)
21937 ; : [346]ALUC=LLHHL :[016] D = #A0650
21938 ;CODES: N / A
21939 ;SYNC: B05J2 (-) T = 2.36 USEC
21940 ;KEY SIG: K3-6 SOB L / K2-7 DAD3 (1) H / K3-8 CIN00 L
21941
21942 21948 050476 012700 000650 T0650: MOV #0650,R0 ;LOAD R0 WITH TEST NO.
21943 21949 050502 013701 050526 MOV @#I0650,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
21944
21945 21951 050506 012702 000005 R0650: MOV #5,R2 ;SET SOB COUNTER = 5
21946 21952 050512 012705 177773 MOV #-5,R5 ;SET UP R5 TO COUNT 5 BRANCHES
21947 21953 050516 000402 BR I0650-2 ;GO DO THE SOB
21948
21949 21955 050520 005205 A0650: INC R5 ;COUNT ONE BRANCH
21950 21956 050522 001412 BEQ E20650 ;BR IF TOO MANY LOOPS BY SOB
21951
21952 21958 050524 000257 I0650: CCC ;SCOPE SYNC
21953 21959 050526 077204 SOB R2,A0650 ;TEST THE SOB
21954 21960 050530 000402 BR SOB2 ;SKIP NEXT TWO WORDS
21955
21956 21962 050532 000474 SOB1: BR SOB3 ;USED BY LAST SOB TEST TO TEST MAX OFFSET
21957 21963 050534 000240 NOP
21958
21959 21965 050536 005702 SOB2: TST R2 ;R2 SHOULD CONTAIN 0
21960 21966 050540 001405 BEQ 00650 ;BR IF IT DOES
21961
21962 21968 050542 104006 E10650: ERROR6 ;SOB COUNTER NOT ZERO
21963 21969 050544 050506 R0650 ;ERROR LOOP RETURN
21964 21970 050546 000402 BR 00650 ;GO TO SCOPE CALL
21965 21971 050550 104006 E20650: ERROR6 ;SOB MADE TOO MANY BRANCHES
21966 21972 050552 050506 R0650 ;ERROR LOOP RETURN
21967 21973 050554 000004 00650: SCOPE ;CALL SCOPE LOOP UTILITY
21968 21974

```

21975
21976
21977
21978
21979
21980
21981
21982
21983
21984
21985
21986
21987
21988
21989
21990
21991
21992
21993
21994
21995
21996
21997
21998
21999
22000
22001
22002
22003
22004
22005
22006
22007
22008
22009
22010
22011

050556 012700 000651
050562 013701 050574
050566 012702 000001
050572 000277
050574 077202
050576 103003
050600 102002
050602 001001
050604 100402
050606 104006
050610 050566
050612 000004

; *****
; .SBTTL T0651 SOB TEST, [R] = 1, FLAGS = 1111
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [130,342,343,345,347,016] FC 1,7
;ACT BUTS: 37[004]100,130 / 12[342]344,345 / 16[345]016,016
;EXEC: [130]ALUC=LHHHH :[343] D = 000000
;CODES: N:C = 1111 (NO CHANGE)
;SYNC: B05J2 (-) T = 2 USEC
;KEY SIG: K3-6 SOB L / K1-7 D(15:00)=0 H

T0651: MOV #0651,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0651,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
R0651: MOV #1,R2 ;SET SOB COUNTER = 1
SCC ;MAKE N:C = 1111
I0651: SOB R2,I0651-2 ;TEST THE SOB
BCC E0651 ;BR IF C = 0
BVC E0651 ;BR IF V = 0
BNE E0651 ;BR IF Z = 0
BMI 00651 ;BR IF N = 1
E0651: ERROR6 ;SOB ALTERED CODES - CLEARED ONE
R0651 ;ERROR LOOP RETURN
00651: SCOPE ;CALL SCOPE LOOP UTILITY

```

22012 ; *****
22013 ; .SBTTL T0652 SOB TEST, [R] = 1, FLAGS = 0000
22014 ; *****
22015
22016 ;MICROPROGRAMMING / LOGIC INFORMATION
22017
22018 ;ROM SEQ: [130,342,343,345,347,016] FC 1,7
22019
22020 ;ACT BUTS: 37[004]100,130 / 12[342]344,345 / 16[345]016,016
22021
22022 ;EXEC: [130]ALUC=LHHHH :[343] D = 000000
22023
22024 ;CODES: N:C = 0000 (NO CHANGE)
22025
22026 ;SYNC: B05J2 (-) T = 2 USEC
22027
22028 ;KEY SIG: K3-6 SOB L / K1-7 D(15:00)=0 H
22029
22030 050614 012700 000652 T0652: MOV #0652,R0 ;LOAD R0 WITH TEST NO.
22031 050620 013701 050632 MOV @#I0652,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22032
22033 050624 012702 000001 R0652: MOV #1,R2 ;SET SOB COUNTER = 1
22034 050630 000257 CCC ;MAKE N:C = 0000
22035
22036 050632 077202 I0652: SOB R2,I0652-2 ;TEST THE SOB
22037
22038 050634 103403 BCS E0652 ;BR IF C = 1
22039 050636 102402 BVS E0652 ;BR IF V = 1
22040 050640 001401 BEQ E0652 ;BR IF Z = 1
22041
22042 050642 100002 BPL 00652 ;BR IF N = 0
22043
22044 050644 104006 E0652: ERROR6 ;SOB ALTERED CODES - SET ONE
22045 050646 050624 R0652 ;ERROR LOOP RETURN
22046
22047 050650 000004 00652: SCOPE ;CALL SCOPE LOOP UTILITY
22048
22049

```

```

22050 ; *****
22051 ; .SBTTL T0653 SOB TEST, [R] = 5, FLAGS = 1111
22052 ; *****
22053 ;MICROPROGRAMMING / LOGIC INFORMATION
22054 ;ROM SEQ: [130,342,343,344,346,016] FC 1,7
22055 ;ACT BUTS: 37[004]100,130 / 12[342]344,344 / 16[344]016,016
22056 ;EXEC: [130]ALUC=LHHHH :[343] D = 000004 (1ST TIME9
22057 ; [346]ALUC=LLHHL :[016] D = #I0653
22058 ;CODES: N:C = 1111 (NO CHANGE)
22059 ;SYNC: B05J2 (-) T = 2.36 USEC
22060 ;KEY SIG: K3-6 SOB L / K2-7 DAD3 (1) H / K3-8 CIN00 L
22061
22062 050652 012700 000653 T0653: MOV #0653,R0 ;LOAD R0 WITH TEST NO.
22063 050656 013701 050670 MOV @#I0653,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22064
22065 050662 012702 000005 R0653: MOV #5,R2 ;SET SOB COUNTER = 5
22066 050666 000277 SCC ;MAKE N:C = 1111
22067
22068 050670 077201 I0653: SOB R2,I0653 ;TEST THE SOB
22069
22070 BCC E0653 ;BR IF C = 0
22071 BVC E0653 ;BR IF V = 0
22072 BNE E0653 ;BR IF Z = 0
22073 BMI 00653 ;BR IF N = 1
22074
22075 050702 104006 E0653: ERROR6 ;SOB ALTERED CODES - CLEARED ONE
22076 050704 050662 R0653 ;ERROR LOOP RETURN
22077
22078 050706 000004 00653: SCOPE ;CALL SCOPE LOOP UTILITY
22079
22080
22081
22082
22083
22084
22085
22086

```

22087
22088
22089
22090
22091
22092
22093
22094
22095
22096
22097
22098
22099
22100
22101
22102
22103
22104
22105
22106
22107
22108
22109
22110
22111
22112
22113
22114
22115
22116
22117
22118
22119
22120
22121
22122
22123

050710 012700 000654
050714 013701 050726
050720 012702 000005
050724 000257
050726 077277
050730 103403
050732 102402
050734 001401
050736 100002
050740 104006
050742 050720
050744 000004

```
; *****  
; .SBTTL T0654 SOB TEST, [R] = 5, FLAGS = 0000  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [130,342,343,344,346,016] FC 1,7  
;ACT BUTS: 37[004]100,130 / 12[342]344,344 / 16[344]016,016  
;EXEC: [130]ALUC=LHHH :[343] D = 000004 (1ST TIME)  
; [346]ALUC=LLHL :[016] D = #I0654  
;CODES: N:C = 0000 (NO CHANGE)  
;SYNC: B05J2 (-) T = 2.36 USEC  
;KEY SIG: K3-6 SOB L / K2-7 DAD3 (1) H / K3-8 CIN00 L  
T0654: MOV #0654,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0654,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
R0654: MOV #5,R2 ;SET SOB COUNTER = 5  
SOB3: CCC ;MAKE N:C = 0000  
I0654: SOB R2,SOB1 ;TEST THE SOB  
BCS E0654 ;BR IF C = 1  
BVS E0654 ;BR IF V = 1  
BEQ E0654 ;BR IF Z = 1  
BPL 00654 ;BR IF N = 0  
E0654: ERROR6 ;SOB ALTERED CODES - SET ONE  
R0654 ;ERROR LOOP RETURN  
00654: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

22124 ; *****
22125 ; .SBTTL T0655 RTS TEST - <N:C> = 1111
22126 ; *****
22127
22128 ;MICROPROGRAMMING / LOGIC INFORMATION
22129
22130 ;ROM SEQ: [124,323,324,325,016] FC 1,6
22131
22132 ;ACT BUTS: 37[004]100,124 / 16[324]016,016
22133
22134 ;EXEC: [323] D = #A0655 / [325] D = 177777
22135
22136 ;CODES: N:C = 1111 (NO CHANGE)
22137
22138 ;SYNC: B05J2 (-) T = 2.5 USEC
22139
22140 ;KEY SIG: K3-6 RTS L / K5-5 BC01 H
22141
22142 050746 012700 000655 T0655: MOV #0655,R0 ;LOAD R0 WITH TEST NO.
22143 050752 013701 051000 MOV @#I0655,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22144 050756 010605 MOV SP,R5 ;SAVE THE SP
22145 050760 012704 177777 R0655: MOV #-1,R4 ;R3 SHOULD GET 177777
22146 050764 010506 MOV R5,SP ;RESET SP FOR ERROR LOOP
22147 050766 012703 051010 MOV #A0655,R3 ;RTS SHOULD LOAD PC FROM [R3]
22148 050772 012746 177777 MOV #-1,-(SP) ;RTS SHOULD LOAD R3 WITH 177777
22149 050776 000277 SCC ;N:C = 1111
22150
22151 051000 000203 I0655: RTS R3 ;TEST THE RTS - GO TO A0655
22152
22153 051002 104005 E10655: ERROR5 ;RTS FAILED TO LOAD THE PC
22154 051004 050760 R0655 ;ERROR LOOP RETURN ADDRESS
22155 051006 000420 BR D0655 ;GO TO EXIT - SCHOOLS OUT
22156
22157 051010 100003 A0655: BPL E20655 ;N:C = 1111 ?
22158 051012 001002 BNE E20655
22159 051014 102001 BVC E20655
22160 051016 103402 BCS B0655
22161
22162 051020 104005 E20655: ERROR5 ;RTS ALTERED CODES - CLEARED ONE
22163 051022 050760 R0655 ;ERROR LOOP RETURN
22164
22165 051024 020403 B0655: CMP R4,R3 ;DID R3 GET LOADED FROM STACK ?
22166 051026 001402 BEQ C0655 ;BR IF YES
22167
22168 051030 104000 E30655: ERROR ;RTS FAILED TO LOAD REG
22169 051032 050760 R0655 ;ERROR LOOP RETURN
22170
22171 051034 020506 C0655: CMP R5,SP ;DID RTS POP THE STACK POINTER ?
22172 051036 001405 BEQ C0655 ;BR IF YES
22173
22174 051040 010504 MOV R5,R4 ;[R4] = S / B SP
22175 051042 010603 MOV SP,R3 ;[R3] = WAS SP
22176 051044 104000 E40655: ERROR ;RTS FAILED TO POP SP
22177 051046 050760 R0655 ;ERROR LOOP RETURN
22178
22179 051050 010506 D0655: MOV R5,SP ;FIX THE SP

```

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 549
CBQEAC.P11 03-JUL-80 08:05 T0655 RTS TEST - <N:C> = 1111

SEQ 0549

22180 051052 000004
22181

00655: SCOPE

;CALL THE SCOPE LOOP UTILITY

```

22182 ; *****
22183 ; .SBTTL T0656 RTS TEST - <N:C> = 0000
22184 ; *****
22185
22186 ;MICROPROGRAMMING / LOGIC INFORMATION
22187
22188 ;ROM SEQ: [124,323,324,325,016] FC 1,6
22189
22190 ;ACT BUTS: 37[004]100,124 / 16[324]016,016
22191
22192 ;EXEC: [323] D = #A0656 / [325] D = 177777
22193
22194 ;CODES: N:C = 0000 (NO CHANGE)
22195
22196 ;SYNC: B05J2 (-) T = 2.5 USEC
22197
22198 ;KEY SIG: K3-6 RTS L / K5-5 BC01 H
22199
22200 051054 012700 000656 T0656: MOV #0656,R0 ;LOAD R0 WITH TEST NO.
22201 051060 013701 051106 MOV @#I0656,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22202 051064 010605 MOV SP,R5 ;SAVE THE SP
22203 051066 012704 177777 R0656: MOV #-1,R4 ;R3 SHOULD GET 177777
22204 051072 010506 MOV R5,SP ;RESET SP FOR ERROR LOOP
22205 051074 012703 051116 MOV #A0656,R3 ;RTS SHOULD LOAD PC FROM [R3]
22206 051100 012746 177777 MOV #-1,-(SP) ;RTS SHOULD LOAD R3 WITH 177777
22207 051104 000257 CCC ;N:C = 0000
22208
22209 051106 000203 I0656: RTS R3 ;TEST THE RTS - GO TO A0656
22210
22211 051110 104005 E10656: ERROR5 ;RTS FAILED TO LOAD THE PC
22212 051112 051066 R0656 ;ERROR LOOP RETURN ADDRESS
22213 051114 000420 BR D0656 ;GO TO EXIT - SCHOOLS OUT
22214
22215 051116 100403 A0656: BMI E20656 ;N:C = 0000 ?
22216 051120 001402 BEQ E20656
22217 051122 102401 BVS E20656
22218 051124 103002 BCC B0656
22219
22220 051126 104005 E20656: ERROR5 ;RTS ALTERED CODES - CLEARED ONE
22221 051130 051066 R0656 ;ERROR LOOP RETURN
22222
22223 051132 020403 B0656: CMP R4,R3 ;DID R3 GET LOADED FROM STACK ?
22224 051134 001402 BEQ C0656 ;BR IF YES
22225
22226 051136 104000 E30656: ERROR ;RTS FAILED TO LOAD REG
22227 051140 051066 R0656 ;ERROR LOOP RETURN
22228
22229 051142 020506 C0656: CMP R5,SP ;DID RTS POP THE STACK POINTER ?
22230 051144 001405 BEQ O0656 ;BR IF YES
22231
22232 051146 010504 MOV R5,R4 ;[R4] = S / B SP
22233 051150 010603 MOV SP,R3 ;[R3] = WAS SP
22234 051152 104000 E40656: ERROR ;RTS FAILED TO POP SP
22235 051154 051066 R0656 ;ERROR LOOP RETURN
22236
22237 051156 010506 D0656: MOV R5,SP ;FIX THE SP

```

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 551
CBQEAC.P11 03-JUL-80 08:05 T0656 RTS TEST - <N:C> = 0000

SEQ 0551

22238 051160 000004
22239

00656: SCOPE

;CALL THE SCOPE LOOP UTILITY

22240
22241
22242
22243
22244
22245
22246
22247
22248
22249
22250
22251
22252
22253
22254
22255
22256
22257
22258
22259
22260
22261
22262
22263
22264
22265
22266
22267
22268
22269
22270
22271
22272
22273
22274
22275
22276
22277
22278
22279
22280
22281
22282
22283
22284
22285
22286
22287
22288
22289
22290
22291
22292
22293

051162 012700 000657
051166 013701 051220
051172 010605
051174 010506
051176 012704 000340
051202 012746 000340
051206 012746 051230
051212 005037 177776
051216 000277

051220 000006

051222 104005
051224 051174
051226 000414

051230 013703 177776
051234 020403
051236 001402

051240 104000

051242 051174

051244 020506
051246 001405

051250 010504
051252 010603
051254 104000
051256 051174

051260 010506
051262 000004

```
; *****  
.SBTTL T0657 RTT TEST - <N:C> = 1111  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [101,320,321,322,017,015,013] FC 1,6,10  
;ACT BUTS: 37[004]100,101 / 26[017]010,013  
;EXEC: [320] D = #A0657 / [322] D = 340  
;CODES: [322] SPS=7 / N:C = 0000  
;SYNC: B05J2 (-) T = 3 USEC  
;KEY SIG: K3-6 RTT H / K3-6 RTI+RTT L / K5-5 BC01 H  
  
T0657: MOV #0657,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0657,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV SP,R5 ;SAVE THE SP  
R0657: 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  
MOV #A0657,-(SP) ;NEW PC S / B = A0657  
CLR @#PSW ;CLEAR THE PSW  
SCC ;N:C = 1111  
  
I0657: RTT ;TEST THE RTT - GO TO A0657  
  
E10657: ERROR5 ;RTT FAILED TO LOAD THE PC  
R0657 ;ERROR LOOP RETURN ADDRESS  
BR C0657 ;GO TO EXIT - SCHOOL'S OUT  
  
A0657: MOV @#PSW,R3 ;SAVE THE PSW  
CMP R4,R3 ;WAS PSW = 340 ?  
BEQ B0657 ;BR IF IT WAS  
  
E20657: ERROR ;RTT FAILED TO LOAD PSW PROPERLY  
;[R3] = WAS PSW  
R0657 ;[R4] = S / B PSW  
;ERROR LOOP RETURN ADDRESS  
  
B0657: CMP R5,SP ;DID RTT UPDATE THE SP ?  
BEQ 00657 ;BR IF YES  
  
E30657: MOV R5,R4 ;[R4] = S / B SP  
MOV SP,R3 ;[R3] = WAS SP  
ERROR ;RTT FAILED TO UPDATE SP  
R0657 ;ERROR LOOP RETURN ADDRESS  
  
C0657: MOV R5,SP ;FIX THE SP  
00657: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

```

22294 ; *****
22295 ; .SBTTL T0660 RTT TEST - <N:C> = 0000
22296 ; *****
22297
22298 ;MICROPROGRAMMING / LOGIC INFORMATION
22299
22300 ;ROM SEQ: [101,320,321,322,017,015,013] FC 1,6,10
22301
22302 ;ACT BUTS: 37[004]100,101 / 26[017]010,013
22303
22304 ;EXEC: [320] D = #A0660 / [322] D = 000017
22305
22306 ;CODES: [322] SPS=7 / N:C = 1111
22307
22308 ;SYNC: B05J2 (-) T = 3 USEC
22309
22310 ;KEY SIG: K3-6 RTI+RTT L / K3-6 RTT H / K5-5 BC01 H
22311
22312 051264 012700 000660 T0660: MOV #0660,R0 ;LOAD R0 WITH TEST NO.
22313 051270 013701 051324 MOV @#I0660,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22314 051274 010605 MOV SP,R5 ;SAVE THE SP
22315 051276 010506 R0660: MOV R5,SP ;RESET SP FOR ERROR LOOP
22316 051300 012704 000017 MOV #017,R4 ;[R4] = S / B PSW AT HTIS POINT
22317 051304 012746 000017 MOV #017,-(SP) ;NEW PSW S / B = 017
22318 051310 012746 051334 MOV #A0660,-(SP) ;NEW PC S / B = A0660
22319 051314 012737 000340 177776 MOV #340,@#PSW ;MAKE [PSW] = 340
22320 051322 000257 CCC ;N:C = 0000
22321
22322 051324 000006 I0660: RTT ;TEST THE RTT - GO TO A0660
22323
22324 051326 104005 E10660: ERROR5 ;RTT FAILED TO LOAD THE PC
22325 051330 051276 R0660 ;ERROR LOOP RETURN ADDRESS
22326 051332 000414 BR C0660 ;GO TO EXIT - SCHOOL'S OUT
22327
22328 051334 013703 177776 A0660: MOV @#PSW,R3 ;SAVE THE PSW
22329 051340 020403 CMP R4,R3 ;WAS PSW = 017 ?
22330 051342 001402 BEQ B0660 ;BR IF IT WAS
22331
22332 051344 104000 E20660: ERROR ;RTT FAILED TO LOAD PSW PROPERLY
22333 ;[R3] = WAS PSW
22334 ;[R4] = S / B PSW
22335 051346 051276 R0660 ;ERROR LOOP RETURN ADDRESS
22336
22337 051350 020506 B0660: CMP R5,SP ;DID RTT UPDATE THE SP ?
22338 051352 001405 BEQ 00660 ;BR IF YES
22339
22340 051354 010504 MOV R5,R4 ;[R4] = S / B SP
22341 051356 010603 MOV SP,R3 ;[R3] = WAS SP
22342 051360 104000 E30660: ERROR ;RTT FAILED TO UPDATE SP
22343 051362 051276 R0660 ;ERROR LOOP RETURN ADDRESS
22344
22345 051364 010506 C0660: MOV R5,SP ;FIX THE SP
22346 051366 000004 00660: SCOPE ;CALL THE SCOPE LOOP UTILITY
22347

```

22348
22349
22350
22351
22352
22353
22354
22355
22356
22357
22358
22359
22360
22361
22362
22363
22364
22365
22366
22367
22368
22369
22370
22371
22372
22373
22374
22375
22376
22377
22378
22379
22380
22381
22382
22383
22384
22385
22386
22387
22388
22389
22390
22391
22392
22393
22394
22395
22396
22397
22398
22399
22400
22401
22402
22403

051370 012700 000661
051374 013701 051420
051400 010604
051402 012703 125252
051406 012705 051444
051412 010337 051432
051416 000257
051420 006404
051422 010406
051424 104005
051426 051406
051430 000435
051432 125252
051434 010406
051436 104005
051440 051406
051442 000430
051444 100403
051446 001402
051450 102401
051452 103005
051454 013702 177776
051460 010406
051462 104006
051464 051406
051466 020627 051434
051472 001405
051474 010602
051476 010406

```
; *****  
; .SBTTL T0661 MARK INSTRUCTION TEST - <N:C>=0000  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [112,353,354,355,356,357,016] FC 1,5  
;ACT BUTS: 37[004]100,112 / 16[356]016,016  
;EXEC: [355] D = #B0661+2 / [356] D = 125252 / [357] D = #A0661  
;CODES: N:C = 0000 (NO CHANGE)  
;SYNC: B05J2 (-) T = 2.6 USEC  
;KEY SIG: K3-6 MARK L / K5-5 BC01 H  
T0661: MOV #0661,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0661,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV SP,R4 ;SAVE SP  
MOV #125252,R3 ;[R5] SHOULD BE 125252  
R0661: MOV #A0661,R5 ;MARK GOES TO A0661 VIA [R5]  
MOV R3,@#B0661 ;INITIALIZE WORD LOADED INTO R5  
CCC ;N:C=0000  
I0661: MARK+4 ;TEST THE MARK  
E10661: MOV R4,SP ;RESET SP  
ERROR5 ;MARK FAILED TO EXECUTE  
R0661 ;ERROR LOOP RETURN ADDRESS  
BR 00661 ;GO TO SCOPE EXIT  
B0661: 125252 ;THIS WORD SHOULD GET LOADED INTO R5  
E20661: MOV R4,SP ;RESET SP  
ERROR5 ;MARK FAILED TO LOAD RC FROM [R5]  
R0661 ;ERROR LOOP RETURN ADDRESS  
BR 00661 ;GO TO SCOPE EXIT  
A0661: BMI D0661 ;N:C=0000?  
BEQ D0661  
BVS D0661  
BCC C0661  
D0661: MOV @#PSW,R2 ;SAVE FLAGS IN R2  
MOV R4,SP ;RESET SP  
E30661: ERROR6 ;MSRK SET A FLAG -[PSW] IN R2  
R0661 ;ERROR LOOP RETURN ADDRESS  
C0661: CMP SP,#B0661+2 ;DID MARK RESET SP?  
BEQ F0661 ;BR IF YES  
MOV SP,R2 ;PUT BAD SP IN R2  
MOV R4,SP ;RESET SP
```



```
22420 ; *****
22421 ; .SBTTL T0662 MARK INSTRUCTION TEST - <N:C>=1111
22422 ; *****
22423 ;MICROPROGRAMMING / LOGIC INFORMATION
22424 ;ROM SEQ: [112,353,354,355,356,357,016] FC 1,5
22425 ;ACT BUTS: 37[004]100,112 / 16[356]016,016
22426 ;EXEC: [355] D = #B0662+2 / [356] D = 125252 / [357] D = #A0662
22427 ;CODES: N:C = 1111 (NO CHANGE)
22428 ;SYNC: B05J2 (-) T = 2.6 USEC
22429 ;KEY SIG: K3-6 MARK L / K5-5 BC01 H
22430
22431 T0662: MOV #0662,R0 ;LOAD R0 WITH TEST NO.
22432 MOV @#I0662,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22433 MOV SP,R4 ;SAVE SP
22434 MOV #125252,R3 ;[R5] SHOULD BE 125252
22435 R0662: MOV #A0662,R5 ;MARK GOES TO A0662 VIA [R5]
22436 MOV R3,@#B0662 ;INITIALIZE WORD LOADED INTO R5
22437 SCC ;N:C=1111
22438
22439 I0662: MARK+4 ;TEST THE MARK
22440
22441 E10662: MOV R4,SP ;RESET SP
22442 ERROR5 ;MARK FAILED TO EXECUTE
22443 R0662 ;ERROR LOOP RETURN ADDRESS
22444 BR 00662 ;GO TO SCOPE EXIT
22445
22446 B0662: 125252 ;THIS WORD SHOULD GET LOADED INTO R5
22447
22448 E20662: MOV R4,SP ;RESET SP
22449 ERROR5 ;MARK FAILED TO LOAD RC FROM [R5]
22450 R0662 ;ERROR LOOP RETURN ADDRESS
22451 BR 00662 ;GO TO SCOPE EXIT
22452
22453 A0662: BPL D0662 ;N:C=1111?
22454 BNE D0662
22455 BVC D0662
22456 BCS C0662
22457
22458 D0662: MOV @#PSW,R2 ;SAVE FLAGS IN R2
22459 MOV R4,SP ;RESET SP
22460 E30662: ERROR6 ;MSRK SET A FLAG -[PSW] IN R2
22461 R0662 ;ERROR LOOP RETURN ADDRESS
22462
22463 C0662: CMP SP,#B0662+2 ;DID MARK RESET SP?
22464 BEQ F0662 ;BR IF YES
22465 MOV SP,R2 ;PUT BAD SP IN R2
22466 MOV R4,SP ;RESET SP
22467
22468 051526 012700 000662
22469 051532 013701 051556
22470 051536 010604
22471 051540 012703 125252
22472 051544 012705 051602
22473 051550 010337 051570
22474 051554 000277
22475 051556 006404
22476 051560 010406
22477 051562 104005
22478 051564 051544
22479 051566 000435
22480 051570 125252
22481 051572 010406
22482 051574 104005
22483 051576 051544
22484 051600 000430
22485 051602 100003
22486 051604 001002
22487 051606 102001
22488 051610 103405
22489 051612 013702 177776
22490 051616 010406
22491 051620 104006
22492 051622 051544
22493 051624 020627 051572
22494 051630 001405
22495 051632 010602
22496 051634 010406
```

22476	051636	104005	E40662:	ERROR5	;MARK FAILED TO RESET SP -[R2]=SP WAS
22477	051640	051544		R0662	;ERROR LOOP RETURN ADDRESS
22478					
22479	051642	000407		BR 00662	;GO TO SCOPE EXIT
22480					
22481	051644	020503	F0662:	CMP R5,R3	;DID MARK RESTORE OLD R5
22482	051646	001404		BEQ G0662	;BR IF YES
22483					
22484	051650	010502		MOV R5,R2	;[R2]=WAS R5
22485	051652	010406		MOV R4,SP	;RESET SP
22486	051654	104007	E50662:	ERROR7	;MARK FAILED TO RESET R5
22487	051656	051544		R0662	;ERROR LOOP RETURN
22488					
22489	051660	010406	G0662:	MOV R4,SP	;RESET SP
22490					
22491	051662	000004	00662:	SCOPE	;CALL THE SCOPE LOOP UTILITY

```

22492 ; *****
22493 ; .SBTTL T0663 BASIC KW11-L RESPONSE TEST
22494 ; *****
22495
22496 051664 012700 000663 T0663: MOV #0663,R0 ;LOAD R0 WITH TEST NO.
22497 051670 013701 051722 MOV @#I0663,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22498 051674 005737 066636 TST @#OPTION ;IS THE KW11 INSTALLED ??
22499 051700 100023 BPL 00663 ;BR IF NOT - SKIP THIS TEST
22500 051702 010605 MOV SP,R5 ;SAVE SP
22501 051704 012702 177546 MOV #LKCSR,R2 ;[R2] = LINE CLOCK ADDRESS
22502 051710 010506 R0663: MOV R5,SP ;RESET SP FOR ERROR LOOP
22503 051712 012737 051726 000004 MOV #A0663,@#4 ;GO TO A0663 IF BUS TIMEOUT
22504 051720 000257 CCC ;SCOPE SYNC
22505
22506 051722 005712 I0663: TST (R2) ;REFERENCE LKCSR ADDR
22507
22508 051724 000405 BR B0663 ;GO TO EXIT
22509
22510 051726 012737 065160 000004 A0663: MOV #BERR,@#4 ;RESTORE TIMEOUT VECTOR
22511 051734 104006 E0663: ERROR6 ;LKCSR FAILED TO RESPOND
22512 051736 051710 R0663 ;ERROR LOOP RETURN ADDRESS
22513
22514 051740 010506 B0663: MOV R5,SP ;RESET SP
22515 051742 012737 065160 000004 MOV #BERR,@#4 ;RESTORE TIMEOUT VECTOR
22516
22517 051750 000004 00663: SCOPE ;CALL THE SCOPE LOOP UTILITY
22518

```

22519
 22520
 22521
 22522
 22523 051752 012700 000664
 22524 051756 013701 052002
 22525 051762 005737 066636
 22526 051766 100012
 22527 051770 012702 177546
 22528 051774 012704 000200
 22529 052000 000257
 22530
 22531 052002 030412
 22532
 22533 052004 001003
 22534
 22535 052006 011203
 22536 052010 104000
 22537 052012 052000
 22538
 22539 052014 000004
 22540

```

; *****
; .SBTTL T0664 KW11-L TEST - LKCSR BIT 7 SET
; *****

T0664:  MOV    #0664,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0664,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        TST    @#OPTION        ;IS THE KW11-L INSTALLED ??
        BPL    00664           ;BR IF NOT - SKIP THIS TEST
        MOV    #LKCSR,R2      ;DEST ADDR = 177546
        MOV    #200,R4        ;[LKCSR] S / B = 200

R0664:  CCC                    ;SCOPE SYNC

I0664:  BIT     R4,(R2)        ;TEST BIT 7 IN LKCSR

        BNE    00664         ;BR IF IT'S SET

E0664:  MOV    (R2),R3        ;GET WAS DATA
        ERROR  R0664        ;BIT 7 NOT SET IN LKCSR
        R0664        ;ERROR LOOP RETURN ADDRESS

00664:  SCOPE                ;CALL THE SCOPE LOOP UTILITY
  
```

```

22541 ; *****
22542 ; .SBTTL T0665 KW11-L TEST - LKCSR BIT 6 CLEAR
22543 ; *****
22544
22545 052016 012700 000665 T0665: MOV #0665,R0 ;LOAD R0 WITH TEST NO.
22546 052022 013701 052046 MOV @#I0665,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22547 052026 005737 066636 TST @#OPTION ;IS THE KW11-L INSTALLED ??
22548 052032 100013 BPL 00665 ;BR IF NOT - SKIP THIS TEST
22549 052034 012702 177546 MOV #LKCSR,R2 ;R2 POINTS TO LKCSR
22550 052040 012704 000200 MOV #200,R4 ;[LKCSR] S / B = 200
22551 052044 000257 R0665: CCC ;SCOPE SYNC
22552
22553 052046 032712 000100 I0665: BIT #100,(R2) ;TEST BIT 6 IN LKCSR
22554
22555 052052 001403 BEQ 00665 ;BR IF CLEAR
22556
22557 052054 011203 MOV (R2),R3 ;GET WAS DATA
22558 052056 104000 E0665: ERROR ;BIT 6 (INTR. ENAB.) IN LKCSR WAS SET
22559 052060 052044 R0665 ;ERROR LOOP RETURN ADDRESS
22560
22561 052062 000004 O0665: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

22562 ; *****
22563 ; .SBTTL T0666 KW11-L TEST - LKCSR BIT 6 SET
22564 ; *****
22565
22566 052064 012700 000666 T0666: MOV #0666,R0 ;LOAD R0 WITH TEST NO.
22567 052070 013701 052142 MOV @#10666,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22568 052074 005737 066636 TST @#OPTION ;IS THE KW11 INSTALLED ??
22569 052100 100037 BPL 00666 ;BR IF NOT - SKIP THIS TEST
22570 052102 010605 MOV SP,R5 ;SAVE SP
22571 052104 012702 177546 MOV #LKCSR,R2 ;R2 POINTS TO LKCSR
22572 052110 012704 000300 MOV #300,R4 ;[LKCSR] S / B = 300
22573 052114 012737 052160 000100 R0666: MOV #A0666,@#100 ;SET UP LCLK VECTOR IN CASE LOGIC
22574 052122 012737 000340 000102 MOV #340,@#102 ;FAULT CAUSES ATL INTERRUPT
22575 052130 010506 MOV R5,SP ;RESET SP FOR ERROR LOOP
22576 052132 012737 000340 177776 MOV #340,@#PSW ;SET PRIORITY TO LEVEL 7
22577 052140 000257 CCC ;SCOPE SYNC
22578
22579 052142 052712 000100 I0666: BIS #100,(R2) ;SET BIT 6 IN LKCSR
22580
22581 052146 020412 CMP R4,(R2) ;RESULT CORRECT?
22582 052150 001403 BEQ A0666 ;BR IF YES
22583
22584 052152 011203 MOV (R2),R3 ;GET WAS DATA
22585 052154 104000 E10666: ERROR ;BIT 6 FAILED TO SET IN LKCSR
22586 052156 052114 R0666 ;ERROR LOOP RETURN ADDRESS
22587
22588
22589 052160 042737 000102 000100 A0666: BIC #102,@#100 ;RESTORE TRAP CATCHER IN KW11-L VECTOR
22590 052166 005037 000102 CLR @#102
22591 052172 042712 000100 BIC #100,(R2) ;TURN OF KW11-L INTR. ENAB.
22592 052176 010506 MOV R5,SP ;RESET SP
22593
22594 052200 000004 00666: SCOPE ;CALL THE SCOPE LOOP UTILITY
22595

```

```

22596 ; *****
22597 ; .SBTTL T0667 KW11-L BASIC INTERRUPT TEST
22598 ; *****
22599
22600
22601 052202 012700 000667 T0667: MOV #0667,R0 ;LOAD R0 WITH TEST NO.
22602 052206 013701 052256 MOV @#I0667,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22603 052212 005737 066636 TST @#OPTION ;IS THE KW11-L INSTALLED ??
22604 052216 100041 BPL 00667 ;BR IF NOT - SKIP THIS TEST
22605 052220 010605 MOV SP,R5 ;SAVE SP
22606 052222 012702 177546 MOV #LKCSR,R2 ;R2 POINTS TO LKCSR
22607 052226 010506 R0667: MOV R5,SP ;RESET SP FOR ERROR LOOP
22608 052230 005004 CLR R4 ;INITIALIZE TIMER
22609 052232 012737 052276 000100 MOV #A0667,@#100 ;SET UP LINE CLOCK VECTOR TO TO
22610 052240 012737 000340 000102 MOV #340,@#102 ;TO A0667 WITH PROCESSOR PRIORITY = 7
22611 052246 005012 CLR (R2) ;CLEAR LKCSR
22612 052250 005037 177776 CLR @#PSW ;SET PRIORITY TO LEVEL 000
22613 052254 000257 CCC ;SCOPE SYNC
22614
22615 052256 052712 000100 I0667: BIS #100,(R2) ;ENABLE LINE CLK INTERRUPT
22616
22617 052262 005304 DEC R4 ;WAIT FOR INTR - REPORT ERROR IF
22618 052264 001376 BNE .-2 ;R4 GOES TO 000000
22619
22620 052266 042712 000100 E0667: BIC #100,(R2) ;TURN OFF INTR. ENAB.
22621 052272 104006 ERROR6 ;KW11-L FAILED TO INTERRUPT
22622 052274 052226 R0667 ;ERROR LOOP RETURN ADDRESS
22623
22624 052276 042712 000100 A0667: BIC #100,(R2) ;TURN OFF INTR. ENAB.
22625 052302 012737 000102 000100 MOV #102,@#100 ;RESTORE TRAP CATCHER IN KW11-L VECTOR
22626 052310 005037 000102 CLR @#102
22627 052314 010506 MOV R5,SP ;RESET SP
22628 052316 005037 177776 CLR @#PSW ;RESET PRIORITY TO LEVEL 0
22629
22630 052322 000004 00667: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

22631 ; *****
22632 ; .SBTTL T0670 RESET TEST - <N:C> = 1111
22633 ; *****
22634
22635 ;MICROPROGRAMMING / LOGIC INFORMATION
22636 ;ROM SEQ: [127,025,040,043,016] FC 1,6
22637
22638 ;ACT BUTS: 37[004]100,127 / 02[025]042,043
22639
22640 ;EXEC: [025] BUT02 TRIGGERS RESET LOGIC ON K5-8
22641
22642 ;CODES: N:C = 1111 (NO CHANGE)
22643
22644 ;SYNC: B05J2 (-) T = 80 MILLISEC
22645
22646 ;KEY SIG: K3-6 RESET L / K3-6 HALT+RESET L / K5-3 BUT02 H / K5-8 RESET RESTA
22647 ;K5-8 INIT*RESET H / K5-8 BUS INIT L / K5-8 P ENDRESET L /
22648 ; K2-8 CLKOFF(1) H
22649
22650
22651
22652 052324 012700 000670 T0670: MOV #0670,R0 ;LOAD R0 WITH TEST NO.
22653 052330 013701 052362 MOV @#I0670,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22654 052334 012737 000001 066664 MOV #1,@#ITCNT ;NO ITERATIONS ON THIS TEST
22655 052342 012702 177564 MOV #XCSR,R2 ;R2 POINTS TO DL11 XCSR
22656 052346 012737 000340 177776 R0670: MOV #340,@#PSW ;MAKE PRTY. BITS ALL 1'S
22657 052354 052712 000004 BIS #4,(R2) ;SET THE DL11 MAINT. BIT
22658 052360 000277 SCC ;N:C = 1111
22659
22660 052362 000005 I0670: RESET ;TEST THE RESET - IT SHOULD CLEAR THE DL11 MAINT BIT
22661
22662 052364 013705 177776 MOV @#PSW,R5 ;SAVE THE PSW
22663 052370 032712 000004 BIT #4,(R2) ;DID MAINT. BIT CLEAR ??
22664 052374 001404 BEQ A0670 ;BR IF YES
22665
22666 052376 042712 000004 E10670: BIC #4,(R2) ;MAKE SURE TO TURN OFF MAINT. BIT
22667 052402 104006 ERROR6 ;RESET FAILED TO CLEAR MAINT BIT
22668 052404 052346 R0670 ;ERROR LOOP RETURN ADDRESS
22669
22670 052406 022705 000357 A0670: CMP #357,R5 ;DID RESET ALTER THE PSW ??
22671 052412 001405 BEQ B0670 ;BR IF NOT
22672
22673 052414 012704 000357 MOV #357,R4 ;[R4] = S/B PSW
22674 052420 010503 MOV R5,R3 ;[R3] = WAS PSW
22675 052422 104000 E20670: ERROR ;RESET ALTERED THE PSW
22676 052424 052346 R0670 ;ERROR LOOP RETURN ADDRESS
22677
22678 052426 005037 177776 B0670: CLR @#PSW ;CLEAR OUT THE PSW
22679 052432 042712 000004 BIC #4,(R2) ;MAKE SURE MAINT BIT IS OFF
22680
22681 052436 000004 00670: SCOPE ;CALL THE SCOPE LOOP UTILITY
22682

```

```

22683 ; *****
22684 ; .SBTTL T0671 RESET TEST - <N:C> = 0000
22685 ; *****
22686
22687 ;MICROPROGRAMMING / LOGIC INFORMATION
22688
22689 ;ROM SEQ: [127,025,040,043,016] FC 1,6
22690
22691 ;ACT BUTS: 37[004]100,127 / 02[025]042,043
22692
22693 ;EXEC: [025] BUT02 TRIGGERS RESET LOGIC ON K5-8
22694
22695 ;CODES: N:C = 0000 (NO CHANGE)
22696
22697 ;SYNC: B05J2 (-) T = 80 MILLISEC
22698
22699 ;KEY SIG: K3-6 RESET L / K3-6 HALT+RESET L / K5-3 BUT02 H / K5-8 RESET RESTA
22700 ;K5-8 INIT*RESET H / K5-8 BUS INIT L / K5-8 P ENDRESET L /
22701 ; K2-8 CLKOFF (1) H
22702
22703 052440 012700 000671 T0671: MOV #0671,R0 ;LOAD R0 WITH TEST NO.
22704 052444 013701 052476 MOV @#I0671,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22705 052450 012737 000001 066664 MOV #1,@#ITCNT ;NO ITERATIONS ON THIS TEST
22706 052456 012702 177564 MOV #XCSR,R2 ;R2 POINTS TO DL11 XCSR
22707 052462 012737 000000 177776 R0671: MOV #0,@#PSW ;MAKE PRY. BITS ALL 0'S
22708 052470 052712 000004 BIS #4,(R2) ;SET THE DL11 MAINT. BIT
22709 052474 000257 CCC ;N:C = 0000
22710
22711 052476 000005 I0671: RESET ;TEST THE RESET - IT SHOULD CLEAR THE DL11 MAINT BIT
22712
22713 052500 013705 177776 MOV @#PSW,R5 ;SAVE THE PSW
22714 052504 032712 000004 BIT #4,(R2) ;DID MAINT. BIT CLEAR ??
22715 052510 001404 BEQ A0671 ;BR IF YES
22716
22717 052512 042712 000004 E10671: BIC #4,(R2) ;MAKE SURE TO TURN OFF MAINT. BIT
22718 052516 104006 ERROR6 ;RESET FAILED TO CLEAR MAINT BIT
22719 052520 052462 R0671 ;ERROR LOOP RETURN ADDRESS
22720
22721 052522 022705 000000 A0671: CMP #0,R5 ;DID RESET ALTER THE PSW ??
22722 052526 001405 BEQ B0671 ;BR IF NOT
22723
22724 052530 012704 000357 MOV #357,R4 ;[R4] = S/B PSW
22725 052534 010503 MOV R5,R3 ;[R3] = WAS PSW
22726 052536 104000 E20671: ERROR ;RESET ALTERED THE PSW
22727 052540 052462 R0671 ;ERROR LOOP RETURN ADDRESS
22728
22729 052542 005037 177776 B0671: CLR @#PSW ;CLEAR OUT THE PSW
22730 052546 042712 000004 BIC #4,(R2) ;MAKE SURE MAINT BIT IS OFF
22731
22732 052552 000004 00671: SCOPE ;CALL THE SCOPE LOOP UTILITY
22733

```

```

22734 ; *****
22735 ; .SBTTL T0672 WAIT INSTRUCTION TEST - [PSW] = 151
22736 ; *****
22737
22738 ;MICROPROGRAMMING / LOGIC INFORMATION
22739
22740 ;ROM SEQ: [114,015,012,020,021:
22741 ; INTR. 014,022,023,007,[TRAP MICROROUTINE]
22742 ; NO INTR. 017,015,012,020,021,017 ETC. FC 1,10,6,10
22743
22744 ;ACT BUTS: 37[004]100,114 / 26[114]010,012 / 25[020]014,014 / 07[022]006,007
22745
22746 ;EXEC BUT 25 IN LOC. 020 SHOULD CAUSE EXIT FROM WAIT LOOP
22747
22748 ;CODES: N / A / N:C=1001 (NO CHANGE)
22749
22750 ;SYNC: B05J2 (-) T=2.24 USEC + INTR. WAIT TIME
22751
22752 ;KEY SIG: K4-5 BRPTR(1) L / K3-6 WAIT L / K4-6 BRQ H / K5-4 -BRSV(0) H
22753
22754 052554 012700 000672 T0672: MOV #0672,R0 ;LOAD R0 WITH TEST NO.
22755 052560 013701 052642 MOV @#I0672,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22756 052564 010605 MOV SP,R5 ;SAVE THE SP
22757 052566 012702 177564 MOV #XCSR,R2 ;R2 POINT TO DL11 XCSR
22758 052572 012737 052662 000064 R0672: MOV #A0672,@#64 ;GO TO A0672 ON DL11 INTR.
22759 052600 012737 000200 000066 MOV #200,@#66 ;AT LEVEL 4
22760 052606 010506 MOV R5,SP ;RESET SP FOR ERROR LOOP
22761 052610 005012 CLR (R2) ;INIT DL11 XCSR
22762 052612 005003 CLR R3 ;INIT TIMER
22763
22764 052614 105712 1$: TSTB (R2) ;DL11 XMIT READY SET ??
22765 052616 100403 BMI 2$ ;BR IF YES
22766 052620 005303 DEC R3 ;COUNT THE TIMER
22767 052622 001374 BNE 1$ ;BR IF NO TIMEOUT
22768 052624 000441 BR E40672 ;GO REPORT TIMEOUT
22769
22770 052626 012737 000140 177776 2$: MOV #140,@#PSW ;SET PSW PRY BITS TO LEVEL 3
22771 052634 000277 SCC ;N:C=1111
22772 052636 152712 000100 BISB #100,(R2) ;ENAB. DL11 INTR - N:C=1001
22773
22774 052642 000001 I0672: WAIT ;TEST THE WAIT-GO TO A0672 ON INTR
22775
22776 052644 012737 000340 177776 MOV #340,@#PSW ;LOCK OUT INTR
22777 052652 005012 CLR (R2) ;TURN OFF DL11 INTR ENAB
22778 052654 104006 E10672: ERROR6 ;WAIT FAILED TO EXECUTE PROPERLY
22779 052656 052572 R0672 ;ERROR LOOP RETURN ADDRESS
22780 052660 000425 BR C0672 ;GO EXIT THIS TEST
22781
22782 052662 042712 000100 A0672: BIC #100,(R2) ;TURN OFF DL11 INTR ENAB
22783 052666 022716 052644 CMP #I0672+2,(SP) ;DID WAIT GET FETCHED ??
22784 052672 001403 BEQ B0672 ;BR IF YES
22785
22786 052674 104006 E20672: ERROR6 ;WAIT NOT FETCHED PROPERLY
22787 052676 052572 R0672 ;ERROR LOOP RETURN ADDRESS
22788 052700 000415 BR C0672 ;GO EXIT THE TEST
22789

```

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 566
 CBQEAC.P11 03-JUL-80 08:05 T0672 WAIT INSTRUCTION TEST - [PSW] = 151

SEQ 0566

```

22790 052702 022766 000151 000002 B0672:  CMP      #151,2(SP)      ;DID 'WAIT' ALTER THE PSW ??
22791 052710 001411                BEQ      C0672        ;BR IF YES
22792
22793 052712 012704 000151                MOV      #151,R4      ;[R4] = S/B PSW
22794 052716 016603 000002                MOV      2(SP),R3     ;[R3] = WAS PSW
22795 052722 104000                E30672: ERROR        ;'WAIT' ALTERED THE PSW
22796 052724 052572                R0672          ;ERROR LOOP RETURN ADDRESS
22797 052726 000402                BR       C0672        ;GOT TO EXIT TEST
22798
22799 052730 104006                E40672: ERROR6       ;DL11 FAILED TO SET READY ON TIME
22800 052732 052572                R0672          ;ERROR LOOP RETURN ADDRESS
22801
22802 052734 010506                C0672:  MOV      R5,SP   ;RESET THE SP
22803 052736 005037 177776                CLR      @#PSW       ;CLEAR OUT THE PSW
22804 052742 005012                CLR      (R2)        ;TURN OFF DL11 INTR.
22805 052744 012737 000066 000064                MOV      #66,@#64    ;RESTORE DL11 VECTOR WITH TRAPCATCHER
22806 052752 005037 000066                CLR      @#66
22807
22808 052756 000004                00672:  SCOPE        ;CALL THE SCOPE LOOP UTILITY
22809
22810

```

22811
22812
22813
22814
22815
22816
22817
22818
22819
22820
22821
22822
22823
22824
22825
22826
22827
22828
22829
22830
22831
22832
22833
22834
22835
22836
22837
22838
22839
22840
22841
22842
22843
22844
22845
22846
22847
22848
22849
22850
22851
22852
22853
22854
22855
22856
22857
22858
22859
22860
22861
22862
22863
22864
22865
22866

052760 012700 000673
052764 013701 053046
052770 010605
052772 012702 177564
052776 012737 053066 000064
053004 012737 000200 000066
053012 010506
053014 005012
053016 005003

053020 105712
053022 100403
053024 005303
053026 001374
053030 000441

053032 012737 000000 177776
053040 000257
053042 152712 000100

053046 000001

053050 012737 000340 177776
053056 005012
053060 104006
053062 052776
053064 000425

053066 042712 000100
053072 022716 053050
053076 001403

053100 104006
053102 052776
053104 000415

; *****
; .SBTTL T0673 WAIT INSTRUCTION TEST - [PSW] = 010
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [114,015,012,020,021:
; INTR. 014,022,023,007,[TRAP MICROROUTINE]
; NO INTR. 017,015,012,020,021,017 ETC. FC 1,10,6,10
;ACT BUTS: 37[004]100,114 / 26[114]010,012 / 25[020]014,014 / 07[022]006,007
;EXEC BUT 25 IN LOC. 020 SHOULD CAUSE EXIT FROM WAIT LOOP
;CODES: N / A / N:C=1000 (NO CHANGE)
;SYNC: B05J2 (-) T=2.24 USEC + INTR. WAIT TIME
;KEY SIG: K4-5 BRPTR(1) L / K3-6 WAIT L / K4-6 BRQ H / K5-4 -BRSV(0) H

T0673: MOV #0673,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0673,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV #XCSR,R2 ;R2 POINT TO DL11 XCSR
R0673: MOV #A0673,@#64 ;GO TO A0673 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

1\$: TSTB (R2) ;DL11 XMIT READY SET ??
BMI 2\$;BR IF YES
DEC R3 ;COUNT THE TIMER
BNE 1\$;BR IF NO TIMEOUT
BR E40673 ;GO REPORT TIMEOUT

2\$: MOV #0,@#PSW ;SET PSW PRY BITS TO LEVEL 0
CCC ;N:C=0000
BISB #100,(R2) ;ENAB. DL11 INTR - N:C=1000

I0673: WAIT ;TEST THE WAIT-GO TO A0673 ON INTR

E10673: MOV #340,@#PSW ;LOCK OUT INTR
CLR (R2) ;TURN OFF DL11 INTR ENAB
ERROR6 ;WAIT FAILED TO EXECUTE PROPERLY
R0673 ;ERROR LOOP RETURN ADDRESS
BR C0673 ;GO EXIT THIS TEST

A0673: BIC #100,(R2) ;TURN OFF DL11 INTR ENAB
CMP #I0673+2,(SP) ;DID WAIT GET FETCHED ??
BEQ B0673 ;BR IF YES

E20673: ERROR6 ;WAIT NOT FETCHED PROPERLY
R0673 ;ERROR LOOP RETURN ADDRESS
BR C0673 ;GO EXIT THE TEST

```

22867 053106 022766 000010 000002 B0673:  CMP    #010,2(SP)    ;DID 'WAIT' ALTER THE PSW ??
22868 053114 001411                BEQ    C0673        ;BR IF YES
22869
22870 053116 012704 000010                MOV    #010,R4      ;[R4] = S/B PSW
22871 053122 016603 000002                MOV    2(SP),R3     ;[R3] = WAS PSW
22872 053126 104000                E30673: ERROR      ;'WAIT' ALTERED THE PSW
22873 053130 052776                R0673              ;ERROR LOOP RETURN ADDRESS
22874 053132 000402                BR     C0673        ;GOT TO EXIT TEST
22875
22876 053134 104006                E40673: ERROR6     ;DL11 FAILED TO SET READY ON TIME
22877 053136 052776                R0673              ;ERROR LOOP RETURN ADDRESS
22878
22879 053140 010506                C0673:  MOV    R5,SP    ;RESET THE SP
22880 053142 005037 177776                CLR    @#PSW        ;CLEAR OUT THE PSW
22881 053146 005012                CLR    (R2)         ;TURN OFF DL11 INTR.
22882 053150 012737 000066 000064                MOV    #66,@#64    ;RESTORE DL11 VECTOR WITH TRAPCATCHER
22883 053156 005037 000066                CLR    @#66
22884
22885 053162 000004                00673:  SCOPE      ;CALL THE SCOPE LOOP UTILITY
22886
22887
22888

```

22889
22890
22891
22892
22893
22894
22895
22896
22897
22898
22899
22900
22901
22902
22903
22904
22905
22906
22907
22908
22909
22910
22911
22912
22913
22914
22915
22916
22917
22918
22919
22920
22921
22922

053164 012700 000674
053170 013701 053240
053174 005737 066636
053200 100041
053202 010605
053204 012702 177546
053210 012737 053260 000100
053216 012737 000340 000102
053224 010506
053226 005004
053230 012737 000000 177776
053236 000257
053240 052712 000100
053244 005304
053246 001376
053250 042712 000100
053254 104006
053256 053224
053260 042712 000100
053264 012737 000102 000100
053272 005037 000102
053276 010506
053300 005037 177776
053304 000004

```
; *****  
.SBTTL T0674 BR PRIORITY ARBITRATION TEST - LEVEL 0 USING KW11-L  
; *****  
T0674: MOV #0674,R0 ;LOAD R0 WITH TEST NO.  
MOV @#10674,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
TST @#OPTION ;IS KW11-L INSTALLED ??  
BPL 00674 ;BR IF NOT - SKIP THIS TEST  
MOV SP,R5 ;SAVE THE SP  
MOV #LKCSR,R2 ;R2 POINTS TO LCLK  
MOV #A0674,@#100 ;GO TO A0674 ON LCLK INTR  
MOV #340,@#102 ;LOCK OUT INTR IN LCLK SERV.  
R0674: MOV R5,SP ;RESET SP FOR ERROR LOOP  
CLR R4 ;INIT R4 AS TIMER  
MOV #0,@#PSW ;SET CPU PRY TO LEVEL 000  
CCC ;SCOPE SYNC  
  
I0674: BIS #100,(R2) ;ENABLE LCLK INTR  
  
DEC R4 ;COUNT TIMER  
BNE .-2 ;LCLK SHOULD INTR BEFORE TIMEOUT  
  
E0674: BIC #100,(R2) ;TURN OFF INTR ENAB.  
ERROR6 ;KW11-L FAILED TO INTR AT LEVEL 0  
R0674 ;ERROR LOOP RETURN ADDRESS  
  
A0674: BIC #100,(R2) ;TURN OFF INTR. ENABLE  
MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECT  
CLR @#102  
MOV R5,SP ;RESET THE SP  
CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0  
  
O0674: SCOPE ;CALL SCOPE LOOP UTILITY
```

```

22923 ; *****
22924 ; .SBTTL T0675 BR PRIORITY ARBITRATION TEST - LEVEL 1 USING KW11-L
22925 ; *****
22926
22927 053306 012700 000675 T0675: MOV #0675,R0 ;LOAD R0 WITH TEST NO.
22928 053312 013701 053362 MOV @#I0675,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22929 053316 005737 066636 TST @#OPTION ;IS KW11-L INSTALLED ??
22930 053322 100041 BPL 00675 ;BR IF NOT - SKIP THIS TEST
22931 053324 010605 MOV SP,R5 ;SAVE THE SP
22932 053326 012702 177546 MOV #LKCSR,R2 ;R2 POINTS TO KW11-L CSR
22933 053332 012737 053402 000100 MOV #A0675,@#100 ;IF INTR OCCURS - GO TO A0675
22934 053340 012737 000340 000102 MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
22935 053346 010506 R0675: MOV R5,SP ;RESET SP FOR ERROR LOOPING
22936 053350 005004 CLR R4 ;INITIALIZE R4 AS TIMER
22937 053352 012737 000040 177776 MOV #40,@#PSW ;SET CPU PRIORITY TO LEVEL 1
22938 053360 000257 CCC ;SCOPE SYNC
22939
22940 053362 052712 000100 I0675: BIS #100,(R2) ;ENABLE KW11-L INTERRUPTS
22941
22942 053366 005304 DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
22943 053370 001376 BNE .-2 ;TIMER FROM GETTING BACK TO 000000
22944
22945 053372 042712 000100 E0675: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
22946 053376 104006 ERROR6 ;KW11-L FAILED TO INTR AT LEVEL 1
22947 053400 053346 R0675 ;ERROR LOOP RETURN ADDRESS
22948
22949 053402 042712 000100 A0675: BIC #100,(R2) ;TURN OFF INTR. ENABLE
22950 053406 012737 000102 000100 MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
22951 053414 005037 000102 CLR @#102
22952 053420 010506 MOV R5,SP ;RESET THE SP
22953 053422 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
22954
22955 053426 000004 O0675: SCOPE ;CALL SCOPE LOOP UTILITY
22956

```

```

22957 ; *****
22958 ; .SBTTL T0676 BR PRIORITY ARBITRATION TEST - LEVEL 2 USING KW11-L
22959 ; *****
22960
22961 053430 012700 000676 T0676: MOV #0676,R0 ;LOAD R0 WITH TEST NO.
22962 053434 013701 053504 MOV @#I0676,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22963 053440 005737 066636 TST @#OPTION ;IS KW11-L INSTALLED ??
22964 053444 100041 BPL 00676 ;BR IF NOT - SKIP THIS TEST
22965 053446 010605 MOV SP,R5 ;SAVE THE SP
22966 053450 012702 177546 MOV #LKCSR,R2 ;R2 POINTS TO KW11-L CSR
22967 053454 012737 053524 000100 MOV #A0676,@#100 ;IF INTR OCCURS - GO TO A0676
22968 053462 012737 000340 000102 MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
22969 053470 010506 R0676: MOV R5,SP ;RESET SP FOR ERROR LOOPING
22970 053472 005004 CLR R4 ;INITIALIZE R4 AS TIMER
22971 053474 012737 000100 177776 MOV #100,@#PSW ;SET CPU PRIORITY TO LEVEL 2
22972 053502 000257 CCC ;SCOPE SYNC
22973
22974 053504 052712 000100 I0676: BIS #100,(R2) ;ENABLE KW11-L INTERRUPTS
22975
22976 053510 005304 DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
22977 053512 001376 BNE .-2 ;TIMER FROM GETTING BACK TO 000000
22978
22979 053514 042712 000100 E0676: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
22980 053520 104006 ERROR6 ;KW11-L FAILED TO INTR AT LEVEL 2
22981 053522 053470 R0676 ;ERROR LOOP RETURN ADDRESS
22982
22983 053524 042712 000100 A0676: BIC #100,(R2) ;TURN OFF INTR. ENABLE
22984 053530 012737 000102 000100 MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
22985 053536 005037 000102 CLR @#102
22986 053542 010506 MOV R5,SP ;RESET THE SP
22987 053544 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
22988
22989 053550 000004 00676: SCOPE ;CALL SCOPE LOOP UTILITY
22990

```

```

22991 ; *****
22992 ; .SBTTL T0677 BR PRIORITY ARBITRATION TEST - LEVEL 3 USING KW11-L
22993 ; *****
22994
22995 053552 012700 000677 T0677: MOV #0677,R0 ;LOAD R0 WITH TEST NO.
22996 053556 013701 053626 MOV @#I0677,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
22997 053562 005737 066636 TST @#OPTION ;IS KW11-L INSTALLED ??
22998 053566 100041 BPL 00677 ;BR IF NOT - SKIP THIS TEST
22999 053570 010605 MOV SP,R5 ;SAVE THE SP
23000 053572 012702 177546 MOV #LKCSR,R2 ;R2 POINTS TO KW11-L CSR
23001 053576 012737 053646 000100 MOV #A0677,@#100 ;IF INTR OCCURS - GO TO A0677
23002 053604 012737 000340 000102 MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
23003 053612 010506 R0677: MOV R5,SP ;RESET SP FOR ERROR LOOPING
23004 053614 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23005 053616 012737 000140 177776 MOV #140,@#PSW ;SET CPU PRIORITY TO LEVEL 3
23006 053624 000257 CCC ;SCOPE SYNC
23007
23008 053626 052712 000100 I0677: BIS #100,(R2) ;ENABLE KW11-L INTERRUPTS
23009
23010 053632 005304 DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
23011 053634 001376 BNE .-2 ;TIMER FROM GETTING BACK TO 000000
23012
23013 053636 042712 000100 E0677: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
23014 053642 104006 ERROR6 ;KW11-L FAILED TO INTR AT LEVEL 3
23015 053644 053612 R0677 ;ERROR LOOP RETURN ADDRESS
23016
23017 053646 042712 000100 A0677: BIC #100,(R2) ;TURN OFF INTR. ENABLE
23018 053652 012737 000102 000100 MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
23019 053660 005037 000102 CLR @#102
23020
23021 053664 010506 MOV R5,SP ;RESET THE SP
23022 053666 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23023
23024 053672 000004 00677: SCOPE ;CALL SCOPE LOOP UTILITY
23025

```

```

23026 ; *****
23027 ; .SBTTL T0700 BR PRIORITY ARBITRATION TEST - LEVEL 4 USING KW11-L
23028 ; *****
23029
23030 053674 012700 000700 T0700: MOV #0700,R0 ;LOAD R0 WITH TEST NO.
23031 053700 013701 053750 MOV @#10700,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23032 053704 005737 066636 TST @#OPTION ;IS KW11-L INSTALLED ??
23033 053710 100041 BPL 00700 ;BR IF NOT - SKIP THIS TEST
23034 053712 010605 MOV SP,R5 ;SAVE THE SP
23035 053714 012702 177546 MOV #LKCSR,R2 ;R2 POINTS TO KW11-L CSR
23036 053720 012737 053770 000100 MOV #A0700,@#100 ;IF INTR OCCURS - GO TO A0700
23037 053726 012737 000340 000102 MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
23038 053734 010506 R0700: MOV R5,SP ;RESET SP FOR ERROR LOOPING
23039 053736 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23040 053740 012737 000200 177776 MOV #200,@#PSW ;SET CPU PRIORITY TO LEVEL 4
23041 053746 000257 CCC ;SCOPE SYNC
23042
23043 053750 052712 000100 I0700: BIS #100,(R2) ;ENABLE KW11-L INTERRUPTS
23044
23045 053754 005304 DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
23046 053756 001376 BNE .-2 ;TIMER FROM GETTING BACK TO 000000
23047
23048 053760 042712 000100 E0700: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
23049 053764 104006 ERROR6 ;KW11-L FAILED TO INTR AT LEVEL 4
23050 053766 053734 R0700 ;ERROR LOOP RETURN ADDRESS
23051
23052 053770 042712 000100 A0700: BIC #100,(R2) ;TURN OFF INTR. ENABLE
23053 053774 012737 000102 000100 MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
23054 054002 005037 000102 CLR @#102
23055 054006 010506 MOV R5,SP ;RESET THE SP
23056 054010 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23057
23058 054014 000004 00700: SCOPE ;CALL SCOPE LOOP UTILITY
23059

```

```

23060 ; *****
23061 ; .SBTTL T0701 BR PRIORITY ARBITRATION TEST - LEVEL 5 USING KW11-L
23062 ; *****
23063
23064 054016 012700 000701 T0701: MOV #0701,R0 ;LOAD R0 WITH TEST NO.
23065 054022 013701 054072 MOV @#10701,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23066 054026 005737 066636 TST @#OPTION ;IS KW11-L INSTALLED ??
23067 054032 100041 BPL 00701 ;BR IF NOT - SKIP THIS TEST
23068 054034 010605 MOV SP,R5 ;SAVE THE SP
23069 054036 012702 177546 MOV #LKCSR,R2 ;R2 POINTS TO KW11-L CSR
23070 054042 012737 054112 000100 MOV #A0701,@#100 ;IF INTR OCCURS - GO TO A0701
23071 054050 012737 000340 000102 MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
23072 054056 010506 R0701: MOV R5,SP ;RESET SP FOR ERROR LOOPING
23073 054060 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23074 054062 012737 000240 177776 MOV #240,@#PSW ;SET CPU PRIORITY TO LEVEL 5
23075 054070 000257 CCC ;SCOPE SYNC
23076
23077 054072 052712 000100 I0701: BIS #100,(R2) ;ENABLE KW11-L INTERRUPTS
23078
23079 054076 005304 DEC R4 ;COUNT THE TIMER - LCLK SHOULD PREVENT
23080 054100 001376 BNE .-2 ;TIMER FROM GETTING BACK TO 000000
23081
23082 054102 042712 000100 E0701: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
23083 054106 104006 ERROR6 ;KW11-L FAILED TO INTR AT LEVEL 5
23084 054110 054056 R0701 ;ERROR LOOP RETURN ADDRESS
23085
23086 054112 042712 000100 A0701: BIC #100,(R2) ;TURN OFF INTR. ENABLE
23087 054116 012737 000102 000100 MOV #102,@#100 ;RESTORE TRAP CATCHER IN THE VECTOR
23088 054124 005037 000102 CLR @#102
23089 054130 010506 MOV R5,SP ;RESET THE SP
23090 054132 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23091
23092 054136 000004 00701: SCOPE ;CALL SCOPE LOOP UTILITY
23093

```

```

23094 ; *****
23095 ; .SBTTL T0702 BR PRIORITY ARBITRATION TEST - LEVEL 0 USING DL11
23096 ; *****
23097
23098 054140 012700 000702 T0702: MOV #0702,R0 ;LOAD R0 WITH TEST NO.
23099 054144 013701 054206 MOV @#I0702,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23100 054150 010605 MOV SP,R5 ;SAVE THE SP
23101 054152 012702 177564 MOV #XCSR,R2 ;R2 POINTS TO DL11 XCSR
23102 054156 012737 054226 000064 MOV #A0702,@#64 ;IF INTR OCCURS - GO TO A0702
23103 054164 012737 000340 000066 MOV #340,@#66 ;WITH CPU PRIORITY AT LEVEL 7
23104 054172 010506 R0702: MOV R5,SP ;RESET SP FOR ERROR LOOPING
23105 054174 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23106 054176 012737 000000 177776 MOV #0,@#PSW ;SET CPU PRIORITY TO LEVEL 0
23107 054204 000257 CCC ;SCOPE SYNC
23108
23109 054206 052712 000100 I0702: BIS #100,(R2) ;ENABLE DL11 INTERRUPTS
23110
23111 054212 005304 DEC R4 ;COUNT THE TIMER - DL11 SHOULD PREVENT
23112 054214 001376 BNE .-2 ;TIMER FROM GETTING BACK TO 000000
23113
23114 054216 042712 000100 E0702: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
23115 054222 104006 ERROR6 ;DL11 FAILED TO INTR AT LEVEL 0
23116 054224 054172 R0702 ;ERROR LOOP RETURN ADDRESS
23117
23118 054226 042712 000100 A0702: BIC #100,(R2) ;TURN OFF INTR. ENABLE
23119 054232 012737 000066 000064 MOV #66,@#64 ;RESTORE TRAP CATCHER IN THE VECTOR
23120 054240 005037 000066 CLR @#66
23121 054244 010506 MOV R5,SP ;RESET THE SP
23122 054246 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23123
23124 054252 000004 00702: SCOPE ;CALL SCOPE LOOP UTILITY
23125

```

```

23126 ; *****
23127 ; .SBTTL T0703 BR PRIORITY ARBITRATION TEST - LEVEL 1 USING DL11
23128 ; *****
23129
23130 054254 012700 000703 T0703: MOV #0703,R0 ;LOAD R0 WITH TEST NO.
23131 054260 013701 054322 MOV @#I0703,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23132 054264 010605 MOV SP,R5 ;SAVE THE SP
23133 054266 012702 177564 MOV #XCSR,R2 ;R2 POINTS TO DL11 XCSR
23134 054272 012737 054342 000064 MOV #A0703,@#64 ;IF INTR OCCURS - GO TO A0703
23135 054300 012737 000340 000066 MOV #340,@#66 ;WITH CPU PRIORITY AT LEVEL 7
23136 054306 010506 R0703: MOV R5,SP ;RESET SP FOR ERROR LOOPING
23137 054310 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23138 054312 012737 000040 177776 MOV #40,@#PSW ;SET CPU PRIORITY TO LEVEL 1
23139 054320 000257 CCC ;SCOPE SYNC
23140
23141 054322 052712 000100 I0703: BIS #100,(R2) ;ENABLE DL11 INTERRUPTS
23142
23143 054326 005304 DEC R4 ;COUNT THE TIMER - DL11 SHOULD PREVENT
23144 054330 001376 BNE .-2 ;TIMER FROM GETTING BACK TO 000000
23145
23146 054332 042712 000100 E0703: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
23147 054336 104006 ERROR6 ;DL11 FAILED TO INTR AT LEVEL 1
23148 054340 054306 R0703 ;ERROR LOOP RETURN ADDRESS
23149
23150 054342 042712 000100 A0703: BIC #100,(R2) ;TURN OFF INTR. ENABLE
23151 054346 012737 000066 000064 MOV #66,@#64 ;RESTORE TRAP CATCHER IN THE VECTOR
23152 054354 005037 000066 CLR @#66
23153 054360 010506 MOV R5,SP ;RESET THE SP
23154 054362 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23155
23156 054366 000004 00703: SCOPE ;CALL SCOPE LOOP UTILITY
23157

```

```

23158 ; *****
23159 ; .SBTTL T0704 BR PRIORITY ARBITRATION TEST - LEVEL 2 USING DL11
23160 ; *****
23161
23162 054370 012700 000704 T0704: MOV #0704,R0 ;LOAD R0 WITH TEST NO.
23163 054374 013701 054450 MOV @#I0704,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23164 054400 032737 020000 066642 BIT #20000,@#BPTLOC ;BREAKPOINT HALT SET ??
23165 054406 001401 BEQ .+4 ;BR IF NOT
23166 054410 000000 HALT ;BREAK-DEPRESS CONTINUE TO RESTART
23167 054412 010605 MOV SP,R5 ;SAVE THE SP
23168 054414 012702 177564 MOV #XCSR,R2 ;R2 POINTS TO DL11 CSR
23169 054420 012737 054470 000064 MOV #A0704,@#64 ;IF INTR OCCURS - GO TO A0704
23170 054426 012737 000340 000066 MOV #340,@#66 ;WITH CPU PRIORITY AT LEVEL 7
23171 054434 010506 R0704: MOV R5,SP ;RESET SP FOR ERROR LOOPING
23172 054436 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23173 054440 012737 000100 177776 MOV #100,@#PSW ;SET CPU PRIORITY TO LEVEL 2
23174 054446 000257 CCC ;SCOPE SYNC
23175
23176 054450 052712 000100 I0704: BIS #100,(R2) ;ENABLE DL11 INTERRUPTS
23177
23178 054454 005304 DEC R4 ;COUNT THE TIMER - DL11 SHOULD PREVENT
23179 054456 001376 BNE .-2 ;TIMER FROM GETTING BACK TO 000000
23180
23181 054460 042712 000100 E0704: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
23182 054464 104006 ERROR6 ;DL11 FAILED TO INTR AT LEVEL 2
23183 054466 054434 R0704 ;ERROR LOOP RETURN ADDRESS
23184
23185 054470 042712 000100 A0704: BIC #100,(R2) ;TURN OFF INTR. ENABLE
23186 054474 012737 000066 000064 MOV #66,@#64 ;RESTORE TRAP CATCHER IN THE VECTOR
23187 054502 005037 000066 CLR @#66
23188 054506 010506 MOV R5,SP ;RESET THE SP
23189 054510 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23190
23191 054514 000004 00704: SCOPE ;CALL SCOPE LOOP UTILITY
23192

```

```

23193 ; *****
23194 ; .SBTTL T0705 BR PRIORITY ARBITRATION TEST - LEVEL 3 USING DL11
23195 ; *****
23196
23197 054516 012700 000705 T0705: MOV #0705,R0 ;LOAD R0 WITH TEST NO.
23198 054522 013701 054564 MOV @#I0705,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23199 054526 010605 MOV SP,R5 ;SAVE THE SP
23200 054530 012702 177564 MOV #XCSR,R2 ;R2 POINTS TO DL11 CSR
23201 054534 012737 054604 000064 MOV #A0705,@#64 ;IF INTR OCCURS - GO TO A0705
23202 054542 012737 000340 000066 MOV #340,@#66 ;WITH CPU PRIORITY AT LEVEL 7
23203 054550 010506 R0705: MOV R5,SP ;RESET SP FOR ERROR LOOPING
23204 054552 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23205 054554 012737 000140 177776 MOV #140,@#PSW ;SET CPU PRIORITY TO LEVEL 3
23206 054562 000257 CCC ;SCOPE SYNC
23207
23208 054564 052712 000100 I0705: BIS #100,(R2) ;ENABLE DL11 INTERRUPTS
23209
23210 054570 005304 DEC R4 ;COUNT THE TIMER - DL11 SHOULD PREVENT
23211 054572 001376 BNE .-2 ;TIMER FROM GETTING BACK TO 000000
23212
23213 054574 042712 000100 E0705: BIC #100,(R2) ;TURN OFF THE INTERRUPT ENABLE
23214 054600 104006 ERROR6 ;DL11 FAILED TO INTR AT LEVEL 3
23215 054602 054550 R0705 ;ERROR LOOP RETURN ADDRESS
23216
23217 054604 042712 000100 A0705: BIC #100,(R2) ;TURN OFF INTR. ENABLE
23218 054610 012737 000066 000064 MOV #66,@#64 ;RESTORE TRAP CATCHER IN THE VECTOR
23219 054616 005037 000066 CLR @#66
23220 054622 010506 MOV R5,SP ;RESET THE SP
23221 054624 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23222
23223 054630 000004 00705: SCOPE ;CALL SCOPE LOOP UTILITY

```

```

23224 ; *****
23225 ; .SBTTL T0706 BR PRIORITY ARBITRATION TEST - LEVEL 7 USING KW11-L
23226 ; *****
23227
23228 054632 012700 000706 T0706: MOV #0706,R0 ;LOAD R0 WITH TEST NO.
23229 054636 013701 054706 MOV @#10706,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23230 054642 005737 066636 TST @#OPTION ;IS KW11-L INSTALLED ??
23231 054646 100042 BPL 00706 ;BR IF NOT - SKIP THIS TEST
23232 054650 010605 MOV SP,R5 ;SAVE THE SP
23233 054652 012702 177546 MOV #LKCSR,R2 ;R2 POINTS TO KW11-L CSR
23234 054656 012737 054720 000100 MOV #A0706,@#100 ;IF INTR OCCURS - GO TO A0706
23235 054664 012737 000340 000102 MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
23236 054672 010506 R0706: MOV R5,SP ;RESET SP FOR ERROR LOOP
23237 054674 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23238 054676 012737 000340 177776 MOV #340,@#PSW ;SET CPU PRIORITY TO LEVEL 7
23239 054704 000257 CCC ;SCOPE SYNC
23240
23241 054706 052712 000100 I0706: BIS #100,(R2) ;ENABLE INTERRUPTS
23242
23243 054712 005304 DEC R4 ;COUNT UNTIL [R4] = 000000 - THEN
23244 054714 001376 BNE .-2 ;CONTINUE - NO INTERRUPT SHOULD OCCUR
23245 054716 000404 BR B0706 ;GO TO EXIT - ALL OK
23246
23247 054720 042712 000100 A0706: BIC #100,(R2) ;TURN OFF THE INTR ENABLE
23248 054724 104006 E0706: ERROR6 ;INTR OCCURRED WITH CPU AT LEVEL 7
23249 054726 054672 R0706 ;ERROR LOOP RETURN ADDRESS
23250
23251 054730 042712 000100 B0706: BIC #100,(R2) ;TURN OFF INTR ENABLE
23252 054734 012737 000102 000100 MOV #102,@#100 ;RESET THE TRAP CATCHER IN THE VECTOR
23253 054742 005037 000102 CLR @#102
23254 054746 010506 MOV R5,SP ;RESET SP JUST IN CASE
23255 054750 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23256
23257 054754 000004 00706: SCOPE ;CALL SCOPE LOOP UTILITY
23258

```

```

23259 ; *****
23260 ; .SBTTL T0707 BR PRIORITY ARBITRATION TEST - LEVEL 6 USING KW11-L
23261 ; *****
23262
23263 054756 012700 000707 T0707: MOV #0707,R0 ;LOAD R0 WITH TEST NO.
23264 054762 013701 055032 MOV @#I0707,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23265 054766 005737 066636 TST @#OPTION ;IS KW11-L INSTALLED ??
23266 054772 100042 BPL 00707 ;BR IF NOT - SKIP THIS TEST
23267 054774 010605 MOV SP,R5 ;SAVE THE SP
23268 054776 012702 177546 MOV #LKCSR,R2 ;R2 POINTS TO KW11-L CSR
23269 055002 012737 055044 000100 MOV #A0707,@#100 ;IF INTR OCCURS - GO TO A0707
23270 055010 012737 000340 000102 MOV #340,@#102 ;WITH CPU PRIORITY AT LEVEL 7
23271 055016 010506 R0707: MOV R5,SP ;RESET SP FOR ERROR LOOP
23272 055020 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23273 055022 012737 000300 177776 MOV #300,@#PSW ;SET CPU PRIORITY TO LEVEL 6
23274 055030 000257 CCC ;SCOPE SYNC
23275
23276 055032 052712 000100 I0707: BIS #100,(R2) ;ENABLE INTERRUPTS
23277
23278 055036 005304 DEC R4 ;COUNT UNTIL [R4] = 000000 - THEN
23279 055040 001376 BNE .-2 ;CONTINUE - NO INTERRUPT SHOULD OCCUR
23280 055042 000404 BR B0707 ;GO TO EXIT - ALL OK
23281
23282 055044 042712 000100 A0707: BIC #100,(R2) ;TURN OFF THE INTR ENABLE
23283 055050 104006 E0707: ERROR6 ;INTR OCCURRED WITH CPU AT LEVEL 6
23284 055052 055016 R0707 ;ERROR LOOP RETURN ADDRESS
23285
23286 055054 042712 000100 B0707: BIC #100,(R2) ;TURN OFF INTR ENABLE
23287 055060 012737 000102 000100 MOV #102,@#100 ;RESET THE TRAP CATCHER IN THE VECTOR
23288 055066 005037 000102 CLR @#102
23289 055072 010506 MOV R5,SP ;RESET SP JUST IN CASE
23290 055074 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23291
23292 055100 000004 00707: SCOPE ;CALL SCOPE LOOP UTILITY
23293

```

```

23294 : *****
23295 : .SBTTL T0710 BR PRIORITY ARBITRATION TEST - LEVEL 7 USING DL11
23296 : *****
23297
23298 055102 012700 000710 T0710: MOV #0710,R0 ;LOAD R0 WITH TEST NO.
23299 055106 013701 055150 MOV @#10710,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23300 055112 010605 MOV SP,R5 ;SAVE THE SP
23301 055114 012702 177564 MOV #XCSR,R2 ;R2 POINTS TO DL11 XCSR
23302 055120 012737 055162 000064 MOV #A0710,@#64 ;IF INTR OCCURS - GO TO A0710
23303 055126 012737 000340 000066 MOV #340,@#66 ;WITH CPU PRIORITY AT LEVEL 7
23304 055134 010506 R0710: MOV R5,SP ;RESET SP FOR ERROR LOOP
23305 055136 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23306 055140 012737 000340 177776 MOV #340,@#PSW ;SET CPU PRIORITY TO LEVEL 7
23307 055146 000257 CCC ;SCOPE SYNC
23308
23309 055150 052712 000100 I0710: BIS #100,(R2) ;ENABLE INTERRUPTS
23310
23311 055154 005304 DEC R4 ;COUNT UNTIL [R4] = 000000 - THEN
23312 055156 001376 BNE .-2 ;CONTINUE - NO INTERRUPT SHOULD OCCUR
23313 055160 000404 BR B0710 ;GO TO EXIT - ALL OK
23314
23315 055162 042712 000100 A0710: BIC #100,(R2) ;TURN OFF THE INTR ENABLE
23316 055166 104006 E0710: ERROR6 ;INTR OCCURRED WITH CPU AT LEVEL 7
23317 055170 055134 R0710 ;ERROR LOOP RETURN ADDRESS
23318
23319 055172 042712 000100 B0710: BIC #100,(R2) ;TURN OFF INTR ENABLE
23320 055176 012737 000066 000064 MOV #66,@#64 ;RESET THE TRAP CATCHER IN THE VECTOR
23321 055204 005037 000066 CLR @#66
23322 055210 010506 MOV R5,SP ;RESET SP JUST IN CASE
23323 055212 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23324
23325 055216 000004 00710: SCOPE ;CALL SCOPE LOOP UTILITY
23326
    
```

```

23327 ; *****
23328 ; .SBTTL T0711 BR PRIORITY ARBITRATION TEST - LEVEL 6 USING DL11
23329 ; *****
23330
23331 055220 012700 000711 T0711: MOV #0711,R0 ;LOAD R0 WITH TEST NO.
23332 055224 013701 055266 MOV @#10711,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23333 055230 010605 MOV SP,R5 ;SAVE THE SP
23334 055232 012702 177564 MOV #XCSR,R2 ;R2 POINTS TO DL11 XCSR
23335 055236 012737 055300 000064 MOV #A0711,@#64 ;IF INTR OCCURS - GO TO A0711
23336 055244 012737 000340 000066 MOV #340,@#66 ;WITH CPU PRIORITY AT LEVEL 7
23337 055252 010506 R0711: MOV R5,SP ;RESET SP FOR ERROR LOOP
23338 055254 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23339 055256 012737 000300 177776 MOV #300,@#PSW ;SET CPU PRIORITY TO LEVEL 6
23340 055264 000257 CCC ;SCOPE SYNC
23341
23342 055266 052712 000100 I0711: BIS #100,(R2) ;ENABLE INTERRUPTS
23343
23344 055272 005304 DEC R4 ;COUNT UNTIL [R4] = 000000 - THEN
23345 055274 001376 BNE -2 ;CONTINUE - NO INTERRUPT SHOULD OCCUR
23346 055276 000404 BR B0711 ;GO TO EXIT - ALL OK
23347
23348 055300 042712 000100 A0711: BIC #100,(R2) ;TURN OFF THE INTR ENABLE
23349 055304 104006 E0711: ERROR6 ;INTR OCCURRED WITH CPU AT LEVEL 6
23350 055306 055252 R0711 ;ERROR LOOP RETURN ADDRESS
23351
23352 055310 042712 000100 B0711: BIC #100,(R2) ;TURN OFF INTR ENABLE
23353 055314 012737 000066 000064 MOV #66,@#64 ;RESET THE TRAP CATCHER IN THE VECTOR
23354 055322 005037 000066 CLR @#66
23355 055326 010506 MOV R5,SP ;RESET SP JUST IN CASE
23356 055330 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23357
23358 055334 000004 00711: SCOPE ;CALL SCOPE LOOP UTILITY
23359

```

```
23360 ; *****
23361 ; .SBTTL T0712 BR PRIORITY ARBITRATION TEST - LEVEL 5 USING DL11
23362 ; *****
23363
23364 055336 012700 000712 T0712: MOV #0712,R0 ;LOAD R0 WITH TEST NO.
23365 055342 013701 055404 MOV @#10712,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23366 055346 010605 MOV SP,R5 ;SAVE THE SP
23367 055350 012702 177564 MOV #XCSR,R2 ;R2 POINTS TO DL11 XCSR
23368 055354 012737 055416 000064 MOV #A0712,@#64 ;IF INTR OCCURS - GO TO A0712
23369 055362 012737 000340 000066 MOV #340,@#66 ;WITH CPU PRIORITY AT LEVEL 7
23370 055370 010506 R0712: MOV R5,SP ;RESET SP FOR ERROR LOOP
23371 055372 005004 CLR R4 ;INITIALIZE R4 AS TIMER
23372 055374 012737 000240 177776 MOV #240,@#PSW ;SET CPU PRIORITY TO LEVEL 5
23373 055402 000257 CCC ;SCOPE SYNC
23374
23375 055404 052712 000100 I0712: BIS #100,(R2) ;ENABLE INTERRUPTS
23376
23377 055410 005304 DEC R4 ;COUNT UNTIL [R4] = 000000 - THEN
23378 055412 001376 BNE .-2 ;CONTINUE - NO INTERRUPT SHOULD OCCUR
23379 055414 000404 BR B0712 ;GO TO EXIT - ALL OK
23380
23381 055416 042712 000100 A0712: BIC #100,(R2) ;TURN OFF THE INTR ENABLE
23382 055422 104006 E0712: ERROR6 ;INTR OCCURRED WITH CPU AT LEVEL 5
23383 055424 055370 R0712 ;ERROR LOOP RETURN ADDRESS
23384
23385 055426 042712 000100 B0712: BIC #100,(R2) ;TURN OFF INTR ENABLE
23386 055432 012737 000066 000064 MOV #66,@#64 ;RESET THE TRAP CATCHER IN THE VECTOR
23387 055440 005037 000066 CLR @#66
23388 055444 010506 MOV R5,SP ;RESET SP JUST IN CASE
23389 055446 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0
23390
23391 055452 000004 00712: SCOPE ;CALL SCOPE LOOP UTILITY
23392
```

```
23393 ; *****  
23394 ; .SBTTL T0713 BR PRIORITY ARBITRATION TEST - LEVEL 4 USING DL11  
23395 ; *****  
23396  
23397 055454 012700 000713 T0713: MOV #0713,R0 ;LOAD R0 WITH TEST NO.  
23398 055460 013701 055522 MOV @#I0713,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
23399 055464 010605 MOV SP,R5 ;SAVE THE SP  
23400 055466 012702 177564 MOV #XCSR,R2 ;R2 POINTS TO DL11 XCSR  
23401 055472 012737 055534 000064 MOV #A0713,@#64 ;IF INTR OCCURS - GO TO A0713  
23402 055500 012737 000340 000066 MOV #340,@#66 ;WITH CPU PRIORITY AT LEVEL 7  
23403 055506 010506 R0713: MOV R5,SP ;RESET SP FOR ERROR LOOP  
23404 055510 005004 CLR R4 ;INITIALIZE R4 AS TIMER  
23405 055512 012737 000200 177776 MOV #200,@#PSW ;SET CPU PRIORITY TO LEVEL 4  
23406 055520 000257 CCC ;SCOPE SYNC  
23407  
23408 055522 052712 000100 I0713: BIS #100,(R2) ;ENABLE INTERRUPTS  
23409  
23410 055526 005304 DEC R4 ;COUNT UNTIL [R4] = 000000 - THEN  
23411 055530 001376 BNE .-2 ;CONTINUE - NO INTERRUPT SHOULD OCCUR  
23412 055532 000404 BR B0713 ;GO TO EXIT - ALL OK  
23413  
23414 055534 042712 000100 A0713: BIC #100,(R2) ;TURN OFF THE INTR ENABLE  
23415 055540 104006 E0713: ERROR6 ;INTR OCCURRED WITH CPU AT LEVEL 4  
23416 055542 055506 R0713 ;ERROR LOOP RETURN ADDRESS  
23417  
23418 055544 042712 000100 B0713: BIC #100,(R2) ;TURN OFF INTR ENABLE  
23419 055550 012737 000066 000064 MOV #66,@#64 ;RESET THE TRAP CATCHER IN THE VECTOR  
23420 055556 005037 000066 CLR @#66  
23421 055562 010506 MOV R5,SP ;RESET SP JUST IN CASE  
23422 055564 005037 177776 CLR @#PSW ;SET CPU PRIORITY BACK TO LEVEL 0  
23423  
23424 055570 000004 00713: SCOPE ;CALL SCOPE LOOP UTILITY  
23425  
23426  
23427
```

23428
23429
23430
23431
23432
23433
23434
23435
23436
23437
23438
23439
23440
23441
23442
23443
23444
23445
23446
23447
23448
23449
23450
23451
23452
23453
23454
23455
23456
23457
23458
23459
23460
23461
23462
23463
23464
23465
23466
23467
23468
23469
23470
23471
23472
23473
23474
23475

055572 012700 000714
055576 013701 055672
055602 005737 066636
055606 100053
055610 012702 177546
055614 010605
055616 012737 055700 000100
055624 012737 000300 000102
055632 010506
055634 005004
055636 005003
055640 012737 000340 177776
055646 052712 000100
055652 042712 000200
055656 105712
055660 100403
055662 005304
055664 001374
055666 000412
055670 000257
055672 005037 177776
055676 005103
055700 005012
055702 005703
055704 001406
055706 104006
055710 055632
055712 000403
055714 005012
055716 104006
055720 055632
055722 010506
055724 012737 000102 000100
055732 005037 000102
055736 000004

```
; *****  
; .SBTTL T0714 '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'  
  
T0714: MOV #0714,R0 ;LOAD R0 WITH THE TEST NO.  
MOV @#10714,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
TST @#OPTION ;IS THE KW11-L INSTALLED ??  
BPL 00714 ;SKIP THIS TEST IF NOT  
MOV #LKCSR,R2 ;R2 POINTS TO KW11-L CSR  
MOV SP,R5 ;SAVE THE SP  
MOV #A0714,@#100 ;SET UP LCLK VECTOR TO GO TO A0714  
MOV #300,@#102  
R0714: 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  
1$: TSTB (R2) ;LCLK READY TO INTR ??  
BMI 2$ ;BR IF YES  
DEC R4 ;COUNT THE TIMER  
BNE 1$ ;BR IF NO TIMEOUT  
BR B0714 ;GO REPORT TIMEOUT  
2$: CCC ;SCOPE SYNC  
  
I0714: CLR @#PSW ;ALLOW INTRS - LCLK SHOULD INTERRUPT  
;BEFORE FETCHING NEXT INSTRUCTION  
;SHOULD NOT BE FETCHED  
A0714: COM R3 ;DISABLE THE LCLK INTR  
CLR (R2) ;DID SOFTWARE FLAG GET SET ??  
TST R3 ;BR IF NOT - IT WORKED OK  
BEQ C0714 ;LCLK FAILED TO INTR ONTIME  
E10714: ERROR6 ;ERROR LOOP RETURN ADDRESS  
R0714 ;GO EXIT  
BR C0714  
  
B0714: CLR (R2) ;DISABLE LCLK INTR  
E20714: ERROR6 ;KW11-L TIMED OUT  
R0714 ;ERROR LOOP RETURN ADDRESS  
  
C0714: MOV R5,SP ;RESET THE SP  
MOV #102,@#100 ;RESTORE THE KW11-L TRAPCATCHER  
CLR @#102  
  
00714: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

T0714 'CLR @#PSW' ALLOWS IMMEDIATE BR-BG-INTR SEQUENCE

SEQ 0586

23476
23477
23478
23479
23480
23481
23482
23483
23484
23485
23486
23487
23488
23489
23490
23491
23492
23493
23494
23495
23496
23497
23498
23499
23500
23501
23502
23503
23504
23505
23506
23507
23508
23509
23510
23511
23512
23513
23514
23515
23516
23517
23518
23519
23520
23521
23522
23523
23524
23525
23526
23527
23528
23529
23530
23531

055740 012700 000715
055744 013701 056106
055750 005737 066636
055754 100133
055756 012702 177546
055762 012703 177564
055766 010605
055770 012737 056116 000100
055776 012737 000300 000102
056004 012737 056152 000064
056012 012737 000200 000066
056020 010506
056022 012737 000340 177776
056030 005037 067560
056034 005037 067564
056040 005004
056042 052713 000100
056046 105713
056050 100403
056052 005304
056054 001374
056056 000444
056060 005004
056062 052712 000100
056066 042712 000200
056072 105712
056074 100403
056076 005304
056100 001374
056102 000437
056104 000257
056106 005037 177776
056112 005137 067560
056116 005013
056120 005012
056122 005737 067560
056126 001403
056130 104007
056132 056020
056134 000426
056136 005737 067564
056142 001423

```
; *****  
; .SBTTL T0715 '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  
  
T0715:  MOV    #0715,R0          ;LOAD R0 WITH THE TEST NO.  
        MOV    @#I0715,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
        TST    @#OPTION        ;IS THE LCLK INSTALLED ??  
        BPL    00715          ;BR IF NOT - SKIP THIS TEST  
        MOV    #LKCSR,R2       ;R2 POINTS TO KW11-L CSR  
        MOV    #XCSR,R3        ;R3 POINTS TO DL11 XCSR  
        MOV    SP,R5           ;SAVE THE SP  
        MOV    #A0715,@#100     ;SET UP THE LCLK VECTOR - GO TO A0715  
        MOV    #300,@#102      ;  
        MOV    #C0715,@#64     ;SET UP THE DL11 VECTOR - GO TO C0715  
        MOV    #200,@#66      ;  
R0715:  MOV    R5,SP           ;RESET SP FOR ERROR LOOPING  
        MOV    #340,@#PSW      ;LOCK OUT ALL INTRs  
        CLR    @#MBUF0         ;INIT TIMER  
        CLR    @#MBUF1        ;CLEAR DL11 INTR FLAG  
        CLR    R4              ;INIT TIMER  
        BIS    #100,(R3)       ;ENABLE DL11 XMIT INTR  
1$:     TSTB   (R3)            ;XMIT READY SET ??  
        BMI    2$              ;BR IF YES  
        DEC    R4              ;COUNT THE TIMER  
        BNE    1$              ;BR IF NO TIMEOUT  
        BR    F0715           ;GO REPORT TIMEOUT FOR DL11  
  
2$:     CLR    R4              ;INIT THE TIMER AGAIN  
        BIS    #100,(R2)       ;ENABLE LCLK INTRs  
        BIC    #200,(R2)       ;CLEAR THE LINE CLOCK READY BIT  
3$:     TSTB   (R2)            ;LCLK READY TO INTR  
        BMI    4$              ;BR IF YES  
        DEC    R4              ;COUNT THE TIMER  
        BNE    3$              ;BR IF NO TIMEOUT  
        BR    G0715           ;GO REPORT KW11-L TIMEOUT  
4$:     CCC  
  
I0715:  CLR    @#PSW          ;ALLOW INTRs - KW SHOULD INTR FIRST  
  
A0715:  COM    @#MBUF0         ;SET SOFTWARE FLAG IF FETCHED  
        CLR    (R3)            ;DISABLE BOTH INTERRUPTS  
        CLR    (R2)            ;  
        TST    @#MBUF0        ;DID SOFTWARE FLAG GET SET ??  
        BEQ    B0715          ;BR IF NOT  
  
E10715: ERROR7  
        R0715  
        BR    H0715          ;KW11-L INTR OCCURRED TOO LATE  
;ERROR LOOP RETURN ADDRESS  
;GO TO EXIT  
  
B0715:  TST    @#MBUF1        ;DID DL11 SOFTWARE FLAG SET ??  
        BEQ    H0715          ;BR IF NOT
```



```
23564 : *****  
23565 : ////////////////COMBINED INSTRUCTION EXERCISER SECTION ////////////////  
23566 : *****  
23567 :  
23568 : *****  
23569 : .SBTTL T0716 'BPT' TRAP LINKAGE TEST  
23570 : *****  
23571 :  
23572 056246 012700 000716 T0716: MOV #0716,R0 ;SAVE THE TEST NO. IN R0  
23573 056252 013701 056272 MOV @#I0716,R1 ;LOAD INSTRUCTION TEST WORD INTO R1  
23574 056256 010605 MOV SP,R5 ;SAVE THE SP  
23575 056260 012737 056300 000014 MOV #A0716,@#14 ;GO TO A0716 ON 'BPT' TRAP  
23576 056266 010506 R0716: MOV R5,SP ;RESET THE SP FOR ERROR LOOPING  
23577 056270 000257 CCC ;SCOPE SYNC  
23578 :  
23579 056272 000003 I0716: BPT ;TEST THE 'BPT' - GO TO A0716  
23580 :  
23581 056274 104005 E0716: ERRORS ;BPT FAILED TO TRAP  
23582 056276 056266 R0716 ;ERROR LOOP RETURN ADDRESS  
23583 :  
23584 056300 010506 A0716: MOV R5,SP ;RESET THE SP  
23585 056302 012737 000016 000014 MOV #16,@#14 ;RESTORE THE VECTOR  
23586 :  
23587 056310 000004 00716: SCOPE ;CALL THE SCOPE LOOP UTILITY  
23588 :  
23589 :  
23590 : *****  
23591 : .SBTTL T0717 RED ZONE OVERFLOW TEST - MOV R,-(SP)  
23592 : *****  
23593 :  
23594 :MICROPROGRAMMING / LOGIC INFORMATION  
23595 :  
23596 :ROM SEQ: [174,257,200,JAMUPP,TRAP MICROROUTINE] FC 1,4,6,10  
23597 :  
23598 :ACT BUTS: 37[004]100,174 / 22[174]200,200 / 01[332]122,123 / 26[123]010,013  
23599 :  
23600 :EXEC: [200] BUS STOP TRIGGERS JAMUPP LOGIC  
23601 :  
23602 :CODES: N / A  
23603 :  
23604 :SYNC: B05J2 (-) T = 10 USEC  
23605 :  
23606 :KEY SIG: K5-5 STACK04 H / K5-5 STPM2 H / K1-7 BOVFLW STOP H / K4-4 OVFLW ER  
23607 :K4-4 BUS STOP H / K4-3 JBERR (1) L / K4-3 JAMUPP L / K4-3 JAM CLK  
23608 :  
23609 056312 012700 000717 T0717: MOV #0717,R0 ;LOAD R0 WITH TEST NO.  
23610 056316 013701 056356 MOV @#I0717,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
23611 056322 010605 MOV SP,R5 ;SAVE SP  
23612 056324 013704 000004 R0717: MOV @#4,R4 ;SAVE T.O. VECTOR  
23613 056330 013703 000336 MOV @#336,R3 ;SAVE VECTOR AT 336  
23614 056334 012737 056372 000004 MOV #A0717,@#4 ;GO TO A0717 ON OVFLW  
23615 056342 012737 125252 000336 MOV #125252,@#336 ;INIT. [336]  
23616 056350 012706 000340 MOV #340,SP ;SET SP TO CAUSE RED ZONE TRAP  
23617 056354 000257 CCC ;SCOPE SYNC  
23618 :  
23619 056356 010046 I0717: MOV R0,-(SP) ;FORCE RED ZONE TRAP - GO TO A0717
```

```

23620
23621 056360 010437 000004          MOV    R4,@#4      ;RESTORE T.O. VECTOR
23622 056364 010506          MOV    R5,SP      ;RESET SP FOR ERROR CALL
23623 056366 104005          E10717: ERROR5    ;MOV FAILED TO CAUSE TRAP
23624 056370 056324          R0717             ;ERROR LOOP RETURN ADDRESS
23625
23626 056372 010437 000004          A0717: MOV    R4,@#4      ;RESTORE T.O. VECTOR
23627 056376 022706 000000          CMP    #0,SP      ;[SP]=0?
23628 056402 001403          BEQ    B0717      ;BE IF YES
23629
23630 056404 010506          E20717: MOV    R5,SP      ;RESET SP FOR ERROR CALL
23631 056406 104005          ERROR5           ;SP NOT BEING JAMMED TO 4
23632 056410 056324          R0717             ;ERROR LOOP RETURN ADDRESS
23633
23634 056412 022737 125252 000336 B0717: CMP    #125252,@#336 ;DID PUSH OCCUR IN YELLOW ZONE?
23635 056420 001403          BEQ    C0717      ;BR IF NOT
23636
23637 056422 010506          E30717: MOV    R5,SP      ;RESET SP FOR ERROR CALL
23638 056424 104005          ERROR5           ;MOV PUSHED INTO YELLOW ZONE
23639 056426 056324          R0717             ;ERROR LOOP RETURN ADDRESS
23640
23641 056430 010337 000336          C0717: MOV    R3,@#336    ;RESTORE VECTOR 336
23642 056434 010506          MOV    R5,SP      ;RESET SP
23643
23644 056436 000004          00717: SCOPE      ;CALL THE SCOPE LOOP UTILITY
23645

```

```
23646 ; *****  
23647 ; .SBTTL T0720 YELLOW ZONE OVERFLOW TEST - MOV R,-(SP)  
23648 ; *****  
23649  
23650 ;MICROPROGRAMMING / LOGIC INFORMATION  
23651  
23652 ;ROM SEQ: [174,257,201,125,375,017,015,010,TRAP MICROROUTINE] FC 1,4,8,10,6  
23653  
23654 ;ACT BUTS: 37[004]100,174 / 22[174]200,201 / 16[125]016,017 / 26[017]010,010  
23655  
23656 ;EXEC: K4-4 CLK BOVFL SETS K5-4 BOVFLW FLOP  
23657  
23658 ;CODES: N/A  
23659  
23660 ;SYNC: B05J2 (-) T= 10USEC  
23661  
23662 ;KEY SIG: K4-4 CLK BOVFLW H / K4-4 CKOVF H / K1-7 BOVFL L  
23663  
23664 056440 012700 000720 T0720: MOV #0720,R0 ;LOAD R0 WITH TEST NO.  
23665 056444 013701 056476 MOV @#I0720,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
23666 056450 010605 MOV SP,R5 ;SAVE SP  
23667 056452 012702 000376 MOV #376,R2 ;R2 POINTS TO STACK  
23668 056456 013704 000004 R0720: MOV @#4,R4 ;SAVE T.O. VECTOR  
23669 056462 012737 056512 000004 MOV #A0720,@#4 ;ON OVFLW - GO TO A0720  
23670 056470 012706 000400 MOV #400,SP ;SET SP TO CAUSE OVFLW  
23671 056474 000257 CCC ;SCOPE SYNC  
23672  
23673 056476 010046 I0720: MOV R0,-(SP) ;FORCE STACK OVFLW - GO TO A0720  
23674  
23675 056500 010437 000004 MOV R4,@#4 ;RESTORE T.O. VECTOR  
23676 056504 010506 MOV R5,SP ;RESET SP FOR ERROR CALL  
23677 056506 104005 E10720: ERROR5 ;STACK OVFLW FAILED TO TRAP  
23678 056510 056456 R0720 ;ERROR LOOP RETURN ADDRESS  
23679  
23680 056512 010437 000004 A0720: MOV R4,@#4 ;RESTORE T.O. VECTOR  
23681 056516 020012 CMP R0,(R2) ;DID [R0] GET PUSHED?  
23682 056520 001403 BEQ B0720 ;BR IF YES  
23683  
23684 056522 010506 E20720: MOV R5,SP ;RESET SP FOR ERROR CALL  
23685 056524 104005 ERROR5 ;MOV FAILED TO PUSH IN YELLOW ZONE  
23686 056526 056456 R0720 ;ERROR LOOP RETURN ADDRESS  
23687  
23688 056530 005706 B0720: TST SP ;[SP]=0?  
23689 056532 001003 BNE C0720 ;BR IF NOT  
23690  
23691 056534 010506 E30720: MOV R5,SP ;RESET SP FOR ERROR CALL  
23692 056536 104005 ERROR5 ;RED ZONE INSTEAD OF YELLOW ZONE  
23693 056540 056456 R0720 ;ERROR LOOP RETURN ADDRESS  
23694  
23695 056542 010506 C0720: MOV R5,SP ;RESET SP  
23696  
23697 056544 000004 O0720: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

```

23698 ; *****
23699 ; .SBTTL T0721 YELLOW ZONE OVERFLOW TEST - (CMP R0,-(SP))
23700 ; *****
23701
23702 056546 012700 000721 T0721: MOV #0721,R0 ;LOAD R0 WITH TEST NO.
23703 056552 013701 056600 MOV @#10721,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23704 056556 010605 MOV SP,R5 ;SAVE THE SP
23705 056560 013704 000004 MOV @#4,R4 ;SAVE TRAP VECTOR
23706 056564 012737 056604 000004 R0721: MOV #A0721,@#4 ;GO TO A0721 IF TRAP SPRUNG
23707 056572 012706 000400 MOV #400,SP ;SET SP TO PUSH INTO 'YELLOW ZONE'
23708 056576 000257 CCC ;SCOPE SYNC
23709
23710 056600 020046 I0721: CMP R0,-(SP) ;TEST THE CMP - NO TRAP SHOULD OCCUR
23711
23712 056602 000405 BR B0721 ;GO TO EXIT TEST
23713
23714 056604 010437 000004 A0721: MOV R4,@#4 ;RESTORE TRAP VECTOR
23715 056610 010506 MOV R5,SP ;RESET THE SP
23716 056612 104005 E0721: ERRORS ;CMP CAUSED OVERFLOW TRAP
23717 056614 056564 R0721 ;ERROR LOOP RETURN ADDRESS
23718
23719 056616 010437 000004 B0721: MOV R4,@#4 ;RESTORE THE VECTOR
23720 056622 010506 MOV R5,SP ;RESET THE SP
23721
23722 056624 000004 00721: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

23723 ; *****
23724 ; .SBTTL T0722 YELLOW ZONE OVERFLOW TEST - (BIT R0,-(SP))
23725 ; *****
23726
23727 056626 012700 000722 T0722: MOV #0722,R0 ;LOAD R0 WITH TEST NO.
23728 056632 013701 056660 MOV @#I0722,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23729 056636 010605 MOV SP,R5 ;SAVE THE SP
23730 056640 013704 000004 MOV @#4,R4 ;SAVE TRAP VECTOR
23731 056644 012737 056664 000004 R0722: MOV #A0722,@#4 ;GO TO A0722 IF TRAP SPRUNG
23732 056652 012706 000400 MOV #400,SP ;SET SP TO PUSH INTO 'YELLOW ZONE'
23733 056656 000257 CCC ;SCOPE SYNC
23734
23735 056660 030046 I0722: BIT R0,-(SP) ;TEST THE BIT - NO TRAP SHOULD OCCUR
23736
23737 056662 000405 BR B0722 ;GO TO EXIT TEST
23738
23739 056664 010437 000004 A0722: MOV R4,@#4 ;RESTORE TRAP VECTOR
23740 056670 010506 MOV R5,SP ;RESET THE SP
23741 056672 104005 E0722: ERROR5 ;BIT CAUSED OVERFLOW TRAP
23742 056674 056644 R0722 ;ERROR LOOP RETURN ADDRESS
23743
23744 056676 010437 000004 B0722: MOV R4,@#4 ;RESTORE THE VECTOR
23745 056702 010506 MOV R5,SP ;RESET THE SP
23746
23747 056704 000004 O0722: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

23748 ; *****
23749 ; .SBTTL T0723 YELLOW ZONE OVERFLOW TEST - (TST -(SP))
23750 ; *****
23751
23752 056706 012700 000723 T0723: MOV #0723,R0 ;LOAD R0 WITH TEST NO.
23753 056712 013701 056740 MOV @#10723,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
23754 056716 010605 MOV SP,R5 ;SAVE THE SP
23755 056720 013704 000004 MOV @#4,R4 ;SAVE TRAP VECTOR
23756 056724 012737 056744 000004 R0723: MOV #A0723,@#4 ;GO TO A0723 IF TRAP SPRUNG
23757 056732 012706 000400 MOV #400,SP ;SET SP TO PUSH INTO 'YELLOW ZONE'
23758 056736 000257 CCC ;SCOPE SYNC
23759
23760 056740 005746 I0723: TST -(SP) ;TEST THE TST - NO TRAP SHOULD OCCUR
23761
23762 056742 000405 BR B0723 ;GO TO EXIT TEST
23763
23764 056744 010437 000004 A0723: MOV R4,@#4 ;RESTORE TRAP VECTOR
23765 056750 010506 MOV R5,SP ;RESET THE SP
23766 056752 104005 E0723: ERROR5 ;TST CAUSED OVERFLOW TRAP
23767 056754 056724 R0723 ;ERROR LOOP RETURN ADDRESS
23768
23769 056756 010437 000004 B0723: MOV R4,@#4 ;RESTORE THE VECTOR
23770 056762 010506 MOV R5,SP ;RESET THE SP
23771
23772 056764 000004 O0723: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

23773
23774
23775
23776
23777
23778
23779
23780
23781
23782
23783
23784
23785
23786
23787
23788
23789
23790
23791
23792
23793
23794
23795
23796
23797
23798
23799
23800
23801
23802
23803
23804
23805
23806
23807
23808
23809
23810
23811

056766 012700 000724
056772 013701 057022
056776 010605
057000 013704 000004
057004 012737 057034 000004
057012 010506
057014 012702 000001
057020 000257
057022 160012
057024 010437 000004
057030 104005
057032 057000
057034 010437 000004
057040 010506
057042 005037 000000
057046 000004

```

; *****
; .SBTTL T0724 ODD ADDRESS ERROR TEST - SUB RA,(RB) - (RB) = ODD
; *****
;MICROPROGRAMMING / LOGIC INFORMATION
;ROM SEQ:
;ACT BUTS:
;EXEC:
;CODES:
;SYNC:
;KEY SIG:
T0724: MOV #0724,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0724,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE SP
R0724: MOV @#4,R4 ;SAVE T.O. VECTOR
MOV #A0724,@#4 ;ON ODD ADDR ERROR - GO TO A0724
MOV R5,SP ;RESET SP FOR ERROR LOOP
MOV #1,R2 ;R2 GETS ODD ADDRESS
CCC ;SCOPE SYNC
I0724: SUB R0,(R2) ;FORCE ODD ADDR ERROR - GO TO A0724
E0724: MOV R4,@#4 ;RESTORE T.O. VECTOR
ERROR5 ;ODD ADDR FAILED TO TRAP
R0724 ;ERROR LOOP RETURN ADDRESS
A0724: MOV R4,@#4 ;RESTORE T.O. VECTOR
MOV R5,SP ;RESET SP
CLR @#0 ;CLR LOC. 0 JUST IN CASE
00724: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```
23812 ; *****  
23813 ; .SBTTL T0725 TEST FOR ODD ADDR. ERROR TRAP FOR DEST. DEFERRED MODES  
23814 ; *****  
23815  
23816 057050 012700 000725 T0725: MOV #0725,R0 ;LOAD R0 WITH TEST NO.  
23817 057054 012702 067565 MOV #MBUF1+1,R2 ;DEST ADDR=MBUF1+1 (ODD)  
23818 057060 012737 057154 000004 MOV #A0725,@#4 ;GO TO A0725 ON ODA TRAP  
23819  
23820 057066 010205 R10725: MOV R2,R5 ;[R5] = DEST. ADDR  
23821 057070 013701 057076 MOV @#I10725,R1 ;[R1] = TEST INSTR  
23822 057074 000257 CCC ;SCOPE SYNC  
23823  
23824 057076 105435 I10725: NEGB @ (R5)+ ;TEST DM=3 TRAP  
23825  
23826 057100 104006 E10725: ERROR6 ;ODA TRAP NOT SPRUNG IN ROM LOC. 163  
23827 057102 057066 R10725 ;ERROR LOOP RETURN ADDRESS  
23828  
23829 057104 012705 067567 R20725: MOV #MBUF1+3,R5 ;[R5] = DEST. ADDR  
23830 057110 013701 057116 MOV @#I20725,R1 ;[R1] = TEST INSTR  
23831 057114 000257 CCC ;SCOPE SYNC  
23832  
23833 057116 105455 I20725: NEGB @-(R5) ;TEST DM=5 TRAP  
23834  
23835 057120 104006 E20725: ERROR6 ;ODA TRAP NOT SPRUNG IN ROM LOC. 165  
23836 057122 057104 R20725 ;ERROR LOOP RETURN ADDRESS  
23837  
23838 057124 010205 R30725: MOV R2,R5 ;[R5] = DEST ADDR  
23839 057126 013701 057134 MOV @#I30725,R1 ;[R1] = TEST INSTR  
23840 057132 000257 CCC ;SCOPE SYNC  
23841  
23842 057134 105475 000000 I30725: NEGB @0(R5) ;TEST DM=7 TRAP  
23843  
23844 057140 104006 E30725: ERROR6 ;ODA TRAP NOT SPRUNG IN ROM LOC 263  
23845 057142 057124 R30725 ;ERROR LOOP RETURN ADDRESS  
23846  
23847 057144 012737 065160 000004 MOV #BERR,@#4 ;RESET T.O. VECTOR  
23848 057152 000403 BR 00725 ;GO TO SCOPE EXIT  
23849  
23850 057154 062716 000004 A0725: ADD #4,(SP) ;MOV RETURN PC AROUND ERROR CALL  
23851 057160 000002 RTI ;RETURN TO NEXT SUB-TEST  
23852  
23853 057162 000004 00725: SCOPE ;CALL SCOPE LOOP UTILITY  
23854
```

```
23855 ; *****  
23856 ; .SBTTL T0726 TEST FOR ODD ADDR ERROR TRAP FOR SOURCE DEFERRED MODES  
23857 ; *****  
23858  
23859 057164 012700 000726 T0726: MOV #0726,R0 ;LOAD R0 WITH TEST NO.  
23860 057170 012702 067565 MOV #MBUF1+1,R2 ;[R2] = SOURCE ADDR. (ODD)  
23861 057174 012737 057270 000004 MOV #A0726,@#4 ;GO TO A0726 ON TRAP  
23862  
23863 057202 010205 R10726: MOV R2,R5 ;[R5] = SOURCE ADDR.  
23864 057204 013701 057212 MOV @#I10726,R1 ;[R1] = TEST INSTR.  
23865 057210 000257 CCC ;SCOPE SYNC  
23866  
23867 057212 113504 I10726: MOVB @ (R5)+,R4 ;TEST SM=3  
23868  
23869 057214 104006 E10726: ERROR6 ;ODA TRAP NOT SPRUNG IN ROM LOC. 143  
23870 057216 057202 R10726 ;ERROR LOOP RETURN ADDRESS  
23871  
23872 057220 012705 067567 R20726: MOV #MBUF1+3,R5 ;[R5] = SOURCE ADDR  
23873 057224 013701 057232 MOV @#I20726,R1 ;[R1] = TEST INSTR  
23874 057230 000257 CCC ;SCOPE SYNC  
23875  
23876 057232 115504 I20726: MOVB @-(R5),R4 ;TEST SM=5  
23877  
23878 057234 104006 E20726: ERROR6 ;ODA TRAP NOT SPRUNG IN ROM LOC 145  
23879 057236 057220 R20726 ;ERROR LOOP RETURN ADDRESS  
23880 057240 010205 R30726: MOV R2,R5 ;[R5] = SOURCE ADDR  
23881 057242 013701 057250 MOV @#I30726,R1 ;[R1] = TEST INSTR  
23882 057246 000257 CCC ;SCOPE SYNC  
23883  
23884 057250 117504 000000 I30726: MOVB @0(R5),R4 ;TEST SM=7  
23885  
23886 057254 104006 E30726: ERROR6 ;ODA TRAP NOT SPRUNG IN ROM LOC 244  
23887 057256 057240 R30726 ;ERROR LOOP RETURN ADDRESS  
23888  
23889 057260 012737 065160 000004 MOV #BERR,@#4 ;RESET T.O. VECTOR  
23890 057266 000403 BR 00726 ;GO TO SCOPE EXIT  
23891  
23892 057270 062716 000004 A0726: ADD #4,(SP) ;MOVE RETURN PC AROUND ERROR CALL  
23893 057274 000002 RTI ;RETURN TO NEXT SUB-TEST  
23894  
23895 057276 000004 00726: SCOPE ;CALL SCOPE LOOP UTILITY  
23896  
23897  
23898
```

```

23899 ; *****
23900 ; .SBTTL T0727 TEST FOR ODD ADDR ERROR TRAP FOR JMP DEST DEFERRED MODES
23901 ; *****
23902
23903 057300 012700 000727 T0727: MOV #0727,R0 ;LOAD R0 WITH TEST NO.
23904 057304 012702 057407 MOV #B0727+3,R2 ;DEST ADDR = B0727+3 (ODD)
23905 057310 012737 057412 000004 MOV #A0727,@#4 ;GO TO A0727 ON ODA TRAP
23906
23907 057316 010205 R10727: MOV R2,R5 ;[R5] = DEST ADDR
23908 057320 013701 057326 MOV @#I10727,R1 ;[R1] = TEST INSTR
23909 057324 000257 CCC ;SCOPE SYNC
23910
23911 057326 000135 I10727: JMP @-(R5)+ ;TEST JMP DM=3
23912
23913 057330 104006 E10727: ERROR6 ;ODA TRAP NOT SPRUNG IN ROM LOC 153
23914 057332 057316 R10727 ;ERROR LOOP RETURN ADDRESS
23915
23916 057334 012705 057407 R20727: MOV #B0727+3,R5 ;[R5] = DEST ADDR
23917 057340 013701 057346 MOV @#I20727,R1 ;[R1] = TEST INSTR
23918 057344 000257 CCC ;SCOPE SYNC
23919
23920 057346 000155 I20727: JMP @-(R5) ;TEST JMP DM=5
23921
23922 057350 104006 E20727: ERROR6 ;ODA TRAP NOT SPRUNG IN ROM LOC 155
23923 057352 057334 R20727 ;ERROR LOOP RETURN ADDRESS
23924
23925 057354 010205 R30727: MOV R2,R5 ;[R5] = DEST ADDR
23926 057356 013701 057364 MOV @#I30727,R1 ;[R1] = TEST INSTR
23927 057362 000257 CCC ;SCOPE SYNC
23928
23929 057364 000175 000000 I30727: JMP @0(R5) ;TEST JMP DM=7
23930
23931 057370 104006 E30727: ERROR6 ;ODA TRAP NOT SPRUNG IN LOC 302
23932 057372 057354 R30727 ;ERROR LOOP RETURN ADDRESS
23933
23934 057374 012737 065160 000004 MOV #BERR,@#4 ;RESET BUS T.O. VECTOR
23935 057402 000421 BR 00727 ;GO TO SCOPE EXIT
23936
23937 057404 000000 B0727: HALT ;CATASTOPHIC ERROR - [PC] QUESTIONABLE.
23938 057406 000000 HALT ;RESTART PROGRAM - DO NOT CONTINUE.
23939 057410 000000 HALT
23940
23941 057412 032716 000001 A0727: BIT #1,(SP) ;TRAP DUE TO ODD PC?
23942 057416 001003 BNE C0727 ;BR IF YES
23943 057420 062716 000004 ADD #4,(SP) ;MOV RETURN PC AROUND ERROR CALL
23944 057424 000002 RTI ;RETURN TO NEXT SUB TEST
23945
23946 057426 011603 C0727: MOV (SP),R3 ;GET ODD PC OFF STACK INTO R3
23947 057430 062706 000004 ADD #4,SP ;FIX SP
23948
23949 057434 104007 E40727: ERROR7 ;PC TRAPPED WITH ODD ADDRESS
23950 057436 057300 T0727
23951
23952 057440 012737 065160 000004 MOV #BERR,@#4 ;RESET T.O. VECTOR
23953
23954 057446 000004 00727: SCOPE ;CALL SCOPE LOOP UTILITY

```

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 598
CBQEAC.P11 03-JUL-80 08:05

E 15

T0727 TEST FOR ODD ADDR ERROR TRAP FOR JMP DEST DEFERRED MODES

SEQ 0598

23955

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 599
 CBQEAC.P11 03-JUL-80 08:05

T0727 TEST FOR ODD ADDR ERROR TRAP FOR JMP DEST DEFERRED MODES

SEQ 0599

```

23956 ; *****
23957 ; .SBTTL T0730 TEST FOR STACK OFLW FOR DEST MODES 1,2,4, AND 6.
23958 ; *****
23959
23960 057450 012700 000730 T0730: MOV #0730,R0 ;LOAD R0 WITH TEST NO.
23961 057454 012737 057602 000004 MOV #A0730,@#4 ;GO TO A0730 ON OVFLW TRAP
23962 057462 010605 MOV SP,R5 ;SAVE SP
23963 057464 012702 000376 MOV #376,R2 ;USE R2 TO SET UP SP TO CAUSE TRAP
23964
23965 057470 013701 057500 R10730: MOV @#I10730,R1 ;[R1] = TEST INSTR.
23966 057474 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
23967 057476 000257 CCC ;SCOPE SYNC
23968
23969 057500 005016 I10730: CLR (SP) ;TEST DM1 - SHOULD SPRING TRAP
23970
23971 057502 010506 MOV R5,SP ;RESET SP
23972 057504 104006 E10730: ERROR6 ;DM1 FAILED TO CAUSE OVERFLOW TRAP
23973 057506 057470 R10730 ;ERROR LOOP RETURN ADDRESS
23974
23975 057510 013701 057520 R20730: MOV @#I20730,R1 ;[R1] = TEST INSTR.
23976 057514 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
23977 057516 000257 CCC ;SCOPE SYNC
23978
23979 057520 005026 I20730: CLR (SP)+ ;TEST DM2 - SHOULD SPRING TRAP
23980
23981 057522 010506 MOV R5,SP ;RESET SP
23982 057524 104006 E20730: ERROR6 ;DM2 FAILED TO CAUSE OVERFLOW TRAP
23983 057526 057510 R20730 ;ERROR LOOP RETURN ADDRESS
23984
23985 057530 013701 057540 R30730: MOV @#I30730,R1 ;[R1] = TEST INSTR.
23986 057534 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
23987 057536 000257 CCC ;SCOPE SYNC
23988
23989 057540 005046 I30730: CLR -(SP) ;TEST DM4 - SHOULD SPRING TRAP
23990
23991 057542 010506 MOV R5,SP ;RESET SP
23992 057544 104006 E30730: ERROR6 ;DM4 FAILED TO CAUSE OVERFLOW TRAP
23993 057546 057530 R30730 ;ERROR LOOP RETURN ADDRESS
23994
23995 057550 013701 057560 R40730: MOV @#I40730,R1 ;[R1] = TEST INSTR.
23996
23997 057554 010206 MOV R2,SP ;SET SP TO CAUSE ERROR
23998 057556 000257 CCC ;SCOPE SYNC
23999
24000 057560 005066 000000 I40730: CLR 0(SP) ;TEST DM6 - SHOULD SPRING TRAP
24001
24002 057564 010506 MOV R5,SP ;RESET SP
24003 057566 104006 E40730: ERROR6 ;DM6 FAILED TO CAUSE OVERFLOW TRAP
24004 057570 057550 R40730 ;ERROR LOOP RETURN ADDRESS
24005
24006 057572 012737 065160 000004 MOV #BERR,@#4 ;RESET BUS T.O. VECTOR
24007 057600 000407 BR 00730 ;GO TO SCOPE EXIT
24008
24009 057602 011604 A0730: MOV (SP),R4 ;GET RETURN PC OFF STACK
24010 057604 062704 ADD #6,R4 ;MOVE RETURN PC AROUND ERROR CALL
24011 057610 010506 MOV R5,SP ;RESET SP

```

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 600
CBQEAC.P11 03-JUL-80 08:05

T0730 TEST FOR STACK OFLW FOR DEST MODES 1,2,4, AND 6.

SEQ 0600

24012	057612	005046	CLR	-(SP)	:PUSH NEW PS ON STACK
24013	057614	010446	MOV	R4,-(SP)	:PUSH RETURN PC ON STACK
24014	057616	000002	RTI		:RETURN TO NEXT SUB-TEST
24015					
24016	057620	000004	00730:	SCOPE	:CALL SCOPE LOOP UTILITY
24017					

```

24018 ; *****
24019 ; .SBTTL T0731 TEST FOR STACK OVFLW FOR MOV DEST MODES 1,2,4, AND 6.
24020 ; *****
24021
24022 057622 012700 000731 T0731: MOV #0731,R0 ;LOAD R0 WITH TEST NO.
24023 057626 012737 057754 000004 MOV #A0731,@#4 ;GO TO A0731 ON STACK OVFLW TRAP
24024 057634 010605 MOV SP,R5 ;SAVE SP
24025 057636 012702 000376 MOV #376,R2 ;USE R2 TO SET UP SP TO CAUSE TRAP
24026
24027 057642 013701 057652 R10731: MOV @#I10731,R1 ;[R1] = TEST INSTR.
24028 057646 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
24029 057650 000257 CCC ;SCOPE SYNC
24030
24031 057652 010016 I10731: MOV R0,(SP) ;TEST MOV DM1 - SHOULD SPRING TRAP
24032
24033 057654 010506 MOV R5,SP ;RESET SP
24034 057656 104006 E10731: ERROR6 ;MOV DM1 FAILED TO SPRING TRAP
24035 057660 057642 R10731 ;ERROR LOOP RETURN ADDRESS
24036
24037 057662 013701 057672 R20731: MOV @#I20731,R1 ;[R1] = TEST INSTR.
24038 057666 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
24039 057670 000257 CCC ;SCOPE SYNC
24040
24041 057672 010026 I20731: MOV R0,(SP)+ ;TEST MOV DM2 - SHOULD SPRING TRAP
24042
24043 057674 010506 MOV R5,SP ;RESET SP
24044 057676 104006 E20731: ERROR6 ;MOV DM2 FAILED TO SPRING TRAP
24045 057700 057662 R20731 ;ERROR LOOP RETURN ADDRESS
24046
24047 057702 013701 057712 R30731: MOV @#I30731,R1 ;[R1] = TEST INSTR.
24048 057706 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
24049 057710 000257 CCC ;SCOPE SYNC
24050
24051 057712 010046 I30731: MOV R0,-(SP) ;TEST MOV DM4 - SHOULD SPRING TRAP
24052
24053 057714 010506 MOV R5,SP ;RESET SP
24054 057716 104006 E30731: ERROR6 ;MOV DM4 FAILED TO SPRING TRAP
24055 057720 057702 R30731 ;ERROR LOOP RETURN ADDRESS
24056
24057 057722 013701 057732 R40731: MOV @#I40731,R1 ;[R1] = TEST INSTR.
24058 057726 010206 MOV R2,SP ;SET UP SP TO CAUSE OVERFLOW
24059 057730 000257 CCC ;SCOPE SYNC
24060
24061 057732 010066 000000 I40731: MOV R0,0(SP) ;TEST MOV DM6 - SHOULD SPRING TRAP
24062
24063 057736 010506 MOV R5,SP ;RESET SP
24064 057740 104006 E40731: ERROR6 ;MOV DM6 FAILED TO CAUSE OVFLW TRAP
24065 057742 057722 R40731 ;ERROR LOOP RETURN ADDRESS
24066
24067 057744 012737 065160 000004 MOV #BERR,@#4 ;RESET T.O. VECTOR
24068 057752 000407 BR 00731 ;GO TO SCOPE EXIT
24069
24070 057754 011604 A0731: MOV (SP),R4 ;GET RETURN PC
24071 057756 062704 000006 ADD #6,R4 ;MOVE RETURN PC AROUND ERROR CALL
24072 057762 010506 MOV R5,SP ;RESET SP
24073 057764 005046 CLR -(SP) ;PUSH NEW PSW
    
```

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 602
CBQEAC.P11 03-JUL-80 08:05

T0731 TEST FOR STACK OVFLW FOR MOV DEST MODES 1,2,4, AND 6.

SEQ 0602

24074 057766 010446
24075 057770 000002
24076
24077 057772 000004
24078

MOV R4,-(SP) ;PUSH RETURN PC
RTI ;RETURN TO NEXT SUB-TEST
00731: SCOPE ;CALL SCOPE LOOP UTILITY

```

24079 ; *****
24080 ; .SBTTL T0732 TEST THAT JSR CAN CAUSE OVERFLOW TRAP
24081 ; *****
24082
24083 057774 012700 000732 T0732: MOV #0732,R0 ;LOAD R0 WITH TEST NO.
24084 060000 012737 060036 000004 MOV #A0732,@#4 ;GO TO A0732 ON OVERFLOW ERROR
24085 060006 010605 MOV SP,R5 ;SAVE SP
24086 060010 013701 060022 MOV @#I0732,R1 ;LOAD R1 WITH TEST INSTR WORD
24087 060014 012706 000400 R0732: MOV #400,SP ;SET THE SP TO CAUSE TRAP
24088 060020 000257 CCC ;SCOPE SYNC
24089
24090 060022 004737 060042 I0732: JSR PC,@#B0732 ;TEST JSR - SHOULD SPRING TRAP
24091
24092 060026 010506 MOV R5,SP ;RESET SP
24093 060030 104005 E10732: ERROR5 ;JSR PUSH DID NOT SPRING OVFL TRAP
24094 060032 060014 R0732 ;ERROR LOOP RETURN ADDRESS
24095
24096 060034 000405 BR C0732 ;GO TO SCOPE EXIT
24097
24098 060036 010506 A0732: MOV R5,SP ;RESET SP
24099 060040 000403 BR C0732 ;GO EXIT TEST - ALL OK
24100
24101 060042 010506 B0732: MOV R5,SP ;RESET SP
24102 060044 104005 E20732: ERROR5 ;JSR PUSH FAILED TO SPRING OVFLW TRAP
24103 060046 060014 R0732 ;ERROR LOOP RETURN ADDRESS
24104
24105 060050 012737 065160 000004 C0732: MOV #BERR,@#4 ;RESET BUS T.O. VECTOR
24106
24107 060056 000004 00732: SCOPE ;CALL SCOPE LOOP UTILITY
24108

```

```

24109 ; *****
24110 ; .SBTTL T0733 TEST THAT 1ST PUSH IN TRAP MICROROUTINE CAUSES OVFLW TRAP
24111 ; *****
24112
24113 060060 012700 000733 T0733: MOV #0733,R0 ;LOAD R0 WITH TEST NO.
24114 060064 013704 000014 MOV @#14,R4 ;SAVE BREAK POINT TRAP VECTOR
24115 060070 013701 060120 MOV @#I0733,R1 ;LOAD R1 WITH TEST INSTR
24116 060074 010605 MOV SP,R5 ;SAVE SP
24117 060076 012737 060132 000004 MOV #A0733,@#4 ;GO TO A0733 ON OVFLW TRAP
24118 060104 012737 060136 000014 MOV #B0733,@#14 ;GO TO B0733 IF BPT SERVICED
24119 060112 012706 000400 R0733: MOV #400,SP ;SET UP SP TO CAUSE OVFLW ON 1ST PUSH
24120 060116 000257 CCC ;SCOPE SYNC
24121
24122 060120 000003 I0733: BPT ;TEST THE BPT - SHOULD CAUSE OVERFLOW TRAP
24123
24124 060122 010506 MOV R5,SP ;RESET SP
24125 060124 104005 E10733: ERROR5 ;BPT FAILED TO TRAP
24126 060126 060112 R0733 ;ERROR LOOP RETURN ADDRESS
24127
24128 060130 000405 BR C0733 ;GO TO SCOPE EXIT
24129
24130 060132 010506 A0733: MOV R5,SP ;RESET SP
24131 060134 000403 BR C0733 ;GO EXIT - ALL OK
24132
24133 060136 010506 B0733: MOV R5,SP ;RESET SP
24134 060140 104005 E20733: ERROR5 ;OVFLW TRAP FAILED TO BUMP BPT SERVICE
24135 060142 060112 R0733 ;ERROR LOOP RETURN ADDRESS
24136
24137 060144 012737 065160 000004 C0733: MOV #BERR,@#4 ;RESET VECTORS
24138 060152 010437 000014 MOV R4,@#14
24139
24140 060156 000004 O0733: SCOPE ;CALL SCOPE LOOP UTILITY
24141

```

```

24142 ; *****
24143 ; .SBTTL T0734 TEST THAT 2ND PUSH IN TRAP MICROUTINE CAUSES OVFLW TRAP
24144 ; *****
24145
24146 060160 012700 000734 T0734: MOV #0734,R0 ;LOAD R0 WITH TEST NO.
24147 060164 013701 060220 MOV @#I0734,R1 ;LOAD R1 WITH TEST INSTR WORD
24148 060170 013704 000014 MOV @#14,R4 ;SAVE BPT VECTOR
24149 060174 010605 MOV SP,R5 ;SAVE SP
24150 060176 012737 060232 000004 MOV #A0734,@#4 ;GO TO A0734 ON STACK OVFLOW
24151 060204 012737 060236 000014 MOV #B0734,@#14 ;GO TO B0734 IF BPT SERVICED
24152 060212 012706 000402 R0734: MOV #402,SP ;SET SP TO CAUSE TRAP ON 2ND PUSH
24153 060216 000257 CCC ;SCOPE SYNC
24154
24155 060220 000003 I0734: BPT ;TEST THE BPT - SHOULD CAUSE OVERFLOW TRAP
24156
24157 060222 010506 MOV R5,SP ;RESET SP
24158 060224 104005 E10734: ERRORS R0734 ;BPT FAILED TO TRAP
24159 060226 060212 ;ERROR LOOP RETURN ADDRESS
24160
24161 060230 000405 BR C0734 ;GO TO SCOPE EXIT
24162
24163 060232 010506 A0734: MOV R5,SP ;RESET SP
24164 060234 000403 BR C0734 ;GO EXIT - ALL OK
24165
24166 060236 010506 B0734: MOV R5,SP ;RESET SP
24167 060240 104005 E20734: ERRORS R0734 ;OVFLW TRAP FAILED TO BUMP BPT SERVICE
24168 060242 060212 ;ERROR LOOP RETURN ADDRESS
24169
24170 060244 012737 065160 000004 C0734: MOV #BERR,@#4 ;RESET VECTORS
24171 060252 010437 000014 MOV R4,@#14
24172
24173 060256 000004 00734: SCOPE ;CALL SCOPE LOOP UTILITY

```

24174
24175
24176
24177
24178
24179
24180
24181
24182
24183
24184
24185
24186
24187
24188
24189
24190
24191
24192
24193
24194
24195
24196
24197
24198
24199
24200
24201
24202
24203
24204
24205
24206
24207
24208
24209
24210
24211

060260 012700 000735
060264 013701 060314
060270 010605
060272 013704 000004
060276 012737 060326 000004
060304 010506
060306 012702 060322
060312 000257
060314 004302
060316 010437 000004
060322 104005
060324 060272
060326 010437 000004
060332 010506
060334 000004

```

; *****
; .SBTTL T0735 ILLEGAL INSTRUCTION TEST - JSR RN,%R
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ:      [150,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6,10
;ACT BUTS:     37[004]100,150 / 01[332]122,123 / 26[123]010,013
;EXEC:         [115] D = NEW PSW / [113] D = OLD PSW / [331] D = #I0735:2 / [333]
;CODES:        [140] SPS=7 / N:C = [LOC 6]
;SYNC:         B05J2 (-) T = 5.8 USEC
;KEY SIG:      K3-6 ILL INSTR L / K5-5 STPM3 H / K3-3 DM=0L / K3-5 JMP+JSR H

T0735:  MOV    #0735,R0          ;LOAD R0 WITH TEST NO.
        MOV    @#I0735,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD
        MOV    SP,R5           ;SAVE SP
R0735:  MOV    @#4,R4           ;SAVE T.O. VECTOR
        MOV    #A0735,@#4      ;ILLEGAL INSTR. TRAP GOES TO A0735
        MOV    R5,SP           ;RESET SP FOR ERROR LOOP
        MOV    #E0735,R2      ;IN CASE JSR JUMPS TO [R2]
        CCC                   ;SCOPE SYNC

I0735:  JSR    R3,R2           ;JSR MODE 0 FORCES TRAP - GO TO A0735

E0735:  MOV    R4,@#4          ;RESTORE T.O. VECTOR
        ERRORS R0735          ;JSR FAILED TO SPRING TRAP
        R0735                 ;ERROR LOOP RETURN ADDRESS

A0735:  MOV    R4,@#4          ;RESTORE VECTOR
        MOV    R5,SP           ;RESET SP

00735:  SCOPE                  ;CALL THE SCOPE LOOP UTILITY

```

24212
24213
24214
24215
24216
24217
24218
24219
24220
24221
24222
24223
24224
24225
24226
24227
24228
24229
24230
24231
24232
24233
24234
24235
24236
24237
24238
24239
24240
24241
24242
24243
24244
24245
24246
24247
24248

060336 012700 000736
060342 013701 060372
060346 010605
060350 013704 000004
060354 012737 060404 000004
060362 010506
060364 012702 060400
060370 000257
060372 000102
060374 010437 000004
060400 104005
060402 060350
060404 010437 000004
060410 010506
060412 000004

```
; *****  
; .SBTTL T0736 ILLEGAL INSTRUCTION TEST - JMP %R  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [150,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6,10  
;ACT BUTS: 37[004]100,150 / 01[332]122,123 / 26[123]010,013  
;EXEC: [115] D = NEW PSW / [113] D = OLD PSW / [311] D = #I0736+2 / [333]  
;CODES: [140] SPS=7 / N:C = [LOC 6]  
;SYNC: B05J2 (-) T = 5.8 USEC  
;KEY SIG: K3-6 ILL INSTR L / K5-5 STPM3 H / K3-3 DM=0L / K3-5 JMP+JSR H  
T0736: MOV #0736,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0736,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV SP,R5 ;SAVE SP  
R0736: MOV @#4,R4 ;SAVE VECTOR POINTER AT LOC. 4  
MOV #A0736,@#4 ;ON TRAP - GO TO A0736  
MOV R5,SP ;RESET SP FOR ERROR LOOP  
MOV #E0736,R2 ;IN CASE IT JUMPS TO ADDR IN RN  
CCC ;SCOPE SYNC  
I0736: JMP R2 ;JMP MODE 0 FORCES TRAP - GO TO A0736  
E0736: MOV R4,@#4 ;RESTORE VECTOR POINTER AT LOC. 4  
ERROR5 ;ILLEGAL INSTR TRAP FAILED  
R0736 ;ERROR LOOP RETURN ADDRESS  
A0736: MOV R4,@#4 ;RESTORE VECTOR POINTER AT LOC. 4  
MOV R5,SP ;RESET SP  
00736: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

24249
24250
24251
24252
24253
24254
24255
24256
24257
24258
24259
24260
24261
24262
24263
24264
24265
24266
24267
24268
24269
24270
24271
24272
24273
24274
24275
24276
24277
24278
24279
24280
24281
24282
24283
24284
24285

; *****
; .SBTTL T0737 BUS TIMEOUT TRAP TEST - TST (R)
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ:

;ACT BUTS:

;EXEC:

;CODES:

;SYNC:

;KEY SIG:

060414 012700 000737
060420 013701 060450
060424 010605
060426 013704 000004
060432 012737 060462 000004
060440 012702 160000
060444 010506
060446 000257
060450 005712
060452 010437 000004
060456 104005
060460 060426
060462 010437 000004
060466 010506
060470 000004

T0737: MOV #0737,R0
MOV @#I0737,R1
MOV SP,R5
R0737: MOV @#4,R4
MOV #A0737,@#4
MOV #160000,R2
MOV R5,SP
CCC
I0737: TST (R2)
E0737: MOV R4,@#4
ERROR5
R0737
A0737: MOV R4,@#4
MOV R5,SP
00737: SCOPE

;LOAD R0 WITH TEST NO.
;LOAD R1 WITH TEST INSTRUCTION WORD
;SAVE SP
;SAVE ORIGINAL T.O. VECTOR POINTER
;ON T.O. TRAP - GO TO A0737
;R3 ADDRESS CAUSES T.O.
;RESET SP FOR ERROR LOOP
;SCOPE SYNC
;FORCE T.O. TRAP - GO TO A0737
;RESTORE T.O. VECTOR
;TIMEOUT TRAP FAILED
;ERROR LOOP RETURN ADDRESS
;RESTORE T.O. VECTOR
;RESET SP
;CALL THE SCOPE LOOP UTILITY

```

24286 ; *****
24287 ; .SBTTL T0740 'T' BIT TRAP TEST
24288 ; *****
24289 ;MICROPROGRAMMING / LOGIC INFORMATION
24290 ;ROM SEQ: [104,373,362,002,015,010,216,215,115,326,327,113,330,331
24291 ;:077,140,332,333,123,015,013] FC 1,7,8,10,6,10
24292 ;ACT BUTS: 37[004]100,104 / 31[104]360,362 / 27[373]000,002 / 26[002]010,010
24293 ;EXEC: [362] BUFP = 002 ('T' BIT TRAP)
24294 ;CODES: N / A
24295 ;SYNC: B05J2 (-) T = 8 USEC
24296 ;KEY SIG: K5-2 PS(T)(1) H / K5-5 STPM3 H / K5-5 STPM2 H
24304 060472 012700 000740 T0740: MOV #0740,R0 ;LOAD R0 WITH TEST NO.
24305 060476 013701 060530 MOV @#I0740,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
24306 060502 010605 MOV SP,R5 ;SAVE SP
24307 060504 010506 R0740: MOV R5,SP ;RESET SP FOR ERROR LOOP
24308 060506 012737 060540 000014 MOV #A0740,@#14 ;GO TO A0740 WHEN 'T' TRAP SPRUNG
24309 060514 012746 000020 MOV #20,-(SP) ;SET 'T' BIT ON STACK
24310 060520 012746 060530 MOV #I0740,-(SP) ;SET UP NEW PC ON STACK
24311 060524 000257 CCC ;SCOPE SYNC
24312 060526 000006 RTT ;TURN ON 'T' BIT - GO TO I0740
24313
24314 060530 005700 I0740: TST R0 ;SPRING 'T' BIT TRAP - GO TO A0740
24315
24316 060532 104005 E10740: ERROR5 ;NO 'T' BIT TRAP OCCURRED
24317 060534 060504 R0740 ;ERROR LOOP RETURN ADDRESS
24318
24319 060536 000406 BR B0740 ;GO EXIT
24320
24321 060540 032766 000020 000002 A0740: BIT #20,2(SP) ;'T' BIT SET IN OLD PSW?
24322 060546 001002 BNE B0740 ;BR IF YES
24323
24324 060550 104000 E20740: ERROR ;#T# BIT NOT SAVED ON STACK
24325 060552 060504 R0740 ;ERROR LOOP RETURN ADDRESS
24326
24327 060554 012737 000016 000014 B0740: MOV #16,@#14 ;RESTORE 'T' BIT TRAP CATCHER
24328 060562 005037 000016 CLR @#16 ;RESET SP
24329 060566 010506 MOV R5,SP
24330
24331 060570 000004 00740: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

24332 ; *****
24333 ; .SBTTL T0741 TEST PUSH INTO PSW WITH [SP] = 000000
24334 ; *****
24335
24336 ;THESE NEXT TWO TESTS VERIFY THAT A 'RED ZONE' TRAP OCCURS IF A
24337 ;PUSH IS ATTEMPTED WITH THE [SP] INITIALLY EQUAL TO 000000,177572.
24338
24339 ;MICROPROGRAMMING / LOGIC INFORMATION
24340
24341 ;ROM SEQ: [142,240,250,174,257,200,JAMUPP,336,317,215,115
24342 ; 326,327,113,330,331,077,140,332,333,123,015,013] FC 1,2,4,6,10
24343 ;ACT BUTS: 37[004]100,142 / 35[240]120,174 / 22[174]200,200 / 01[332]122,123
24344 ; / 26[123]010,013
24345
24346 ;EXEC: [200] BUS STOP TRIGGERS JAMUPP LOGIC
24347
24348 ;CODES: N / A
24349
24350 ;SYNC: B05J2 (-) T = 10 USEC
24351
24352 ;KEY SIG: K4-6 PROC ADRS H / K4-4 BUS STOP H / K4-4 OVFLW ERR L / K4-3 JBERR
24353 ; K4-3 JAMUPP L / K4-3 JAM CLK H / K4-3 PJAMSTART H / K5-5 STACK04
24354
24355 060572 012700 000741 T0741: MOV #0741,R0 ;LOAD R0 WITH TEST NO.
24356 060576 013701 060622 MOV @#10741,R1 ;LOAD R1 WITH COPY OF TEST INSTRUCTION
24357 060602 010605 MOV SP,R5 ;SAVE THE SP
24358 060604 013704 000004 R0741: MOV @#4,R4 ;SAVE THE BUS ERROR VECTOR
24359 060610 012737 060646 000004 MOV #A0741,@#4 ;'RED ZONE' TRAP GOES TO A0741
24360 060616 005006 CLR SP ;MAKE SP = 000000
24361 060620 000257 CCC ;SCOPE SYNC
24362
24363 060622 012746 177777 I0741: MOV #-1,-(SP) ;ATTEMPT PUSH INTO PSW - SHOULD CAUSE
24364 ;'RED ZONE' TRAP TO BE SPRUNG
24365
24366 060626 010437 000004 MOV R4,@#4 ;RESTORE BUS ERROR VECTOR
24367 060632 005004 CLR R4 ;[R4] = S / B SP
24368 060634 010603 MOV SP,R3 ;[R3] = WAS SP
24369 060636 010506 MOV R5,SP ;RESET THE SP
24370 060640 104000 E10741: ERROR ;TRAP NOT SPRUNG
24371 060642 060604 R0741 ;ERROR LOOP RETURN ADDRESS
24372 060644 000413 BR 00741 ;GO TO SCOPE EXIT - SCHOOL'S OUT
24373
24374 060646 022706 000000 A0741: CMP #0,SP ;WAS IT A RED ZONE TRAP ?
24375 060652 001407 BEQ B0741 ;BR IF YES
24376
24377 060654 010437 000004 MOV R4,@#4 ;RESTORE BUS ERROR VECTOR
24378 060660 005004 CLR R4 ;[R4] = S / B SP
24379 060662 010603 MOV SP,R3 ;[R3] = WAS SP
24380 060664 010506 MOV R5,SP ;RESET THE SP
24381 060666 104000 E20741: ERROR ;TRAP SPRUNG BUT NOT RED ZONE
24382 060670 060604 R0741 ;ERROR LOOP RETURN ADDRESS
24383
24384 060672 010506 B0741: MOV R5,SP ;FIX UP THE SP
24385
24386 060674 000004 00741: SCOPE ;CALL THE SCOPE LOOP UTILITY
24387

```

24388
24389
24390
24391
24392
24393
24394
24395
24396
24397
24398
24399
24400
24401
24402
24403
24404
24405
24406
24407
24408
24409
24410
24411
24412
24413
24414
24415
24416
24417
24418
24419
24420
24421
24422
24423
24424
24425
24426
24427
24428
24429
24430
24431
24432
24433
24434
24435
24436
24437
24438
24439
24440

060676 012700 000742
060702 013701 060730
060706 010605
060710 013704 000004
060714 012737 060754 000004
060722 012706 177572
060726 000257
060730 012746 177777
060734 010437 000004
060740 005004
060742 010603
060744 010506
060746 104000
060750 060710
060752 000413
060754 022706 000000
060760 001407
060762 010437 000004
060766 005004
060770 010603
060772 010506
060774 104000
060776 060710
061000 010506
061002 000004

000004

```
; *****  
; .SBTTL T0742 TEST PUSH INTO SR WITH [SP] = 177572  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [142,240,250,174,257,200,JAMUPP,336,317,215,115  
; : 326,327,113,330,331,077,140,332,333,123,015,013] FC 1,2,4,6,10  
;ACT BUTS: 37[004]100,142 / 35[240]120,174 / 22[174]200,200 / 01[332]122,123  
; / 26[123]010,013  
;EXEC: [200] BUS STOP TRIGGERS JAMUPP LOGIC  
;CODES: N / A  
;SYNC: B05J2 (-) T = 10 USEC  
;KEY SIG: K4-6 PROC ADRS H / K4-4 BUS STOP / K4-4 OVFLW ERR L / K4-3 JBERR(1  
; :4-3 JAMUPP L / K4-3 JAM CLK H / K4-3 PJAMSTART H / K5-5 STACK04 H  
T0742: MOV #0742,R0 ;LOAD R0 WITH TEST NO.  
MOV @#10742,R1 ;LOAD R1 WITH COPY OF TEST INSTRUCTION  
MOV SP,R5 ;SAVE THE SP  
R0742: MOV @#4,R4 ;SAVE THE BUS ERROR VECTOR  
MOV #A0742,@#4 ;'RED ZONE' TRAP GOES TO A0742  
MOV #177572,SP ;MAKE SP=177572  
CCC ;SCOPE SYNC  
I0742: MOV #-1,-(SP) ;ATTEMPT PUSH INTO SR - SHOULD CAUSE  
; 'RED ZONE' TRAP TO BE SPRUNG  
MOV R4,@#4 ;RESTORE BUS ERROR VECTOR  
CLR R4 ;[R4] = S / B SP  
MOV SP,R3 ;[R3] = WAS SP  
MOV R5,SP ;RESET THE SP  
E10742: ERROR ;TRAP NOT SPRUNG  
R0742 ;ERROR LOOP RETURN ADDRESS  
BR 00742 ;GO TO SCOPE EXIT - SCHOOL'S OUT  
A0742: CMP #0,SP ;WAS IT A RED ZONE TRAP ?  
BEQ B0742 ;BR IF YES  
MOV R4,@#4 ;RESTORE BUS ERROR VECTOR  
CLR R4 ;[R4]= S / B SP  
MOV SP,R3 ;[R3] = WAS SP  
MOV R5,SP ;RESET THE SP  
E20742: ERROR ;TRAP SPRUNG BUT NOT RED ZONE  
R0742 ;ERROR LOOP RETURN  
B0742: MOV R5,SP ;FIX UP THE SP  
00742: SCOPE ;CALL THE SCOPE LOOP UTILITY
```

```

24441 ; *****
24442 ; .SBTTL T0743 TEST PUSH INTO SLR WITH [SP] = 177776
24443 ; *****
24444
24445 ;MICROPROGRAMMING / LOGIC INFORMATION
24446
24447 ;ROM SEQ: [142,240,250,174,257,200,JAMUPP,336,317,215,115
24448 ; 326,327,113,330,331,077,140,332,333,123,015,013] FC 1,2,4,6,10
24449 ;ACT BUTS: 37[004]100,142 / 35[240]120,174 / 22[174]200,200 / 01[332]122,123
24450 ; / 26[123]010,013
24451
24452 ;EXEC: [200] BUS STOP TRIGGERS JAMUPP LOGIC
24453
24454 ;CODES: N / A
24455
24456 ;SYNC: B05J2 (-) T = 10 USEC
24457
24458 ;KEY SIG: K4-6 PROC ADRS H / K4-4 BUS STOP / K4-4 OVFLW ERR L / K4-3 JBERR(1
24459 ;4-3 JAMUPP L / K4-3 JAM CLK H / K4-3 PJAMSTART H / K5-5 STACK04 H
24460
24461
24462 061004 012700 000743 T0743: MOV #0743,R0 ;LOAD R0 WITH TEST NO.
24463 061010 013701 061046 MOV @#I0743,R1 ;LOAD R1 WITH COPY OF TEST INSTRUCTION
24464 061014 010605 MOV SP,R5 ;SAVE THE SP
24465 061016 032737 000004 066636 BIT #4,@#OPTION ;IS KJ11 INSTALLED ??
24466 061024 001435 BEQ 00743 ;BR IF NOT - SKIP THIS TEST
24467 061026 013704 000004 R0743: MOV @#4,R4 ;SAVE THE BUS ERROR VECTOR
24468 061032 012737 061072 000004 MOV #A0743,@#4 ;'RED ZONE' TRAP GOES TO A0743
24469 061040 012706 177776 MOV #177776,SP ;MAKE SP=177776
24470 061044 000257 CCC ;SCOPE SYNC
24471
24472 061046 012746 177777 I0743: MOV #-1,-(SP) ;ATTEMPT PUSH INTO SR - SHOULD CAUSE
24473 ;'RED ZONE' TRAP TO BE SPRUNG
24474
24475 061052 010437 000004 MOV R4,@#4 ;RESTORE BUS ERROR VECTOR
24476 061056 005004 CLR R4 ;[R4] = S / B SP
24477 061060 010603 MOV SP,R3 ;[R3] = WAS SP
24478 061062 010506 MOV R5,SP ;RESET THE SP
24479 061064 104000 E10743: ERROR ;TRAP NOT SPRUNG
24480 061066 061026 R0743 ;ERROR LOOP RETURN ADDRESS
24481 061070 000413 BR 00743 ;GO TO SCOPE EXIT - SCHOOL'S OUT
24482
24483 061072 022706 000000 A0743: CMP #0,SP ;WAS IT A RED ZONE TRAP ?
24484 061076 001407 BEQ B0743 ;BR IF YES
24485
24486 061100 010437 000004 MOV R4,@#4 ;RESTORE BUS ERROR VECTOR
24487 061104 005004 CLR R4 ;[R4]= S / B SP
24488 061106 010603 MOV SP,R3 ;[R3] = WAS SP
24489 061110 010506 MOV R5,SP ;RESET THE SP
24490 061112 104000 E20743: ERROR ;TRAP SPRUNG BUT NOT RED ZONE
24491 061114 061026 R0743 ;ERROR LOOP RETURN
24492
24493 061116 010506 B0743: MOV R5,SP ;FIX UP THE SP
24494
24495 061120 000004 00743: SCOPE ;CALL THE SCOPE LOOP UTILITY

```

```

24496 061122 013737 000010 067564      MOV    @#10,@#MBUF1      ;SAVE RSVD INSTR VECTOR
24497 061130 013737 000012 067566      MOV    @#12,@#MBUF1+2
24498
24499
24500      ; *****
24501      ; .SBTTL T0744 RSVD INSTRUCTION TEST - 000007 THRU 000077
24502      ; *****
24503      ;MICROPROGRAMMING / LOGIC INFORMATION
24504
24505      ;ROM SEQ:      [100,126,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6
24506
24507      ;ACT BUTS:     37[004]100,100 / 01[332]122,123 / 26[123]010,013
24508
24509      ;EXEC:        [115] D = 000000 / [113] D = OLD PSW / [331] D = #I0744+2 / [333]
24510
24511      ;CODES:       [140] SPS=7 / N:C = 0000
24512
24513      ;SYNC:        B05J2 (-) T = 5.8 USEC
24514
24515      ;KEY SIG:     K3-6 RSVD INSTR L / K5-5 BC01 H / K5-3 BUT01 H / K5-3 BUT03 H
24516      ; K5-3 BUT04 H
24517
24518 061136 012700 000744      T0744: MOV    #0744,R0      ;LOAD R0 WITH TEST NO.
24519 061142 010605      MOV    SP,R5      ;SAVE THE SP
24520 061144 012737 061200 000010      MOV    #A0744,@#10      ;SET UP RSVD INSTR. TRAP VECTOR
24521 061152 005037 000012      CLR    @#12
24522 061156 012701 000007      MOV    #7,R1      ;SET UP FIRST ONE IN GROUP
24523 061162 010506      R0744: MOV    R5,SP      ;RESET SP FOR ERROR LOOP AND NEW INSTR
24524 061164 010137 061172      MOV    R1,@#I0744      ;LOAD NEW INSTR
24525 061170 000257      CCC
24526
24527 061172 000007      I0744: 000007      ;SCOPE SYNC
24528
24529
24530 061174 104005      E0744: ERROR5      ;TEST THE RSVD INSTR - THIS LOCATION
24531 061176 061162      R0744      ;GETS CHANGED EACH PASS THROUGH
24532
24533 061200 005201      A0744: INC    R1      ;RSVD INSTR. IN R1 FAILED TO TRAP
24534 061202 022701 000100      CMP    #100,R1      ;ERROR LOOP RETURN
24535 061206 001365      BNE    R0744      ;GENERATE NEW RSVD INSTR
24536
24537 061210 010506      MOV    R5,SP      ;AT END OF THIS GROUP ??
24538 061212 000004      00744: SCOPE      ;BR IF NOT
                ;MAKE SURE TO RESET THE SP
                ;CALL THE SCOPE LOOP UTILITY

```

24539
24540
24541
24542
24543
24544
24545
24546
24547
24548
24549
24550
24551
24552
24553
24554
24555
24556
24557
24558
24559
24560
24561
24562
24563
24564
24565
24566
24567
24568
24569
24570
24571
24572
24573
24574
24575
24576
24577
24578
24579
24580
24581
24582
24583

; *****
; .SBTTL T0745 RSVD INSTRUCTION TEST - 000210 THRU 000237
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [100,126,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6
;ACT BUTS: 37[004]100,100 / 01[332]122,123 / 26[123]010,013
;EXEC: [115] D = 000000 / [113] D = OLD PSW / [331] D = #I0745+2 / [333]
;CODES: [140] SPS=7 / N:C = 0000
;SYNC: B05J2 (-) 1 = 5.8 USEC
;KEY SIG: K3-6 RSVD INSTR L / K5-5 BC01 H / K5-3 BUT01 H / K5-3 BUT03 H
; K5-3 BUT04 H
;K5-5 STPM3 H

061214 012700 000745
061220 010605
061222 032737 040000 066642
061230 001401
061232 000000
061234 012737 061270 000010
061242 005037 000012
061246 012701 000210
061252 010506
061254 010137 061262
061260 000257
061262 000210
061264 104005
061266 061252
061270 005201
061272 022701 000240
061276 001365
061300 010506
061302 000004

T0745: MOV #0745,R0 ;LOAD R0 WITH TEST NO.
MOV SP,R5 ;SAVE THE SP
BIT #40000,@#BPTLOC ;BREAKPOINT HALT SET ??
BEQ .+4 ;BR IF NOT
HALT ;BREAK-DEPRESS CONTINUE TO RESTART
MOV #A0745,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
CLR @#12
R0745: MOV #210,R1 ;SET UP FIRST ONE IN GROUP
MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
MOV R1,@#I0745 ;LOAD NEW INSTR
CCC ;SCOPE SYNC
I0745: 000210 ;TEST THE RSVD INSTR - THIS LOCATION
;GETS CHANGED EACH PASS THROUGH
E0745: ERROR5 ;RSVD INSTR. IN R1 FAILED TO TRAP
R0745 ;ERROR LOOP RETURN
A0745: INC R1 ;GENERATE NEW RSVD INSTR
CMP #240,R1 ;AT END OF THIS GROUP ??
BNE R0745 ;BR IF NOT
00745: MOV R5,SP ;MAKE SURE TO RESET THE SP
SCOPE ;CALL THE SCOPE LOOP UTILITY

24584
24585
24586
24587
24588
24589
24590
24591
24592
24593
24594
24595
24596
24597
24598
24599
24600
24601
24602
24603
24604
24605
24606
24607
24608
24609
24610
24611
24612
24613
24614
24615
24616
24617
24618
24619
24620
24621
24622
24623
24624
24625
24626
24627

061304 012700 000746
061310 105737 066636
061314 100424
061316 010605
061320 012737 061354 000010
061326 005037 000012
061332 012701 000065
061336 010506
061340 010137 061346
061344 000257
061346 000065
061350 104005
061352 061336
061354 005201
061356 022701 000067
061362 001365
061364 010506
061366 000004

; *****
; .SBTTL T0746 RSVD INSTRUCTION TEST - 000065 THRU 000066
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [100,126,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6
;ACT BUTS: 37[004]100,100 / 01[332]122,123 / 26[123]010,013
;EXEC: [115] D = 000000 / [113] D = OLD PSW / [331] D = #I0746+2 / [333]
;CODES: [140] SPS=7 / N:C = 0000
;SYNC: B05J2 (-) T = 5.8 USEC
;KEY SIG: K3-6 RSVD INSTR L / K5-5 BC01 H / K5-3 BUT01 H / K5-3 BUT03 H
; K5-3 BUT04 H
;K5-5 STPM3 H

```
T0746: MOV #0746,R0 ;LOAD R0 WITH TEST NO.
        TSTB @#OPTION ;KT11D INSTALLED ??
        BMI 00746 ;SKIP THIS TEST IF YES
        MOV SP,R5 ;SAVE THE SP
        MOV #A0746,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
        CLR @#12
R0746: MOV #65,R1 ;SET UP FIRST ONE IN GROUP
        MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
        MOV R1,@#I0746 ;LOAD NEW INSTR
        CCC ;SCOPE SYNC

I0746: 000065 ;TEST THE RSVD INSTR - THIS LOCATION
        ;GETS CHANGED EACH PASS THROUGH

E0746: ERROR5 ;RSVD INSTR. IN R1 FAILED TO TRAP
        R0746 ;ERROR LOOP RETURN

A0746: INC R1 ;GENERATE NEW RSVD INSTR
        CMP #67,R1 ;AT END OF THIS GROUP ??
        BNE R0746 ;BR IF NOT

00746: MOV R5,SP ;MAKE SURE TO RESET THE SP
        SCOPE ;CALL THE SCOPE LOOP UTILITY
```

24628
24629
24630
24631
24632
24633
24634
24635
24636
24637
24638
24639
24640
24641
24642
24643
24644
24645
24646
24647
24648
24649
24650
24651
24652
24653
24654
24655
24656
24657
24658
24659
24660
24661
24662
24663
24664
24665
24666
24667
24668
24669
24670
24671

; *****
; .SBTTL T0747 RSVD INSTRUCTION TEST - 007000 THRU 007777
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [100,126,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6
;ACT BUTS: 37[004]100,100 / 01[332]122,123 / 26[123]010,013
;EXEC: [115] D = 000000 / [113] D = OLD PSW / [331] D = #I0747+2 / [333]
;CODES: [140] SPS=7 / N:C = 0000
;SYNC: B05J2 (-) T = 5.8 USEC
;KEY SIG: K3-6 RSVD INSTR L / K5-5 BC01 H / K5-3 BUT01 H / K5-3 BUT03 H
; K5-3 BUT04 H
;K5-5 STPM3 H

061370 012700 000747
061374 032737 000001 066636
061402 001024
061404 010605
061406 012737 061442 000010
061414 005037 000012
061420 012701 007000
061424 010506
061426 010137 061434
061432 000257
061434 007000
061436 104005
061440 061424
061442 005201
061444 022701 010000
061450 001365
061452 010506
061454 000004

T0747: MOV #0747,R0 ;LOAD R0 WITH TEST NO.
BIT #1,@#OPTION ;KE11-E (EIS) INSTALLED ??
BNE 00747 ;SKIP THIS TEST IF YES
MOV SP,R5 ;SAVE THE SP
MOV #A0747,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
CLR @#12
MOV #7000,R1 ;SET UP FIRST ONE IN GROUP
R0747: MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
MOV R1,@#I0747 ;LOAD NEW INSTR
CCC ;SCOPE SYNC
I0747: 007000 ;TEST THE RSVD INSTR - THIS LOCATION
;GETS CHANGED EACH PASS THROUGH
E0747: ERROR5 ;RSVD INSTR. IN R1 FAILED TO TRAP
R0747 ;ERROR LOOP RETURN
A0747: INC R1 ;GENERATE NEW RSVD INSTR
CMP #10000,R1 ;AT END OF THIS GROUP ??
BNE R0747 ;BR IF NOT
00747: MOV R5,SP ;MAKE SURE TO RESET THE SP
SCOPE ;CALL THE SCOPE LOOP UTILITY

24672
24673
24674
24675
24676
24677
24678
24679
24680
24681
24682
24683
24684
24685
24686
24687
24688
24689
24690
24691
24692
24693
24694
24695
24696
24697
24698
24699
24700
24701
24702
24703
24704
24705
24706
24707
24708
24709
24710
24711
24712
24713
24714

061456 012700 000750
061462 032737 000001 066636
061470 001024
061472 010605
061474 012737 061530 000010
061502 005037 000012
061506 012701 070000
061512 010506
061514 010137 061522
061520 000257
061522 070000
061524 104005
061526 061512
061530 005201
061532 022701 074000
061536 001365
061540 010506
061542 000004

```

; *****
; .SBTTL T0750 RSVD INSTRUCTION TEST - 070000 THRU 073777
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ:      [100,126,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6
;ACT BUTS:     37[004]100,100 / 01[332]122,123 / 26[123]010,013
;EXEC:         [115] D = 000000 / [113] D = OLD PSW / [331] D = #10750+2 / [333]
;CODES:        [140] SPS=7 / N:C = 0000
;SYNC:         B05J2 (-) T = 5.8 USEC
;KEY SIG:      K3-6 RSVD INSTR L / K5-5 BC01 H / K5-3 BUT01 H / K5-3 BUT03 H
                : K5-3 BUT04 H
                :K5-5 STPM3 H

T0750:  MOV    #0750,R0          ;LOAD R0 WITH TEST NO.
        BIT    #1,@#OPTION      ;IS THE KE11-E INSTALLED ??
        BNE   00750             ;BR IF YES - SKIP THIS TEST
        MOV   SP,R5             ;SAVE THE SP
        MOV   #A0750,@#10      ;SET UP RSVD INSTR. TRAP VECTOR
        CLR   @#12
        MOV   #70000,R1        ;SET UP FIRST ONE IN GROUP
R0750:  MOV   R5,SP             ;RESET SP FOR ERROR LOOP AND NEW INSTR
        MOV   R1,@#10750      ;LOAD NEW INSTR
        CCC
        I0750: 070000          ;SCOPE SYNC
                                ;TEST THE RSVD INSTR - THIS LOCATION
                                ;GETS CHANGED EACH PASS THROUGH
        E0750: ERROR5         ;RSVD INSTR. IN R1 FAILED TO TRAP.
        R0750          ;ERROR LOOP RETURN
        A0750:  INC    R1        ;GENERATE NEW RSVD INSTR
        CMP    #74000,R1      ;AT END OF THIS GROUP ??
        BNE   R0750          ;BR IF NOT
        O0750:  MOV   R5,SP    ;MAKE SURE TO RESET THE SP
        SCOPE          ;CALL THE SCOPE LOOP UTILITY

```

```

24715 ; *****
24716 ; .SBTTL T0751 RSVD INSTRUCTION TEST - 075000 THRU 075037
24717 ; *****
24718 ;MICROPROGRAMMING / LOGIC INFORMATION
24719 ;ROM SEQ: [100,126,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6
24720 ;ACT BUTS: 37[004]100,100 / 01[332]122,123 / 26[123]010,013
24721 ;EXEC: [115] D = 000000 / [113] D = OLD PSW / [331] D = #I0751+2 / [333]
24722 ;CODES: [140] SPS=7 / N:C = 0000
24723 ;SYNC: B05J2 (-) T = 5.8 USEC
24724 ;KEY SIG: K3-6 RSVD INSTR L / K5-5 BC01 H / K5-3 BUT01 H / K5-3 BUT03 H
24725 ; K5-3 BUT04 H
24726 ;K5-5 STPM3 H
24727
24728
24729
24730
24731
24732
24733
24734
24735 061544 012700 000751 T0751: MOV #0751,R0 ;LOAD R0 WITH TEST NO.
24736 061550 032737 000002 066636 BIT #2,@#OPTION ;KE11-F (FIS) INSTALLED ??
24737 061556 001024 BNE 00751 ;SKIP TEST IF YES
24738 061560 010605 MOV SP,R5 ;SAVE THE SP
24739 061562 012737 061616 000010 MOV #A0751,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
24740 061570 005037 000012 CLR @#12
24741 061574 012701 075000 MOV #75000,R1 ;SET UP FIRST ONE IN GROUP
24742 061600 010506 R0751: MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
24743 061602 010137 061610 MOV R1,@#I0751 ;LOAD NEW INSTR
24744 061606 000257 CCC ;SCOPE SYNC
24745
24746 061610 075000 I0751: 75000 ;TEST THE RSVD INSTR - THIS LOCATION
24747 ;GETS CHANGED EACH PASS THROUGH
24748
24749 061612 104005 E0751: ERROR5 ;RSVD INSTR. IN R1 FAILED TO TRAP
24750 061614 061600 R0751 ;ERROR LOOP RETURN
24751
24752 061616 005201 A0751: INC R1 ;GENERATE NEW RSVD INSTR
24753 061620 022701 075040 CMP #75040,R1 ;AT END OF THIS GROUP ??
24754 061624 001365 BNE R0751 ;BR IF NOT
24755
24756 061626 010506 00751: MOV R5,SP ;MAKE SURE TO RESET THE SP
24757 061630 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
24758

```

```
24759 ; *****  
24760 ; .SBTTL T0752 RSVD INSTRUCTION TEST - 075040 THRU 076777  
24761 ; *****  
24762  
24763 ;MICROPROGRAMMING / LOGIC INFORMATION  
24764  
24765 ;ROM SEQ: [100,126,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6  
24766  
24767 ;ACT BUTS: 37[004]100,100 / 01[332]122,123 / 26[123]010,013  
24768  
24769 ;EXEC: [115] D = 000000 / [113] D = OLD PSW / [331] D = #I0752+2 / [333]  
24770  
24771 ;CODES: [140] SPS=7 / N:C = 0000  
24772  
24773 ;SYNC: B05J2 (-) T = 5.8 USEC  
24774  
24775 ;KEY SIG: K3-6 RSVD INSTR L / K5-5 BC01 H / K5-3 BUT01 H / K5-3 BUT03 H  
24776 ; K5-3 BUT04 H  
24777 ;K5-5 STPM3 H  
24778  
24779 061632 012700 000752 T0752: MOV #0752,R0 ;LOAD R0 WITH TEST NO.  
24780 061636 010605 MOV SP,R5 ;SAVE THE SP  
24781 061640 012737 061674 000010 MOV #A0752,@#10 ;SET UP RSVD INSTR. TRAP VECTOR  
24782 061646 005037 000012 CLR @#12  
24783 061652 012701 075040 MOV #75040,R1 ;SET UP FIRST ONE IN GROUP  
24784 061656 010506 R0752: MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR  
24785 061660 010137 061666 MOV R1,@#I0752 ;LOAD NEW INSTR  
24786 061664 000257 CCC ;SCOPE SYNC  
24787  
24788 061666 075040 I0752: 75040 ;TEST THE RSVD INSTR - THIS LOCATION  
24789 ;GETS CHANGED EACH PASS THROUGH  
24790  
24791 061670 104005 E0752: ERROR5 ;RSVD INSTR. IN R1 FAILED TO TRAP  
24792 061672 061656 R0752 ;ERROR LOOP RETURN  
24793  
24794 061674 005201 A0752: INC R1 ;GENERATE NEW RSVD INSTR  
24795 061676 022701 077000 CMP #77000,R1 ;AT END OF THIS GROUP ??  
24796 061702 001365 JNE R0752 ;BR IF NOT  
24797  
24798 061704 010506 00752: MOV R5,SP ;MAKE SURE TO RESET THE SP  
24799 061706 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY  
24800
```

```
24801 ; *****
24802 ; .SBTTL T0753 RSVD INSTRUCTION TEST - 106400 THRU 107777
24803 ; *****
24804
24805 ;MICROPROGRAMMING / LOGIC INFORMATION
24806
24807 ;ROM SEQ: [100,126,007,115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,6
24808
24809 ;ACT BUTS: 37[004]100,100 / 01[332]122,123 / 26[123]010,013
24810
24811 ;EXEC: [115] D = 000000 / [113] D = OLD PSW / [331] D = #I0753+2 / [333]
24812
24813 ;CODES: [140] SPS=7 / N:C = 0000
24814
24815 ;SYNC: B05J2 (-) T = 5.8 USEC
24816
24817 ;KEY SIG: K3-6 RSVD INSTR L / K5-5 BC01 H / K5-3 BUT01 H / K5-3 BUT03 H
24818 ; K5-3 BUT04 H
24819 ;K5-5 STPM3 H
24820
24821 061710 012700 000753 T0753: MOV #0753,R0 ;LOAD R0 WITH TEST NO.
24822 061714 032737 000200 066636 BIT #200,@#OPTION ;IS THE KT11-D INSTALLED ??
24823 061722 001024 BNE 00753 ;BR IF YES - SKIP THIS TEST
24824 061724 010605 MOV SP,R5 ;SAVE THE SP
24825 061726 012737 061762 000010 MOV #A0753,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
24826 061734 005037 000012 CLR @#12
24827 061740 012701 106400 MOV #106400,R1 ;SET UP FIRST ONE IN GROUP
24828 061744 010506 R0753: MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
24829 061746 010137 061754 MOV R1,@#I0753 ;LOAD NEW INSTR
24830 061752 000257 CCC ;SCOPE SYNC
24831
24832 061754 106400 I0753: 106400 ;TEST THE RSVD INSTR - THIS LOCATION
24833 ;GETS CHANGED EACH PASS THROUGH
24834
24835 061756 104005 E0753: ERROR5 ;RSVD INSTR. IN R1 FAILED TO TRAP
24836 061760 061744 R0753 ;ERROR LOOP RETURN
24837
24838 061762 005201 A0753: INC R1 ;GENERATE NEW RSVD INSTR
24839 061764 022701 110000 CMP #110000,R1 ;AT END OF THIS GROUP ??
24840 061770 001365 BNE R0753 ;BR IF NOT
24841
24842 061772 010506 00753: MOV R5,SP ;MAKE SURE TO RESET THE SP
24843 061774 000004 SCOPE ;CALL THE SCOPE LOOP UTILITY
24844
```

24845
24846
24847
24848
24849
24850
24851
24852
24853
24854
24855
24856
24857
24858
24859
24860
24861
24862
24863
24864
24865
24866
24867
24868
24869
24870
24871
24872
24873
24874
24875
24876
24877
24878
24879
24880
24881
24882
24883
24884
24885
24886
24887
24888

; *****
; .SBTTL T0754 RSVD INSTRUCTION TEST - 170000 THRU 177777
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [100,126,007,1115,326,327,113,330,331,077,140,332,333,123,015,013] FC 1,
;ACT BUTS: 37[004]100,100 / 01[332]122,123 / 26[123]010,013
;EXEC: [115] D = 000000 / [113] D = OLD PSW / [331] D = #I0754+2 / [333]
;CODES: [140] SPS=7 / N:C = 0000
;SYNC: B05J2 (-) T = 5.8 USEC
;KEY SIG: K3-6 RSVD INSTR L / K5-5 BC01 H / K5-3 BUT01 H / K5-3 BUT03 H
; K5-3 BUT04 H
;K5-5 STPM3 H

061776 012700 000754
062002 010605
062004 012737 062040 000010
062012 005037 000012
062016 012701 170000
062022 010506
062024 010137 062032
062030 000257
062032 170000
062034 104005
062036 062022
062040 005201
062042 022701 000000
062046 001365
062050 010506
062052 000004
062054 013737 067564 000010
062062 013737 067566 000012

T0754: MOV #0754,R0 ;LOAD R0 WITH TEST NO.
MOV SP,R5 ;SAVE THE SP
MOV #A0754,@#10 ;SET UP RSVD INSTR. TRAP VECTOR
CLR @#12
MOV #170000,R1 ;SET UP FIRST ONE IN GROUP
R0754: MOV R5,SP ;RESET SP FOR ERROR LOOP AND NEW INSTR
MOV R1,@#I0754 ;LOAD NEW INSTR
CCC ;SCOPE SYNC
I0754: 170000 ;TEST THE RSVD INSTR - THIS LOCATION
;GETS CHANGED EACH PASS THROUGH
E0754: ERROR5 ;RSVD INSTR. IN R1 FAILED TO TRAP
R0754 ;ERROR LOOP RETURN
A0754: INC R1 ;GENERATE NEW RSVD INSTR
CMP #0,R1 ;AT END OF THIS GROUP ??
BNE R0754 ;BR IF NOT
00754: MOV R5,SP ;MAKE SURE TO RESET THE SP
SCOPE ;CALL THE SCOPE LOOP UTILITY
MOV @#MBUF1,@#10 ;RESTORE RSVD INSTR VECTOR
MOV @#MBUF1+2,@#12

```
24889
24890 062070 012737 065026 000014 TSET: MOV #TBSER,@#14 ;SET UP THE 'T' BIT TRAP VECTOR
24891 062076 012737 000340 000016 MOV #340,@#16 ;PRIORITY 7
24892 062104 012737 062104 066654 DMRET: MOV #DMRET,@#RETURN ;INITIALIZE SCOPE LOOP RETURN
24893
24894 ; *****
24895 ;SBTTL T0755 BUT SERVICE TEST IN ROM LOCATION 373 - (TST %R)
24896 ; *****
24897
24898 ;THIS NEXT GROUP OF 28 SEQUENTIAL TESTS VERIFIES THAT A 'T' BIT
24899 ;TRAP CAN BE SERVICED IN EACH MICROWORD THAT DOES A 'BUT SERVICE'
24900 ;EACH ROUTINE ENTERS THE TRAP MICROUTINE WHEN THE TRAP IS SPRUNG
24901 ;WHICH GENERATES THE FOLLOWING ROM SEQUENCE:
24902
24903 ;[010,216,215,115,326,327,113,330,331,077,140,332,333,
24904 ;123,015,013] FC 6,10,1
24905
24906 ;MICROPROGRAMMING / LOGIC INFORMATION
24907
24908 ;ROM SEQ: [104,373,362,002,015,TRAP MICROUTINE] FC 1,7,8,10,6
24909
24910 ;ACT BUTS: 37[004]100,104 / 27[373]000,002 / 26[002]010,010
24911
24912 ;EXEC: [362] BUPP=002 ('T' BIT TRAP)
24913
24914 ;CODES: N / A
24915
24916 ;SYNC: B05J2 (-)
24917
24918 ;KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
24919 ; K5-5 STPM2 H
24920
24921 062112 012700 000755 T0755: MOV #0755,R0 ;LOAD R0 WITH TEST NO.
24922 062116 013701 062142 MOV @#I0755,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
24923 062122 010605 MOV SP,R5 ;SAVE THE SP
24924 062124 010506 R0755: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
24925 062126 012746 000020 MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
24926 062132 012746 062142 MOV #I0755,-(SP) ;MAKE NEW PC = I0755
24927 062136 000257 CCC ;SCOPE SYNC
24928 062140 000006 RTT ;SET 'T' BIT - GO TO I0755
24929
24930 062142 005700 I0755: TST R0 ;TST INSTRUCTION SHOULD SPRING TRAP
24931
24932 062144 104005 E0755: ERRORS ;BUT SERVICE IN 373 FAILED
24933 062146 062124 R0755 ;ERROR LOOP RETURN
24934
24935 062150 000004 O0755: SCOPE ;CALL SCOPE LOOP UTILITY
24936
```

24937
24938
24939
24940
24941
24942
24943
24944
24945
24946
24947
24948
24949
24950
24951
24952
24953
24954
24955
24956
24957
24958
24959
24960
24961
24962
24963
24964
24965
24966
24967
24968
24969
24970
24971
24972

: *****
:SBTTL T0756 BUT SERVICE TEST IN ROM LOCATION 366 - (BISB RA,(RB))
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [161,266,267,270,230,254,074,366,375,017,015,
TRAP MICROROUTINE] FC 1,3,8,10,6
:ACT BUTS: 37[004]100,161 / 33[266]220,237 / 34[237]220,230
:EXEC: 16[366]016,017 / 26[017]010,010
[375] BUPP=017 ('T' BIT TRAP)
:CODES: N / A
:SYNC: B05J2 (-)
:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

062152 012700 000756
062156 013701 062206
062162 010605
062164 010506
062166 012702 067561
062172 012746 000020
062176 012746 062206
062202 000257
062204 000006
062206 150012
062210 104005
062212 062164
062214 000004

T0756: MOV #0756,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0756,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R0756: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #MBUF0+1,R2 ;DEST ADDR = MBUF0
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0756,-(SP) ;MAKE NEW PC = I0756
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0756
I0756: BISB R0,(R2) ;BISB INSTRUCTION SHOULD SPRING TRAP
E0756: ERROR5 ;BUT SERVICE IN 366 FAILED
R0756 ;ERROR LOOP RETURN
00756: SCOPE ;CALL SCOPE LOOP UTILITY

24973
24974
24975
24976
24977
24978
24979
24980
24981
24982
24983
24984
24985
24986
24987
24988
24989
24990
24991
24992
24993
24994
24995
24996
24997
24998
24999
25000
25001
25002
25003
25004
25005
25006
25007
25008
25009
25010

062216 012700 000757
062222 013701 062252
062226 010605
052230 010506
062232 012702 067560
062236 012746 000020
062242 012746 062252
062246 000257
062250 000006
062252 006712
062254 104005
062256 062230
062260 000004

: *****
:SBTTL T0757 BUT SERVICE TEST IN ROM LOCATION 367 - (SXT (RN))
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [161,266,267,234,367,375,017,015,TRAP MICROROUTINE]
:FC 1,3,8,10,6
:ACT BUTS: 37[004]100,161 / 33[266]220,234 / 16[367]016,017
:26[017]010,010
:EXEC: [375] BUPP=017 ('T' BIT TRAP)
:CODES: N / A
:SYNC: B05J2 (-)
:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
:K5-5 STPM2 H

T0757: MOV #0757,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0757,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R0757: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #MBUFO,R2 ;DEST ADDR = MBUFO
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0757,-(SP) ;MAKE NEW PC = I0757
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0757
I0757: SXT (R2) ;SXT INSTRUCTION SHOULD SPRING TRAP
E0757: ERROR5 ;BUT SERVICE IN 367 FAILED
R0757 ;ERROR LOOP RETURN
00757: SCOPE ;CALL SCOPE LOOP UTILITY

25011
25012
25013
25014
25015
25016
25017
25018
25019
25020
25021
25022
25023
25024
25025
25026
25027
25028
25029
25030
25031
25032
25033
25034
25035
25036
25037
25038
25039
25040
25041
25042
25043
25044
25045

: *****
:SBTTL T0760 BUT SERVICE TEST IN ROM LOCATION 132 - (SXT %R)
: *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [132,360,002,015,TRAP MICROROUTINE] FC 1,8,10,6
;ACT BUTS: 37[004]100,132 / 27[132]000,002 / 26[002]010,010
;EXEC: [360] BUPP=002 ('T' BIT TRAP)
;CODES: N / A
;SYNC: B05J2 (-)
;KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
; K5-5 STPM2 H

062262 012700 000760
062266 013701 062312
062272 010605
062274 010506
062276 012746 000020
062302 012746 062312
062306 000257
062310 000006
062312 006702
062314 104005
062316 062274
062320 000004

T0760: MOV #0760,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0760,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R0760: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0760,-(SP) ;MAKE NEW PC = I0760
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0760
I0760: SXT R2 ;SXT INSTRUCTION SHOULD SPRING TRAP
E0760: ERROR5 ;BUT SERVICE IN 132 FAILED
R0760 ;ERROR LOOP RETURN
00760: SCOPE ;CALL SCOPE LOOP UTILITY

25046
25047
25048
25049
25050
25051
25052
25053
25054
25055
25056
25057
25058
25059
25060
25061
25062
25063
25064
25065
25066
25067
25068
25069
25070
25071
25072
25073
25074
25075
25076
25077
25078
25079
25080
25081

: *****
:SBTTL T0761 BUT SERVICE TEST IN ROM LOCATION 372 - (NEG %R)
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [105,372,360,002,015,TRAP MICROROUTINE] FC 1,7,8,10,6

:ACT BUTS: 37[004]100,105 / 31[105]360,360 / 27[372]000,002
: 26[002]010,010

:EXEC: [360] BUPP=002 ('T' BIT TRAP)

:CODES: N / A

:SYNC: B05J2 (-)

:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

062322 012700 000761
062326 013701 062352
062332 010605
062334 010506
062336 012746 000020
062342 012746 062352
062346 000257
062350 000006
062352 005402
062354 104005
062356 062334
062360 000004

T0761: MOV #0761,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0761,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R0761: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0761,-(SP) ;MAKE NEW PC = I0761
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0761

I0761: NEG R2 ;NEG INSTRUCTION SHOULD SPRING TRAP

E0761: ERROR5 ;BUT SERVICE IN 372 FAILED
R0761 ;ERROR LOOP RETURN

O0761: SCOPE ;CALL SCOPE LOOP UTILITY

25082
25083
25084
25085
25086
25087
25088
25089
25090
25091
25092
25093
25094
25095
25096
25097
25098
25099
25100
25101
25102
25103
25104
25105
25106
25107
25108
25109
25110
25111
25112
25113
25114
25115
25116
25117
25118
25119

062362 012700 000762
062366 013701 062416
062372 010605
062374 010506
062376 012703 067560
062402 012746 000020
062406 012746 062416
062412 000257
062414 000006
062416 161302
062420 104005
062422 062374
062424 000004

```
; *****  
;SBTTL T0762 BUT SERVICE TEST IN ROM LOCATION 370 - (SUB (RA),RB)  
; *****  
;MICROPROGRAMMING / LOGIC INFORMATION  
;ROM SEQ: [141,247,250,121,370,360,002,015,TRAP MICROROUTINE]  
; FC 1,2,8,10,6  
;ACT BUTS: 37[004]100,141 / 35[247]120,121 / 27[370]000,002  
; 26[002]010,010  
;EXQC: [360] BUPP=002 ('T' BIT TRAP)  
;CODES: N / A  
;SYNC: B05J2 (-)  
;KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H  
; K5-5 STPM2 H  
T0762: MOV #0762,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0762,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV SP,R5 ;SAVE THE SP  
R0762: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING  
MOV #M000,R3 ;SOURCE ADDR = M000  
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW  
MOV #I0762,-(SP) ;MAKE NEW PC = I0762  
CCC ;SCOPE SYNC  
RTT ;SET 'T' BIT - GO TO I0762  
I0762: SUB (R3),R2 ;SUB INSTRUCTION SHOULD SPRING TRAP  
E0762: ERROR5 ;BUT SERVICE IN 370 FAILED  
R0762 ;ERROR LOOP RETURN  
00762: SCOPE ;CALL SCOPE LOOP UTILITY
```

25120
25121
25122
25123
25124
25125
25126
25127
25128
25129
25130
25131
25132
25133
25134
25135
25136
25137
25138
25139
25140
25141
25142
25143
25144
25145
25146
25147
25148
25149
25150
25151
25152
25153
25154
25155
25156
25157

: *****
:SBTTL T0763 BUT SERVICE TEST IN ROM LOCATION 371 - (ADD (RA),RB)
: *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [141,247,250,120,371,360,002,015,TRAP MICROROUTINE]
: FC 1,2,8,10,6

;ACT BUTS: 37[004]100,141 / 35[247]120,120 / 31[120]360,360
: 27[371]000,002 / 26[002]010,010

;EXEC: [350] BUPP=002 ('T' BIT TRAP)

;CODES: N / A

;SYNC: B05J2 (-)

;KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

062426 012700 000763
062432 013701 062462
062436 010605
062440 010506
062442 012705 067560
062446 012746 000020
062452 012746 062462
062456 000257
062460 000006
062462 061502
062464 104005
062466 062440
062470 000004

T0763: MOV #0763,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0763,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R0763: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #M0763,R5 ;SOURCE ADDR = M0763
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0763,-(SP) ;MAKE NEW PC = I0763
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0763
I0763: ADD (R5),R2 ;ADD INSTRUCTION SHOULD SPRING TRAP
E0763: ERROR5 ;BUT SERVICE IN 371 FAILED
R0763 ;ERROR LOOP RETURN
00763: SCOPE ;CALL SCOPE LOOP UTILITY

25158
25159
25160
25161
25162
25163
25164
25165
25166
25167
25168
25169
25170
25171
25172
25173
25174
25175
25176
25177
25178
25179
25180
25181
25182
25183
25184
25185
25186
25187
25188
25189
25190
25191
25192

062472 012700 000764
062476 013701 062522
062502 010605
062504 010506
062506 012746 000020
062512 012746 062522
062516 000257
062520 000006

062522 000302

062524 104005
062526 062504

062530 000004

: *****
:SBTTL T0764 BUT SERVICE TEST IN ROM LOCATION 135 - (SWAB %R)
: *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [134,135,360,002,015,TRAP MICROROUTINE] FC 1,7,8,10,6
;ACT BUTS: 37[004]100,134 / 27[135]000,002 / 26[002]010,010
;EXEC: [135] BUPP=002 ('T' BIT TRAP)
;CODES: N / A
;SYNC: B05J2 (-)
;KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
; K5-5 STPM2 H

T0764: MOV #0764,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0764,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R0764: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0764,-(SP) ;MAKE NEW PC = I0764
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0764

I0764: SWAB R2 ;SWAB INSTRUCTION SHOULD SPRING TRAP

E0764: ERROR5 ;BUT SERVICE IN 135 FAILED
R0764 ;ERROR LOOP RETURN

00764: SCOPE ;CALL SCOPE LOOP UTILITY

25193
25194
25195
25196
25197
25198
25199
25200
25201
25202
25203
25204
25205
25206
25207
25208
25209
25210
25211
25212
25213
25214
25215
25216
25217
25218
25219
25220
25221
25222
25223
25224
25225
25226
25227

062532 012700 000765
062536 013701 062562
062542 010605
062544 010506
062546 012746 000020
062552 012746 062562
062556 000257
062560 000006
062562 160304
062564 104005
062566 062544
062570 000004

```
: *****  
:SBTTL T0765 BUT SERVICE TEST IN ROM LOCATION 363 - (SUB RA,RB)  
: *****  
:MICROPROGRAMMING / LOGIC INFORMATION  
:ROM SEQ: [103,363,360,002,015,TRAP MICROROUTINE] FC 1,7,8,10,6  
:ACT BUTS: 37[004]100,103 / 27[363]000,002 / 26[002]010,010  
:EXEC: [360] BUPP=002 ('T' BIT TRAP)  
:CODES: N / A  
:SYNC: B05J2 (-)  
:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H  
: K5-5 STPM2 H  
T0765: MOV #0765,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0765,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV SP,R5 ;SAVE THE SP  
R0765: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING  
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW  
MOV #I0765,-(SP) ;MAKE NEW PC = I0765  
CCC ;SCOPE SYNC  
RTT ;SET 'T' BIT - GO TO I0765  
I0765: SUB R3,R4 ;SUB INSTRUCTION SHOULD SPRING TRAP  
E0765: ERROR5 ;BUT SERVICE IN 363 FAILED  
R0765 ;ERROR LOOP RETURN  
00765: SCOPE ;CALL SCOPE LOOP UTILITY
```

```
25228 : *****
25229 :SBTTL T0766 BUT SERVICE TEST IN ROM LOCAT'ON 364 - (ADD RA,RB)
25230 : *****
25231
25232 ;MICROPROGRAMMING / LOGIC INFORMATION
25233
25234 ;ROM SEQ: [102,364,360,002,015,TRAP MICROROUTINE] FC 1,7,8,10,6
25235
25236 ;ACT BUTS: 37[004]100,102 / 31[102]360,360 / 27[364]000,002
25237 ;
25238
25239 ;EXEC: [360] BUPP=002 ('T' BIT TRAP)
25240
25241 ;CODES: N / A
25242
25243 ;SYNC: B05J2 (-)
25244
25245 ;KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
25246 ;
25247
25248 062572 012700 000766 T0766: MOV #0766,R0 ;LOAD R0 WITH TEST NO.
25249 062576 013701 062622 MOV @#I0766,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
25250 062602 010605 MOV SP,R5 ;SAVE THE SP
25251 062604 010506 R0766: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
25252 062606 012746 000020 MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
25253 062612 012746 062622 MOV #I0766,-(SP) ;MAKE NEW PC = I0766
25254 062616 000257 CCC ;SCOPE SYNC
25255 062620 000006 RTT ;SET 'T' BIT - GO TO I0766
25256
25257 062622 060304 I0766: ADD R3,R4 ;ADD INSTRUCTION SHOULD SPRING TRAP
25258
25259 062624 104005 E0766: ERROR5 ;BUT SERVICE IN 364 FAILED
25260 062626 062604 R0766 ;ERROR LOOP RETURN
25261
25262 062630 000004 O0766: SCOPE ;CALL SCOPE LOOP UTILITY
25263
```

25264
25265
25266
25267
25268
25269
25270
25271
25272
25273
25274
25275
25276
25277
25278
25279
25280
25281
25282
25283
25284
25285
25286
25287
25288
25288
25289
25290
25291
25292
25293
25294
25295
25296
25297
25298
25299
25300

062632 012700 000767
062636 013701 062666
062642 010605
062644 012702 067560
062650 010506
062652 012746 000020
062656 012746 062666
062662 000257
062664 000006

062666 010012

062670 104005
062672 062650

062674 000004

```
: *****  
:SBTTL T0767 BUT SERVICE TEST IN ROM LOCATION 125 - (MOV RA,(RB))  
: *****  
  
:MICROPROGRAMMING / LOGIC INFORMATION  
  
:ROM SEQ:      [171,257,201,125,375,017,015,TRAP MICROROUTINE] FC 1,4,8,10,6  
  
:ACT BUTS:     37[004]100,171 / 22[171]200,201 / 16[125]016,017  
:              26[017]010,010  
  
:EXEC:         [375] BUPP=017 ('T' BIT TRAP)  
  
:CODES:        N / A  
  
:SYNC:         B05J2 (-)  
  
:KEY SIG:      K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H  
:              K5-5 STPM2 H  
  
T0767:  MOV      #0767,R0          ;LOAD R0 WITH TEST NO.  
        MOV      @#I0767,R1      ;LOAD R1 WITH TEST INSTRUCTION WORD  
        MOV      SP,R5          ;SAVE THE SP  
        MOV      #MBUFO,R2      ;DEST ADDR = MBUFO  
R0767:  MOV      R5,SP          ;RESTORE SP FOR ERROR LOOPING  
        MOV      #20,-(SP)      ;SET 'T' BIT IN THE NEW PSW  
        MOV      #I0767,-(SP)   ;MAKE NEW PC = I0767  
        CCC  
        RTT          ;SCOPE SYNC  
        ;SET 'T' BIT - GO TO I0767  
  
I0767:  MOV      R0,(R2)        ;MOV INSTRUCTION SHOULD SPRING TRAP  
  
E0767:  ERRORS  
        R0767          ;BUT SERVICE IN 125 FAILED  
        ;ERROR LOOP RETURN  
  
00767:  SCOPE          ;CALL SCOPE LOOP UTILITY
```

25301
25302
25303
25304
25305
25306
25307
25308
25309
25310
25311
25312
25313
25314
25315
25316
25317
25318
25319
25320
25321
25322
25323
25324
25325
25326
25327
25328
25329
25330
25331
25332
25333
25334
25335

: *****
:SBTTL T0770 BUT SERVICE TEST IN ROM LOCATION 170 - (MOV RA,RB)
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [170,204,002,015,TRAP MICROROUTINE] FC 1,4,10,6

:ACT BUTS: 37[004]100,170 / 20[170]000,002 / 26[002]010,010

:EXEC: [204] BUPP=002 ('T' BIT TRAP)

:CODES: N / A

:SYNC: B05J2 (-)

:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

062676 012700 000770
062702 013701 062726
062706 010605
062710 010506
062712 012746 000020
062716 012746 062726
062722 000257
062724 000006
062726 010003
062730 104005
062732 062710
062734 000004

T0770: MOV #0770,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0770,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R0770: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0770,-(SP) ;MAKE NEW PC = I0770
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0770

I0770: MOV R0,R3 ;MOV INSTRUCTION SHOULD SPRING TRAP

E0770: ERROR5 ;BUT SERVICE IN 170 FAILED
R0770 ;ERROR LOOP RETURN

00770: SCOPE ;CALL SCOPE LOOP UTILITY

T0770 BUT SERVICE TEST IN ROM LOCATION 170 - (MOV RA,RB)

SEQ 0634

25336
25337
25338
25339
25340
25341
25342
25343
25344
25345
25346
25347
25348
25349
25350
25351
25352
25353
25354
25355
25356
25357
25358
25359
25360
25361
25362
25363
25364
25365
25366
25367
25368
25369
25370
25371
25372
25373

: *****
:SBTTL T0771 BUT SERVICE TEST IN ROM LOCATION 160 - (MOV (RA),RB)
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [141,247,250,160,204,002,015,TRAP MICROROUTINE]
: FC 1,2,4,10,6

:ACT BUTS: 37[004]100,141 / 35[247]120,160 / 20[160]000,002
: 26[002]010,010

:EXEC: [204] BUPP=002 ('T' BIT TRAP)

:CODES: N / A

:SYNC: B05J2 (-)

:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

062736 012700 000771
062742 013701 062772
062746 010605
062750 012703 067570
062754 010506
062756 012746 000020
062762 012746 062772
062766 000257
062770 000006
062772 011304
062774 104005
062776 062754
063000 000004

T0771: MOV #0771,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0771,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV #DWTA,R3 ;SOURCE ADDR = DWTA
R0771: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0771,-(SP) ;MAKE NEW PC = I0771
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0771
I0771: MOV (R3),R4 ;MOV INSTRUCTION SHOULD SPRING TRAP
E0771: ERROR5 ;BUT SERVICE IN 160 FAILED
R0771 ;ERROR LOOP RETURN
00771: SCOPE ;CALL SCOPE LOOP UTILITY

T0771 BUT SERVICE TEST IN ROM LOCATION 160 - (MOV (RA),RB)

SEQ 0635

25374
25375
25376
25377
25378
25379
25380
25381
25382
25383
25384
25385
25386
25387
25388
25389
25390
25391
25392
25393
25394 063002 012700 000772
25395 063006 013701 063032
25396 063012 010605
25397 063014 010506
25398 063016 012746 000020
25399 063022 012746 063032
25400 063026 000257
25401 063030 000006
25402
25403 063032 110003
25404
25405 063034 104005
25406 063036 063014
25407
25408 063040 000004
25409

```
: *****  
:SBTTL T0772 BUT SERVICE TEST IN ROM LOCATION 003 - (MOVB RA,RB)  
: *****  
:MICROPROGRAMMING / LOGIC INFORMATION  
:ROM SEQ: [170,204,003,204,002,015,TRAP MICROROUTINE] FC 1,4,10,6  
:ACT BUTS: 37[004]100,170 / 20[170]000,003 / 27[003]000,002  
:          26[002]010,010  
:EXEC: [204] 2ND TIME THRU BUPP=002 ('T' BIT TRAP)  
:CODES: N / A  
:SYNC: B05J2 (-)  
:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H  
:          K5-5 STPM2 H  
T0772: MOV #0772,R0 ;LOAD R0 WITH TEST NO.  
MOV @#I0772,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
MOV SP,R5 ;SAVE THE SP  
R0772: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING  
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW  
MOV #I0772,-(SP) ;MAKE NEW PC = I0772  
CCC ;SCOPE SYNC  
RTT ;SET 'T' BIT - GO TO I0772  
I0772: MOVB R0,R3 ;MOVB INSTRUCTION SHOULD SPRING TRAP  
E0772: ERROR5 ;BUT SERVICE IN 003 FAILED  
R0772 ;ERROR LOOP RETURN  
00772: SCOPE ;CALL SCOPE LOOP UTILITY
```

T0772 BUT SERVICE TEST IN ROM LOCATION 003 - (MOVB RA,RB)

SEQ 0636

25410
25411
25412
25413
25414
25415
25416
25417
25418
25419
25420
25421
25422
25423
25424
25425
25426
25427
25428
25429
25430
25431
25432
25433
25434
25435
25436
25437
25438
25439
25440
25441
25442
25443
25444

; *****
;SBTTL T0773 BUT SERVICE TEST IN ROM LOCATION 271 - (ROR %R)
; *****

;MICROPROGRAMMING / LOGIC INFORMATION

;ROM SEQ: [106,271,274,002,015,TRAP MICROROUTINE] FC 1,9,10,6
;ACT BUTS: 37[004]100,106 / 27[271]000,002 / 26[002]010,010
;EXEC: [274] BUPP=002 ('T' BIT TRAP)
;CODES: N / A
;SYNC: B05J2 (-)
;KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
; K5-5 STPM2 H

063042 012700 000773
063046 013701 063072
063052 010605
063054 010506
063056 012746 000020
063062 012746 063072
063066 000257
063070 000006
063072 006003
063074 104005
063076 063054
063100 000004

T0773: MOV #0773,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0773,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R0773: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0773,-(SP) ;MAKE NEW PC = I0773
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0773
I0773: ROR R3 ;ROR INSTRUCTION SHOULD SPRING TRAP
E0773: ERROR5 ;BUT SERVICE IN 271 FAILED
R0773 ;ERROR LOOP RETURN
00773: SCOPE ;CALL SCOPE LOOP UTILITY

25445
25446
25447
25448
25449
25450
25451
25452
25453
25454
25455
25456
25457
25458
25459
25460
25461
25462
25463
25464
25465
25466
25467
25468
25469
25470
25471
25472
25473
25474
25475
25476
25477
25478
25479

: *****
:SBTTL T0774 BUT SERVICE TEST IN ROM LOCATION 273 - (RORB %R)
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [107,272,273,274,002,015,TRAP MICROROUTINE] FC 1,9,10,6

:ACT BUTS: [274] BUPP=002 ('T' BIT TRAP)

:EXEC: Z

:CODES: N / A

:SYNC: B05J2 (-)

:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

063102 012700 000774
063106 013701 063132
063112 010605
063114 010506
063116 012746 000020
063122 012746 063132
063126 000257
063130 000006

063132 106003

063134 104005
063136 063114

063140 000004

T0774: MOV #0774,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0774,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R0774: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0774,-(SP) ;MAKE NEW PC = I0774
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0774

I0774: RORB R3 ;RORB INSTRUCTION SHOULD SPRING TRAP

E0774: ERROR5 ;BUT SERVICE IN 273 FAILED
R0774 ;ERROR LOOP RETURN

00774: SCOPE ;CALL SCOPE LOOP UTILITY

```
25480 ; *****  
25481 ; SBTTL T0775 BUT SERVICE TEST IN ROM LOCATION 277 - (ROR (RN))  
25482 ; *****  
25483  
25484 ;MICROPROGRAMMING / LOGIC INFORMATION  
25485  
25486 ;ROM SEQ: [161,266,267,232,275,277,376,017,015,TRAP MICROROUTINE]  
25487 ; FC 1,3,9,10,6  
25488  
25489 ;ACT BUTS: 37[004]100,161 / 33[266]220,232 / 16[277]016,017  
25490 ; 26[017]010,010  
25491  
25492 ;EXEC: [376] BUPP = 017 ('T' BIT TRAP)  
25493  
25494 ;CODES: N / A  
25495  
25496 ;SYNC: B05J2 (-)  
25497  
25498 ;KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H  
25499 ; K5-5 STPM2 H  
25500  
25501 063142 012700 000775 T0775: MOV #0775,R0 ;LOAD R0 WITH TEST NO.  
25502 063146 013701 063176 MOV @#I0775,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
25503 063152 010605 MOV SP,R5 ;SAVE THE SP  
25504 063154 012703 067560 MOV #MBUF0,R3 ;DEST ADDR = MBUF0  
25505 063160 010506 R0775: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING  
25506 063162 012746 000020 MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW  
25507 063166 012746 063176 MOV #I0775,-(SP) ;MAKE NEW PC = I0775  
25508 063172 000257 CCC ;SCOPE SYNC  
25509 063174 000006 RTT ;SET 'T' BIT - GO TO I0775  
25510  
25511 063176 006013 I0775: ROR (R3) ;ROR INSTRUCTION SHOULD SPRING TRAP  
25512  
25513 063200 104005 E0775: ERROR5 ;BUT SERVICE IN 277 FAILED  
25514 063202 063160 R0775 ;ERROR LOOP RETURN  
25515  
25516 063204 000004 O0775: SCOPE ;CALL SCOPE LOOP UTILITY  
25517
```

```

25518 : *****
25519 :.SBTTL T0776 BUT SERVICE TEST IN ROM LOCATION 374 - (NEGB (RN))
25520 : *****
25521
25522 ;MICROPROGRAMMING / LOGIC INFORMATION
25523
25524 ;ROM SEQ: [161,266,267,223,253,075,374,375,017,015,TRAP MICROROUTINE]
25525 ; FC 1,3,9,8,10,6
25526
25527 ;ACT BUTS: 37[004]100,161 / 33[266]220,223 / 16[374]016,017
25528 ;
25529
25530 ;EXEC: [375] BUPP=017 ('T' BIT TRAP)
25531
25532 ;CODES: N / A
25533
25534 ;SYNC: B05J2 (-)
25535
25536 ;KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
25537 ; K5-5 STPM2 H
25538
25539 063206 012700 000776 T0776: MOV #0776,R0 ;LOAD R0 WITH TEST NO.
25540 063212 013701 063242 MOV @#I0776,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
25541 063216 010605 MOV SP,R5 ;SAVE THE SP
25542 063220 012703 067561 MOV #MBUF0+1,R3 ;DEST ADDR = MBUF0+1 (ODD)
25543 063224 010506 R0776: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
25544 063226 012746 000020 MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
25545 063232 012746 063242 MOV #I0776,-(SP) ;MAKE NEW PC = I0776
25546 063236 000257 CCC ;SCOPE SYNC
25547 063240 000006 RTT ;SET 'T' BIT - GO TO I0776
25548
25549 063242 105413 I0776: NEGB (R3) ;NEGB INSTRUCTION SHOULD SPRING TRAP
25550
25551 063244 104005 E0776: ERROR5 ;BUT SERVICE IN 374 FAILED
25552 063246 063224 R0776 ;ERROR LOOP RETURN
25553
25554 063250 000004 O0776: SCOPE ;CALL SCOPE LOOP UTILITY
25555

```

25556
25557
25558
25559
25560
25561
25562
25563
25564
25565
25566
25567
25568
25569
25570
25571
25572
25573
25574
25575
25576
25577
25578
25579
25580
25581
25582
25583
25584
25585
25586
25587
25588
25589
25590
25591
25592

: *****
:SBTTL T0777 BUT SERVICE TEST IN ROM LOCATION 306 - (JMP (RN))
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [151,300,306,313,017,015,TRAP MICROROUTINE] FC 1,5,10,6
:ACT BUTS: 37[004]100,151 / 15[151]306,306 / 16[306]016,017
: : 26[017]010,010
:EXEC: [313] BUPP=017 ('T' BIT TRAP)
:CODES: N / A
:SYNC: B05J2 (-)
:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: : K5-5 STPM2 H

063252 012700 000777
063256 013701 063306
063262 010605
063264 012702 063310
063270 010506
063272 012746 000020
063276 012746 063306
063302 000257
063304 000006
063306 000112
063310 104005
063312 063270
063314 000004

T0777: MOV #0777,R0 ;LOAD R0 WITH TEST NO.
MOV @#I0777,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
MOV #E0777,R2 ;DEST ADDR = E0777 FOR JMP
R0777: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I0777,-(SP) ;MAKE NEW PC = I0777
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I0777
I0777: JMP (R2) ;JMP INSTRUCTION SHOULD SPRING TRAP
E0777: ERROR5 ;BUT SERVICE IN 306 FAILED
R0777 ;ERROR LOOP RETURN
00777: SCOPE ;CALL SCOPE LOOP UTILITY

```
25593 : *****  
25594 :SBTTL T1000 BUT SERVICE TEST IN ROM LOCATION 110 - (BVS A)  
25595 : *****  
25596  
25597 ;MICROPROGRAMMING / LOGIC INFORMATION  
25598  
25599 ;ROM SEQ: [110,347,017,015,TRAP MICROROUTINE] FC 1,7,10,6  
25600  
25601 ;ACT BUTS: 37[004]100,110 / 16[110]016,017 / 26[017]010,010  
25602  
25603 ;EXEC: [347] BUPP=017 ('T' BIT TRAP)  
25604  
25605 ;CODES: N / A  
25606  
25607 ;SYNC: B05J2 (-)  
25608  
25609 ;KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H  
25610 : K5-5 STPM2 H  
25611  
25612 063316 012700 001000 T1000: MOV #1000,R0 ;LOAD R0 WITH TEST NO.  
25613 063322 013701 063346 MOV @#I1000,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD  
25614 063326 010605 MOV SP,R5 ;SAVE THE SP  
25615 063330 010506 R1000: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING  
25616 063332 012746 000020 MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW  
25617 063336 012746 063346 MOV #I1000,-(SP) ;MAKE NEW PC = I1000  
25618 063342 000257 CCC ;SCOPE SYNC  
25619 063344 000006 RTT ;SET 'T' BIT - GO TO I1000  
25620  
25621 063346 102400 I1000: BVS E1000 ;BVS INSTRUCTION SHOULD SPRING TRAP  
25622  
25623 063350 104005 E1000: ERROR5 ;BUT SERVICE IN 110 FAILED  
25624 063352 063330 R1000 ;ERROR LOOP RETURN  
25625  
25626 063354 000004 O1000: SCOPE ;CALL SCOPE LOOP UTILITY  
25627
```

25628
25629
25630
25631
25632
25633
25634
25635
25636
25637
25638
25639
25640
25641
25642
25643
25644
25645
25646
25647
25648
25649
25650
25651
25652
25653
25654
25655
25656
25657
25658
25659
25660
25661
25662

: *****
:SBTTL T1001 BUT SERVICE TEST IN ROM LOCATION 340 - (BR A)
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [111,340,341,017,015,TRAP MICROROUTINE] FC 1,7,10,6
:ACT BUTS: 37[004]100,111 / 16[340]016,017 / 26[017]010,010
:EXEC: [341] BUPP=017 ('T' BIT TRAP)
:CODES: N / A
:SYNC: B05J2 (-)
:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

063356 012700 001001
063362 013701 063406
063366 010605
063370 010506
063372 012746 000020
063376 012746 063406
063402 000257
063404 000006
063406 000400
063410 104005
063412 063370
063414 000004

T1001: MOV #1001,R0 ;LOAD R0 WITH TEST NO.
MOV @#I1001,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R1001: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I1001,-(SP) ;MAKE NEW PC = I1001
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I1001
I1001: BR E1001 ;BR INSTRUCTION SHOULD SPRING TRAP
E1001: ERROR5 ;BUT SERVICE IN 340 FAILED
R1001 ;ERROR LOOP RETURN
01001: SCOPE ;CALL SCOPE LOOP UTILITY

T1001 BUT SERVICE TEST IN ROM LOCATION 340 - (BR A)

SEQ 0643

25663
25664
25665
25666
25667
25668
25669
25670
25671
25672
25673
25674
25675
25676
25677
25678
25679
25680
25681
25682
25683
25684
25685
25686
25687
25688
25689
25690
25691
25692
25693
25694
25695
25696

: *****
:SBTTL T1002 BUT SERVICE TEST IN ROM LOCATION 350 - (CCC)
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [116,350,351,017,015,TRAP MICROROUTINE] FC 1,7,10,6

:ACT BUTS: 37[004]100,116 / 16[350]016,017 / 26[017]010,010

:EXEC [351] BUPP=017 ('T' BIT TRAP)

:CODES: N / A

:SYNC: B05J2 (-)

:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

063416 012700 001002
063422 013701 063444
063426 010605
063430 010506
063432 012746 000020
063436 012746 063444
063442 000006
063444 000257
063446 104005
063450 063430
063452 000004

T1002: MOV #1002,R0 ;LOAD R0 WITH TEST NO.
MOV @#I1002,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R1002: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I1002,-(SP) ;MAKE NEW PC = I1002
RTT ;SET 'T' BIT - GO TO I1002
I1002: CCC ;CCC INSTRUCTION SHOULD SPRING TRAP
E1002: ERROR5 ;BUT SERVICE IN 350 FAILED
R1002 ;ERROR LOOP RETURN
O1002: SCOPE ;CALL SCOPE LOOP UTILITY

25697
25698
25699
25700
25701
25702
25703
25704
25705
25706
25707
25708
25709
25710
25711
25712
25713
25714
25715
25716
25717
25718
25719
25720
25721
25722
25723
25724
25725
25726
25727
25728
25729
25730

063454 012700 001003
063460 013701 063502
063464 010605
063466 010506
063470 012746 000020
063474 012746 063502
063500 000006
063502 000277
063504 104005
063506 063466
063510 000004

: *****
:SBTTL T1003 BUT SERVICE TEST IN ROM LOCATION 117 - (SCC)
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [117,352,017,015,TRAP MICROROUTINE] FC 1,7,10,6

:ACT BUTS: 37[004]100,117 / 16[117]016,017 / 26[017]010,010

:EXEC: [352] BUPP=017 ('T' BIT TRAP)

:CODES: N / A

:SYNC: B05J2 (-)

:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

T1003: MOV #1003,R0 ;LOAD R0 WITH TEST NO.
MOV @#I1003,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R1003: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I1003,-(SP) ;MAKE NEW PC = I1003
RTT ;SET 'T' BIT - GO TO I1003

I1003: SCC ;SCC INSTRUCTION SHOULD SPRING TRAP

E1003: ERROR5 ;BUT SERVICE IN 117 FAILED
R1003 ;ERROR LOOP RETURN

O1003: SCOPE ;CALL SCOPE LOOP UTILITY

25731
25732
25733
25734
25735
25736
25737
25738
25739
25740
25741
25742
25743
25744
25745
25746
25747
25748
25749
25750
25751
25752
25753
25754
25755
25756
25757
25758
25759
25760
25761
25762
25763
25764
25765
25766

063512 012700 001004
063516 013701 063546
063522 010605
063524 010506
063526 012746 063550
063532 012746 000020
063536 012746 063546
063542 000257
063544 000006
063546 000207
063550 104005
063552 063524
063554 000004

: *****
:SBTTL T1004 BUT SERVICE TEST IN ROM LOCATION 324 - (RTS PC)
: *****
:MICROPROGRAMMING / LOGIC INFORMATION
:ROM SEQ: [124,323,324,325,017,015,TRAP MICROPROUTINE] FC 1,6,10,6
:ACT BUTS: 37[004]100,124 / 16[324]016,017 / 26[017]010,010
:EXEC: [325] BUPP=017 ('T' BIT TRAP)
:CODES: N / A
:SYNC: B05J2 (-)
:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

T1004: MOV #1004,R0 ;LOAD R0 WITH TEST NO.
MOV @#I1004,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R5 ;SAVE THE SP
R1004: MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
MOV #E1004,-(SP) ;RTS WILL LOAD PC WITH E1004
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I1004,.(SP) ;MAKE NEW PC = I1004
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I1004
I1004: RTS PC ;RTS INSTRUCTION SHOULD SPRING TRAP
E1004: ERROR5 ;BUT SERVICE IN 324 FAILED
R1004 ;ERROR LOOP RETURN
O1004: SCOPE ;CALL SCOPE LOOP UTILITY

25767
25768
25769
25770
25771
25772
25773
25774
25775
25776
25777
25778
25779
25780
25781
25782
25783
25784
25785
25786
25787
25788
25789
25790
25791
25792
25793
25794
25795
25796
25797
25798
25799
25800
25801
25802
25803
25804
25805
25806

```

: *****
:SBTTL T1005 BUT SERVICE TEST IN ROM LOCATION 345 - (SOB RN,A)
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ:      [130,342,343,345,347,017,015,TRAP MICROROUTINE] FC 1,7,10,6
:ACT BUTS:     37[004]100,130 / 16[345]016,017 / 26[017]010,010
:EXEC:         [347] BUPP=017 ('T' BIT TRAP)
:CODES:        N / A
:SYNC:         B05J2 (-)
:KEY SIG:      K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
:              K5-5 STPM2 H

```

```

063556 012700 001005
063562 013701 063626
063566 032737 100000 066642
063574 001401
063576 000000
063600 010605
063602 010506
063604 012703 000001
063610 012746 000020
063614 012746 063626
063620 000257
063622 000006
063624 000401
063626 077302
063630 104005
063632 063602
063634 000004

```

```

T1005: MOV #1005,R0 ;LOAD R0 WITH TEST NO.
        MOV @#I1005,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
        BIT #100000,@#BPTLOC ;BREAKPOINT HALT SET ??
        BEQ .+4 ;BR IF NOT
        HALT ;BREAK-DEPRESS CONTINUE TO RESTART
R1005: MOV SP,R5 ;SAVE THE SP
        MOV R5,SP ;RESTORE SP FOR ERROR LOOPING
        MOV #1,R3 ;SOB COUNT = +1
        MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
        MOV #I1005,-(SP) ;MAKE NEW PC = I1005
        CCC ;SCOPE SYNC
        RTT ;SET 'T' BIT - GO TO I1005
        BR E1005
I1005: SOB R3,I1005-2 ;SOB INSTRUCTION SHOULD SPRING TRAP
E1005: ERRORS ;BUT SERVICE IN 345 FAILED
        R1005 ;ERROR LOOP RETURN
O1005: SCOPE ;CALL SCOPE LOOP UTILITY

```


T1006 BUT SERVICE TEST IN ROM LOCATION 344 - (SOB RA,A)

SEQ 0648

25846
25847
25848
25849
25850
25851
25852
25853
25854
25855
25856
25857
25858
25859
25860
25861
25862
25863
25864
25865
25866
25867
25868
25869
25870
25871
25872
25873
25874
25875
25876
25877
25878
25879
25880
25881
25882
25883
25884
25885
25886
25887
25888
25889
25890
25891
25892
25893
25894
25895

: *****
:SBTTL T1007 BUT SERVICE TEST IN ROM LOCATION 356 - (MARK 0)
: *****

:MICROPROGRAMMING / LOGIC INFORMATION

:ROM SEQ: [112,353,354,355,356,357,017,015,TRAP MICROROUTINE]
: FC 1,5,10,6
:ACT BUTS: 37[004]100,112 / 016[356]016,017 / 26[017]010,010
:EXEC: [357] BUPP = 017 ('T' BIT TRAP)
:CODES: N / A
:SYNC: B05J2 (-)
:KEY SIG: K5-2 PS (T) (1) H / K3-7 SERVICE H / K5-5 STPM3 H
: K5-5 STPM2 H

T1007: MOV #1007,R0 ;LOAD R0 WITH TEST NO.
MOV @#I1007,R1 ;LOAD R1 WITH TEST INSTRUCTION WORD
MOV SP,R4 ;SAVE THE SP
R1007: MOV R4,SP ;RESTORE SP FOR ERROR LOOPING
MOV #20,-(SP) ;SET 'T' BIT IN THE NEW PSW
MOV #I1007,-(SP) ;MAKE NEW PC = I1007
MOV #A1007,R5 ;MARK GOES TO A1007 IF TRAP NOT SPRUNG
CCC ;SCOPE SYNC
RTT ;SET 'T' BIT - GO TO I1007
I1007: MARK+1 ;MRK INSTRUCTION SHOULD SPRING TRAP
0 ;'T' BIT SERVICE WILL PUSH THE 'PSW'
0 ;AND THE 'PC' IN THESE LOCATIONS
BR A1007 ;JUST IN CASE MARK FAILS
A1007: MOV R4,SP ;RESET THE SP
BR E1007 ;GO REPORT ERROR
BR B1007 ;'T' TRAP WORKED - GO TO EXIT
E1007: ERROR5 ;MRK FAILED TO SPRING 'T' TRAP
R1007 ;ERROR LOOP RETURN ADDRESS
B1007: MOV R4,SP ;RESET THE SP IF ALL OK
O1007: SCOPE ;CALL SCOPE LOOP UTILITY

25896
25897
25898
25899
25900
25901
25902
25903
25904
25905
25906
25907
25908
25909
25910
25911
25912
25913
25914
25915
25916
25917
25918
25919
25920
25921
25922
25923
25924
25925
25926
25927
25928
25929
25930
25931
25932
25933
25934
25935
25936
25937
25938
25939
25940
25941
25942
25943
25944
25945
25946
25947
25948
25949
25950
25951

: *****
: .SBTTL T1010 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:

```

:ALUADD:      NULL
:              SRC OP1
:              DST OP1
:              SUM1
:              SRC OP2
:              DST OP2
:              SUM2
:              ETC.

```

: UPON DETECTION OF AN ERROR THE PRINTOUT HAS THE FOLLOWING SIG-
: NIFICANCE IN COLUMNS 5-8:

```

:      COL5 [R1] = SOURCE OPR
:      COL6 [R2] = DEST OPR
:      COL7 [R3] = WAS SUM
:      COL8 [R4] = S / B SUM

```

: 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.

```

T1010: MOV      #1010,R0          ;LOAD R0 WITH TEST NO.
        MOV      #ALUADD,R5     ;R5 POINTS TO TABLE OF NO.S
L1010: TST      (R5)+           ;POINT TO A SRC OP
        CMP      #ALUADD+62,R5  ;DONE ALL NO.S IN TABLE ?
        BEQ      01010          ;BR IF YES
        MOV      (R5)+,R1       ;LOAD SRC OP
        MOV      (R5)+,R3       ;LOAD DEST OP
        CCC
I1010: ADD      R1,R3           ;TEST THE ADD FUNCTION

        CMP      (R5),R3        ;CORRECT SUM ?
        BEQ      L1010          ;GO ADD NEXT PAIR IF YES

```

063772	012700	001010
063776	012705	067604
064002	005725	
064004	022705	067666
064010	001416	
064012	012501	
064014	012503	
064016	000257	
064020	060103	
064022	021503	
064024	001766	

```

25952
25953 064026 011504          MOV      (R5),R4          ;GET S / B SUM
25954 064030 014502          MOV      -(R5),R2        ;GET DEST OP
25955 064032 104000          E1010:  ERROR          ;ALU ADD OPERATION FAILED
25956 064034 064042          R1010:  R1010          ;ERROR LOOP RETURN ADDRESS
25957
25958 064036 005725          TST      (R5)+          ;CORRECT R5 POINTER
25959 064040 000760          BR       L1010          ;GO DO NEXT PAIR
25960
25961 064042 024545          R1010:  CMP      -(R5),-(R5) ;RESET R5 TO POINT TO BAD GUYS
25962 064044 000756          BR       L1010          ;GO REPEAT FAILING PAIR
25963
25964 064046 000004          01010:  SCOPE          ;CALL SCOPE LOOP UTILITY
25965

```

25966
25967
25968
25969
25970
25971
25972
25973
25974
25975
25976
25977
25978
25979
25980
25981
25982
25983
25984
25985
25986
25987
25988
25989
25990
25991
25992
25993
25994
25995
25996
25997
25998
25999
26000
26001
26002
26003
26004
26005
26006
26007
26008
26009
26010
26011
26012
26013
26014
26015
26016
26017
26018
26019
26020
26021

: *****
: .SBTTL T1011 ALU SUB FUNCTION TEST
: *****

: THIS TEST VERIFIES THAT THE ALU ADD FUNCTION CAN RESPOND CORRECTLY
: TO THE 8 POSSIBLE COMBINATIONS THAT COULD OCCUR AT THE INPUTS OF
: EACH OF THE 16 BIT POSITIONS AS DESCRIBED BELOW:

	AIN	BIN	CIN
:	0	0	0
:	0	0	1
:	0	1	0
:	0	1	1
:	1	0	0
:	1	0	1
:	1	1	0
:	1	1	1

: 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.

```

: UPON DETECTION OF AN ERROR THE PRINTOUT HAS THE FOLLOWING SIG-
: NIFICANCE IN COLUMNS 5-8:

```

:          COL5 [R1] = SOURCE OPR
:          COL6 [R2] = DEST OPR
:          COL7 [R3] = WAS DIFFERENCE
:          COL8 [R4] = S / B DIFFERENCE

```

: 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.

```

26008 064050 012700 001011
26009 064054 012705 070032
26010 064060 005725
26011 064062 022705 070114
26012 064066 001416
26013 064070 012501
26014 064072 012503
26015 064074 000257
26016
26017 064076 160103
26018
26019 064100 021503
26020 064102 001766
26021

```

```

T1011: MOV #1011,R0          ;LOAD R0 WITH TEST NO.
        MOV #ALUSUB,R5      ;R5 POINTS TO TABLE OF NO.S
L1011:  TST (R5)+           ;POINT TO A SRC OP
        CMP #ALUSUB+62,R5   ;DONE ALL NO.S IN TABLE ?
        BEQ 01011          ;BR IF YES
        MOV (R5)+,R1        ;LOAD SRC OP
        MOV (R5)+,R3        ;LOAD DEST OP
        CCC                 ;SCOPE SYNC

I1011:  SUB R1,R3           ;TEST THE SUB FUNCTION

        CMP (R5),R3        ;CORRECT DIFF. ?
        BEQ L1011          ;GO SUB NEXT PAIR IF YES

```

26022	064104	011504		MOV	(R5),R4	:GET S / B DIFF
26023	064106	014502		MOV	-(R5),R2	:GET DEST OP
26024	064110	104000	E1011:	ERROR		:ALU SUB OPERATION FAILED
26025	064112	064120		R1011		:ERROR LOOP RETURN ADDRESS
26026						
26027	064114	005725		TST	(R5)+	:CORRECT R5 POINTER
26028	064116	000760		BR	L1011	:GO DO NEXT PAIR
26029						
26030	064120	024545	R1011:	CMP	-(R5),-(R5)	:RESET R5 TO POINT TO BAD GUYS
26031	064122	000756		BR	L1011	:GO REPEAT FAILING PAIR
26032						
26033	064124	000004	01011:	SCOPE		:CALL SCOPE LOOP UTILITY
26034						
26035						
26036						
26037						
26038						

26039
26040
26041
26042
26043
26044
26045
26046
26047
26048
26049
26050
26051
26052
26053
26054
26055
26056
26057
26058
26059
26060
26061
26062
26063
26064
26065
26066
26067
26068
26069
26070
26071
26072
26073
26074
26075
26076
26077
26078
26079
26080
26081
26082
26083
26084
26085
26086
26087
26088
26089
26090
26091
26092
26093
26094

; *****
; .SBTTL T1012 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
:000000	177777	177777
:177777	000000	000000
:125252	125252	000000
:052525	052525	000000
:125252	052525	052525
:052525	125252	125252

; THE 8 PAIRS OF NO.S AND THE ANSWERS ARE STORED IN A TEBLE TAGGED
; 'ANDTAB' IN THE FOLLOWING PATTERN:

:ANDTAB: NULL
: SRC OP1
: DST OP1
: ANS1
: SRC OP2
: DST OP2
: ANS2
: ETC.

; WHEN AN ERROR IS REPORTED THE PRINTOUT IN COL. 5-8 HAS THE
; FOLLOWING SIGNIFICANCE:

: COL 5 [R1] = SOURCE OPR
: COL 6 [R2] = DEST OPR
: COL 7 [R3] = WAS ANSWER
: COL 8 [R4] = S / B ANSWER

; 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

T1012: MOV #1012,R0 ;LOAD R0 WITH TEST NO.
MOV #ANDTAB,R5 ;R5 POINTS TO TABLE OF TEST NO.S
L1012: TST (R5)+ ;POINT TO A SOURCE OPR
CMP #ANDTAB+62,R5 ;DONE ALL COMBINATIONS ?
BEQ 01012 ;BR IF YES
MOV (R5)+,R1 ;LOAD THE SRC OP
MOV (R5)+,R3 ;LOAD THE DEST OP
CCC ;SCOPE SYNC
I1012: BIC R1,R3 ;TEST THE 'AND'
CMP R3,(R5) ;RESULT CORRECT ?
BEQ L1012 ;BR IF YES - GET THE NEXT PAIR

064126 012700 001012
064132 012705 067666
064136 005725
064140 022705 067750
064144 001416
064146 012501
064150 012503
064152 000257
064154 040103
064156 020315
064160 001766

26095	064162	011504		MOV	(R5),R4		:GET THE S / B DATA
26096	064164	014502		MOV	-(R5),R2		:GET DEST OP
26097	064166	104000	E1012:	ERROR			:ALU 'AND' FAILED
26098	064170	064176		R1012			:ERROR LOOP RETURN
26099							
26100	064172	005725		TST	(R5)+		:CORRECT R5 POINTER
26101	064174	000760		BR	L1012		:GO GET NEXT PAIR
26102							
26103	064176	024545	R1012:	CMP	-(R5),-(R5)		:RESET R5 TO POINT BACK TO BAD GUYS
26104	064200	000756		BR	L1012		:GO REPEAT THE BAD GUYS
26105							
26106	064202	000004	01012:	SCOPE			:CALL SCOPE LOOP UTILITY
26107							
26108							

26109
26110
26111
26112
26113
26114
26115
26116
26117
26118
26119
26120
26121
26122
26123
26124
26125
26126
26127
26128
26129
26130
26131
26132
26133
26134
26135
26136
26137
26138
26139
26140
26141
26142
26143
26144
26145
26146
26147
26148
26149
26150
26151
26152
26153
26154
26155
26156
26157
26158
26159
26160
26161
26162
26163
26164

: *****
: .SBTTL T1013 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
:000000	000000	000000
:177777	177777	177777
:000000	177777	177777
:177777	000000	177777
:125252	125252	125252
:052525	052525	052525
:125252	052525	177777
:052525	125252	177777

: 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.

: WHEN AN ERROR IS REPORTED THE PRINTOUT IN COL. 5-8 HAS THE
: FOLLOWING SIGNIFICANCE:

: COL 5 [R1] = SOURCE OPR
: COL 6 [R2] = DEST OPR
: COL 7 [R3] = WAS ANSWER
: COL 8 [R4] = S / B ANSWER

: 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

```

T1013: MOV #1013,R0 ;LOAD R0 WITH TEST NO.
MOV #ORTAB,R5 ;R5 POINTS TO TABLE OF TEST NO.S
L1013: TST (R5)+ ;POINT TO A SOURCE OPR
CMP #ORTAB+62,R5 ;DONE ALL COMBINATIONS ?
BEQ 01013 ;BR IF YES
MOV (R5)+,R1 ;LOAD THE SRC OP
MOV (R5)+,R3 ;LOAD THE DEST OP
CCC ;SCOPE SYNC

I1013: BIS R1,R3 ;TEST THE 'OR'

CMP R3,(R5) ;RESULT CORRECT ?
BEQ L1013 ;BR IF YES - GET THE NEXT PAIR

```

064204	012700	001013
064210	012705	067750
064214	005725	
064216	022705	070032
064222	001416	
064224	012501	
064226	012503	
064230	000257	
064232	050103	
064234	020315	
064236	001766	

```

26165 064240 011504          MOV      (R5),R4          ;GET THE S / B DATA
26166 064242 014502          MOV      -(R5),R2        ;GET DEST OP
26167 064244 104000          E1013:  ERROR          ;ALU 'DR' FAILED
26168 064246 064254          R1013:  R1013          ;ERROR LOOP RETURN
26169
26170 064250 005725          TST      (R5)+          ;CORRECT R5 POINTER
26171 064252 000760          BR       L1013          ;GO GET NEXT PAIR
26172
26173 064254 024545          R1013:  CMP      -(R5),-(R5) ;RESET R5 TO POINT BACK TO BAD GUYS
26174 064256 000756          BR       L1013          ;GO REPEAT THE BAD GUYS
26175
26176 064260 000004          01013:  SCOPE          ;CALL SCOPE LOOP UTILITY
26177
26178
26179
26180
26181
26182
26183

```

26184
26185
26186
26187
26188
26189
26190
26191
26192
26193
26194
26195
26196
26197
26198
26199
26200
26201
26202
26203
26204
26205
26206
26207
26208
26209
26210
26211
26212 064262 012700 001014
26213 064266 005001
26214 064270 005002
26215 064272 005004
26216 064274 010203
26217 064276 000257
26218
26219 064300 060103
26220
26221
26222 064302 020403
26223 064304 001403
26224
26225 064306 104000
26226 064310 064274
26227
26228 064312 000404
26229
26230 064314 005201
26231 064316 100402
26232 064320 005302
26233 064322 000764
26234
26235 064324 000004

```

: *****
: .SBTTL T1014 INC / DEC / ADD TEST - CYCLE NO.S 000000-077777
: *****
: 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 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 SCOPE LOOP UTILITY IAS CALLED TO REINITIALIZE THE TEST
- : 2. SW09=1 THE ROUTINE LOCKS ON THE FAILING PAIR OF OPERANDS UNTIL THE ERROR GOES AWAY OR SW09 IS RESET

: THE SIGNIFICANCE OF THE PRINTOUT IN COLUMNS 5 - 8:

```

: COLUMN 5 [R1] SOURCE OP
: COLUMN 6 [R2] DEST OP
: COLUMN 7 [R3] WAS ANSWER
: COLUMN 8 [R4] S / B ANSWER (ALWAYS 000000)

```

```

T1014: MOV #1014,R0 ;LOAD R0 WITH TEST NO.
CLR R1 ;INITIALIZE REGS TO 000000
CLR R2
CLR R4
R1014: MOV R2,R3 ;LOAD DEST OPERAND
CCC ;SCOPE SYNC
I1014: ADD R1,R3 ;ADD THE TWO TEST NO.S
;RESULT S / B = 000000
CMP R4,R3 ;RESULT = 000000 ?
BEQ A1014 ;BR IF YES
E1014: ERROR ;INCORRECT RESULT IN R3
R1014 ;ERROR LOOP RETURN
BR 01014 ;GO TO SCOPE EXIT
A1014: INC R1 ;ADD 1 TO SOURCE OP
BMI 01014 ;GET OUT IF IT WENT NEGATIVE
DEC R2 ;SUB 1 FROM THE DEST OP
BR R1014 ;GO ADD THE TWO NO.S
01014: SCOPE ;CALL SCOPE LOOP UTILITY

```

26236
26237
26238
26239
26240
26241
26242
26243
26244
26245
26246
26247
26248
26249
26250
26251
26252
26253
26254
26255
26256
26257
26258
26259
26260
26261
26262
26263
26264
26265
26266
26267
26268
26269
26270
26271
26272
26273
26274
26275
26276
26277
26278
26279
26280
26281
26282
26283
26284
26285
26286
26287
26288

: *****
: .SBTTL T1015 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:

- : 1. SW09=0 THE SCOPE LOOP UTILITY IAS CALLED TO REINITIALIZE
: THE TEST
- : 2. SW09=1 THE ROUTINE LOCKS ON THE FAILING PAIR OF OPERANDS
: UNTIL THE ERROR GOES AWAY OR SW09 IS RESET

: THE SIGNIFICANCE OF THE PRINTOUT IN COLUMNS 5 - 8:

: COLUMN 5 [R1] SOURCE OP
 : COLUMN 6 [R2] DEST OP
 : COLUMN 7 [R3] WAS ANSWER
 : COLUMN 8 [R4] S / B ANSWER (ALWAYS 000000)

064326 012700 001015
 064332 005001
 064334 005002
 064336 005004
 064340 010203
 064342 000257
 064344 060103
 064346 020403
 064350 001403
 064352 104000
 064354 064340
 064356 000404
 064360 005202
 064362 100402
 064364 005301
 064366 000764
 064370 000004

T1015: MOV #1015,R0 ;LOAD R0 WITH TEST NO.
 CLR R1 ;INITIALIZE REGS TO 000000
 CLR R2
 CLR R4
 R1015: MOV R2,R3 ;LOAD DEST OPERAND
 CCC ;SCOPE SYNC
 I1015: ADD R1,R3 ;ADD THE TWO TEST NO.S
 ;RESULT S / B = 000000
 CMP R4,R3 ;RESULT = 000000 ?
 BEQ A1015 ;BR IF YES
 E1015: ERROR ;INCORRECT RESULT IN R3
 R1015 ;ERROR LOOP RETURN
 BR 01015 ;GO TO SCOPE EXIT
 A1015: INC R2 ;ADD 1 TO DEST. OP
 BMI 01015 ;GET OUT IF IT WENT NEGATIVE
 DEC R1 ;SUB 1 FROM THE SOURCE OP
 BR R1015 ;GO ADD THE TWO NO.S
 01015: SCOPE ;CALL SCOPE LOOP UTILITY

```

26289 ; *****
26290 ; .SBTTL END OF PASS SERVICE ROUTINE
26291 ; *****
26292
26293 064372 010037 066706 ENDPS: MOV R0,@#LAST ;SAVE LAST TEST NO. FOR
26294 ;MISSED TEST ERROR CHECK
26295 064376 004737 064564 JSR PC,@#MISS ;GO CHECK FOR MISSED TESTS
26296 064402 005237 066670 1$: INC @#PASCNT ;UPDATE THE PASS COUNTER
26297 064406 032737 010000 177570 BIT #SW12,@#SR ;INHIBIT END PASS PRINTOUT ?
26298 064414 001035 BNE 6$ ;BR IF YES
26299 064416 104400 TYPE
26300 064420 067147 EOP1
26301 064422 013702 066670 MOV @#PASCNT,R2 ;CONVERT AND PRINT PASS COUNT
26302 064426 004737 065760 JSR PC,@#OTA
26303 064432 104400 TYPE
26304 064434 067136 DIGITS
26305 064436 104400 TYPE
26306 064440 067176 EOP2
26307 064442 013702 066666 MOV @#ERRCNT,R2 ;CONVERT AND PRINT ERROR COUNT
26308 064446 004737 065760 JSR PC,@#OTA
26309 064452 104400 TYPE
26310 064454 067136 DIGITS
26311 064456 005737 066672 TST @#PFCNT ;ANY POWER FAILS LOGGED THIS PASS?
26312 064462 001410 BEQ 2$ ;BR IF NONE
26313 064464 013702 066672 MOV @#PFCNT,R2 ;GET THE PWR FAIL COUNT
26314 064470 004737 065760 JSR PC,@#OTA ;GO CONVERT PFCNT
26315 064474 104400 TYPE
26316 064476 067163 PFMESS
26317 064500 104400 TYPE ;TYPE 'PFCNT = NNNNNN
26318 064502 067136 DIGITS
26319 064504 104400 2$: TYPE
26320 064506 066726 CRLF
26321 064510 012737 000040 066662 6$: MOV #32.,@#ICOUNT
26322 064516 012737 000040 066664 MOV #32.,@#ITCNT
26323 064524 012701 066674 PFRET: MOV #PRIFLG,R1 ;R1 POINTS TO BEGINNING OF FLAGS
26324 064530 005021 1$: CLR (R1)+ ;CLEAR A FLAG
26325 064532 022701 066722 CMP #ONCE,R1 ;ALL DONE?
26326 064536 001374 BNE 1$ ;BR IF NOT
26327 064540 000005 RESET ;CLEAR THE WORLD PRIOR TO RESTART
26328 064542 013702 000042 MOV @#42,R2 ;CHECK FOR XXDP/ACT11 HOOK
26329 064546 001404 BEQ DONE1 ;BR IF NO HOOK FOUND
26330 064550 004712 LOGICAL:JSR PC,(R2) ;RETURN TO XXDP OR ACT MONITOR
26331 064552 000240 NOP
26332 064554 000240 NOP
26333 064556 000240 NOP
26334 064560 000137 003034 DONE1: JMP @#BEGIN ;GO RESTART AT THE BEGINNING

```

```

26335 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
26336 .SBTTL / / / / / UTILITIES / / / / /
26337 ;XXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXXX
26338 .SBTTL
26339 ; *****
26340 .SBTTL SUBROUTINE TO CHECK FOR AND REPORT MISSED TESTS
26341 ; *****
26342
26343 ;THERE IS A BYTE TABLE TAGGED 'STAB1' THAT IS MAINTAINED BY THE SCOPE
26344 ;LOOP SERVICE ROUTINE. EACH TIME A TEST IN THE 'CIT' OR 'IEX' SECTION
26345 ;IS ENTERED A CORRESPONDING BYTE IN THE TABLE IS SET TO 377.
26346 ;DURING THE 'BIT' SECTION OF THE PROGRAM A 'MOVB #377,STAB1(R0)'
26347 ;IS USED TO FLAG EACH TEST ENTERED SINCE THE 'SCOPE' TRAP HAS NOT BEEN
26348 ;VERIFIED YET.
26349 ;EACH ENTRY INDEX CORRESPONDS TO AN OCTAL TEST #. THE TABLE IS CLEARED
26350 ;UPON ENTRY INTO THE 'CIT' SECTION AND MAINTAINED BY THE SCOPE LOOP
26351 ;SERVICE UNTIL END OF PASS SERVICE. THIS ROUTINE IS CALLED THEN TO
26352 ;SCAN THE TABLE AND REPORT ANY MISSED TESTS. THIS ERROR CHECKING CAN
26353 ;BE INHIBITED BY SETTING SW12 TO A'1'. THE ERROR PRINTOUT HAS THE
26354 ;FORMAT SHOWN BELOW:
26355
26356
26357 : MISSED TEST
26358 : NNN
26359 : MMM
26360 : XXX
26361
26362 : ETC
26363
26364 : WHERE: THE HEADER IS PRINTED ONLY ONCE AND NNN,MMM,XXX,
26365 : ETC ARE THE OCTAL NO.S OF THE MISSED TESTS.
26366
26367
26368 064564 032737 010000 177570 MISS: BIT #SW12,@#SR ;INHIBIT MISSED TEST PRINTOUT ?
26369 064572 001044 BNE 5$ ;BR IF YES
26370 064574 032737 020000 177570 BIT #SW13,@#SR ;INHIBIT PRINT ?
26371 064602 001040 BNE 5$ ;BR IF YES
26372 064604 013700 066704 MOV @#FIRST,R0 ;USE TEST NO. AS INDEX TO ERROR TABLE
26373 064610 013701 066706 MOV @#LAST,R1 ;USE [R1] TO INDICATE END OF MISSED
26374 ;TEST ERROR TABLE
26375 064614 122760 000377 070140 2$: CMPB #377,STAB1(R0) ;WAS TEST FLAG = 377 ?
26376 064622 001424 BEQ 4$ ;BR IF YES - TEST EXECUTED
26377 064624 005737 066710 TST @#MISFLG ;HEADER PRINTED ?
26378 064630 001004 BNE 3$ ;BR IF YES - PRINT ONLY ONCE
26379 064632 104400 TYPE ;GO TYPE 'MISSED TESTS'
26380 064634 067455 MISHDR
26381 064636 005137 066710 COM @#MISFLG ;SET FLAG TO PREVENT PRINTING AGAIN
26382 064642 010002 3$: MOV R0,R2 ;GET THE TEST NO.
26383 064644 004737 065760 JSR PC,@#OTA ;GO CONVERT AND PRINT THE NO.
26384 064650 104400 TYPE
26385 064652 067136 DIGITS
26386 064654 104400 TYPE
26387 064656 066726 CRLF
26388 064660 022737 177777 066666 CMP #-1,@#ERRCNT ;MAX ERROR COUNT ??
26389 064666 001402 BEQ 4$ ;BR IF YES
26390 064670 005237 066666 INC @#ERRCNT ;COUNT THE ERROR

```

26391	064674	020001
26392	064676	001402
26393	064700	005200
26394	064702	000744
26395	064704	000207
26396		

4\$:	CMP	R0,R1
	BEQ	5\$
	INC	R0
	BR	2\$
5\$:	RTS	PC

```

:DONE ALL TABLE ENTRIES ?
:BR IF YES
:INDEX POINTS TO NEXT TEST FLAG
:GO CHECK THE NEXT FLAG
:RETURN TO END OF PASS SERVICE

```

```

26397 ; *****
26398 ; .SBTTL POWER FAIL SUBROUTINE
26399 ; *****
26400
26401 064706 012737 064746 000024 PDWN: MOV #PUP,@#24 ;SET UP POWER FAIL VECTOR TO GO
26402 064714 012737 000340 000026 MOV #340,@#26 ;TO PUP ON POWER UP - LEVEL 7
26403 064722 005237 066672 INC @#PFCNT ;COUNT THE POWER FAIL
26404 064726 022737 177777 066666 CMP #-1,@#ERRCNT ;MAX ERROR COUNT ??
26405 064734 001402 BEQ 1$ ;BR IF YES
26406 064736 005237 066666 INC @#ERRCNT ;COUNT THE ERROR
26407 064742 000000 1$: HALT ;PWR RESTART SHOULD GO TO 'PUP'
26408 064744 000776 BR 1$ ;HANG IF CONTINUE DEPRESSED
26409
26410 064746 012737 000340 177776 PUP: MOV #340,@#PSW ;SET PRIORITY TO LEVEL 7 JUST IN CASE
26411 064754 012706 001000 MOV #BT001,SP ;RESET THE STACK POINTER
26412 064760 005037 067560 CLR @#MBUF0 ;INIT STALL COUNTER
26413 064764 005337 067560 1$: DEC @#MBUF0 ;COUNT ONE TIME
26414 064770 001375 BNE 1$ ;BR IF NOT BACK TO 000000
26415 064772 012737 064706 000024 MOV #PDWN,@#24 ;SET UP POWER FAIL VECTOR
26416 065000 012737 000340 000026 MOV #340,@#26 ;PRIORITY LEVEL 7
26417 065006 032737 010000 177570 BIT #SW12,@#SR ;INHIBIT POWER FAIL MESSAGE ?
26418 065014 001002 BNE 2$ ;BR IF YES
26419 065016 104400 TYPE ;GO PRINT POWER MESSAGE
26420 065020 067416 PFMSG
26421 065022 000137 064524 2$: JMP @#PFRET ;GO CLEANUP AND ATTEMPT RESTART
26422
26423
26424 ; *****
26425 ; .SBTTL 'T' BIT SERVICE ROUTINE
26426 ; *****
26427
26428 065026 062716 000004 TBSE: ADD #4,(SP) ;MOVE RETURN PC AROUND ERROR WORDS
26429 065032 042766 000020 000002 BIC #20,2(SP) ;TURN OFF THE 'T' BIT
26430 065040 000006 RTT ;RETURN TO THE CALLING TEST
26431
26432

```

```

26433 ; *****
26434 ; .SBTTL RSVD INSTRUCTION TRAP SERVICE ROUTINE
26435 ; *****
26436
26437 ; THIS ROUTINE SERVICES UNEXPECTED RESERVED INSTRUCTION TRAP ERRORS
26438 ; IT RESULTS IN PRINTING THE ERROR MESSAGE: 'TRAPPED TO 10 PC=XXXXXX'
26439 ; WHERE XXXXXX IS THE ADDRESS CONTAINING THE INSTRUCTION WORD THAT
26440 ; SPRUNG THE TRAP. AFTER PRINTING THE ERROR MESSAGE AN ATTEMPT IS
26441 ; MADE TO RESTART THE PROGRAM AT THE BEGINNING.
26442
26443 ; IF THE TRAP IS SPRUNG WHILE IN THE PROCESS OF TRYING TO SERVICE A
26444 ; PREVIOUS RSVD INSTRUCTION TRAP OR AN UNEXPECTED BUS ERROR THE PROGRAM
26445 ; WILL HALT. AFTER THE HALT THE STACK WILL CONTAIN INFORMATION RELATIVE
26446 ; TO THE TWO SUCCESSIVE TRAPS AS SHOWN BELOW:
26447
26448 ;[SP] PC+2 OF 2ND TRAP
26449 ;[SP]+2 PSW
26450 ;[SP]+4 PC+2 OF 1ST TRAP
26451 ;[SP]+6 PSW
26452
26453 ; LOCATION 'CATERR' CAN BE EXAMINED TO OBTAIN THE FOLLOWING
26454 ; INFORMATION:
26455
26456 ;[CATERR]=401 RSVD INSTR TRAP COMBINED WITH A BUS ERROR
26457 ; TRAP (PC AT TIME OF ERROR HALT INDICATES
26458 ; WHICH OCCURRED FIRST)
26459 ;[CATERR]=2 TWO SUCCESSIVE BUS ERROR TRAPS
26460 ;[CATERR]=1000 TWO SUCCESSIVE RSVD INSTR TRAPS
26461
26462 ; THE CONTENTS OF R0 (DISPLAYED IN THE DATA LIGHTS) AT THE TIME OF THE
26463 ; HALT PROVIDES FURTHER INFORMATION AS TO THE LAST TEST BEING EXECUTED
26464 ; WHEN THE TRAPS OCCURRED.
26465
26466
26467 ; THESE TWO INSTRUCTIONS ARE USED BY THE BASIC INSTRUCTION
26468 ; TESTS TO VERIFY THE RSVD INSTR TRAP MECHANISM PRIOR TO ACTIVATING THE SERVICE
26469 ; ROUTINE
26470
26471 065042 005167 001644 RSVTST: COM RSVFLG ; SET RSVD INSTR TRAP TEST FLAG
26472 065046 000002 RTI ; RETURN TO BASIC TEST
26473
26474 065050 005737 066720 RSERR: TST @#CATERR ; ANY PENDING CATASTROPHIC ERRORS
26475 065054 001032 BNE 2$ ; BE IF YES
26476 065056 105237 066721 INCB @#1+CATERR ; SET RSVD INSTR FLAG
26477 065062 032737 010000 177570 BIT #SW12,@#SR ; INHIBIT ERROR PRINT ?
26478 065070 001020 BNE 1$ ; BR IF YES
26479 065072 104400 TYPE ; GO TYPE 'TRAPPED TO 10 PC='
26480 065074 067476 RMSG MOV (SP),R2 ; GET, CONVERT AND PRINT CONTENTS
26481 065076 011602 TST -(R2) ; OF THE OC
26482 065100 005742 JSR PC,@#OTA ; OF THE OC
26483 065102 004737 065760 TYPE ; OUTPUT CR / LF
26484 065106 104400 DIGITS
26485 065110 067136 TYPE
26486 065112 104400 CRLF
26487 065114 066726 CRLF
26488 065116 022737 177777 066666 CMP #-1,@#ERRCNT ; MAX ERROR COUNT ??

```

26489	065124	001402		BEQ	1\$:BR IF YES
26490	065126	005237	066666	INC	@#ERRCNT		:COUNT THE ERROR
26491	065132	012706	001000	1\$:	MOV	#BT001,SP	
26492	065136	000137	064524		JMP	@#PFRET	:GO ATTEMPT RESTART
26493	065142	105237	066721	2\$:	INCB	@#1+CATERR	:INCREMENT RSVD INSTR FLAG
26494	065146	000000			HALT		:CATASTROPHIC ERROR HALT
26495	065150	000770			BR	1\$:DEPRESSING CONTINUE WILL CAUSE
26496							:ATTEMPT TO RESTART.
26497							

26498
26499
26500
26501
26502
26503
26504
26505
26506
26507
26508
26509
26510
26511
26512
26513
26514
26515
26516
26517
26518
26519
26520
26521
26522
26523
26524
26525
26526
26527
26528
26529
26530
26531
26532
26533
26534
26535
26536
26537
26538
26539
26540
26541
26542
26543
26544
26545
26546
26547
26548
26549
26550
26551
26552
26553

```
; *****  
; .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 FOLLOING  
; 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 R0 (DISPLAYED IN THE DATA LIGHTS) AT THE TIME OF  
; THE HALT PROVIDED FURTHER INFORMATION AS TO THE TEST IN PROGRESS  
; WHEN THE TRAPS OCCURRED.
```

```
; THE CONTENTS OF THE SP CAN BE USED TO INDICATE IF STACK OVERFLOW CAUSED  
; THE BUSS ERROR TRAP(S) AS SHOWN BELOW:
```

```
;400>[SP]>336 YELLOW ZONE  
;[SP]=0 RED ZONE
```

```
; THESE TWO INSTRUCTIONS ARE USED BY THE BASIC INSTRUCTION TESTS TO  
; VERIFY THAT THE BUS ERROR TRAP MECHANISM WORKS PRIOR TO ACTIVATING  
; THE SERVICE ROUTINE
```

```
BETST: COM BERFLG ;SET BUS ERROR TRAP TEST FLAG  
RTI ;RETURN TO BASIC TEST
```

```
BERR: TST @#CATERR ;ANY CATASTROPHIC ERRORS PENDING?  
BNE 2$ ;BR IF YES  
INCB @#CATERR ;SET CATASTROPHIC ERROR FLAG  
BIT #SW12,@#SR ;INHIBIT ERROR PRINT  
BNE 1$ ;BR IF YES  
TYPE ;PRINT 'TRAP TO 4' MESSAGE  
BEMSG
```

```
MOV (SP),R2 ;GET TRAP PC FROM STACK
```

```
065152 005167 001536  
065156 000002  
  
065160 005737 066720  
065164 001031  
065166 105237 066720  
065172 032737 010000 177570  
065200 001017  
065202 104400  
065204 067430  
065206 011602
```

```

26554 065210 004737 065760      JSR    PC,@#OTA      ;CONVERT IT TO ASCII
26555 065214 104400              TYPE              ;GO TYPE TRAP PC
26556 065216 067136              DIGITS
26557 065220 104400              TYPE
26558 065222 066726              CRLF              ;GO OUTPUT CR / LF
26559 065224 022737 177777 066666  CMP    #-1,@#ERRCNT ;MAX ERROR COUNT ??
26560 065232 001402              BEQ    1$           ;BR IF YES
26561 065234 005237 066666              INC    @#ERRCNT    ;COUNT THE ERROR
26562 065240 012706 001000              1$:  MOV    #BT001,SP ;RESET THE STACK POINTER
26563 065244 000137 064524              JMP    @#PFRET     ;GO CLEANUP FOR RESTART
26564
26565 065250 105237 066720              2$:  INCB   @#CATERR ;SET CATASTROPHIC ERROR FLAG
26566 065254 000000              HALT
26567 065256 000770              BR     1$          ;DEPRESS CONTINUE TO ATTEMPT RESTART
26568
26569
26570
26571

```

26572
26573
26574
26575
26576
26577
26578
26579
26580
26581
26582
26583
26584
26585
26586
26587
26588
26589
26590
26591
26592
26593
26594
26595
26596
26597
26598
26599
26600
26601
26602
26603
26604
26605
26606
26607
26608
26609
26610
26611
26612
26613
26614
26615

: *****
: .SBTTL SCOPE SERVICE ROUTINE
: *****

: THIS UTILITY IS CALLED BY AN IOT=SCOPE INSTRUCTION AT THE END
: OF EACH TEST IN THE COMPREHENSIVE INSTRUCTION TEST AND COM-
: BINED INSTRUCTION EXERCISER TEST SECTIONS OF THE PROGRAM.
: IT IS DESIGNED TO IMPLEMENT THE CONSOLE SWITCH OPTIONS DEFINED
: BELOW:

: SW14 = 1 LOOP ON CURRENT TEST
: SW11 = 1 INHIBIT SUB-TEST ITERATIONS
: SW10 = 1 LOOP ON TEST SELECTED BY SR<09:00>

065260 005137 066702
065264 000002

065266 112760 000377 070140
065274 032737 040000 177570
065302 001403
065304 013716 066654
065310 000002
065312 032737 002000 177570
065320 001412
065322 013737 177570 066700
065330 042737 176000 066700
065336 020037 066700
065342 001760
065344 000407
065346 032737 004000 177570
065354 001003
065356 005337 066664
065362 001350
065364 013737 066662 066664
065372 011637 066654
065376 000002

SCOPEA: COM @#SCOFLG ;THESE TWO ILSTRUCTIONS ARE
RTI ;USED IN THE BASIC TESTS TO
;VERIFY THE IOT LINKAGE

SCOPEB: MOVB #377,STAB1(R0) ;SET FLAG IN MISSED TEST TABLE
BIT #SW14,@#SR ;LOOP ON CURRENT TEST ?
BEQ 2\$;BR IF NO - SW14=0
1\$: MOV @#RETURN,(SP) ;SET UP RTN PC ON STK TO LOOP
RTI ;RETURN TO CURRENT TEST
2\$: BIT #SW10,@#SR ;LOOP ON SELECTED TEST ?
BEQ 3\$;BR IF NO - SW10=0
MOV @#SR,@#SELTST ;GET CONTENTS OF SWITCHES
BIC #176000,@#SELTST ;MASK OUT SR<15:10>
CMP R0,@#SELTST ;IS THIS THE SELECTED TEST ?
BEQ 1\$;BR IF YES
BR 4\$;GO EXIT TO NEXT TEST
3\$: BIT #SW11,@#SR ;INHIBIT ITERATIONS ?
BNE 4\$;BR IF YES - SW11=1
DEC @#ITCNT ;COUNT ONE TIME
BNE 1\$;BR IF NOT DONE - DO IT AGIN
4\$: MOV @#ICOUNT,@#ITCNT ;RESET ITERATION COUNTER
MOV (SP),@#RETURN ;SET UP NEW SCOPE RETURN
RTI ;RETURN TO DO NEXT SEQ. TEST

```
26616 ; *****
26617 ; .SBTTL ERROR SERVICE ROUTINE
26618 ; *****
26619
26620 ;THIS UTILITY IS CALLED BY AN ERRORX = EMTX INSTRUCTION OF THE
26621 ;FOLLOWING FORMAT:
26622
26623 ; E'N: ERRORX
26624 ; R'N
26625
26626 ; WHERE: X REPRESENTS THE CODING OF THE LOW BYTE IN THE EMT
26627 ; AND INDICATES THE NO. OF COLUMNS TO BE PRINTED:
26628
26629 ; X=0 ALL 8 COLUMNS
26630 ; 7 1ST SEVEN COLUMNS
26631 ; 6 1ST SIX COLUMNS
26632 ; 5 1ST FIVE COLUMNS
26633 ; 4 1ST FOUR COLUMNS
26634 ; 3 1ST THREE COLUMNS
26635 ; 2 1ST TWO COLUMNS
26636 ; 1 1ST COLUMN ONLY
26637
26638 ; R'N IS THE ADDRESS WHERE CONTROL IS RETURNED AFTER
26639 ; THE ERROR SERVICE IF SW09=1 (LOOP ON HARD ERROR).
26640 ; IF SWITCH 09 IS RESET CONTROL ISRETURNED TO
26641 ; E'N+4.
26642
26643 ;IT IS DESIGNED TO SERVICE THE SWITCH OPTIONS DEFINED BELOW:
26644
26645 ; SW15=1 HALT ON ERROR - TESTED AFTER
26646 ; PRINTING - DEPRESSING CONTINUE RESUMES NORMAL
26647 ; EXECUTION
26648
26649 ; SW13=1 INHIBIT ALL ERROR PRINTOUTS EXCEPT:
26650
26651 ; 1)BELL ON ERROR
26652 ; 2)FAULT #
26653 ; 3)INTRODUCTORY MESSAGE
26654 ; 4)ANY CATASTROPHIC ERROR MESSAGE
26655
26656
26657 ; SW09=1 LOCK ON HARD ERRORS
26658
26659 ;THIS UTILITY ALSO CALLS THE OTA (OCTAL TO ASCII) AND TYPE (PRINT)
26660 ;UTILITIES TO FORMAT AND REPORT THE ERRORS
26661
26662
26663 065400 005137 066676 ERRB: COM @#ERRFLG ;THESE TWO INSTRUCTIONS ARE USED
26664 065404 000002 RTI ;IN THE BASIC TESTS TO VERIFY THE EMT
26665
26666 065406 022737 177777 066666 ERRB: CMP #-1,@#ERRCNT ;ERROR COUNT = 177777 ??
26667 065414 001402 BEQ 1$ ;BR IF YES - FREEZE ERRCNT
26668 065416 005237 066666 INC @#ERRCNT ;COUNT THIS ERROR
26669 065422 010246 1$: MOV R2,-(SP) ;SAVE R2 ON THE STACK
26670 065424 016602 000002 MOV 2(SP),R2 ;GET THE PC+2 OF ERROR CALL
26671 065430 011237 066656 MOV (R2),@#ERRTN ;GET THE ERROR LOOP ADDRESS
```

26672	065434	014237	066640		MOV	-(R2),@#COLCNT	:GET THE ERROR EMT CALL
26673	065440	042737	177770	066640	BIC	#177770,@#COLCNT	:MASK OUT BITS <15:03>
26674	065446	032737	020000	177570	3\$: BIT	#SW13,@#SR	:INHIBIT ERROR PRINTOUT ?
26675	065454	001103			BNE	6\$:BR IF YES
26676	065456	005737	066716		TST	@#ERFLG1	:ERROR HEADER PRINTED FLAG SET ??
26677	065462	001006			BNE	4\$:BR IF IT IS-PRINTER HEADER ONCE PER PASS
26678	065464	104400			TYPE		:CALL THE TYPE UTILITY
26679	065466	066732			ERHDR1		:ADDR OF ERROR HEADER MSG 1
26680	065470	104400			TYPE		:CALL THE TYPE UTILITY
26681	065472	067034			ERHDR2		:ADDR OF ERROR HEADER MSG2
26682	065474	005137	066716		COM	@#ERFLG1	:FLAG THAT THE HEADER WAS PRINTED ONCE
26683	065500	104400			4\$: TYPE		:START WITH A CR / LF
26684	065502	066726			CRLF		
26685	065504	004737	065746		JSR	PC,@#8\$:GO OUTPUT COLUMN ONE
26686	065510	022737	000001	066640	CMP	#1,@#COLCNT	:WAS IT AN ERROR1 CALL ?
26687	065516	001460			BEQ	5\$:BR IF YES
26688							
26689	065520	016602	000004		MOV	4(SP),R2	:GET THE ERROR PSW
26690	065524	004737	065746		JSR	PC,@#8\$:GO OUTPUT COLUMN TWO
26691	065530	022737	000002	066640	CMP	#2,@#COLCNT	:WAS IT AN ERROR2 CALL ?
26692	065536	001450			BEQ	5\$:BR IF YES
26693							
26694	065540	010602			MOV	SP,R2	:GET THE CONTENTS OF THE SP
26695	065542	005722			TST	(R2)+	:CORRECT IT
26696	065544	004737	065746		JSR	PC,@#8\$:GO OUTPUT COLUMN 3
26697	065550	022737	000003	066640	CMP	#3,@#COLCNT	:WAS IT AN ERROR3 CALL ?
26698	065556	001440			BEQ	5\$:BR IF YES
26699							
26700	065560	010002			MOV	R0,R2	:GET THE TEST # IN R0
26701	065562	004737	065746		JSR	PC,@#8\$:GO OUTPUT COLUMN 4
26702	065566	022737	000004	066640	CMP	#4,@#COLCNT	:WAS IT AN ERROR4 CALL ?
26703	065574	001431			BEQ	5\$:BR IF YES
26704							
26705	065576	010102			MOV	R1,R2	:GET THE COPY OF THE INSTRUCTION
26706	065600	004737	065746		JSR	PC,@#8\$:GO OUTPUT COLUMN 5
26707	065604	022737	000005	066640	CMP	#5,@#COLCNT	:WAS IT AN ERROR5 CALL ?
26708	065612	001422			BEQ	5\$:BR IF YES
26709							
26710	065614	012602			MOV	(SP)+,R2	:GET THE ORIGINAL R2 OFF STACK
26711	065616	004737	065746		JSR	PC,@#8\$:GO OUTPUT COLUMN 6
26712	065622	010246			MOV	R2,-(SP)	:MAINTAIN COMPATABILITY
26713	065624	022737	000006	066640	CMP	#6,@#COLCNT	:WAS IT AN ERROR6 CALL ?
26714	065632	001412			BEQ	5\$:BR IF YES
26715							
26716	065634	010302			MOV	R3,R2	:GET THE CONTENTS OF R3
26717	065636	004737	065746		JSR	PC,@#8\$:GO OUTPUT COLUMN 7
26718	065642	022737	000007	066640	CMP	#7,@#COLCNT	:WAS IT AN ERROR7 CALL ?
26719	065650	001403			BEQ	5\$:BR IF YES
26720							
26721	065652	010402			MOV	R4,R2	:GET THE CONTENTS OF R4
26722	065654	004737	065746		JSR	PC,@#8\$:GO OUTPUT COLUMN 8
26723	065660	104400			5\$: TYPE		:CALL THE TYPE UTILITY
26724	065662	066726			CRLF		:ADDRESS OF THE CR / LF MESSAGE
26725	065664	012602			6\$: MOV	(SP)+,R2	:RESTORE THE INTEGRITY OF R2
26726	065666	032737	002000	177570	BIT	#SW10,@#SR	:LOOP ON SELECTED TEST ???
26727	065674	001012			BNE	7\$:BR IF YES - SW09 PART OF TEST NO.

26728	065676	032737	001000	177570		BIT	#SW09,@#SR	:LOOP ON HARD ERRORS ???
26729	065704	001406				BEQ	7\$:BR IF NOT - REWARD TO THE FIRST
26730								:MAINTENANCE MAN WHO CALLS THE
26731								:AUTHOR OF THIS PROGRAM IN
26732								:MARLBORO,MASS.
26733	065706	013716	066656			MOV	@#ERRTN,(SP)	:PUT ERROR RETURN ADDR ON STACK
26734	065712	042766	000020	000002		BIC	#20,2(SP)	:CLEAR 'T' BIT ON THE STACK IN CASE
26735								:THIS ERROR OCCURRED IN 'T' BIT TESTS
26736	065720	000002				RTI		:RETURN TO LOOP ON THIS ERROR CALL
26737	065722	005737	177570		7\$:	TST	@#SR	:HALT AFTER PRINTING ???
26738	065726	100001				BPL	+.4	:BR IF NOT
26739	065730	000000				HALT		:DEPRESS CONTINUE TO RESUME TEST
26740								:AFTER THE HALT THE ADDRESS DISPLAY
26741								:CONTAINS #(7\$+10) AND THE DATA DISP-
26742								:LAY CONTAINS THE NUMBER OF THE FAIL-
26743								:ING TEST
26744	065732	062716	000002			ADD	#2,(SP)	:MOVE RETURN PC AROUND RN
26745	065736	042766	000020	000002		BIC	#20,2(SP)	:CLEAR 'T' BIT ON STACK IN CASE THIS
26746								:ERROR OCCURRED IN 'T' BIT TESTS
26747	065744	000002				RTI		:RETURN TO INSTRUCTION AFTER ERROR
26748								
26749								:CONVERSION AND OUTPUT CALLS
26750								
26751	065746	004737	065760		8\$:	JSR	PC,@#OTA	: CONVERT [R2] TO SIX ASCII CHARS.
26752	065752	104400				TYPE		:CALL THE TYPE UTILITY
26753	065754	067136				DIGITS		:ADDRESS OF THE ASCII BUFFER
26754	065756	000207				RTS	PC	:RETURN TO CALLER ABOVE
26755								
26756								
26757								

26758
26759
26760
26761
26762
26763
26764
26765
26766
26767
26768
26769
26770
26771
26772
26773
26774
26775
26776
26777
26778
26779
26780
26781
26782
26783
26784
26785
26786
26787
26788
26789
26790
26791
26792
26793

065760 004737 066146
065764 012704 067136
065770 005003
065772 010201
065774 006302
065776 006103
066000 012700 000006
066004 000404
066006 006302
066010 006103
066012 005301
066014 001374
066016 012701 000003
066022 116324 066644
066026 005003
066030 005300
066032 001365
066034 004737 066166
066040 000207

; *****
; .SBTTL OCTAL TO ASCII CONVERSION ROUTINE
; *****

; THIS ROUTINE CONVERTS THE 16 BIT OCTAL NUMBER IN R2 TO ITS 6 CHAR.
; ASCII EQUIVALENT AND STORES THE CHARACTERS IN AN EIGHT CHAR. BUFFER
; THAT STARTS AT THE ADDRESS TAGGED 'DIGITS' - THE 7TH AND 8TH CHAR.
; POSITIONS ARE LOADED WITH A SPACE CHAR. FOLLOWED BY A ZERO TERMINATOR
; BYTE - THE ROUTINE IS CALLED VIA A 'JSR PC,OTA'

```
OTA: JSR PC,@SAVR ;GO SAVE THE REGISTERS R0 THRU R5
      MOV #DIGITS,R4 ;SET UP R4 TO POINT TO ASCII BUFFER
      CLR R3 ;INITIALIZE R3 FOR USE AS INDEX REG.
      ;TO ASCII CONVERSION TABLE
      MOV R2,R1 ;SAVE ORIGINAL NUMBER IN R1
1$: ASL R2 ;MOST SIGNIFICANT BIT GOES INTO 'C'
    ROL R3 ;R3 CONTAINS THE MOST SIGNIFICANT BIT
    MOV #6,R0 ;COUNT SIX DIGITS CONVERSION
    BR 3$ ;CONVERT AND LOAD THE 1ST DIGIT
2$: ASL R2 ;SHIFT BITS OUT OF R2 INTO 'C'
    ROL R3 ;SHIFT 'C' INTO THE LSB POS. OF R3
    DEC R1 ;COUNT ONE BIT SHIFTED
    BNE 2$ ;BR UNTIL 3 BITS SHIFTED
3$: MOV #3,R1 ;INITIALIZE BIT SHIFT COUNTER
    MOVB DIGTAB(R3),(R4)+ ;MOVE CHAR. TO DIGIT BUFFER
    CLR R3 ;CLEAR INDEX TO CONVERSION TABLE
    DEC R0 ;COUNT ONE CHAR.
    BNE 2$ ;BR UNTIL 6 CHARS CONVERTED
    JSR PC,@RESTR ;GO RESTORE THE REGS
    RTS PC ;RETURN TO CALLER
```

```
26794 ; *****  
26795 ; .SBTTL PRINT SUBROUTINE  
26796 ; *****  
26797  
26798 ; THIS ROUTINE IS CALLED TO PRINT ALL ASCII MESSAGES - IT IS CALLED VIA  
26799 ; A TYPE = TRAP = 104400 INSTRUCTION AS SHOWN BELOW:  
26800  
26801 ; TYPE  
26802 ; ADDR  
26803  
26804 ; WHERE 'ADDR' IS THE STARTING ADDRESS OF THE MESSAGE BUFFER  
26805 ; TO BE PRINTED - THE SUBROUTINE WILL CONTINUE TO PRINT CHAR-  
26806 ; ACTERS UNTIL IT FINDS A '000' BYTE TERMINATOR  
26807  
26808 ; IF FILLER CHARACTERS ARE REQUIRED THE LOCATION TAGGED 'FILLS' MUST  
26809 ; CONTAIN THE FOLLOWING INFORMATION:  
26810  
26811 ; FILLS - ODD BYTE = NO. OF FILLERS REQUIRED  
26812 ; FILLS - EVEN BYTE = FILLER CHARACTER DESIRED  
26813  
26814 ; THE DEFAULT VALUE OF FILLS IS 0,0  
26815  
26816 066042 005137 066674 PRINA: COM @#PRIFLG ; THESE TWO INSTRUCTIONS ARE  
26817 066046 000002 RTI ; USED BY THE BASIC TESTS TO VERIFY  
26818 ; THE TRAP INSTRUCTION  
26819  
26820 066050 010046 PRINT: MOV R0,-(SP) ; SAVE R0 ON THE STACK  
26821 066052 017600 000002 MOV @2(SP),R0 ; SET R0 TO POINT TO THE MESSAGE BUFFER  
26822 066056 062766 000002 000002 ADD #2,2(SP) ; ADJUST THE RETURN PC TO POINT TO THE  
26823 ; INSTRUCTION FOLLOWING THE CALL  
26824 066064 112046 1$: MOVB (R0)+,-(SP) ; PUSH CHAR. TO BE TYPED ON THE STACK  
26825 ; AND UPDATE THE BUFFER POINTER  
26826 066066 001003 BNE 2$ ; BRANCH IF NOT A 000 TERMINATOR  
26827 066070 005726 TST (SP)+ ; POP TERMINATOR OFF THE STACK  
26828 066072 012600 MOV (SP)+,R0 ; RESTORE THE ORIGINAL R0  
26829 066074 000002 RTI ; RETURN TO CALLER  
26830 066076 004737 066130 2$: JSR PC,@#5$ ; GO TYPE THE CHARACTER  
26831 066102 122726 000012 3$: CMPB #12,(SP)+ ; WAS CHAR TYPED A LINE FEED ?  
26832 066106 001366 BNE 1$ ; BRANCH IF NOT  
26833 066110 013746 066660 MOV @#FILLS,-(SP) ; GET THE FILLER COUNT AND CHARACTER  
26834 066114 105366 000001 4$: DECB 1(SP) ; COUNT ONE FILLER OUT  
26835 066120 002770 BLT 3$ ; BR IF NO MORE FILLERS NEEDED  
26836 066122 004737 066130 JSR PC,@#5$ ; GO TYPE THE FILLER  
26837 066126 000772 BR 4$ ; GO COUNT AND TEST FILLER COUNT  
26838  
26839 066130 105737 177564 5$: TSTB @#XCSR ; OUTPUT DEVICE READY  
26840 066134 100375 BPL 5$ ; BRANCH BACK IF NOT READY - NOTE THE  
26841 ; PROGRAM WILL HANG HERE IN THE EVENT THAT  
26842 ; THE DL11 READY LOGIC FAILS DURING RUN  
26843 066136 116637 000002 177566 MOVB 2(SP),@#XDBR ; OUTPUT THE CHARACTER  
26844 066144 000207 RTS PC ; RETURN TO CALLER
```

26845
26846
26847
26848
26849
26850
26851
26852 066146 010546
26853 066150 010446
26854 066152 010346
26855 066154 010246
26856 066156 010146
26857 066160 010046
26858 066162 016607 000014
26859
26860
26861

: .SBTTL SAVE REGISTER SUBROUTINE
: *****

: THIS ROUTINE IS CALLED BY THE 'DTA' UTILITY AND WILL SAVE GENERAL
: REGISTERS R0 THRU R5 ON THE PROCESSOR STACK

SAVR: MOV R5,-(SP) ;PUSH R5 ON THE STACK
MOV R4,-(SP) ;PUSH R4
MOV R3,-(SP) ;PUSH R3
MOV R2,-(SP) ;PUSH R2
MOV R1,-(SP) ;PUSH R1
MOV R0,-(SP) ;PUSH R0
MOV 14(SP),PC ;RETURN TO THE INSTRUCTION THAT
;FOLLOWS THE 'JSR PC,SAVR' CALL

26862
26863
26864
26865
26866
26867
26868
26869
26870
26871
26872
26873
26874
26875
26876
26877
26878
26879

066166 012666 000014
066172 012600
066174 012601
066176 012602
066200 012603
066202 012604
066204 012605
066206 000207

: *****
: .SBTTL RESTORE REGISTERS SUB-ROUTINE
: *****

: THIS ROUTINE COMPLEMENTS THE SAVE REGISTER ROUTINE AND WILL RESTORE
: GENERAL REGISTERS R0 THRU R5 FROM THE STACK - IT IS CALLED VIA A
: JSR PC,RESTR.

RESTR: MOV (SP)+,14(SP) ;REPOSITION THE RETURN PC ON THE STACK
MOV (SP)+,R0 ;RESTORE R0 - R5
MOV (SP)+,R1
MOV (SP)+,R2
MOV (SP)+,R3
MOV (SP)+,R4
MOV (SP)+,R5
RTS PC ;RETURN TO THE INSTRUCTION THAT
;FOLLOWS THE 'JSR RESTR' CALL

```

26880 ; *****
26881 ; .SBTTL ROUTINES TO CHECK FOR AND FLAG 11/40 OPTIONS
26882 ; *****
26883
26884 066210 005737 066670 TSTOPT: TST @#PASCNT ;1ST PASS ??
26885 066214 001027 BNE 1$ ;BR IF NOT
26886 066216 005037 066636 CLR @#OPTION ;CLEAR THE OPTION FLAG WORD
26887 066222 032737 010000 177570 BIT #SW12,@#SR ;INHIBIT PRINTING ??
26888 066230 001002 BNE .+6 ;BR IF YES
26889 066232 104400 TYPE ;TYPE OPTIONS AVAIL. HEADER
26890 066234 067246 OPT1
26891 066236 004737 066276 JSR PC,@#CHKKT ;GO TEST FOR KT11-D OPTION
26892 066242 004737 066352 JSR PC,@#CHKKJ ;GO TEST FOR KJ11-A OPTION
26893 066246 004737 066426 JSR PC,@#CHKKF ;GO TEST FOR KE11-F OPTION
26894 066252 004737 066504 JSR PC,@#CHKKE ;GO TEST FOR KE11-E OPTION
26895 066256 004737 066562 JSR PC,@#CHKKW ;GO TEST FOR KW11-L OPTION
26896 066262 005737 066636 TST @#OPTION ;ANY OPTIONS FOUND
26897 066266 001002 BNE 1$ ;BR IF YES
26898 066270 104400 TYPE ;GO TYPE 'NONE'
26899 066272 067401 OPT7
26900 066274 000207 1$: RTS PC ;RETURN TO 'CIT' START-UP
26901
26902 066276 013704 000004 CHKKT: MOV @#4,R4 ;SAVE THE TIMEOUT VECTOR
26903 066302 012737 066340 000004 MOV #1$,@#4 ;GO TO 1$ IF TRAP OCCURS
26904 066310 005737 177572 TST @#177572 ;REFERENCE KT11 SR0
26905 066314 052737 000200 066636 BIS #200,@#OPTION ;SET BIT7 IF KT IS THERE
26906 066322 032737 010000 177570 BIT #SW12,@#SR ;INHIBIT PRINTING ?
26907 066330 001002 BNE .+6 ;BR IF YES
26908 066332 104400 TYPE ;GO TYPE 'KT11-D'
26909 066334 067312 OPT2
26910 066336 000402 BR 2$
26911 066340 062706 000004 1$: ADD #4,SP ;FIX UP THE SP
26912 066344 010437 000004 2$: MOV R4,@#4 ;RESTORE THE TIMEOUT VECTOR
26913 066350 000207 RTS PC ;RETURN TO TSTOPT ROUTINE
26914
26915 066352 013704 000004 CHKKJ: MOV @#4,R4 ;SAVE THE TIMEOUT VECTOR
26916 066356 012737 066414 000004 MOV #1$,@#4 ;GO TO 1$ IF TRAP OCCURS
26917 066364 005737 177774 TST @#177774 ;REFERENCE KJ11 REG.
26918 066370 052737 000004 066636 BIS #4,@#OPTION ;SET BIT2 IF KJ IS THERE
26919 066376 032737 010000 177570 BIT #SW12,@#SR ;INHIBIT PRINTING ?
26920 066404 001002 BNE .+6 ;BR IF YES
26921 066406 104400 TYPE ;GO TYPE 'KJ11-A'
26922 066410 067325 OPT3
26923 066412 000402 BR 2$
26924 066414 062706 000004 1$: ADD #4,SP ;FIX UP THE SP
26925 066420 010437 000004 2$: MOV R4,@#4 ;RESTORE THE TIMEOUT VECTOR
26926 066424 000207 RTS PC ;RETURN TO TSTOPT ROUTINE
26927
26928
26929 066426 013704 000010 CHKKF: MOV @#10,R4 ;SAVE THE RSVD INSTR VECTOR
26930 066432 012737 066472 000010 MOV #1$,@#10 ;GO TO 1$ IF TRAP OCCURS
26931 066440 012702 067560 MOV #MBUF0,R2 ;SET UP FOR FADD TRY
26932 066444 075002 FADD R2 ;TEST FOR THE KE11-F OPTION
26933 066446 052737 000002 066636 BIS #2,@#OPTION ;SET BIT1 IF IT RESPONDS
26934 066454 032737 010000 177570 BIT #SW12,@#SR ;INHIBIT PRINTING ?
26935 066462 001002 BNE .+6 ;BR IF YES

```

```

26936 066464 104400          TYPE          ;GO TYPE 'KE11-F'
26937 066466 067340          OPT4
26938 066470 000402          BR          2$
26939 066472 062706 000004 1$: ADD      #4,SP      ;FIX UP THE SP
26940 066476 010437 000010 2$: MOV      R4,@#10    ;RESTORE THE RSVD INSTR VECTOR
26941 066502 000207          RTS          PC      ;RETURN TO TSTOPT ROUTINE
26942
26943 066504 013704 000010 000010 CHKKE: MOV     @#10,R4    ;SAVE THE RSVD INSTR VECTOR
26944 066510 012737 066550 000010 MOV     #1,@#10    ;GO TO 1$ IF TRAP OCCURS
26945 066516 005001          CLR      R1      ;SET UP TO TRY ASH INSTR.
26946 066520 005002          CLR      R2
26947 066522 072201          ASH     R1,R2    ;TRY A KE INSTR
26948 066524 052737 000001 066636 BIS     #1,@#OPTION ;SET BIT0 IF KE IS THERE
26949 066532 032737 010000 177570 BIT     #SW12,@#SR  ;INHIBIT PRINTING ?
26950 066540 001002          BNE     .+6      ;BR IF YES
26951 066542 104400          TYPE          ;GO TYPE 'KE11-E'
26952 066544 067353          OPT5
26953 066546 000402          BR          2$
26954 066550 062706 000004 1$: ADD      #4,SP      ;FIX UP THE SP
26955 066554 010437 000010 2$: MOV      R4,@#10    ;RESTORE THE RSVD INSTR VECTOR
26956 066560 000207          RTS          PC      ;RETURN TO TSTOPT ROUTINE
26957
26958 066562 013704 000004 000004 CHKKW: MOV     @#4,R4    ;SAVE THE TIMEOUT VECTOR
26959 066566 012737 066624 000004 MOV     #1,@#4     ;GO TO 1$ IF TRAP OCCURS
26960 066574 005737 177546 TST     @#177546   ;REFERENCE KW11-L CSR
26961 066600 052737 100000 066636 BIS     #100000,@#OPTION ;SET BIT15 IF KW IS THERE
26962 066606 032737 010000 177570 BIT     #SW12,@#SR  ;INHIBIT PRINTING ?
26963 066614 001002          BNE     .+6      ;BR IF YES
26964 066616 104400          TYPE          ;GO TYPE 'KW11-L'
26965 066620 067366          OPT6
26966 066622 000402          BR          2$
26967 066624 062706 000004 1$: ADD      #4,SP      ;FIX UP THE SP
26968 066630 010437 000004 2$: MOV      R4,@#4     ;RESTORE THE TIMEOUT VECTOR
26969 066634 000207          RTS          PC      ;RETURN TO TSTOPT ROUTINE
26970

```

```

26971 ;FLAGS, CONSTANTS, AND VARIABLES
26972
26973 066636 000000 OPTION: 0 ;SET UP ON ENTRY TO 'CIT' SECTION TO
26974 ;SPECIFY OPTIONS INSTALLED:
26975 ;BIT15=1 KW11-L INSTALLED
26976 ;BIT07=1 KT11-D INSTALLED
26977 ;BIT02=1 KJ11-A INSTALLED
26978 ;BIT01=1 KE11-F INSTALLED
26979 ;BIT00=1 KE11-E INSTALLED
26980 066640 000000 COLCNT: 0 ;USED BY ERROR SERVICE TO STORE COLUMN COUNT
26981 066642 000000 BPTLOC: 0 ;STORES 16 USER DEFINED MAINTENANCE
26982 ;BREAKPOINTS
26983 066644 030460 DIGTAB: '01 ;OCTAL TO ASCII CONVERSION TABLE
26984 066646 031462 '23
26985 066650 032464 '45
26986 066652 033466 '67
26987 066654 000000 RETURN: 0 ;USED BY SCOPE TO STORE RETURN ADDRESS
26988 066656 000000 ERRTN: 0 ;USED BY ERROR SERV. TO STORE ERROR RETURN ADDR.
26989 066660 002400 FILLS: 02400 ;STORES FILL CHAR AND FILL COUNT
26990 066662 000001 ICOUNT: 1 ;ITERATION COUNTERS
26991 066664 000001 ITCNT: 1
26992 066666 000000 ERRCNT: 0 ;STORES TOTAL ERROR COUNT
26993 066670 000000 PASCNT: 0 ;STORES TOTAL NO. OF PASSES
26994 066672 000000 PFCNT: 0 ;KEEPS COUNT OF # OF PURFAILS
26995 066674 000000 PRIFLG: 0 ;FLAG USED BY BASIC TESTS FOR TRAP TEST
26996 066676 000000 ERRFLG: 0 ;FLAG USED BY BASIC TESTS FOR EMT TEST
26997 066700 000000 SELTST: 0 ;STORES SR<8:0> FOR LOOP ON SELECTED TEST
26998 066702 000000 SCOFLG: 0 ;USED BY BASIC TESTS FOR IOT TEST
26999 066704 000000 FIRST: 0 ;CONTAINS FIRST TEST # LOGGED IN MISSED TEST TABLE
27000 066706 000000 LAST: 0 ;CONTAINS LAST TEST # LOGGED IN MISSED TEST TABLE
27001 066710 000000 MISFLG: 0 ;FLAGS MISSED TEST ERROR HEADER PRINTED
27002 066712 000000 RSVFLG: 0 ;FLAG USED BY BASIC TEST OF RSVD INSTR TRAP
27003 066714 000000 BERFLG: 0 ;FLAG USED BY BASIC TEST OF BUS ERROR TRAPS
27004 066716 000000 ERFLG1: 0 ;ALLOWS ONLY 1 ERROR HEADER PER PASS
27005 066720 000000 CATERR: 0 ;FLAGS USED BY BUS ERROR AND RSVD INSTR TRAP
27006 ;SERVICE ROUTINES
27007 066722 000000 ONCE: 0 ;FLAGS PROGRAM TITLE HAS BEEN PRINTED

```

```

27008      ;MESSAGE TABLES
27009
27010      .EVEN
27011 066724 000007
27012 066726 005015
27013 066730 000000
27014 066732
27015 066732 005015 024040 041520
27016 066740 020051 020040 024040
27017 066746 051520 020051 020040
27018 066754 024040 050123 020051
27019 066762 020040 052040 051505
27020 066770 020124 020040 024040
27021 066776 051111 020051 020040
27022 067004 042040 051505 020124
27023 067012 020040 053440 051501
27024 067020 020040 020040 051440
27025 067026 027440 041040 000040
27026 067034
27027 067034 005015 024040 033522
27028 067042 020051 020040 024040
27029 067050 051520 024527 020040
27030 067056 024040 033122 020051
27031 067064 020040 024040 030122
27032 067072 020051 020040 024040
27033 067100 030522 020051 020040
27034 067106 024040 031122 020051
27035 067114 020040 024040 031522
27036 067122 020051 020040 024040
27037 067130 032122 006451 000012
27038 067136 030060 030060 030060
27039 067144 020040 000
27040 067147 015 050012 051501
27041 067154 047103 020124 020075
27042 067162 000
27043 067163 040 050040 041506
27044 067170 052116 036440 000040
27045 067176 020040 051105 041522
27046 067204 052116 036440 000040
27047 067212 005015 041103 042521
27048 067220 041501 020060 042113
27049 067226 030461 040455 041440
27050 067234 052520 042040 040511
27051 067242 006507 000012
27052 067246 005015 042120 030520
27053 067254 027461 030064 044440
27054 067262 052116 051105 040516
27055 067270 020114 050117 044524
27056 067276 047117 020123 047506
27057 067304 047125 006504 000012
27058 067312 005015 052113 030461
27059 067320 042055 005015 000
27060 067325 015 045412 030512
27061 067332 026461 006501 000012
27062 067340 005015 042513 030461
27063 067346 043055 005015 000

      ERHDR1:
      .ASCIZ '<15><12>' (PC) (PS) (SP) TEST (IR) DEST WAS S / B '

      ERHDR2:
      .ASCIZ '<15><12>' (R7) (PSW) (R6) (R0) (R1) (R2) (R3) (R4)'<15><12>'

      DIGITS: .ASCIZ '000000 '
      EOP1: .ASCIZ '<15><12>'PASCNT = '
      PFMESS: .ASCIZ ' PFCNT = '
      EOP2: .ASCIZ ' ERRCNT = '
      IDENT1: .ASCIZ '<15><12>'CBQEAC0 KD11-A CPU DIAG'<15><12>'

      OPT1: .ASCIZ '<15><12>'PDP11/40 INTERNAL OPTIONS FOUND'<15><12>'

      OPT2: .ASCIZ '<15><12>'KT11-D'<15><12>'
      OPT3: .ASCIZ '<15><12>'KJ11-A'<15><12>'
      OPT4: .ASCIZ '<15><12>'KE11-F'<15><12>'

```

27064	067353	015	045412	030505	OPT5:	.ASCIZ	<15><12>'KE11-E'<15><12>
27065	067360	026461	006505	000012			
27066	067366	005015	053513	030461	OPT6:	.ASCIZ	<15><12>'KW11-L'<15><12>
27067	067374	046055	005015	000			
27068	067401	015	047012	047117	OPT7:	.ASCIZ	<15><12>'NONE FOUND'
27069	067406	020105	047506	047125			
27070	067414	000104					
27071	067416	005015	047520	042527	PFMSG:	.ASCIZ	<15><12>'POWER' <15><12>
27072	067424	006522	000012				
27073	067430	005015	051124	050101	BEMSG:	.ASCIZ	<15><12>'TRAPPED TO 4 PC = '
27074	067436	042520	020104	047524			
27075	067444	032040	050040	020103			
27076	067452	020075	000				
27077	067455	015	046412	051511	MISHDR:	.ASCIZ	<15><12>'MISSED TESTS'<15><12>
27078	067462	042523	020104	042524			
27079	067470	052123	006523	000012			
27080	067476	005015	051124	050101	RSMSG:	.ASCIZ	<15><12>'TRAPPED TO 10 PC = '
27081	067504	042520	020104	047524			
27082	067512	030440	020060	041520			
27083	067520	036440	000040				
27084							.EVEN

```

27085 ;COMMON DATA STRUCTURES AND MISCELLANEOUS TABLES
27086
27087 067524 177400 OBUF: 177400 ;DL11 OUTPUT TEST BUFFER
27088 067526 177400 177400
27089 067530 177400 177400
27090 067532 177400 177400
27091
27092 067534 000004 IBUF: .BLKW 4 ;DL11 INPUT TEST BUFFER
27093
27094 067544 067570 ATA: DWTA
27095 067546 070114 DWTB
27096 067550 070130 DBTA
27097 067552 070134 DBTB
27098 067554 067560 MBUF0
27099 067556 067564 MBUF1
27100
27101 067560 000000 MBUF0: 0
27102 067562 000000 0
27103 067564 000000 MBUF1: 0
27104 067566 000000 0
27105 067570 000000 DWTA: 0
27106 067572 177777 -1
27107 067574 177400 177400
27108 067576 000377 377
27109 067600 125252 125252
27110 067602 052525 052525
27111
27112 ;THIS TABLE OF 8 ENTRIES IS USED BY THE ALU ADD TEST IN THE
27113 ;COMBINED INSTRUCTION TESTS
27114
27115 067604 000000 ALUADD: 000000 ;NULL
27116 067606 000000 000000 ;SRC OP1
27117 067610 000000 000000 ;DST OP1
27118 067612 000000 000000 ;ANS1
27119 067614 177777 177777 ;SRC OP2
27120 067616 177777 177777 ;DST OP2
27121 067620 177776 177776 ;ANS2
27122 067622 125252 125252 ;SRC OP3
27123 067624 052525 052525 ;DST OP3
27124 067626 177777 177777 ;ANS3
27125 067630 052525 052525 ;SRC OP4
27126 067632 125252 125252 ;DST OP4
27127 067634 177777 177777 ;ANS4
27128 067636 125252 125252 ;SRC OP5
27129 067640 125252 125252 ;DST OP5
27130 067642 052524 052524 ;ANS5
27131 067644 052525 052525 ;SRC OP6
27132 067646 052525 052525 ;DST OP6
27133 067650 125252 125252 ;ANS6
27134 067652 052525 052525 ;SRC OP7
27135 067654 125253 125253 ;DST OP7
27136 067656 000000 000000 ;ANS7
27137 067660 125253 125253 ;SRC OP8
27138 067662 052525 052525 ;DST OP8
27139 067664 000000 000000 ;ANS8
27140

```

27141
27142
27143
27144
27145 067666 000000
27146 067670 000000
27147 067672 000000
27148 067674 000000
27149 067676 177777
27150 067700 177777
27151 067702 000000
27152 067704 000000
27153 067706 177777
27154 067710 177777
27155 067712 177777
27156 067714 000000
27157 067716 000000
27158 067720 125252
27159 067722 125252
27160 067724 000000
27161 067726 052525
27162 067730 052525
27163 067732 000000
27164 067734 125252
27165 067736 052525
27166 067740 052525
27167 067742 052525
27168 067744 125252
27169 067746 125252

:THIS TABLE OF 8 ENTRIES IS USED BY THE ALU 'AND' TESTS IN THE
:COMBINED INSTRUCTION EXERCISER TESTS

ANDTAB: 000000	:NULL ENTRY
000000	:SRC OP1
000000	:DEST OP1
000000	:ANS1
177777	:SRC OP2
177777	:DST OP2
000000	:ANS2
000000	:SRC OP3
177777	:DST OP3
177777	:ANS3
177777	:SRC OP4
000000	:DST OP4
000000	:ANS4
125252	:SRC OP5
125252	:DST OP5
000000	:ANS5
052525	:SRC OP6
052525	:DST OP6
000000	:ANS6
125252	:SRC OP7
052525	:DST OP7
052525	:ANS7
052525	:SRC OP8
125252	:DST OP8
125252	:ANS8

27170
27171
27172
27173
27174 067750 000000
27175 067752 000000
27176 067754 000000
27177 067756 000000
27178 067760 177777
27179 067762 177777
27180 067764 177777
27181 067766 000000
27182 067770 177777
27183 067772 177777
27184 067774 177777
27185 067776 000000
27186 070000 177777
27187 070002 125252
27188 070004 125252
27189 070006 125252
27190 070010 052525
27191 070012 052525
27192 070014 052525
27193 070016 125252
27194 070020 052525
27195 070022 177777
27196 070024 052525

:THIS TABLE OF 8 ENTRIES IS USED BY THE ALU 'OR' TEST IN THE
:COMBINED INSTRUCTION EXERCISER TEST

ORTAB: 000000	:NULL ENTRY
000000	:SRC OP1
000000	:DEST OP1
000000	:ANS1
177777	:SRC OP2
177777	:DST OP2
177777	:ANS2
000000	:SRC OP3
177777	:DST OP3
177777	:ANS3
177777	:SRC OP4
000000	:DST OP4
177777	:ANS4
125252	:SRC OP5
125252	:DST OP5
125252	:ANS5
052525	:SRC OP6
052525	:DST OP6
052525	:ANS6
125252	:SRC OP7
052525	:DST OP7
177777	:ANS7
052525	:SRC OP8

```

27197 070026 125252          125252          :DST OP8
27198 070030 177777          177777          :ANS8
27199
27200
27201
27202          :THIS TABLE OF 8 ENTRIES IS USED BYTHE ALU SUB TEST IN THE
27203          :COMBINED INSTRUCTION EXERCISER TESTS
27204 070032 000000          ALUSUB: 000000          :NULL
27205 070034 000000          000000          :SRC OP1
27206 070036 000000          000000          :DST OP1
27207 070040 000000          000000          :ANS1
27208 070042 177777          177777          :SRC OP2
27209 070044 177777          177777          :DST OP2
27210 070046 000000          000000          :ANS2
27211 070050 125252          125252          :SRC OP3
27212 070052 052525          052525          :DST OP3
27213 070054 125253          125253          :ANS3
27214 070056 052525          052525          :SRC OP4
27215 070060 125252          125252          :DST OP4
27216 070062 052525          052525          :ANS4
27217 070064 125252          125252          :SRC OP5
27218 070066 125252          125252          :DST OP5
27219 070070 000000          000000          :ANS5
27220 070072 052525          052525          :SRC OP6
27221 070074 052525          052525          :DST OP6
27222 070076 000000          000000          :ANS6
27223 070100 052525          052525          :SRC OP7
27224 070102 125253          125253          :DST OP7
27225 070104 052526          052526          :ANS7
27226 070106 125253          125253          :SRC OP8
27227 070110 052525          052525          :DST OP8
27228 070112 125252          125252          :ANS8
27229
27230 070114 000000          DWTB: 0
27231 070116 000001          1
27232 070120 000400          400
27233 070122 177401          177401
27234 070124 052526          52526
27235 070126 125253          125253
27236
27237 070130          .EVEN
27238 070130          000          DBTA:
27239 070133          125          .BYTE 000,377,252,125
27240 070134
27241 070134          000          001          120          DBTB:
27242 070137          253          .BYTE 000,001,120,253
27243 070140 001022          STAB1: .BLKB 530.          :RESERVE 530. BYTE TABLE FOR
27244
27245 071162 000000          STAB2: 0          :LOGGING ANY MISSED TESTS
27246
27247          000001          .END

```


A0127	012110	7456	7462#
A0130	012206	7493	7504#
A0131	012334	7558	7563#
A0132	012412	7587	7598#
A0133	012530	7644	7651#
A0136	012766	7768	7773#
A0140	013156	7850	7856#
A015	001256	3326	3331#
A0160	014252	8431	8436#
A0162	014354	8508	8513#
A017	001340	3397	3402#
A020	001372	3434	3439#
A021	001422	3470	3475#
A0224	015724	9625	9630#
A0226	016036	9705	9710#
A023	001512	3542	3547#
A0230	016156	9790	9795#
A0232	016352	9898	9903#
A0234	016504	9984	9989#
A0235	016564	10031	10036#
A0236	016644	10080	10085#
A0237	017006	10155	10160#
A0240	017062	10201	10206#
A0241	017142	10247	10252#
A0242	017222	10293	10298#
A0243	017300	10340	10345#
A0244	017354	10386	10391#
A0245	017434	10432	10437#
A0246	017514	10478	10483#
A0247	017572	10524	10529#
A0250	017652	10570	10575#
A0251	017730	10615	10620#
A0252	020010	10660	10665#
A0253	020066	10705	10710#
A0254	020146	10750	10755#
A0255	020226	10795	10800#
A0256	020306	10840	10845#
A0257	020366	10886	10891#
A0260	020450	10932	10937#
A0261	020530	10978	10983#
A0262	020612	11024	11029#
A0263	020674	11070	11075#
A0264	020750	11112	11117#
A0265	021040	11160	11165#
A0266	021120	11202	11207#
A0267	021212	11250	11255#
A0270	021272	11293	11298#
A0271	021364	11341	11346#
A0272	021440	11382	11387#
A0273	021526	11429	11434#
A0274	021616	11476	11481#
A0275	021674	11520	11525#
A0276	021752	11564	11568#
A0277	022026	11606	11610#
A0300	022106	11649	11653#
A0301	022164	11693	11697#

A0302	022242	11736	11740#
A0303	022322	11779	11783#
A0304	022400	11822	11826#
A0305	022460	11865	11869#
A0306	022534	11907	11911#
A0307	022612	11950	11954#
A0310	022672	11993	11997#
A0311	022746	12036	12040#
A0312	023026	12080	12084#
A0313	023106	12124	12128#
A0314	023162	12168	12172#
A0315	023242	12211	12215#
A0316	023320	12254	12258#
A0317	023400	12297	12301#
A032	001720	3754	3759#
A0320	023456	12340	12344#
A0321	023536	12383	12387#
A0322	023616	12426	12430#
A0323	023674	12468	12472#
A0324	023750	12511	12516#
A0325	024032	12558	12563#
A0326	024112	12603	12607#
A0327	024170	12646	12650#
A033	001762	3793	3798#
A0330	024264	12696	12700#
A0331	024344	12741	12745#
A0332	024424	12785	12789#
A0333	024506	12829	12833#
A0334	024600	12876	12880#
A0335	024662	12920	12924#
A0336	024740	12963	12967#
A0337	025020	13007	13011#
A034	002024	3830	3835#
A0340	025102	13051	13055#
A0341	025160	13095	13099#
A0342	025242	13139	13143#
A0343	025324	13183	13187#
A0344	025402	13227	13231#
A0345	025464	13272	13276#
A0346	025544	13316	13320#
A0347	025626	13359	13363#
A0350	025706	13403	13407#
A0351	025770	13447	13451#
A0352	026052	13491	13495#
A0353	026132	13534	13538#
A0354	026214	13579	13584#
A0355	026274	13623	13628#
A0356	026354	13668	13673#
A0357	026434	13712	13717#
A0360	026514	13756	13761#
A0361	026576	13799	13804#
A0363	026750	13879	13884#
A0365	027062	13958	13963#
A0367	027214	14037	14042#
A0371	027346	14118	14123#
A0372	027432	14162	14167#

A0373	027506	14203	14208#
A0374	027576	14249	14254#
A0375	027662	14292	14297#
A0376	027740	14334	14339#
A0377	030024	14377	14382#
A0404	030352	14567	14575#
A0405	030444	14613	14621#
A0406	030540	14659	14667#
A0407	030634	14705	14713#
A041	002172	3985	3990#
A0410	030730	14751	14759#
A0411	031022	14797	14805#
A0412	031114	14843	14851#
A0413	031210	14889	14897#
A0420	031542	15091	15096#
A0421	031622	15136	15141#
A0422	031706	15181	15186#
A0423	031772	15227	15232#
A0424	032050	15272	15277#
A0425	032134	15318	15323#
A0426	032222	15364	15369#
A0427	032310	15410	15415#
A0430	032372	15459	15464#
A0431	032452	15507	15512#
A0432	032536	15555	15560#
A0433	032622	15603	15608#
A0434	032706	15651	15656#
A0435	032770	15697	15702#
A0436	033054	15742	15747#
A0437	033140	15786	15791#
A0440	033224	15831	15836#
A0441	033304	15875	15880#
A0442	033372	15919	15924#
A0443	033462	15964	15969#
A0444	033570	16015	16020#
A0445	033650	16057	16063#
A0446	033732	16100	16106#
A0447	034014	16143	16149#
A0450	034074	16187	16193#
A0453	034300	16311	16316#
A0454	034376	16360	16365#
A0455	034466	16408	16413#
A0456	034556	16457	16462#
A0457	034646	16506	16511#
A046	002374	4141	4146#
A0460	034750	16561	16566#
A0461	035052	16616	16621#
A0462	035154	16671	16676#
A0463	035260	16726	16731#
A0464	035364	16782	16787#
A0465	035470	16837	16842#
A0466	035574	16893	16898#
A0467	035700	16948	16953#
A0470	036002	17003	17008#
A0471	036104	17058	17063#
A0472	036206	17113	17118#

A0473	036310	17169	17174#
A0474	036414	17225	17230#
A0475	036520	17280	17285#
A0476	036624	17336	17341#
A0477	036732	17392	17397#
A0500	037026	17442	17447#
A0501	037122	17492	17497#
A0502	037216	17542	17547#
A0503	037312	17592	17597#
A0504	037420	17645	17650#
A0505	037514	17695	17700#
A0506	037610	17745	17750#
A0515	040224	18016	18021#
A0516	040306	18063	18068#
A0521	040474	18180	18185#
A0522	040552	18221	18226#
A0524	040702	18303	18309#
A0525	040766	18348	18353#
A0526	041050	18394	18399#
A0527	041134	18439	18444#
A0537	041672	18755	18760#
A0540	041750	18799	18804#
A0541	042034	18843	18848#
A0542	042112	18887	18892#
A0543	042176	18932	18937#
A0544	042274	18981	18986#
A0545	042356	19026	19031#
A0546	042440	19071	19076#
A0547	042522	19116	19121#
A0550	042606	19160	19165#
A0551	042666	19205	19210#
A0552	042754	19250	19255#
A0553	043034	19295	19300#
A0554	043122	19340	19345#
A0555	043210	19385	19390#
A0556	043276	19430	19435#
A0557	043362	19476	19481#
A0560	043446	19521	19526#
A0561	043534	19566	19571#
A0562	043614	19610	19615#
A0563	043704	19655	19660#
A0564	043764	19699	19704#
A0565	044046	19743	19748#
A0566	044130	19787	19792#
A0567	044212	19831	19836#
A0570	044274	19876	19881#
A0571	044362	19921	19926#
A0572	044446	19965	19970#
A0573	044534	20010	20015#
A0574	044626	20056	20061#
A0575	044720	20103	20108#
A0576	045012	20149	20154#
A0577	045104	20195	20200#
A0600	045174	20242	20247#
A0601	045266	20289	20294#
A0602	045356	20335	20340#

A0617	046376	20834	20843#	
A0620	046442	20875	20884#	
A0621	046506	20915	20924#	20932
A0622	046564	20962	20971#	
A0623	046630	21012#	21026	
A0624	046716	21066#	21075	
A0625	047002	21111	21116#	
A0626	047060	21154	21163#	
A0627	047124	21204#	21218	
A0630	047212	21255#	21264	
A0631	047270	21295	21301#	
A0632	047344	21341	21347#	
A0633	047426	21397#	21406	
A0634	047512	21448#	21457	
A0635	047566	21484	21492#	
A0636	047644	21523	21533#	
A0637	047722	21569	21577#	
A0640	047770	21607	21615#	
A0641	050040	21647	21655#	
A0642	050110	21696#	21705	
A0643	050170	21742#		
A0644	050260	21790#	21799	
A0645	050342	21831	21836#	
A0646	050414	21878#	21887	
A0650	050520	21955#	21959	
A0655	051010	22147	22157#	
A0656	051116	22205	22215#	
A0657	051230	22264	22274#	
A0660	051334	22318	22328#	
A0661	051444	22370	22390#	
A0662	051602	22442	22462#	
A0663	051726	22503	22510#	
A0666	052160	22573	22582	22589#
A0667	052276	22609	22624#	
A0670	052406	22664	22670#	
A0671	052522	22715	22721#	
A0672	052662	22758	22782#	
A0673	053066	22835	22859#	
A0674	053260	22899	22915#	
A0675	053402	22933	22949#	
A0676	053524	22967	22983#	
A0677	053646	23001	23017#	
A0700	053770	23036	23052#	
A0701	054112	23070	23086#	
A0702	054226	23102	23118#	
A0703	054342	23134	23150#	
A0704	054470	23169	23185#	
A0705	054604	23201	23217#	
A0706	054720	23234	23247#	
A0707	055044	23269	23282#	
A0710	055162	23302	23315#	
A0711	055300	23335	23348#	
A0712	055416	23368	23381#	
A0713	055534	23401	23414#	
A0714	055700	23442	23460#	
A0715	056116	23492	23521#	

BT017	001312	3365	3388#	3400	3407
BT020	001352	3404	3427#	3437	3443
BT021	001402	3440	3463#	3473	3480
BT022	001434	3477	3502#	3512	
BT023	001460	3509	3532#	3545	3552
BT024	001524	3549	3573#	3582	
BT025	001550	3579	3604#	3612	
BT026	001570	3609	3632#	3640	
BT027	001610	3637	3661#	3669	
BT030	001630	3666	3690#	3698	
BT031	001650	3695	3718#	3726	
BT032	001670	3723	3747#	3757	3764
BT033	001732	3761	3786#	3796	3803
BT034	001774	3800	3823#	3833	3840
BT035	002036	3837	3860#	3869	
BT036	002056	3866	3889#	3898	
BT037	002076	3895	3918#	3927	
BT040	002116	3924	3947#	3958	
BT041	002144	3955	3977#	3988	3995
BT042	002204	3992	4016#	4026	
BT043	002224	4023	4046#	4057	
BT044	002246	4054	4077#		
BT045	002274	4084	4087	4106#	4116
BT046	002322	4113	4122#	4144	4150
B0024	004164	4833	4838#		
B0077	007524	6418	6423#		
B0100	007612	6464	6469#		
B0101	007716	6518	6523#		
B0102	010022	6572	6577#		
B0103	010124	6626	6631#		
B0116	011270	7108	7114#		
B0117	011370	7163	7169#		
B0126	012042	7421	7427#		
B0130	012230	7506	7512#		
B0132	012434	7600	7606#		
B0133	012546	7652	7658#		
B0230	016224	9812	9817#		
B0236	016712	10102	10107#		
B0264	020762	11118	11123#		
B0266	021132	11208	11213#		
B0270	021304	11299	11304#		
B0272	021452	11388	11393#		
B0327	024202	12651	12656#		
B0457	034660	16512	16517#		
B0460	034762	16567	16572#		
B0461	035064	16622	16627#		
B0462	035166	16677	16682#		
B0463	035272	16732	16737#		
B0464	035376	16788	16793#		
B0465	035502	16843	16848#		
B0466	035606	16899	16904#		
B0467	035710	16954	16959#		
B0470	036012	17009	17014#		
B0471	036114	17064	17069#		
B0472	036216	17119	17124#		
B0473	036322	17175	17180#		



ERRA 065400
ERRB 065406
ERRCNT 066666
ERRFLG 066676
ERROR = 104000

7723	26663#													
8023	26666#													
4160	26307	26388	26390*	26404	26406*	26488	26490*	26559	26561*	26666	26668*	26992#		
7725*	7730*	26663*	26996#											
2930#	7728	9491	9524	9627	9633	9668	9707	9714	9750	9792	9798	9814		
9820	9826	9861	9900	9906	9942	9986	9992	10033	10039	10082	10088	10104		
10110	10116	10157	10163	10203	10209	10249	10255	10295	10301	10342	10348	10388		
10394	10434	10440	10480	10486	10526	10532	10578	10623	10668	10713	10758	10803		
10849	10895	10941	10987	11033	11079	11127	11169	11217	11259	11308	11350	11397		
11438	11484	11528	11571	11613	11656	11700	11743	11786	11829	11872	11914	11957		
12000	12043	12087	12131	12175	12218	12262	12304	12347	12390	12433	12475	12520		
12567	12611	12660	12704	12749	12793	12837	12884	12928	12971	13015	13059	13103		
13147	13191	13235	13280	13324	13367	13411	13455	13499	13542	13581	13587	13625		
13631	13670	13676	13714	13720	13758	13764	13808	13846	13888	13924	13967	14004		
14046	14082	14126	14170	14205	14251	14294	14336	14379	14421	14458	14495	14532		
14571	14617	14663	14709	14755	14801	14847	14893	14939	14978	15017	15056	15093		
15099	15138	15144	15183	15189	15229	15235	15274	15281	15320	15327	15366	15373		
15412	15419	15461	15467	15509	15515	15557	15563	15605	15611	15653	15659	15706		
15751	15795	15839	15883	15927	15973	16024	16060	16103	16146	16190	16234	16272		
16321	16370	16418	16467	16522	16577	16632	16687	16742	16798	16853	16909	16964		
17019	17074	17129	17185	17241	17296	17352	17402	17452	17502	17552	17602	17655		
17705	17755	17792	17829	17866	17904	17941	17980	18024	18071	18109	18145	18182		
18223	18267	18306	18357	18403	18448	18487	18526	18566	18604	18642	18680	18719		
18763	18807	18851	18895	18941	18989	19034	19079	19124	19169	19214	19259	19304		
19349	19394	19439	19485	19530	19574	19618	19663	19707	19751	19795	19839	19884		
19929	19974	20019	20065	20112	20158	20204	20251	20298	20344	20383	20425	20464		
20503	20542	20581	20620	20659	20696	20734	20770	20809	22168	22176	22226	22234		
22278	22288	22332	22342	22536	22558	22585	22675	22726	22795	22872	24324	24370		
24381	24424	24435	24479	24490	25955	26024	26097	26167	26225	26277				
2931#														
2932#														
2933#														
2934#														
2935#	8068	8099	8104	8109	8114	8146	8175	8205	8236	8268	8301	8333		
8365	8402	8433	8442	8479	8510	8519	8550	8583	8614	8646	8678	8710		
8742	8775	8808	8844	8877	8906	8940	8973	9006	9037	9070	9102	9134		
9167	9200	9231	9264	9297	9330	9362	9395	9428	9460	9556	9588	10572		
10617	10662	10707	10752	10797	10842	10888	10934	10980	11026	11072	11114	11120		
11162	11204	11210	11252	11295	11301	11343	11384	11390	11431	11478	11522	11566		
11608	11651	11695	11738	11781	11824	11867	11909	11952	11995	12038	12082	12126		
12170	12213	12256	12299	12342	12385	12428	12470	12513	12560	12605	12648	12653		
12698	12743	12787	12831	12878	12922	12965	13009	13053	13097	13141	13185	13229		
13274	13318	13361	13405	13449	13493	13536	13801	13881	13960	14039	14120	14164		
14211	14257	14300	14342	14385	14578	14624	14670	14716	14762	14808	14854	14900		
15699	15744	15788	15833	15877	15921	15966	16017	16066	16109	16152	16196	16313		
16362	16410	16459	16508	16514	16563	16569	16618	16624	16673	16679	16728	16734		
16784	16790	16839	16845	16895	16901	16950	16956	17005	17011	17060	17066	17115		
17121	17171	17177	17227	17233	17282	17288	17338	17344	17394	17444	17494	17544		
17594	17647	17697	17747	18018	18065	18188	18229	18312	18350	18396	18441	18757		
18801	18845	18889	18934	18983	19028	19073	19118	19162	19207	19252	19297	19342		
19387	19432	19478	19523	19568	19612	19657	19701	19745	19789	19833	19878	19923		
19967	20012	20058	20105	20151	20197	20244	20291	20337	21489	21497	21530	21536		
21542	21574	21582	21612	21619	21652	21660	21693	21701	21706	21739	21743	21751		
21787	21795	21800	21833	21842	21874	21883	21888	22153	22162	22211	22220	22270		
22324	22377	22385	22404	22449	22457	22476	23581	23623	23631	23638	23677	23685		
23692	23716	23741	23766	23803	24093	24102	24125	24134	24158	24167	24204	24242		

ERROR1= 104001
ERROR2= 104002
ERROR3= 104003
ERROR4= 104004
ERROR5= 104005

E0072	007172	6229#					
E0073	007236	6263#					
E0075	007356	6337#					
E0076	007430	6373#					
E010	001126	3173	3176#				
E011	001146	3207	3210#				
E0116	011252	7104#					
E0117	011352	7159#					
E012	001166	3239#					
E0120	011450	7209#					
E0121	011522	7243#					
E0122	011574	7277#					
E0123	011646	7312#					
E0127	012126	7467#					
E013	001212	3270#					
E0131	012330	7560#					
E0133	012542	7655#					
E0134	012630	7697#					
E0135	012702	7733#					
E0136	012762	7770#					
E0137	013060	7811#					
E014	001234	3300#					
E0140	013150	7838	7852#				
E0141	013236	7891#					
E0142	013276	7924#					
E0143	013340	7957#					
E0145	013670	8062	8063	8064	8065	8068#	
E0147	013760	8143	8146#				
E0150	014002	8175#					
E0151	014026	8202	8205#				
E0152	014050	8236#					
E0153	014074	8265	8268#				
E0154	014116	8301#					
E0155	014142	8330	8333#				
E0156	014164	8365#					
E0157	014224	8394	8402#				
E016	001306	3362	3363	3364	3367#		
E0161	014326	8471	8479#				
E0163	014412	8550#					
E0164	014440	8580	8583#				
E0165	014466	8611	8614#				
E0166	014512	8646#					
E0167	014536	8675	8678#				
E0170	014562	8710#					
E0171	014606	8742#					
E0172	014634	8772	8775#				
E0173	014662	8805	8808#				
E0174	014722	8841	8844#				
E0175	014750	8874	8877#				
E0176	014772	8906#					
E0177	015020	8937	8940#				
E0200	015046	8970	8973#				
E0201	015074	9003	9006#				
E0202	015120	9037#					
E0203	015144	9067	9070#				
E0204	015170	9102#					

E0205	015214	9134#			
E0206	015240	9167#			
E0207	015266	9197	9200#		
E0210	015310	9231#			
E0211	015336	9261	9264#		
E0212	015364	9294	9297#		
E0213	015412	9326	9330#		
E0214	015436	9359	9362#		
E0215	015462	9395#			
E0216	015506	9428#			
E0217	015532	9460#			
E022	001454	3506	3507	3508	3511#
E0220	015554	9491#			
E0221	015602	9521	9524#		
E0222	015626	9553	9556#		
E0223	015652	9588#			
E0225	015766	9668#			
E0227	016104	9750#			
E0231	016302	9861#			
E0232	001520	3551#			
E0233	016416	9942#			
E024	001544	3581#			
E025	001564	3608	3611#		
E026	001604	3636	3639#		
E027	001624	3668#			
E030	001644	3697#			
E031	001664	3725#			
E035	002052	3868#			
E036	002072	3897#			
E0362	026656	13846#			
E0364	027004	13924#			
E0366	027142	14004#			
E037	002112	3926#			
E0370	027270	14082#			
E040	002140	3957#			
E0400	030100	14421#			
E0401	030152	14458#			
E0402	030220	14495#			
E0403	030266	14532#			
E0414	031272	14939#			
E0415	031346	14978#			
E0416	031420	15017#			
E0417	031472	15056#			
E042	002220	4025#			
E043	002242	4056#			
E044	002270	4086#			
E045	002316	4115#			
E0451	034154	16234#			
E0452	034226	16272#			
E0507	037662	17792#			
E0510	037724	17829#			
E0511	037770	17866#			
E0512	040034	17904#			
E0513	040100	17941#			
E0514	040146	17980#			
E0517	040356	18109#			

E0520	040424	18145#			
E0523	040626	18267#			
E0530	041216	18487#			
E0531	041272	18526#			
E0532	041342	18566#			
E0533	041412	18604#			
E0534	041464	18642#			
E0535	041536	18680#			
E0536	041612	18719#			
E0603	045430	20383#			
E0604	045510	20425#			
E0605	045560	20464#			
E0606	045632	20503#			
E0607	045704	20542#			
E0610	045756	20581#			
E0611	046032	20620#			
E0612	046106	20659#			
E0613	046152	20696#			
E0614	046224	20734#			
E0615	046270	20770#			
E0616	046342	20809#			
E0647	050462	21920#	21925		
E0651	050606	22001	22002	22003	22006#
E0652	050644	22038	22039	22040	22044#
E0653	050702	22076	22077	22078	22081#
E0654	050740	22113	22114	22115	22118#
E0663	051734	22511#			
E0664	052010	22536#			
E0665	052056	22558#			
E0667	052272	22621#			
E0674	053254	22912#			
E0675	053376	22946#			
E0676	053520	22980#			
E0677	053642	23014#			
E0700	053764	23049#			
E0701	054106	23083#			
E0702	054222	23115#			
E0703	054336	23147#			
E0704	054464	23182#			
E0705	054600	23214#			
E0706	054724	23248#			
E0707	055050	23283#			
E0710	055166	23316#			
E0711	055304	23349#			
E0712	055422	23382#			
E0713	055540	23415#			
E0716	056274	23581#			
E0721	056612	23716#			
E0722	056672	23741#			
E0723	056752	23766#			
E0724	057030	23803#			
E0735	060322	24198	24204#		
E0736	060400	24236	24242#		
E0737	060456	24279#			
E0744	061174	24530#			
E0745	061264	24574#			

E0746	061350	24618#	
E0747	061436	24662#	
E0750	061524	24706#	
E0751	061612	24749#	
E0752	061670	24791#	
E0753	061756	24835#	
E0754	062034	24877#	
E0755	062144	24932#	
E0756	062210	24968#	
E0757	062254	25006#	
E0760	062314	25041#	
E0761	062354	25077#	
E0762	062420	25115#	
E0763	062464	25153#	
E0764	062524	25188#	
E0765	062564	25223#	
E0766	062624	25259#	
E0767	062670	25296#	
E0770	062730	25331#	
E0771	062774	25369#	
E0772	063034	25405#	
E0773	063074	25440#	
E0774	063134	25475#	
E0775	063200	25513#	
E0776	063244	25551#	
E0777	063310	25579	25588#
E1000	063350	25621	25623#
E1001	063410	25656	25658#
E10012	003350	4456#	
E10013	003410	4494#	
E1002	063446	25692#	
E10021	003760	4710	4713#
E10022	004030	4752#	
E10023	004076	4791#	
E10024	004150	4829#	
E1003	063504	25726#	
E10032	004522	5046#	
E10033	004572	5084#	
E10034	004642	5121#	
E10035	004712	5159#	
E10036	004756	5196#	
E10037	005032	5235#	
E1004	063550	25754	25762#
E10040	005116	5277#	
E10041	005170	5316#	
E10046	005506	5501#	
E1005	063630	25798	25802#
E10053	006000	5673#	
E10054	006060	5715#	
E10056	006210	5791#	
E10057	006266	5830#	
E1006	063672	25835#	25839
E10062	006476	5942#	
E10063	006552	5982#	
E1007	063762	25884	25886#
E10074	007276	6296#	

E10077	007510	6414#			
E1010	064032	25955#			
E10100	007574	6460#			
E10101	007700	6514#			
E10102	010004	6568#			
E10103	010106	6622#			
E10104	010224	6680#			
E10105	010314	6723#			
E10106	010404	6766#			
E10107	010474	6808#			
E1011	064110	26024#			
E10110	010562	6850#			
E10111	010650	6893#			
E10112	010740	6935#			
E10113	011030	6978#			
E10114	011106	7019#			
E10115	011164	7060#			
E10116	011264	7111#			
E10117	011364	7166#			
E1012	064166	26097#			
E10124	011700	7342#			
E10125	011740	7377#	7380		
E10126	012014	7416#			
E1013	064244	26167#			
E10130	012202	7501#	7512		
E10132	012406	7595#	7606		
E1014	064306	26225#			
E10146	013712	8099#			
E1015	064352	26277#			
E1015A	001252	3328#			
E10160	014246	8433#			
E10162	014350	8510#			
E1017	001334	3394	3395	3396	3399#
E1020	001366	3436#			
F1021	001416	3472#			
E10224	015720	9622	9623	9624	9627#
E10226	016032	9702	9703	9704	9707#
E1023	001506	3539	3540	3541	3544#
E10230	016152	9787	9788	9789	9792#
E10232	016346	9895	9896	9897	9900#
E10234	016500	9981	9982	9983	9986#
E10235	016560	10028	10029	10030	10033#
E10236	016640	10077	10078	10079	10082#
E10237	017002	10152	10153	10154	10157#
E10240	017056	10198	10199	10200	10203#
E10241	017136	10244	10245	10246	10249#
E10242	017216	10290	10291	10292	10295#
E10243	017274	10337	10338	10339	10342#
E10244	017350	10383	10384	10385	10388#
E10245	017430	10429	10430	10431	10434#
E10246	017510	10475	10476	10477	10480#
E10247	017566	10521	10522	10523	10526#
E10250	017646	10567	10568	10569	10572#
E10251	017724	10612	10613	10614	10617#
E10252	020004	10657	10658	10659	10662#
E10253	020062	10702	10703	10704	10707#

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 701
 CBQEAC.P11 03-JUL-80 08:05 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0700

E10254	020142	10747	10748	10749	10752#
E10255	020222	10792	10793	10794	10797#
E10256	020302	10837	10838	10839	10842#
E10257	020362	10883	10884	10885	10888#
E10260	020444	10929	10930	10931	10934#
E10261	020524	10975	10976	10977	10980#
E10262	020606	11021	11022	11023	11026#
E10263	020670	11067	11068	11069	11072#
E10264	020744	11114#			
E10265	021034	11162#			
E10266	021114	11204#			
E10267	021206	11252#			
E10270	021266	11295#			
E10271	021360	11343#			
E10272	021434	11384#			
E10273	021522	11431#			
E10274	021612	11473	11474	11475	11478#
E10275	021670	11517	11518	11519	11522#
E10276	021746	11561	11562	11563	11566#
E10277	022022	11603	11604	11605	11608#
E10300	022102	11646	11647	11648	11651#
E10301	022160	11690	11691	11692	11695#
E10302	022236	11733	11734	11735	11738#
E10303	022316	11776	11777	11778	11781#
E10304	022374	11819	11820	11821	11824#
E10305	022454	11862	11863	11864	11867#
E10306	022530	11904	11905	11906	11909#
E10307	022606	11947	11948	11949	11952#
E10310	022666	11990	11991	11992	11995#
E10311	022742	12033	12034	12035	12038#
E10312	023022	12077	12078	12079	12082#
E10313	023102	12121	12122	12123	12126#
E10314	023156	12165	12166	12167	12170#
E10315	023236	12208	12209	12210	12213#
E10316	023314	12251	12252	12253	12256#
E10317	023374	12294	12295	12296	12299#
E1032	001714	3756#			
E10320	023452	12337	12338	12339	12342#
E10321	023532	12380	12381	12382	12385#
E10322	023612	12423	12424	12425	12428#
E10323	023670	12465	12466	12467	12470#
E10324	023744	12508	12509	12510	12513#
E10325	024026	12555	12556	12557	12560#
E10326	024106	12600	12601	12602	12605#
E10327	024164	12643	12644	12645	12648#
E1033	001756	3795#			
E10330	024260	12693	12694	12695	12698#
E10331	024340	12738	12739	12740	12743#
E10332	024420	12782	12783	12784	12787#
E10333	024502	12826	12827	12828	12831#
E10334	024574	12873	12874	12875	12878#
E10335	024656	12917	12918	12919	12922#
E10336	024734	12960	12961	12962	12965#
E10337	025014	13004	13005	13006	13009#
E1034	002020	3832#			
E10340	025076	13048	13049	13050	13053#

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 702
 CBQEAC.P11 03-JUL-80 08:05 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0701

E10341	025154	13092	13093	13094	13097#
E10342	025236	13136	13137	13138	13141#
E10343	025320	13180	13181	13182	13185#
E10344	025376	13224	13225	13226	13229#
E10345	025460	13269	13270	13271	13274#
E10346	025540	13313	13314	13315	13318#
E10347	025622	13356	13357	13358	13361#
E10350	025702	13400	13401	13402	13405#
E10351	025764	13444	13445	13446	13449#
E10352	026046	13488	13489	13490	13493#
E10353	026126	13531	13532	13533	13536#
E10354	026210	13576	13577	13578	13581#
E10355	026270	13620	13621	13622	13625#
E10356	026350	13665	13666	13667	13670#
E10357	026430	13709	13710	13711	13714#
E10360	026510	13753	13754	13755	13758#
E10361	026572	13801#			
E10363	026724	13881#			
E10365	027056	13960#			
E10367	027210	14039#			
E10371	027342	14115	14116	14117	14120#
E10372	027426	14159	14160	14161	14164#
E10373	027502	14205#			
E10374	027572	14251#			
E10375	027656	14294#			
E10376	027734	14336#			
E10377	030020	14379#			
E10404	030342	14571#			
E10405	030434	14617#			
E10406	030530	14663#			
E10407	030624	14709#			
E1041	002166	3987#			
E10410	030720	14755#			
E10411	031012	14801#			
E10412	031104	14847#			
E10413	031200	14893#			
E10420	031536	15088	15089	15090	15093#
E10421	031616	15133	15134	15135	15138#
E10422	031702	15178	15179	15180	15183#
E10423	031766	15224	15225	15226	15229#
E10424	032044	15269	15270	15271	15274#
E10425	032130	15315	15316	15317	15320#
E10426	032216	15361	15362	15363	15366#
E10427	032304	15407	15408	15409	15412#
E10430	032366	15456	15457	15458	15461#
E10431	032446	15504	15505	15506	15509#
E10432	032532	15552	15553	15554	15557#
E10433	032616	15600	15601	15602	15605#
E10434	032702	15648	15649	15650	15653#
E10435	032764	15694	15695	15696	15699#
E10436	033050	15739	15740	15741	15744#
E10437	033134	15783	15784	15785	15788#
E10440	033220	15828	15829	15830	15833#
E10441	033300	15872	15873	15874	15877#
E10442	033366	15916	15917	15918	15921#
E10443	033456	15961	15962	15963	15966#

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 703
 CBQEAC.P11 03-JUL-80 08:05 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0702

E10444	033564	16012	16013	16014	16017#
E10445	033644	16060#			
E10446	033726	16103#			
E10447	034010	16146#			
E10450	034070	16190#			
E10453	034302	16308	16309	16310	16313#
E10454	034372	16357	16358	16359	16362#
E10455	034462	16405	16406	16407	16410#
E10456	034552	16454	16455	16456	16459#
E10457	034642	16503	16504	16505	16508#
E1046	002370	4143#			
E10460	034744	16558	16559	16560	16563#
E10461	035046	16613	16614	16615	16618#
E10462	035150	16668	16669	16670	16673#
E10463	035254	16723	16724	16725	16728#
E10464	035360	16779	16780	16781	16784#
E10465	035464	16834	16835	16836	16839#
E10466	035570	16890	16891	16892	16895#
E10467	035674	16945	16946	16947	16950#
E10470	035776	17000	17001	17002	17005#
E10471	036100	17055	17056	17057	17060#
E10472	036202	17110	17111	17112	17115#
E10473	036304	17166	17167	17168	17171#
E10474	036410	17222	17223	17224	17227#
E10475	036514	17277	17278	17279	17282#
E10476	036620	17333	17334	17335	17338#
E10477	036726	17389	17390	17391	17394#
E10500	037022	17439	17440	17441	17444#
E10501	037116	17489	17490	17491	17494#
E10502	037212	17539	17540	17541	17544#
E10503	037306	17589	17590	17591	17594#
E10504	037414	17642	17643	17644	17647#
E10505	037510	17692	17693	17694	17697#
E10506	037604	17742	17743	17744	17747#
E10515	040220	18013	18014	18015	18018#
E10516	040302	18060	18061	18062	18065#
E10521	040470	18182#			
E10522	040546	18223#			
E10524	040676	18306#			
E10525	040762	18350#			
E10526	041044	18396#			
E10527	041130	18441#			
E10537	041666	18752	18753	18754	18757#
E10540	041744	18796	18797	18798	18801#
E10541	042030	18840	18841	18842	18845#
E10542	042106	18884	18885	18886	18889#
E10543	042172	18929	18930	18931	18934#
E10544	042270	18978	18979	18980	18983#
E10545	042352	19023	19024	19025	19028#
E10546	042434	19068	19069	19070	19073#
E10547	042516	19113	19114	19115	19118#
E10550	042602	19157	19158	19159	19162#
E10551	042662	19202	19203	19204	19207#
E10552	042750	19247	19248	19249	19252#
E10553	043030	19292	19293	19294	19297#
E10554	043116	19337	19338	19339	19342#

E10555	043204	19382	19383	19384	19387#
E10556	043272	19427	19428	19429	19432#
E10557	043356	19473	19474	19475	19478#
E10560	043442	19518	19519	19520	19523#
E10561	043530	19563	19564	19565	19568#
E10562	043610	19607	19608	19609	19612#
E10563	043700	19652	19653	19654	19657#
E10564	043760	19696	19697	19698	19701#
E10565	044042	19740	19741	19742	19745#
E10566	044124	19784	19785	19786	19789#
E10567	044206	19828	19829	19830	19833#
E10570	044270	19873	19874	19875	19878#
E10571	044356	19918	19919	19920	19923#
E10572	044442	19962	19963	19964	19967#
E10573	044530	20007	20008	20009	20012#
E10574	044622	20053	20054	20055	20058#
E10575	044714	20100	20101	20102	20105#
E10576	045006	20146	20147	20148	20151#
E10577	045100	20192	20193	20194	20197#
E10600	045170	20239	20240	20241	20244#
E10601	045262	20286	20287	20288	20291#
E10602	045352	20332	20333	20334	20337#
E10617	046370	20839#			
E10620	046434	20880#			
E10621	046500	20920#			
E10622	046556	20967#			
E10623	046622	21008#			
E10624	046710	21061#			
E10625	046764	21107#			
E10626	047052	21159#			
E10627	047116	21200#			
E10630	047204	21251#			
E10631	047262	21297#			
E10632	047336	21343#			
E10633	047412	21389#			
E10634	047476	21440#			
E10635	047562	21489#			
E10636	047640	21530#	21539		
E10637	047716	21574#			
E10640	047764	21612#			
E10641	050034	21652#			
E10642	050104	21693#			
E10643	050164	21739#			
E10644	050254	21787#			
E10645	050336	21828	21831	21833#	
E10646	050410	21869	21872	21874#	
E10650	050542	21968#			
E10655	051002	22153#			
E10656	051110	22211#			
E10657	051222	22270#			
E10660	051326	22324#			
E10661	051424	22377#			
E10662	051562	22449#			
E10666	052154	22585#			
E10670	052402	22667#			
E10671	052516	22718#			

E10672	052654	22778#
E10673	053060	22855#
E10714	055706	23463#
E10715	056130	23526#
E10717	056366	23623#
E10720	056506	23677#
E10725	057100	23826#
E10726	057214	23869#
E10727	057330	23913#
E10730	057504	23972#
E10731	057656	24034#
E10732	060030	24093#
E10733	060124	24125#
E10734	060224	24158#
E10740	060532	24316#
E10741	060640	24370#
E10742	060746	24424#
E10743	061064	24479#
E20012	003360	4463#
E20013	003420	4500#
E2002	001014	3006#
E20021	003772	4719#
E20022	004042	4759#
E20023	004110	4797#
E20024	004160	4835#
E20032	004532	5052#
E20033	004602	5090#
E20034	004652	5127#
E20035	004722	5165#
E20036	004770	5203#
E20037	005044	5242#
E20040	005130	5284#
E20041	005202	5323#
E20046	005520	5507#
E20053	006010	5679#
E20054	006070	5721#
E20056	006222	5797#
E20057	006300	5836#
E20062	006510	5948#
E20063	006564	5989#
E20074	007310	6303#
E20077	007520	6420#
E20100	007606	6466#
E20101	007712	6520#
E20102	010016	6574#
E20103	010120	6628#
E20104	010236	6686#
E20105	010326	6729#
E20106	010416	6772#
E20107	010506	6814#
E20110	010574	6856#
E20111	010662	6899#
E20112	010752	6941#
E20113	011042	6984#
E20114	011120	7025#
E20115	011176	7066#

E20116	011274	7117#
E20117	011374	7172#
E20124	011710	7348#
E20125	011752	7384#
E20126	012036	7424#
E20130	012224	7509#
E20132	012430	7603#
E20146	013720	8104#
E2015	001262	3335#
E20160	014270	8442#
E20162	014370	8519#
E2017	001346	3406#
E2020	001376	3442#
E2021	001430	3479#
E20224	015730	9633#
E20226	016044	9714#
E20230	016162	9798#
E20232	016356	9906#
E20234	016510	9992#
E20235	016570	10039#
E20236	016650	10088#
E20237	017012	10163#
E20240	017066	10209#
E20241	017146	10255#
E20242	017226	10301#
E20243	017304	10348#
E20244	017360	10394#
E20245	017440	10440#
E20246	017520	10486#
E20247	017576	10532#
E20250	017656	10578#
E20251	017734	10623#
E20252	020014	10668#
E20253	020072	10713#
E20254	020152	10758#
E20255	020232	10803#
E20256	020314	10849#
E20257	020374	10895#
E20260	020456	10941#
E20261	020536	10987#
E20262	020620	11033#
E20263	020702	11079#
E20264	020756	11120#
E20265	021046	11169#
E20266	021126	11210#
E20267	021220	11259#
E20270	021300	11301#
E20271	021372	11350#
E20272	021446	11390#
E20273	021534	11438#
E20274	021622	11484#
E20275	021700	11528#
E20276	021756	11571#
E20277	022032	11613#
E20300	022112	11656#
E20301	022170	11700#

E20302	022246	11743#
E20303	022326	11786#
E20304	022404	11829#
E20305	022464	11872#
E20306	022540	11914#
E20307	022616	11957#
E20310	022676	12000#
E20311	022752	12043#
E20312	023032	12087#
E20313	023112	12131#
E20314	023166	12175#
E20315	023246	12218#
E20316	023324	12262#
E20317	023404	12304#
E2032	001726	3763#
E20320	023462	12347#
E20321	023542	12390#
E20322	023622	12433#
E20323	023700	12475#
E20324	023756	12520#
E20325	024040	12567#
E20326	024120	12611#
E20327	024176	12653#
E2033	001770	3802#
E20330	024272	12704#
E20331	024352	12749#
E20332	024432	12793#
E20333	024514	12837#
E20334	024606	12884#
E20335	024670	12928#
E20336	024746	12971#
E20337	025026	13015#
E2034	002032	3839#
E20340	025110	13059#
E20341	025166	13103#
E20342	025250	13147#
E20343	025332	13191#
E20344	025410	13235#
E20345	025472	13280#
E20346	025552	13324#
E20347	025634	13367#
E20350	025714	13411#
E20351	025776	13455#
E20352	026060	13499#
E20353	026140	13542#
E20354	026220	13587#
E20355	026300	13631#
E20356	026360	13676#
E20357	026440	13720#
E20360	026520	13764#
E20361	026604	13808#
E20363	026736	13888#
E20365	027070	13967#
E20367	027222	14046#
E20371	027352	14126#
E20372	027436	14170#

E20373	027514	14211#
E20374	027604	14257#
E20375	027670	14300#
E20376	027746	14342#
E20377	030032	14385#
E20404	030360	14578#
E20405	030452	14624#
E20406	030546	14670#
E20407	030642	14716#
E2041	002200	3994#
E20410	030734	14762#
E20411	031026	14808#
E20412	031122	14854#
E20413	031216	14900#
E20420	031546	15099#
E20421	031626	15144#
E20422	031712	15189#
E20423	031776	15235#
E20424	032056	15281#
E20425	032142	15327#
E20426	032230	15373#
E20427	032316	15419#
E20430	032376	15467#
E20431	032456	15515#
E20432	032542	15563#
E20433	032626	15611#
E20434	032712	15659#
E20435	032776	15706#
E20436	033062	15751#
E20437	033146	15795#
E20440	033230	15839#
E20441	033310	15883#
E20442	033376	15927#
E20443	033470	15973#
E20444	033576	16024#
E20445	033656	16066#
E20446	033740	16109#
E20447	034020	16152#
E20450	034102	16196#
E20453	034316	16321#
E20454	034406	16370#
E20455	034476	16418#
E20456	034566	16467#
E20457	034654	16514#
E2046	002402	4149#
E20460	034756	16569#
E20461	035060	16624#
E20462	035162	16679#
E20463	035266	16734#
E20464	035372	16790#
E20465	035476	16845#
E20466	035602	16901#
E20467	035704	16956#
E20470	036006	17011#
E20471	036110	17066#
E20472	036212	17121#

E20473	036316	17177#
E20474	036422	17233#
E20475	036526	17288#
E20476	036632	17344#
E20477	036742	17402#
E20500	037036	17452#
E20501	037132	17502#
E20502	037226	17552#
E20503	037322	17602#
E20504	037430	17655#
E20505	037524	17705#
E20506	037620	17755#
E20515	040230	18024#
E20516	040312	18071#
E20521	040502	18188#
E20522	040560	18229#
E20524	040710	18312#
E20525	040774	18357#
E20526	041056	18403#
E20527	041142	18448#
E20537	041676	18763#
E20540	041754	18807#
E20541	042040	18851#
E20542	042116	18895#
E20543	042204	18941#
E20544	042300	18989#
E20545	042362	19034#
E20546	042444	19079#
E20547	042526	19124#
E20550	042614	19169#
E20551	042674	19214#
E20552	042762	19259#
E20553	043042	19304#
E20554	043130	19349#
E20555	043216	19394#
E20556	043304	19439#
E20557	043370	19485#
E20560	043454	19530#
E20561	043540	19574#
E20562	043620	19618#
E20563	043710	19663#
E20564	043770	19707#
E20565	044052	19751#
E20566	044134	19795#
E20567	044216	19839#
E20570	044300	19884#
E20571	044366	19929#
E20572	044454	19974#
E20573	044542	20019#
E20574	044634	20065#
E20575	044726	20112#
E20576	045020	20158#
E20577	045112	20204#
E20600	045202	20251#
E20601	045274	20298#
E20602	045364	20344#

.MAIN. MACY11 30A(1052) 03-JUL-80 08:08 PAGE 710
 CBQEAC.P11 03-JUL-80 08:05 CROSS REFERENCE TABLE -- USER SYMBOLS

SEQ 0709

E20617	046406	20843	20844	20845	20848#
E20620	046452	20884	20885	20886	20889#
E20621	046516	20924	20925	20926	20929#
E20622	046574	20971	20972	20973	20976#
E20623	046640	21012	21013	21014	21017#
E20624	046726	21066	21067	21068	21071#
E20625	046774	21102	21112#	21124	
E20626	047070	21163	21164	21165	21168#
E20627	047134	21204	21205	21206	21209#
E20630	047222	21255	21256	21257	21260#
E20631	047300	21301	21302	21303	21306#
E20632	047354	21347	21348	21349	21352#
E20633	047420	21384	21387	21393#	
E20634	047504	21435	21438	21444#	
E20635	047576	21497#			
E20636	047652	21536#			
E20637	047732	21577	21578	21579	21582#
E20640	050000	21615	21616	21617	21619#
E20641	050050	21660#			
E20642	050120	21701#			
E20643	050172	21734	21743#		
E20644	050270	21795#			
E20645	050352	21842#			
E20646	050424	21883#			
E20650	050550	21956	21971#		
E20655	051020	22157	22158	22159	22162#
E20656	051126	22215	22216	22217	22220#
E20657	051240	22278#			
E20660	051344	22332#			
E20661	051436	22385#			
E20662	051574	22457#			
E20670	052422	22675#			
E20671	052536	22726#			
E20672	052674	22786#			
E20673	053100	22863#			
E20714	055716	23468#			
E20715	056144	23533#			
E20717	056406	23631#			
E20720	056524	23685#			
E20725	057120	23835#			
E20726	057234	23878#			
E20727	057350	23922#			
E20730	057524	23982#			
E20731	057676	24044#			
E20732	060044	24102#			
E20733	060140	24134#			
E20734	060240	24167#			
E20740	060550	24324#			
E20741	060666	24381#			
E20742	060774	24435#			
E20743	061112	24490#			
E30024	004172	4842#			
E30077	007532	6427#			
E30100	007620	6472#			
E30101	007724	6526#			
E30102	010030	6580#			

E30103	010132	6634#			
E30126	012046	7430#			
E30130	012242	7516#			
E30132	012446	7610#			
E30146	013726	8109#			
E30230	016220	9809	9810	9811	9814#
E30236	016706	10099	10100	10101	10104#
E30264	020770	11127#			
E30266	021140	11217#			
E30270	021312	11308#			
E30272	021460	11397#			
E30327	024210	12660#			
E30457	034670	16522#			
E30460	034772	16577#			
E30461	035074	16632#			
E30462	035176	16687#			
E30463	035302	16742#			
E30464	035406	16798#			
E30465	035512	16853#			
E30466	035616	16909#			
E30467	035720	16964#			
E30470	036022	17019#			
E30471	036124	17074#			
E30472	036226	17129#			
E30473	036332	17185#			
E30474	036436	17241#			
E30475	036542	17296#			
E30476	036646	17352#			
E30621	046530	20935#			
E30623	046652	21023#			
E30624	046736	21076#			
E30625	047012	21116	21117	21118	21121#
E30627	047146	21215#			
E30631	047306	21292	21295	21310#	
E30632	047362	21338	21341	21356#	
E30633	047436	21397	21398	21399	21402#
E30634	047522	21448	21449	21450	21453#
E30636	047664	21542#			
E30642	050130	21706#			
E30643	050206	21751#			
E30644	050300	21782	21800#		
E30646	050434	21888#			
E30655	051030	22168#			
E30656	051136	22226#			
E30657	051254	22288#			
E30660	051360	22342#			
E30661	051462	22397#			
E30662	051620	22469#			
E30672	052722	22795#			
E30673	053126	22872#			
E30715	056162	23540#			
E30717	056424	23638#			
E30720	056536	23692#			
E30725	057140	23844#			
E30726	057254	23886#			
E30727	057370	23931#			

E30730	057544	23992#							
E30731	057716	24054#							
E40100	007632	6479#							
E40101	007736	6533#							
E40102	010042	6587#							
E40103	010144	6641#							
E40130	012256	7523#							
E40132	012462	7617#							
E40146	013734	8114#							
E40230	016232	9820#							
E40236	016720	10110#							
E40623	046662	21027#							
E40625	047024	21127#							
E40633	047446	21408#							
E40634	047532	21459#							
E40655	051044	22176#							
E40656	051152	22234#							
E40661	051500	22404#							
E40662	051636	22476#							
E40672	052730	22768	22799#						
E40673	053134	22845	22876#						
E40715	056174	23546#							
E40727	057434	23949#							
E40730	057566	24003#							
E40731	057740	24064#							
E50230	016244	9826#							
E50236	016732	10116#							
E50661	051516	22414#							
E50662	051654	22486#							
E50715	056206	23552#							
FILLS	066660	26833	26989#						
FIRST	066704	4669	26372	26999#					
F0661	051506	22401	22409#						
F0662	051644	22473	22481#						
F0715	056170	23506	23544#						
G0661	051522	22410	22417#						
G0662	051660	22482	22489#						
G0715	056202	23515	23550#						
H0715	056212	23528	23531	23535	23542	23548	23555#		
IBUF	067534	7978	7999	27092#					
ICOUNT	066662	4158*	26321*	26609	26990#				
IDENT1	067212	8035	27047#						
INIT	003000	4147	4157#						
ITCNT	066664	4159*	22654*	22705*	26322*	26607*	26609*	26991#	
I0001	003046	4187#							
I0002	003062	4213#							
I0003	003100	4241#							
I0004	003114	4266#							
I0005	003136	4295#							
I0006	003160	4324#							
I0007	003214	4355#							
I0010	003252	4387#							
I0011	003310	4420#							
I0012	003342	4452#							
I0013	003402	4490#							
I0014	003444	4527#							

I0015	003500	4560#	
I0016	003544	4597#	
I0017	003612	4632#	
I002	001012	2998	3004#
I0020	003706	4676#	
I0021	003750	4708#	
I0022	004022	4748#	
I0023	004072	4787#	
I0024	004144	4825#	
I0025	004226	4872#	
I0026	004274	4906#	
I0027	004340	4940#	
I0030	004400	4973#	
I0031	004450	5008#	
I0032	004516	5042#	
I0033	004566	5080#	
I0034	004636	5117#	
I0035	004706	5155#	
I0036	004752	5192#	
I0037	005026	5231#	
I004	001034	3056#	
I0040	005110	5273#	
I0041	005162	5312#	
I0042	005240	5353#	
I0043	005312	5391#	
I0044	005360	5425#	
I0045	005430	5461#	
I0046	005500	5496#	
I0047	005552	5535#	
I005	001052	3085#	
I0050	005612	5563	5566#
I0051	005652	5598#	
I0052	005722	5633#	
I0053	005770	5667#	
I0054	006046	5709#	
I0055	006126	5750#	
I0056	006176	5785#	
I0057	006260	5825#	
I006	001070	3114#	
I0060	006336	5865#	
I0061	006412	5900#	
I0062	006464	5936#	
I0063	006544	5978#	
I0064	006620	6018#	
I0065	006660	6051#	
I0066	006722	6084#	
I0067	006764	6116#	
I007	001104	3142#	
I0070	007034	6150#	
I0071	007104	6185#	
I0072	007152	6221#	
I0073	007226	6259#	
I0074	007266	6292#	
I0075	007344	6331#	
I0076	007414	6367#	
I0077	007500	6409#	

I010	001120	3171#	
I0100	007570	6456#	
I0101	007674	6510#	
I0102	010000	6564#	
I0103	010102	6618#	
I0104	010212	6674#	
I0105	010302	6717#	
I0106	010372	6760#	
I0107	010462	6802#	
I011	001136	3204#	
I0110	010550	6844#	
I0111	010636	6887#	
I0112	010726	6929#	
I0113	011016	6972#	
I0114	011100	7014#	
I0115	011156	7055#	
I0116	011242	7099#	
I0117	011342	7154#	
I012	001156	3234#	
I0120	011434	7203#	
I0121	011506	7237#	
I0122	011560	7271#	
I0123	011632	7306#	
I0124	011676	7340#	
I0125	011734	7375#	
I0126	012012	7414#	
I0127	012106	7460#	
I013	001200	3264#	
I0130	012200	7499#	
I0131	012320	7555#	
I0132	012404	7593#	
I0133	012526	7649#	
I0134	012620	7692#	
I0135	012672	7728#	
I0136	012752	7756	7765#
I0137	013046	7797	7806#
I014	001230	3296#	
I0140	013136	7845#	
I0141	013224	7884#	7888
I0142	013270	7919#	
I0143	013332	7952#	
I0145	013656	8059	8062#
I0147	013754	8139	8143#
I015	001246	3324#	
I0150	014000	8171	8174#
I0151	014022	8198	8202#
I0152	014046	8230	8234#
I0153	014070	8261	8265#
I0154	014114	8295	8299#
I0155	014136	8326	8330#
I0156	014162	8359	8363#
I0157	014204	8390	8394#
I016	001274	3360#	
I0160	014244	8427	8431#
I0161	014310	8467	8471#
I0162	014346	8504	8508#

I0163	014410	8544	8548#
I0164	014434	8575	8580#
I0165	014462	8606	8611#
I0166	014510	8609	8644#
I0167	014532	8671	8675#
I017	001322	3392#	
I0170	014560	8703	8708#
I0171	014604	8735	8740#
I0172	014630	8767	8772#
I0173	014656	8800	8805#
I0174	014716	8833	8841#
I0175	014744	8869	8874#
I0176	014770	8900	8904#
I0177	015014	8932	8937#
I020	001362	3432#	
I0200	015042	8965	8970#
I0201	015070	8998	9003#
I0202	015116	9030	9035#
I0203	015140	9063	9067#
I0204	015166	9095	9100#
I0205	015212	9127	9132#
I0206	015236	9160	9165#
I0207	015262	9192	9197#
I021	001412	3468#	
I0210	015306	9225	9229#
I0211	015332	9256	9261#
I0212	015360	9289	9294#
I0213	015406	9321	9326#
I0214	015432	9355	9359#
I0215	015460	9388	9393#
I0216	015504	9421	9426#
I0217	015530	9453	9458#
I022	001436	3504#	
I0220	015552	9485	9489#
I0221	015576	9516	9521#
I0222	015622	9549	9553#
I0223	015650	9581	9586#
I0224	015706	9613	9620#
I0225	015762	9658	9665#
I0226	016020	9693	9700#
I0227	016100	9739	9747#
I023	001474	3537#	
I0231	016276	9851	9858#
I0232	016334	9886	9893#
I0233	016412	9931	9939#
I0234	016466	9968	9979#
I0235	016546	10018	10026#
I0237	016770	10142	10150#
I024	001534	3577#	
I0240	017044	10188	10196#
I0241	017124	10234	10242#
I0242	017204	10280	10288#
I0243	017262	10327	10335#
I0244	017336	10373	10381#
I0245	017416	10419	10427#
I0246	017476	10465	10473#

10247	017554	10511	10519#
1025	001552	3606#	
10250	017634	10558	10565#
10251	017712	10603	10610#
10252	017772	10648	10655#
10253	020050	10693	10700#
10254	020130	10738	10745#
10255	020210	10783	10790#
10256	020270	10828	10835#
10257	020350	10874	10881#
1026	001572	3634#	
10260	020432	10920	10927#
10261	020512	10966	10973#
10262	020574	11012	11019#
10263	020656	11058	11065#
10264	020740	11103	11110#
10265	021030	11150	11158#
10266	021110	11192	11200#
10267	021202	11240	11248#
1027	001616	3664#	
10270	021262	11282	11291#
10271	021354	11331	11339#
10272	021430	11373	11380#
10273	021516	11420	11427#
10274	021600	11461	11471#
10275	021656	11507	11515#
10276	021734	11552	11559#
10277	022010	11595	11601#
1030	001636	3693#	
10300	022070	11637	11644#
10301	022146	11681	11688#
10302	022224	11724	11731#
10303	022304	11767	11774#
10304	022362	11810	11817#
10305	022442	11853	11860#
10306	022516	11896	11902#
10307	022574	11938	11945#
1031	001656	3721#	
10310	022654	11981	11988#
10311	022730	12024	12031#
10312	023010	12068	12075#
10313	023070	12112	12119#
10314	023144	12156	12163#
10315	023224	12199	12206#
10316	023302	12242	12249#
10317	023362	12285	12292#
1032	001710	3752#	
10320	023440	12328	12335#
10321	023520	12371	12378#
10322	023600	12414	12421#
10323	023656	12456	12463#
10324	023732	12499	12506#
10325	024014	12545	12553#
10326	024074	12591	12598#
10327	024152	12635	12641#
1033	001752	3791#	

I0330	024246	12684	12691#
I0331	024326	12729	12736#
I0332	024406	12773	12780#
I0333	024470	12817	12824#
I0334	024562	12861	12871#
I0335	024644	12908	12915#
I0336	024722	12952	12958#
I0337	025002	12995	13002#
I034	002014	3828#	
I0340	025064	13039	13046#
I0341	025142	13083	13090#
I0342	025224	13127	13134#
I0343	025306	13171	13178#
I0344	025364	13215	13222#
I0345	025446	13260	13267#
I0346	025526	13304	13311#
I0347	025610	13347	13354#
I035	002046	3864#	
I0350	025670	13391	13398#
I0351	025752	13435	13442#
I0352	026034	13479	13486#
I0353	026114	13522	13529#
I0354	026176	13567	13574#
I0355	026256	13611	13618#
I0356	026336	13656	13663#
I0357	026416	13700	13707#
I036	002066	3893#	
I0360	026476	13744	13751#
I0361	026562	13788	13796#
I0362	026646	13832	13840#
I0363	026714	13869	13876#
I0364	026774	13911	13918#
I0365	027046	13947	13955#
I0366	027132	13990	13998#
I0367	027200	14027	14034#
I037	002106	3922#	
I0370	027260	14069	14076#
I0371	027330	14105	14113#
I0372	027414	14149	14157#
I0373	027474	14193	14200#
I0374	027564	14236	14246#
I0375	027646	14281	14289#
I0376	027726	14324	14331#
I0377	030010	14366	14374#
I040	002130	3952#	
I0400	030070	14409	14416#
I0401	030142	14445	14453#
I0402	030210	14482	14489#
I0403	030256	14519	14526#
I0404	030326	14556	14564#
I0405	030420	14602	14610#
I0406	030514	14648	14656#
I0407	030610	14694	14702#
I041	002156	3982#	
I0410	030704	14740	14748#
I0411	030776	14786	14794#

10412	031070	14832	14840#
10413	031164	14878	14886#
10414	031260	14925	14933#
10415	031334	14964	14972#
10416	031406	15003	15011#
10417	031460	15042	15050#
1042	002212	4020#	
10420	031524	15079	15086#
10421	031604	15123	15131#
10422	031670	15168	15176#
10423	031754	15214	15222#
10424	032032	15259	15267#
10425	032116	15305	15313#
10426	032204	15351	15359#
10427	032272	15397	15405#
1043	002234	4051#	
10430	032354	15445	15454#
10431	032434	15493	15502#
10432	032520	15541	15550#
10433	032604	15589	15598#
10434	032670	15637	15646#
10435	032752	15684	15692#
10436	033036	15729	15737#
10437	033122	15774	15781#
1044	002260	4081#	
10440	033206	15818	15826#
10441	033266	15862	15870#
10442	033354	15906	15914#
10443	033444	15950	15959#
10444	033552	15997	16010#
10445	033634	16047	16054#
10446	033716	16090	16097#
10447	034000	16133	16140#
1045	002306	4110#	
10450	034060	16177	16184#
10451	034142	16221	16228#
10452	034214	16259	16266#
10453	034270	16296	16306#
10454	034360	16345	16355#
10455	034450	16393	16403#
10456	034540	16442	16452#
10457	034630	16491	16501#
1046	002352	4134#	
10460	034732	16546	16556#
10461	035034	16601	16611#
10462	035136	16656	16666#
10463	035242	16711	16721#
10464	035346	16767	16777#
10465	035452	16822	16832#
10466	035556	16878	16888#
10467	035662	16933	16943#
10470	035764	16988	16998#
10471	036066	17043	17053#
10472	036170	17098	17108#
10473	036272	17154	17164#
10474	036376	17210	17220#

10475	036502	17265	17275#
10476	036606	17321	17331#
10477	036712	17377	17387#
10500	037006	17427	17437#
10501	037102	17477	17487#
10502	037176	17527	17537#
10503	037272	17577	17587#
10504	037400	17627	17640#
10505	037474	17680	17690#
10506	037570	17730	17740#
10507	037652	17779	17786#
10510	037714	17816	17823#
10511	037760	17853	17860#
10512	040024	17891	17898#
10513	040070	17928	17935#
10514	040134	17967	17974#
10515	040206	18003	18011#
10516	040270	18049	18058#
10517	040350	18096	18104#
10520	040416	18133	18140#
10521	040462	18170	18177#
10522	040540	18211	18218#
10523	040616	18254	18261#
10524	040666	18292	18300#
10525	040752	18337	18345#
10526	041036	18383	18391#
10527	041120	18428	18436#
10530	041204	18473	18481#
10531	041260	18512	18520#
10532	041332	18552	18560#
10533	041402	18590	18598#
10534	041454	18628	18636#
10535	041526	18666	18674#
10536	041600	18705	18713#
10537	041654	18742	18750#
10540	041732	18786	18794#
10541	042016	18830	18838#
10542	042074	18874	18882#
10543	042160	18919	18927#
10544	042256	18965	18976#
10545	042340	19013	19021#
10546	042422	19058	19066#
10547	042504	19103	19111#
10550	042570	19147	19155#
10551	042650	19192	19200#
10552	042736	19237	19245#
10553	043016	19282	19290#
10554	043104	19327	19335#
10555	043172	19372	19380#
10556	043260	19417	19425#
10557	043344	19463	19471#
10560	043430	19509	19516#
10561	043516	19553	19561#
10562	043576	19597	19605#
10563	043666	19641	19650#
10564	043746	19686	19694#

10565	044030	19730	19738#	
10566	044112	19774	19782#	
10567	044174	19818	19826#	
10570	044256	19863	19871#	
10571	044344	19908	19916#	
10572	044430	19952	19960#	
10573	044516	19997	20005#	
10574	044610	20042	20051#	
10575	044702	20089	20098#	
10576	044774	20135	20144#	
10577	045066	20181	20190#	
10600	045156	20228	20237#	
10601	045250	20275	20284#	
10602	045340	20322	20330#	
10603	045422	20371	20378#	
10604	045500	20409	20419#	
10605	045550	20450	20458#	
10606	045622	20489	20497#	
10607	045674	20528	20536#	
10610	045746	20567	20575#	
10611	046020	20606	20614#	
10612	046074	20645	20653#	
10613	046142	20683	20690#	
10614	046214	20720	20728#	
10615	046260	20757	20764#	
10616	046332	20795	20803#	
10617	046366	20832	20837#	
10620	046432	20873	20878#	
10621	046476	20913	20918#	
10622	046554	20960	20965#	
10623	046620	21001	21006#	
10624	046706	21054	21059#	
10625	046762	21101	21105#	
10626	047050	21152	21157#	
10627	047114	21193	21198#	
10630	047202	21244	21249#	
10631	047256	21290	21295#	
10632	047332	21336	21341#	
10633	047406	21382	21387#	
10634	047472	21433	21438#	
10635	047560	21482	21487#	
10636	047636	21521	21528#	
10637	047714	21567	21572#	
10640	047762	21605	21610#	
10641	050032	21644	21650#	
10642	050102	21685	21691#	
10643	050162	21731	21737#	
10644	050252	21776	21785#	
10645	050332	21825	21831#	
10646	050404	21866	21872#	
10647	050472	21915	21918	21925#
10650	050526	21949	21953	21959#
10651	050574	21994	21999#	
10652	050632	22031	22036#	
10653	050670	22069	22074#	
10654	050726	22106	22111#	

I0655	051000	22143	22151#	
I0656	051106	22201	22209#	
I0657	051220	22259	22268#	
I0660	051324	22313	22322#	
I0661	051420	22367	22374#	
I0662	051556	22439	22446#	
I0663	051722	22497	22506#	
I0664	052002	22524	22531#	
I0665	052046	22546	22553#	
I0666	052142	22567	22579#	
I0667	052256	22602	22615#	
I0670	052362	22653	22660#	
I0671	052476	22704	22711#	
I0672	052642	22755	22774#	22783
I0673	053046	22832	22851#	22860
I0674	053240	22894	22906#	
I0675	053362	22928	22940#	
I0676	053504	22962	22974#	
I0677	053626	22996	23008#	
I0700	053750	23031	23043#	
I0701	054072	23065	23077#	
I0702	054206	23099	23109#	
I0703	054322	23131	23141#	
I0704	054450	23163	23176#	
I0705	054564	23198	23208#	
I0706	054706	23229	23241#	
I0707	055032	23264	23276#	
I0710	055150	23299	23309#	
I0711	055266	23332	23342#	
I0712	055404	23365	23375#	
I0713	055522	23398	23408#	
I0714	055672	23437	23457#	
I0715	056106	23486	23518#	
I0716	056272	23573	23579#	
I0717	056356	23610	23619#	
I0720	056476	23665	23673#	
I0721	056600	23703	23710#	
I0722	056660	23728	23735#	
I0723	056740	23753	23760#	
I0724	057022	23792	23800#	
I0732	060022	24086	24090#	
I0733	060120	24115	24122#	
I0734	060220	24147	24155#	
I0735	060314	24193	24201#	
I0736	060372	24231	24239#	
I0737	060450	24268	24276#	
I0740	060530	24305	24310	24314#
I0741	060622	24356	24363#	
I0742	060730	24410	24417#	
I0743	061046	24463	24472#	
I0744	061172	24524*	24527#	
I0745	061262	24568*	24571#	
I0746	061346	24612*	24615#	
I0747	061434	24656*	24659#	
I0750	061522	24700*	24703#	
I0751	061610	24743*	24746#	

I0752	061666	24785*	24788#	
I0753	061754	24829*	24832#	
I0754	062032	24871*	24874#	
I0755	062142	24922	24926	24930#
I0756	062206	24957	24962	24966#
I0757	062252	24995	25000	25004#
I0760	062312	25031	25035	25039#
I0761	062352	25067	25071	25075#
I0762	062416	25104	25109	25113#
I0763	062462	25142	25147	25151#
I0764	062522	25178	25182	25186#
I0765	062562	25213	25217	25221#
I0766	062622	25249	25253	25257#
I0767	062666	25285	25290	25294#
I0770	062726	25321	25325	25329#
I0771	062772	25358	25363	25367#
I0772	063032	25395	25399	25403#
I0773	063072	25430	25434	25438#
I0774	063132	25465	25469	25473#
I0775	063176	25502	25507	25511#
I0776	063242	25540	25545	25549#
I0777	063306	25577	25582	25586#
I1000	063346	25613	25617	25621#
I1001	063406	25648	25652	25656#
I1002	063444	25683	25687	25690#
I1003	063502	25717	25721	25724#
I1004	063546	25751	25756	25760#
I1005	063626	25787	25795	25800#
I1006	063700	25827	25832	25839#
I1007	063744	25868	25872	25877#
I1010	064020	25947#		
I1011	064076	26017#		
I1012	064154	26090#		
I1013	064232	26160#		
I1014	064300	26219#		
I10146	013710	8093	8097#	
I1015	064344	26271#		
I10230	016140	9777	9785#	
I10236	016626	10067	10075#	
I10725	057076	23821	23824#	
I10726	057212	23864	23867#	
I10727	057326	23908	23911#	
I10730	057500	23965	23969#	
I10731	057652	24027	24031#	
I20146	013716	8097	8102#	
I20230	016206	9802	9807#	
I20236	016674	10091	10097#	
I20725	057116	23830	23833#	
I20726	057232	23873	23876#	
I20727	057346	23917	23920#	
I20730	057520	23975	23979#	
I20731	057672	24037	24041#	
I30146	013724	8102	8107#	
I30725	057134	23839	23842#	
I30726	057250	23881	23884#	
I30727	057364	23926	23929#	

00221	015606	9522	9527#
00222	015632	9554	9559#
00223	015656	9586	9591#
00224	015734	9631	9636#
00225	015772	9666	9671#
00226	016050	9712	9717#
00227	016110	9748	9753#
00230	016250	9824	9829#
00231	016306	9859	9864#
00232	016362	9904	9909#
00233	016422	9940	9945#
00234	016514	9990	9995#
00235	016574	10037	10042#
00236	016736	10114	10119#
00237	017016	10161	10166#
00240	017072	10207	10212#
00241	017152	10253	10258#
00242	017232	10299	10304#
00243	017310	10346	10351#
00244	017364	10392	10397#
00245	017444	10438	10443#
00246	017524	10484	10489#
00247	017602	10530	10535#
00250	017662	10576	10581#
00251	017740	10621	10626#
00252	020020	10666	10671#
00253	020076	10711	10716#
00254	020156	10756	10761#
00255	020236	10801	10806#
00256	020320	10846	10852#
00257	020400	10892	10898#
00260	020462	10938	10944#
00261	020542	10984	10990#
00262	020624	11030	11036#
00263	020706	11076	11082#
00264	020774	11124	11130#
00265	021052	11166	11172#
00266	021144	11214	11220#
00267	021224	11256	11262#
00270	021316	11305	11311#
00271	021376	11347	11353#
00272	021464	11394	11400#
00273	021540	11435	11441#
00274	021626	11482	11487#
00275	021704	11526	11531#
00276	021762	11569	11574#
00277	022036	11611	11616#
00300	022116	11654	11659#
00301	022174	11698	11703#
00302	022252	11741	11746#
00303	022332	11784	11789#
00304	022410	11827	11832#
00305	022470	11870	11875#
00306	022544	11912	11917#
00307	022622	11955	11960#
00310	022702	11998	12003#

00311	022756	12041	12046#
00312	023036	12085	12090#
00313	023116	12129	12134#
00314	023172	12173	12178#
00315	023252	12216	12221#
00316	023330	12259	12265#
00317	023410	12302	12307#
00320	023466	12345	12350#
00321	023546	12388	12393#
00322	023626	12431	12436#
00323	023704	12473	12478#
00324	023762	12517	12523#
00325	024044	12564	12570#
00326	024124	12608	12614#
00327	024214	12657	12663#
00330	024276	12701	12707#
00331	024356	12746	12752#
00332	024436	12790	12796#
00333	024520	12834	12840#
00334	024612	12881	12887#
00335	024674	12925	12931#
00336	024752	12968	12974#
00337	025032	13012	13018#
00340	025114	13056	13062#
00341	025172	13100	13106#
00342	025254	13144	13150#
00343	025336	13188	13194#
00344	025414	13232	13238#
00345	025476	13277	13283#
00346	025556	13321	13327#
00347	025640	13364	13370#
00350	025720	13408	13414#
00351	026002	13452	13458#
00352	026064	13496	13502#
00353	026144	13539	13545#
00354	026224	13585	13590#
00355	026304	13629	13634#
00356	026364	13674	13679#
00357	026444	13718	13723#
00360	026524	13762	13767#
00361	026610	13805	13811#
00362	026662	13843	13849#
00363	026742	13885	13891#
00364	027010	13921	13927#
00365	027074	13964	13970#
00366	027146	14001	14007#
00367	027226	14043	14049#
00370	027274	14079	14085#
00371	027356	14124	14129#
00372	027442	14168	14173#
00373	027520	14209	14214#
00374	027610	14255	14260#
00375	027674	14298	14303#
00376	027752	14340	14345#
00377	030036	14383	14388#
00400	030104	14419	14424#

00401	030156	14456	14461#
00402	030224	14492	14498#
00403	030272	14529	14535#
00404	030364	14576	14581#
00405	030456	14622	14627#
00406	030552	14668	14673#
00407	030646	14714	14719#
00410	030740	14760	14765#
00411	031032	14806	14811#
00412	031126	14852	14857#
00413	031222	14898	14903#
00414	031276	14936	14942#
00415	031352	14975	14981#
00416	031424	15014	15020#
00417	031476	15053	15059#
00420	031552	15097	15102#
00421	031632	15142	15147#
00422	031716	15187	15192#
00423	032002	15233	15238#
00424	032062	15278	15284#
00425	032146	15324	15330#
00426	032234	15370	15376#
00427	032322	15416	15422#
00430	032402	15465	15470#
00431	032462	15513	15518#
00432	032546	15561	15566#
00433	032632	15609	15614#
00434	032716	15657	15662#
00435	033002	15703	15709#
00436	033066	15748	15754#
00437	033152	15792	15798#
00440	033234	15837	15842#
00441	033314	15881	15886#
00442	033402	15925	15930#
00443	033474	15970	15976#
00444	033602	16021	16027#
00445	033662	16064	16069#
00446	033744	16107	16112#
00447	034024	16150	16155#
00450	034106	16194	16199#
00451	034160	16231	16237#
00452	034232	16269	16275#
00453	034322	16317	16324#
00454	034412	16366	16373#
00455	034502	16414	16421#
00456	034572	16463	16470#
00457	034674	16518	16525#
00460	034776	16573	16580#
00461	035100	16628	16635#
00462	035202	16683	16690#
00463	035306	16738	16745#
00464	035412	16794	16801#
00465	035516	16849	16856#
00466	035622	16905	16912#
00467	035724	16960	16967#
00470	036026	17015	17022#

00471	036130	17070	17077#
00472	036232	17125	17132#
00473	036336	17181	17188#
00474	036442	17237	17244#
00475	036546	17292	17299#
00476	036652	17348	17355#
00477	036746	17398	17405#
00500	037042	17448	17455#
00501	037136	17498	17505#
00502	037232	17548	17555#
00503	037326	17598	17605#
00504	037434	17651	17658#
00505	037530	17701	17708#
00506	037624	17751	17758#
00507	037666	17789	17795#
00510	037730	17826	17832#
00511	037774	17863	17869#
00512	040040	17901	17907#
00513	040104	17938	17944#
00514	040152	17977	17983#
00515	040234	18022	18027#
00516	040316	18069	18074#
00517	040362	18107	18112#
00520	040430	18143	18148#
00521	040506	18186	18191#
00522	040564	18227	18232#
00523	040632	18264	18270#
00524	040714	18310	18315#
00525	041000	18354	18361#
00526	041062	18400	18406#
00527	041146	18445	18451#
00530	041222	18484	18490#
00531	041276	18523	18529#
00532	041346	18563	18569#
00533	041416	18601	18607#
00534	041470	18639	18645#
00535	041542	18677	18683#
00536	041616	18716	18722#
00537	041702	18761	18766#
00540	041760	18805	18810#
00541	042044	18849	18854#
00542	042122	18893	18898#
00543	042210	18938	18944#
00544	042304	18987	18992#
00545	042366	19032	19037#
00546	042450	19077	19082#
00547	042532	19122	19127#
00550	042620	19166	19172#
00551	042700	19211	19217#
00552	042766	19256	19262#
00553	043046	19301	19307#
00554	043134	19346	19352#
00555	043222	19391	19396#
00556	043310	19436	19442#
00557	043374	19482	19488#
00560	043460	19527	19533#

00561	043544	19572	19577#		
00562	043624	19616	19621#		
00563	043714	19661	19666#		
00564	043774	19705	19710#		
00565	044056	19749	19754#		
00566	044140	19793	19797#		
00567	044222	19837	19842#		
00570	044304	19882	19887#		
00571	044372	19927	19932#		
00572	044460	19971	19977#		
00573	044546	20016	20022#		
00574	044640	20062	20068#		
00575	044732	20109	20115#		
00576	045024	20155	20161#		
00577	045116	20201	20207#		
00600	045206	20248	20254#		
00601	045300	20295	20301#		
00602	045370	20341	20347#		
00603	045434	20381	20386#		
00604	045514	20422	20428#		
00605	045564	20461	20467#		
00606	045636	20500	20506#		
00607	045710	20539	20545#		
00610	045762	20578	20584#		
00611	046036	20617	20623#		
00612	046112	20656	20662#		
00613	046156	20693	20699#		
00614	046230	20731	20737#		
00615	046274	20767	20773#		
00616	046346	20806	20812#		
00617	046412	20841	20846	20851#	
00620	046456	20882	20887	20892#	
00621	046534	20922	20933	20938#	
00622	046600	20969	20974	20979#	
00623	046666	21010	21021	21025	21032#
00624	046742	21064	21069	21073	21079#
00625	047030	21109	21114	21125	21130#
00626	047074	21161	21166	21171#	
00627	047162	21202	21213	21217	21222#
00630	047236	21253	21258	21262	21268#
00631	047312	21299	21304	21307	21314#
00632	047366	21345	21350	21354	21360#
00633	047452	21391	21395	21400	21404
00634	047536	21442	21446	21451	21411#
00635	047604	21494	21501#		21462#
00636	047672	21546#			
00637	047740	21586#			
00640	050006	21623#			
00641	050056	21657	21665#		
00642	050136	21698	21711#		
00643	050214	21748	21756#		
00644	050306	21792	21805#		
00645	050360	21838	21846#		
00646	050442	21880	21893#		
00647	050474	21922	21927#		
00650	050554	21966	21970	21973#	

00651	050612	22004	22009#		
00652	050650	22042	22047#		
00653	050706	22079	22084#		
00654	050744	22116	22121#		
00655	051052	22172	22180#		
00656	051160	22230	22238#		
00657	051262	22284	22292#		
00660	051366	22338	22346#		
00661	051524	22380	22388	22407	22419#
00662	051662	22452	22460	22479	22491#
00663	051750	22499	22517#		
00664	052014	22526	22533	22539#	
00665	052062	22548	22555	22561#	
00666	052200	22569	22594#		
00667	052322	22604	22630#		
00670	052436	22681#			
00671	052552	22732#			
00672	052756	22808#			
00673	053162	22885#			
00674	053304	22896	22921#		
00675	053426	22930	22955#		
00676	053550	22964	22989#		
00677	053672	22998	23024#		
00700	054014	23033	23058#		
00701	054136	23067	23092#		
00702	054252	23124#			
00703	054366	23156#			
00704	054514	23191#			
00705	054630	23223#			
00706	054754	23231	23257#		
00707	055100	23266	23292#		
00710	055216	23325#			
00711	055334	23358#			
00712	055452	23391#			
00713	055570	23424#			
00714	055736	23439	23475#		
00715	056244	23488	23562#		
00716	056310	23587#			
00717	056436	23644#			
00720	056544	23697#			
00721	056624	23722#			
00722	056704	23747#			
00723	056764	23772#			
00724	057046	23810#			
00725	057162	23848	23853#		
00726	057276	23890	23895#		
00727	057446	23935	23954#		
00730	057620	24007	24016#		
00731	057772	24068	24077#		
00732	060056	24107#			
00733	060156	24140#			
00734	060256	24173#			
00735	060334	24210#			
00736	060412	24248#			
00737	060470	24284#			
00740	060570	24331#			

R0051	005646	5595#	5605			
R0052	005716	5630#	5639			
R0053	005760	5663#	5674	5680		
R0054	006036	5705#	5716	5722		
R0055	006122	5747#	5757			
R0056	006166	5781#	5792	5798		
R0057	006250	5821#	5831	5837		
R0060	006332	5862#	5871			
R0061	006406	5897#	5906			
R0062	006454	5932#	5943	5949		
R0063	006536	5974#	5983	5990		
R0064	006612	6014#	6023			
R0065	006654	6048#	6056			
R0066	006716	6081#	6089			
R0067	006760	6113#	6123			
R0070	007026	6147#	6157			
R0071	007076	6181#	6191			
R0072	007140	6216#	6230			
R0073	007222	6256#	6264			
R0074	007262	6289#	6297	6304		
R0075	007336	6328#	6338			
R0076	007404	6363#	6374			
R0077	007472	6404#	6415	6421	6428	
R0100	007560	6452#	6461	6467	6473	6481
R0101	007662	6506#	6515	6521	6527	6535
R0102	007766	6560#	6569	6575	6581	6589
R0103	010072	6614#	6623	6629	6635	6643
R0104	010200	6670#	6681	6687		
R0105	010270	6713#	6724	6730		
R0106	010360	6756#	6767	6773		
R0107	010450	6798#	6809	6815		
R0110	010540	6840#	6851	6857		
R0111	010626	6883#	6894	6900		
R0112	010714	6925#	6936	6942		
R0113	011004	6968#	6979	6985		
R0114	011070	7010#	7020	7026		
R0115	011146	7051#	7061	7067		
R0116	011222	7092#	7105	7112	7118	
R0117	011322	7147#	7160	7167	7173	
R0120	011430	7200#	7210			
R0121	011502	7234#	7244			
R0122	011554	7268#	7278			
R0123	011626	7303#	7313			
R0124	011666	7336#	7343	7349		
R0125	011730	7372#	7378	7385		
R0126	011772	7408#	7417	7425	7431	
R0127	012066	7454#	7468			
R0130	012146	7492#	7502	7510	7517	7524
R0131	012276	7549#	7561			
R0132	012352	7586#	7596	7604	7611	7618
R0133	012502	7643#	7656			
R0134	012576	7686#	7698			
R0135	012650	7722#	7734			
R0136	012742	7760#				
R0137	013036	7801#				
R0140	013126	7841#	7854			

R0141	013214	7880#	7892			
R0142	013264	7916#	7925			
R0143	013324	7949#	7958			
R0145	013654	8060#	8069			
R0146	013706	8095#	8100	8105	8110	8115
R0147	013752	8141#	8147			
R0150	013776	8173#	8176			
R0151	014020	8200#	8206			
R0152	014044	8232#	8237			
R0153	014066	8263#	8269			
R0154	014112	8297#	8302			
R0155	014134	8328#	8334			
R0156	014160	8361#	8366			
R0157	014202	8392#	8403			
R0160	014242	8429#	8434	8443		
R0161	014306	8469#	8480			
R0162	014344	8506#	8511	8520		
R0163	014406	8546#	8551			
R0164	014430	8577#	8584			
R0165	014456	8608#	8615			
R0166	014504	8641#	8647			
R0167	014530	8673#	8679			
R0170	014554	8705#	8711			
R0171	014600	8737#	8743			
R0172	014624	8769#	8776			
R0173	014652	8802#	8809			
R0174	014712	8838#	8845			
R0175	014740	8871#	8878			
R0176	014766	8902#	8907			
R0177	015010	8934#	8941			
R0200	015036	8967#	8974			
R0201	015064	9000#	9007			
R0202	015112	9032#	9038			
R0203	015136	9065#	9071			
R0204	015162	9097#	9103			
R0205	015206	9129#	9135			
R0206	015232	9162#	9168			
R0207	015256	9194#	9201			
R0210	015304	9227#	9232			
R0211	015326	9258#	9265			
R0212	015354	9291#	9298			
R0213	015402	9323#	9331			
R0214	015430	9357#	9363			
R0215	015454	9390#	9396			
R0216	015500	9423#	9429			
R0217	015524	9455#	9461			
R0220	015550	9487#	9492			
R0221	015572	9518#	9525			
R0222	015620	9551#	9557			
R0223	015644	9583#	9589			
R0224	015676	9616#	9628	9634		
R0225	015754	9662#	9669			
R0226	016014	9697#	9708	9715		
R0227	016072	9743#	9751			
R0231	016270	9855#	9862			
R0232	016330	9890#	9901	9907		

R0233	016404	9935#	9943	
R0234	016456	9975#	9987	9993
R0235	016536	10022#	10034	10040
R0237	016760	10146#	10158	10164
R0240	017036	10192#	10204	10210
R0241	017114	10238#	10250	10256
R0242	017174	10284#	10296	10302
R0243	017254	10331#	10343	10349
R0244	017330	10377#	10389	10395
R0245	017406	10423#	10435	10441
R0246	017466	10469#	10481	10487
R0247	017546	10515#	10527	10533
R0250	017624	10561#	10573	10579
R0251	017702	10606#	10618	10624
R0252	017762	10651#	10663	10669
R0253	020040	10696#	10708	10714
R0254	020120	10741#	10753	10759
R0255	020200	10786#	10798	10804
R0256	020260	10831#	10843	10850
R0257	020340	10877#	10889	10896
R0260	020422	10923#	10935	10942
R0261	020502	10969#	10981	10988
R0262	020564	11015#	11027	11034
R0263	020646	11061#	11073	11080
R0264	020730	11106#	11115	11121 11128
R0265	021016	11153#	11163	11170
R0266	021100	11196#	11205	11211 11218
R0267	021172	11244#	11253	11260
R0270	021252	11287#	11296	11302 11309
R0271	021344	11335#	11344	11351
R0272	021420	11376#	11385	11391 11398
R0273	021506	11423#	11432	11439
R0274	021572	11467#	11479	11485
R0275	021650	11511#	11523	11529
R0276	021724	11555#	11567	11572
R0277	022002	11598#	11609	11614
R0300	022060	11640#	11652	11657
R0301	022136	11684#	11696	11701
R0302	022214	11727#	11739	11744
R0303	022274	11770#	11782	11787
R0304	022352	11813#	11825	11830
R0305	022432	11856#	11868	11873
R0306	022512	11899#	11910	11915
R0307	022564	11941#	11953	11958
R0310	022644	11984#	11996	12001
R0311	022722	12027#	12039	12044
R0312	023000	12071#	12083	12088
R0313	023060	12115#	12127	12132
R0314	023136	12159#	12171	12176
R0315	023214	12202#	12214	12219
R0316	023272	12245#	12257	12263
R0317	023352	12288#	12300	12305
R0320	023430	12331#	12343	12348
R0321	023510	12374#	12386	12391
R0322	023570	12417#	12429	12434
R0323	023650	12459#	12471	12476

R0324	023724	12502#	12514	12521	
R0325	024004	12549#	12561	12568	
R0326	024064	12594#	12606	12612	
R0327	024144	12638#	12649	12654	12661
R0330	024236	12687#	12699	12705	
R0331	024316	12732#	12744	12750	
R0332	024376	12776#	12788	12794	
R0333	024460	12820#	12832	12838	
R0334	024552	12867#	12879	12885	
R0335	024634	12911#	12923	12929	
R0336	024716	12955#	12966	12972	
R0337	024772	12998#	13010	13016	
R0340	025054	13042#	13054	13060	
R0341	025134	13086#	13098	13104	
R0342	025214	13130#	13142	13148	
R0343	025276	13174#	13186	13192	
R0344	025356	13218#	13230	13236	
R0345	025436	13263#	13275	13281	
R0346	025516	13307#	13319	13325	
R0347	025600	13350#	13362	13368	
R0350	025660	13394#	13406	13412	
R0351	025742	13438#	13450	13456	
R0352	026024	13482#	13494	13500	
R0353	026106	13525#	13537	13543	
R0354	026166	13570#	13582	13588	
R0355	026246	13614#	13626	13632	
R0356	026326	13659#	13671	13677	
R0357	026406	13703#	13715	13721	
R0360	026466	13747#	13759	13765	
R0361	026552	13792#	13802	13809	
R0362	026636	13836#	13847		
R0363	026704	13872#	13882	13889	
R0364	026764	13914#	13925		
R0365	027036	13951#	13961	13968	
R0366	027122	13994#	14005		
R0367	027170	14030#	14040	14047	
R0370	027250	14072#	14083		
R0371	027320	14109#	14121	14127	
R0372	027404	14153#	14165	14171	
R0373	027464	14196#	14206	14212	
R0374	027554	14242#	14252	14258	
R0375	027632	14284#	14295	14301	
R0376	027716	14327#	14337	14343	
R0377	027774	14369#	14380	14386	
R0400	030060	14412#	14422		
R0401	030126	14448#	14459		
R0402	030200	14485#	14496		
R0403	030246	14522#	14533		
R0404	030314	14559#	14572	14579	
R0405	030406	14605#	14618	14625	
R0406	030500	14651#	14664	14671	
R0407	030574	14697#	14710	14717	
R0410	030670	14743#	14756	14763	
R0411	030762	14789#	14802	14809	
R0412	031054	14835#	14848	14855	
R0413	031150	14881#	14894	14901	

RC414	031244	14928#	14940		
R0415	031320	14967#	14979		
R0416	031374	15006#	15018		
R0417	031446	15045#	15057		
R0420	031516	15082#	15094	15100	
R0421	031576	15127#	15139	15145	
R0422	031660	15172#	15184	15190	
R0423	031744	15218#	15230	15236	
R0424	032024	15263#	15275	15282	
R0425	032106	15309#	15321	15328	
R0426	032174	15355#	15367	15374	
R0427	032262	15401#	15413	15420	
R0430	032342	15449#	15462	15468	
R0431	032422	15497#	15510	15516	
R0432	032504	15545#	15558	15564	
R0433	032570	15593#	15606	15612	
R0434	032654	15641#	15654	15660	
R0435	032744	15688#	15700	15707	
R0436	033026	15733#	15745	15752	
R0437	033114	15778#	15789	15796	
R0440	033200	15822#	15834	15840	
R0441	033260	15866#	15878	15884	
R0442	033342	15910#	15922	15928	
R0443	033430	15954#	15967	15974	
R0444	033534	16004#	16018	16025	
R0445	033624	16050#	16061	16067	
R0446	033704	16093#	16104	16110	
R0447	033766	16136#	16147	16153	
R0450	034046	16180#	16191	16197	
R0451	034130	16224#	16235		
R0452	034202	16262#	16273		
R0453	034260	16301#	16314	16322	
R0454	034350	16350#	16363	16371	
R0455	034436	16397#	16411	16419	
R0456	034526	16446#	16460	16468	
R0457	034620	16496#	16509	16515	16523
R0460	034722	16551#	16564	16570	16578
R0461	035022	16605#	16619	16625	16633
R0462	035124	16660#	16674	16680	16688
R0463	035230	16716#	16729	16735	16743
R0464	035334	16772#	16785	16791	16799
R0465	035436	16826#	16840	16846	16854
R0466	035542	16882#	16896	16902	16910
R0467	035650	16938#	16951	16957	16965
R0470	035752	16993#	17006	17012	17020
R0471	036052	17047#	17061	17067	17075
R0472	036154	17102#	17116	17122	17130
R0473	036260	17159#	17172	17178	17186
R0474	036364	17215#	17228	17234	17242
R0475	036466	17269#	17283	17289	17297
R0476	036572	17325#	17339	17345	17353
R0477	036700	17382#	17395	17403	
R0500	036774	17432#	17445	17453	
R0501	037066	17481#	17495	17503	
R0502	037162	17531#	17545	17553	
R0503	037260	17582#	17595	17603	

R0504	037366	17635#	17648	17656
R0505	037460	17684#	17698	17706
R0506	037554	17734#	17748	17756
R0507	037644	17782#	17793	
R0510	037706	17819#	17830	
R0511	037750	17856#	17867	
R0512	040014	17894#	17905	
R0513	040060	17931#	17942	
R0514	040130	17971#	17981	
R0515	040200	18007#	18019	18025
R0516	040260	18053#	18066	18072
R0517	040342	18100#	18110	
R0520	040410	18137#	18146	
R0521	040452	18173#	18183	18189
R0522	040526	18214#	18224	18230
R0523	040612	18258#	18268	
R0524	040660	18296#	18307	18313
R0525	040742	18341#	18351	18358
R0526	041026	18387#	18397	18404
R0527	041110	18432#	18442	18449
R0530	041174	18477#	18488	
R0531	041250	18516#	18527	
R0532	041324	18556#	18567	
R0533	041374	18594#	18605	
R0534	041444	18632#	18643	
R0535	041516	18670#	18681	
R0536	041570	18709#	18720	
R0537	041644	18746#	18758	18764
R0540	041724	18790#	18802	18808
R0541	042006	18834#	18846	18852
R0542	042066	18878#	18890	18896
R0543	042150	18923#	18935	18942
R0544	042250	18972#	18984	18990
R0545	042330	19017#	19029	19035
R0546	042414	19062#	19074	19080
R0547	042476	19107#	19119	19125
R0550	042560	19151#	19163	19170
R0551	042642	19196#	19208	19215
R0552	042726	19241#	19253	19260
R0553	043010	19286#	19298	19305
R0554	043074	19331#	19343	19350
R0555	043162	19376#	19388	19395
R0556	043250	19421#	19433	19440
R0557	043334	19467#	19479	19486
R0560	043422	19513#	19524	19531
R0561	043506	19557#	19569	19575
R0562	043570	19601#	19613	19619
R0563	043652	19645#	19658	19664
R0564	043740	19690#	19702	19708
R0565	044022	19734#	19746	19752
R0566	044104	19778#	19790	19796
R0567	044166	19822#	19834	19840
R0570	044250	19867#	19879	19885
R0571	044332	19912#	19924	19930
R0572	044420	19956#	19968	19975
R0573	044506	20001#	20013	20020

R0574	044574	20046#	20059	20066				
R0575	044666	20093#	20106	20113				
R0576	044760	20139#	20152	20159				
R0577	045052	20185#	20198	20205				
R0600	045144	20232#	20245	20252				
R0601	045234	20279#	20292	20299				
R0602	045326	20326#	20338	20345				
R0603	045412	20374#	20384					
R0604	045470	20415#	20426					
R0605	045536	20453#	20465					
R0606	045606	20492#	20504					
R0607	045660	20531#	20543					
R0610	045732	20570#	20582					
R0611	046004	20609#	20621					
R0612	046060	20648#	20660					
R0613	046134	20686#	20697					
R0614	046204	20724#	20735					
R0615	046252	20760#	20771					
R0616	046316	20798#	20810					
R0617	046360	20834#	20840	20849				
R0620	046424	20875#	20881	20890				
R0621	046470	20915#	20921	20930	20936			
R0622	046546	20962#	20968	20977				
R0623	046612	21003#	21009	21018	21024	21028		
R0624	046700	21056#	21062	21072	21077			
R0625	046754	21102#	21108	21113	21122	21128		
R0626	047042	21154#	21160	21169				
R0627	047106	21195#	21201	21210	21216	21220		
R0630	047174	21246#	21252	21261	21266			
R0631	047250	21292#	21298	21307	21311			
R0632	047324	21338#	21344	21353	21357			
R0633	047400	21384#	21390	21394	21403	21409		
R0634	047464	21435#	21441	21445	21454	21460		
R0635	047552	21484#	21490	21498				
R0636	047620	21523#	21531	21537	21543			
R0637	047706	21569#	21575	21583				
R0640	047754	21607#	21613	21620				
R0641	050022	21646#	21653	21661				
R0642	050072	21687#	21694	21702	21707			
R0643	050152	21733#	21740	21744	21752			
R0644	050242	21781#	21788	21796	21801			
R0645	050322	21827#	21834	21843				
R0646	050374	21868#	21876	21884	21889			
R0647	050454	21917#	21921					
R0650	050506	21951#	21969	21972				
R0651	050566	21996#	22007					
R0652	050624	22033#	22045					
R0653	050662	22071#	22082					
R0654	050720	22108#	22119					
R0655	050760	22145#	22154	22163	22169	22177		
R0656	051066	22203#	22212	22221	22227	22235		
R0657	051174	22261#	22271	22281	22289			
R0660	051276	22315#	22325	22335	22343			
R0661	051406	22370#	22378	22386	22398	22405	22415	
R0662	051544	22442#	22450	22458	22470	22477	22487	
R0663	051710	22502#	22512					

R0664	052000	22529#	22537				
R0665	052044	22551#	22559				
R0666	052114	22573#	22586				
R0667	052226	22607#	22622				
R0670	052346	22656#	22668	22676			
R0671	052462	22707#	22719	22727			
R0672	052572	22758#	22779	22787	22796	22800	
R0673	052776	22835#	22856	22864	22873	22877	
R0674	053224	22901#	22913				
R0675	053346	22935#	22947				
R0676	053470	22969#	22981				
R0677	053612	23003#	23015				
R0700	053734	23038#	23050				
R0701	054056	23072#	23084				
R0702	054172	23104#	23116				
R0703	054306	23136#	23148				
R0704	054434	23171#	23183				
R0705	054550	23203#	23215				
R0706	054672	23236#	23249				
R0707	055016	23271#	23284				
R0710	055134	23304#	23317				
R0711	055252	23337#	23350				
R0712	055370	23370#	23383				
R0713	055506	23403#	23416				
R0714	055632	23444#	23464	23469			
R0715	056020	23496#	23527	23534	23541	23547	23553
R0716	056266	23576#	23582				
R0717	056324	23612#	23624	23632	23639		
R0720	056456	23668#	23678	23686	23693		
R0721	056564	23706#	23717				
R0722	056644	23731#	23742				
R0723	056724	23756#	23767				
R0724	057000	23794#	23804				
R0732	060014	24087#	24094	24103			
R0733	060112	24119#	24126	24135			
R0734	060212	24152#	24159	24168			
R0735	060272	24195#	24205				
R0736	060350	24233#	24243				
R0737	060426	24270#	24280				
R0740	060504	24307#	24317	24325			
R0741	060604	24358#	24371	24382			
R0742	060710	24412#	24425	24436			
R0743	061026	24467#	24480	24491			
R0744	061162	24523#	24531	24535			
R0745	061252	24567#	24575	24579			
R0746	061336	24611#	24619	24623			
R0747	061424	24655#	24663	24667			
R0750	061512	24699#	24707	24711			
R0751	061600	24742#	24750	24754			
R0752	061656	24784#	24792	24796			
R0753	061744	24828#	24836	24840			
R0754	062022	24870#	24878	24882			
R0755	062124	24924#	24933				
R0756	062164	24959#	24969				
R0757	062230	24997#	25007				
R0760	062274	25033#	25042				

CROSS REFERENCE TABLE -- USER SYMBOLS

9753	9829	9864	9909	9945	9995	10042	10119	10166	10212	10258	10304	10351	
10397	10443	10489	10535	10581	10626	10671	10716	10761	10806	10852	10898	10944	
10990	11036	11082	11130	11172	11220	11262	11311	11353	11400	11441	11487	11531	
11574	11616	11659	11703	11746	11789	11832	11875	11917	11960	12003	12046	12090	
12134	12178	12221	12265	12307	12350	12393	12436	12478	12523	12570	12614	12663	
12707	12752	12796	12840	12887	12931	12974	13018	13062	13106	13150	13194	13238	
13283	13327	13370	13414	13458	13502	13545	13590	13634	13679	13723	13767	13811	
13849	13891	13927	13970	14007	14049	14085	14129	14173	14214	14260	14303	14345	
14388	14424	14461	14498	14535	14581	14627	14673	14719	14765	14811	14857	14903	
14942	14981	15020	15059	15102	15147	15192	15238	15284	15330	15376	15422	15470	
15518	15566	15614	15662	15709	15754	15798	15842	15886	15930	15976	16027	16069	
16112	16155	16199	16237	16275	16324	16373	16421	16470	16525	16580	16635	16690	
16745	16801	16856	16912	16967	17022	17077	17132	17188	17244	17299	17355	17405	
17455	17505	17555	17605	17658	17708	17758	17795	17832	17869	17907	17944	17983	
18027	18074	18112	18148	18191	18232	18270	18315	18361	18406	18451	18490	18529	
18569	18607	18645	18683	18722	18766	18810	18854	18898	18944	18992	19037	19082	
19127	19172	19217	19262	19307	19352	19396	19442	19488	19533	19577	19621	19666	
19710	19754	19797	19842	19887	19932	19977	20022	20068	20115	20161	20207	20254	
20301	20347	20386	20428	20467	20506	20545	20584	20623	20662	20699	20737	20773	
20812	20851	20892	20938	20979	21032	21079	21130	21171	21222	21268	21314	21360	
21411	21462	21501	21546	21586	21623	21665	21711	21756	21805	21846	21893	21927	
21973	22009	22047	22084	22121	22180	22238	22292	22346	22419	22491	22517	22539	
22561	22594	22630	22681	22732	22808	22885	22921	22955	22989	23024	23058	23092	
23124	23156	23191	23223	23257	23292	23325	23358	23391	23424	23475	23562	23587	
23644	23697	23722	23747	23772	23810	23853	23895	23954	24016	24077	24107	24140	
24173	24210	24248	24284	24331	24386	24440	24495	24538	24582	24626	24670	24714	
24757	24799	24843	24885	24935	24971	25009	25044	25080	25118	25156	25191	25226	
25262	25299	25334	25372	25408	25443	25478	25516	25554	25591	25626	25661	25695	
25729	25765	25805	25844	25891	25964	26033	26106	26176	26235	26287			
SCOPEA 065260	7551	26589#											
SCOPEB 065266	8021	26593#											
SELTST 066700	26600*	26601*	26602	26997#									
SOB1 050532	21962#	22111											
SOB2 050536	21960	21965#											
SOB3 050724	21962	22109#											
SR = 177570	2911#	8029	26297	26368	26370	26417	26477	26549	26594	26598	26600	26605	26674
STAB1 070140	26726	26728	26737	26887	26906	26919	26934	26949	26962				
	4643	4668*	4702*	4742*	4781*	4819*	4865*	4900*	4934*	4967*	5001*	5036*	5074*
	5111*	5149*	5187*	5225*	5265*	5307*	5346*	5385*	5418*	5454*	5489*	5529*	5562*
	5592*	5626*	5660*	5702*	5743*	5778*	5818*	5858*	5891*	5928*	5971*	6011*	6045*
	6078*	6110*	6144*	6178*	6213*	6253*	6286*	6325*	6360*	6396*	6449*	6503*	6557*
	6611*	6666*	6709*	6752*	6794*	6836*	6879*	6921*	6964*	7007*	7048*	7089*	7144*
	7196*	7230*	7264*	7299*	7334*	7370*	7406*	7452*	7490*	7547*	7584*	7641*	7681*
	7720*	7755*	7796*	7834*	7877*	7913*	7946*	7974*	26375	26593*	27243#		
STAB2 071162	4644	27245#											
STACKL= 001000	2953#	3292	4146										
SW09 = 001000	2922#	26728											
SW10 = 002000	2921#	26598	26726										
SW11 = 004000	2920#	26605											
SW12 = 010000	2919#	8029	26297	26368	26417	26477	26549	26887	26906	26919	26934	26949	26962
SW13 = 020000	2918#	26370	26674										
SW14 = 040000	2917#	26594											
SW15 = 100000	2916#												
TBSER 065026	24890	26428#											
TSET 062070	24890#												
TSTOPT 066210	8036	26884#											

SCOPEA 065260
SCOPEB 065266
SELTST 066700
SOB1 050532
SOB2 050536
SOB3 050724
SR = 177570
STAB1 070140
STAB2 071162
STACKL= 001000
SW09 = 001000
SW10 = 002000
SW11 = 004000
SW12 = 010000
SW13 = 020000
SW14 = 040000
SW15 = 100000
TBSER 065026
TSET 062070
TSTOPT 066210

T0066	006674	6053	6077#		
T0067	006736	6086	6109#		
T0070	007004	6119	6143#		
T0071	007054	6153	6177#		
T0072	007120	6188	6212#		
T0073	007200	6252#			
T0074	007242	6261	6285#		
T0075	007314	6300	6324#		
T0076	007362	6334	6359#		
T0077	007434	6370	6395#		
T0100	007536	6424	6448#		
T0101	007640	6476	6502#		
T0102	007744	6530	6556#		
T0103	010050	6584	6610#		
T0104	010152	6638	6665#		
T0105	010242	6684	6708#		
T0106	010332	6727	6751#		
T0107	010422	6770	6793#		
T0110	010512	6812	6835#		
T0111	010600	6854	6878#		
T0112	010666	6897	6920#		
T0113	010756	6939	6963#		
T0114	011046	6982	7006#		
T0115	011124	7023	7047#		
T0116	011202	7064	7088#		
T0117	011302	7143#			
T0120	011402	7195#			
T0121	011454	7206	7229#		
T0122	011526	7240	7263#		
T0123	011600	7274	7298#		
T0124	011652	7309	7333#		
T0125	011714	7346	7369#		
T0126	011756	7381	7405#		
T0127	012052	7428	7451#		
T0130	012132	7464	7489#		
T0131	012262	7520	7546#		
T0132	012336	7583#			
T0133	012466	7614	7640#		
T0134	012550	7680#			
T0135	012634	7695	7719#		
T0136	012706	7731	7754#	7771	
T0137	013002	7795#	7812		
T0140	013064	7809	7833#		
T0141	013172	7876#			
T0142	013242	7886	7912#		
T0143	013302	7921	7945#		
T0144	013344	7954	7973#	7991	8013
T0145	013644	8058#			
T0146	013676	8092#			
T0147	013742	8138#			
T0150	013766	8170#			
T0151	014010	8197#			
T0152	014034	8229#			
T0153	014056	8260#			
T0154	014102	8294#			
T0155	014124	8325#			

T0156	014150	8358#
T0157	014172	8389#
T0160	014232	8426#
T0161	014276	8466#
T0162	014334	8503#
T0163	014376	8543#
T0164	014420	8574#
T0165	014446	8605#
T0166	014474	8638#
T0167	014520	8670#
T0170	014544	8702#
T0171	014570	8734#
T0172	014614	8766#
T0173	014642	8799#
T0174	014670	8832#
T0175	014730	8868#
T0176	014756	8899#
T0177	015000	8931#
T0200	015026	8964#
T0201	015054	8997#
T0202	015102	9029#
T0203	015126	9062#
T0204	015152	9094#
T0205	015176	9126#
T0206	015222	9159#
T0207	015246	9191#
T0210	015274	9224#
T0211	015316	9255#
T0212	015344	9288#
T0213	015372	9320#
T0214	015420	9354#
T0215	015444	9387#
T0216	015470	9420#
T0217	015514	9452#
T0220	015540	9484#
T0221	015562	9515#
T0222	015610	9548#
T0223	015634	9580#
T0224	015660	9612#
T0225	015736	9657#
T0226	015774	9692#
T0227	016052	9738#
T0230	016112	9776#
T0231	016252	9850#
T0232	016310	9885#
T0233	016364	9930#
T0234	016424	9967#
T0235	016516	10017#
T0236	016576	10066#
T0237	016740	10141#
T0240	017020	10187#
T0241	017074	10233#
T0242	017154	10279#
T0243	017234	10326#
T0244	017312	10372#
T0245	017366	10418#

T0246	017446	10464#
T0247	017526	10510#
T0250	017604	10557#
T0251	017664	10602#
T0252	017742	10647#
T0253	020022	10692#
T0254	020100	10737#
T0255	020160	10782#
T0256	020240	10827#
T0257	020322	10873#
T0260	020402	10919#
T0261	020464	10965#
T0262	020544	11011#
T0263	020626	11057#
T0264	020710	11102#
T0265	020776	11149#
T0266	021054	11191#
T0267	021146	11239#
T0270	021226	11281#
T0271	021320	11330#
T0272	021400	11372#
T0273	021466	11419#
T0274	021542	11460#
T0275	021630	11506#
T0276	021706	11551#
T0277	021764	11594#
T0300	022040	11636#
T0301	022120	11680#
T0302	022176	11723#
T0303	022254	11766#
T0304	022334	11809#
T0305	022412	11852#
T0306	022472	11895#
T0307	022546	11937#
T0310	022624	11980#
T0311	022704	12023#
T0312	022760	12067#
T0313	023040	12111#
T0314	023120	12155#
T0315	023174	12198#
T0316	023254	12241#
T0317	023332	12284#
T0320	023412	12327#
T0321	023470	12370#
T0322	023550	12413#
T0323	023630	12455#
T0324	023706	12498#
T0325	023764	12544#
T0326	024046	12590#
T0327	024126	12634#
T0330	024216	12683#
T0331	024300	12728#
T0332	024360	12772#
T0333	024440	12816#
T0334	024522	12860#
T0335	024614	12907#

T0336	024676	12951#
T0337	024754	12994#
T0340	025034	13038#
T0341	025116	13082#
T0342	025174	13126#
T0343	025256	13170#
T0344	025340	13214#
T0345	025416	13259#
T0346	025500	13303#
T0347	025560	13346#
T0350	025642	13390#
T0351	025722	13434#
T0352	026004	13478#
T0353	026066	13521#
T0354	026146	13566#
T0355	026226	13610#
T0356	026306	13655#
T0357	026366	13699#
T0360	026446	13743#
T0361	026526	13787#
T0362	026612	13831#
T0363	026664	13868#
T0364	026744	13910#
T0365	027012	13946#
T0366	027076	13989#
T0367	027150	14026#
T0370	027230	14068#
T0371	027276	14104#
T0372	027360	14148#
T0373	027444	14192#
T0374	027522	14235#
T0375	027612	14280#
T0376	027676	14323#
T0377	027754	14365#
T0400	030040	14408#
T0401	030106	14444#
T0402	030160	14481#
T0403	030226	14518#
T0404	030274	14555#
T0405	030366	14601#
T0406	030460	14647#
T0407	030554	14693#
T0410	030650	14739#
T0411	030742	14785#
T0412	031034	14831#
T0413	031130	14877#
T0414	031224	14924#
T0415	031300	14963#
T0416	031354	15002#
T0417	031426	15041#
T0420	031500	15078#
T0421	031554	15122#
T0422	031634	15167#
T0423	031720	15213#
T0424	032004	15258#
T0425	032064	15304#

T0426	032150	15350#
T0427	032236	15396#
T0430	032324	15444#
T0431	032404	15492#
T0432	032464	15540#
T0433	032550	15588#
T0434	032634	15636#
T0435	032720	15683#
T0436	033004	15728#
T0437	033070	15773#
T0440	033154	15817#
T0441	033236	15861#
T0442	033316	15905#
T0443	033404	15949#
T0444	033476	15996#
T0445	033604	16046#
T0446	033664	16089#
T0447	033746	16132#
T0450	034026	16176#
T0451	034110	16220#
T0452	034162	16258#
T0453	034234	16295#
T0454	034324	16344#
T0455	034414	16392#
T0456	034504	16441#
T0457	034574	16490#
T0460	034676	16545#
T0461	035000	16600#
T0462	035102	16655#
T0463	035204	16710#
T0464	035310	16766#
T0465	035414	16821#
T0466	035520	16877#
T0467	035624	16932#
T0470	035726	16987#
T0471	036030	17042#
T0472	036132	17097#
T0473	036234	17153#
T0474	036340	17209#
T0475	036444	17264#
T0476	036550	17320#
T0477	036654	17376#
T0500	036750	17426#
T0501	037044	17476#
T0502	037140	17526#
T0503	037234	17576#
T0504	037330	17626#
T0505	037436	17679#
T0506	037532	17729#
T0507	037626	17778#
T0510	037670	17815#
T0511	037732	17852#
T0512	037776	17890#
T0513	040042	17927#
T0514	040106	17966#
T0515	040154	18002#

T0516	040236	18048#
T0517	040320	18095#
T0520	040364	18132#
T0521	040432	18169#
T0522	040510	18210#
T0523	040566	18253#
T0524	040634	18291#
T0525	040716	18336#
T0526	041002	18382#
T0527	041064	18427#
T0530	041150	18472#
T0531	041224	18511#
T0532	041300	18551#
T0533	041350	18589#
T0534	041420	18627#
T0535	041472	18665#
T0536	041544	18704#
T0537	041620	18741#
T0540	041704	18785#
T0541	041762	18829#
T0542	042046	18873#
T0543	042124	18918#
T0544	042212	18964#
T0545	042306	19012#
T0546	042370	19057#
T0547	042452	19102#
T0550	042534	19146#
T0551	042622	19191#
T0552	042702	19236#
T0553	042770	19281#
T0554	043050	19326#
T0555	043136	19371#
T0556	043224	19416#
T0557	043312	19462#
T0560	043376	19508#
T0561	043462	19552#
T0562	043546	19596#
T0563	043626	19640#
T0564	043716	19685#
T0565	043776	19729#
T0566	044060	19773#
T0567	044142	19817#
T0570	044224	19862#
T0571	044306	19907#
T0572	044374	19951#
T0573	044462	19996#
T0574	044550	20041#
T0575	044642	20088#
T0576	044734	20134#
T0577	045026	20180#
T0600	045120	20227#
T0601	045210	20274#
T0602	045302	20321#
T0603	045372	20370#
T0604	045436	20408#
T0605	045516	20449#

T0606	045566	20488#
T0607	045640	20527#
T0610	045712	20566#
T0611	045764	20605#
T0612	046040	20644#
T0613	046114	20682#
T0614	046160	20719#
T0615	046232	20756#
T0616	046276	20794#
T0617	046350	20831#
T0620	046414	20872#
T0621	046460	20898#
T0622	046536	20959#
T0623	046602	21000#
T0624	046670	21053#
T0625	046744	21100#
T0626	047032	21151#
T0627	047076	21192#
T0630	047164	21243#
T0631	047240	21289#
T0632	047314	21335#
T0633	047370	21381#
T0634	047454	21432#
T0635	047540	21481#
T0636	047606	21520#
T0637	047674	21552#
T0640	047742	21590#
T0641	050010	21643#
T0642	050060	21684#
T0643	050140	21730#
T0644	050216	21775#
T0645	050310	21824#
T0646	050362	21865#
T0647	050444	21914#
T0650	050476	21948#
T0651	050556	21993#
T0652	050614	22030#
T0653	050652	22068#
T0654	050710	22105#
T0655	050746	22142#
T0656	051054	22200#
T0657	051162	22258#
T0660	051264	22312#
T0661	051370	22366#
T0662	051526	22438#
T0663	051664	22496#
T0664	051752	22523#
T0665	052016	22545#
T0666	052064	22566#
T0667	052202	22601#
T0670	052324	22652#
T0671	052440	22703#
T0672	052554	22754#
T0673	052760	22831#
T0674	053164	22893#
T0675	053306	22927#

T0676	053430	22961#	
T0677	053552	22995#	
T0700	053674	23030#	
T0701	054016	23064#	
T0702	054140	23098#	
T0703	054254	23130#	
T0704	054370	23162#	
T0705	054516	23197#	
T0706	054632	23228#	
T0707	054756	23263#	
T0710	055102	23298#	
T0711	055220	23331#	
T0712	055336	23364#	
T0713	055454	23397#	
T0714	055572	23436#	
T0715	055740	23485#	
T0716	056246	23572#	
T0717	056312	23609#	
T0720	056440	23664#	
T0721	056546	23702#	
T0722	056626	23727#	
T0723	056706	23752#	
T0724	056766	23791#	
T0725	057050	23816#	
T0726	057164	23859#	
T0727	057300	23903#	23950
T0730	057450	23960#	
T0731	057622	24022#	
T0732	057774	24083#	
T0733	060060	24113#	
T0734	060160	24146#	
T0735	060260	24192#	
T0736	060336	24230#	
T0737	060414	24267#	
T0740	060472	24304#	
T0741	060572	24355#	
T0742	060676	24409#	
T0743	061004	24462#	
T0744	061136	24518#	
T0745	061214	24559#	
T0746	061304	24604#	
T0747	061370	24648#	
T0750	061456	24692#	
T0751	061544	24735#	
T0752	061632	24779#	
T0753	061710	24821#	
T0754	061776	24865#	
T0755	062112	24921#	
T0756	062152	24956#	
T0757	062216	24994#	
T0760	062262	25030#	
T0761	062322	25066#	
T0762	062362	25103#	
T0763	062426	25141#	
T0764	062472	25177#	
T0765	062532	25212#	

T0766	062572	25248#												
T0767	062632	25284#												
T0770	062676	25320#												
T0771	062736	25357#												
T0772	063002	25394#												
T0773	063042	25429#												
T0774	063102	25464#												
T0775	063142	25501#												
T0776	063206	25539#												
T0777	063252	25576#												
T1000	063316	25612#												
T1001	063356	25647#												
T1002	063416	25682#												
T1003	063454	25716#												
T1004	063512	25750#												
T1005	063556	25786#												
T1006	063636	25826#												
T1007	063710	25867#												
T1010	063772	25938#												
T1011	064050	26008#												
T1012	064126	26081#												
T1013	064204	26151#												
T1014	064262	26212#												
T1015	064326	26264#												
XCSR = 177564		2949#	7878	7914	7947	22655	22706	22757	22834	23101	23133	23168	23200	23301
		23334	23367	23400	23490	26839								
XDBR = 177566		2950#	26843*											
. = 071164		2579#	2608	2615	2622	2624	2631	2638	2645	2652	2654	2656	2658	2660
		2662	2664	2666	2668	2670	2672	2674	2676	2678	2680	2682	2684	2686
		2688	2690	2692	2694	2696	2698	2700	2703	2705	2707	2709	2711	2713
		2715	2717	2719	2721	2723	2725	2727	2729	2731	2733	2735	2737	2739
		2741	2743	2745	2747	2749	2751	2753	2755	2757	2759	2761	2763	2765
		2767	2769	2771	2773	2775	2777	2779	2781	2783	2785	2787	2789	2791
		2793	2795	2797	2799	2801	2803	2805	2807	2809	2811	2813	2815	2817
		2819	2821	2823	2825	2827	2829	2831	2833	2835	2837	2839	2841	2843
		2845	2847	2849	2851	2853	2855	2857	2859	2861	2863	2865	2867	2869
		2871	2873	2875	2877	2879	2881	2883	2885	2887	2889	2891	2951#	4152#
		5267	6398	7683	8836	9971	11463	12863	14238	15999	17629	18967	20411	21778
		22618	22909	22943	22977	23011	23046	23080	23112	23144	23165	23179	23211	23244
		23279	23312	23345	23378	23411	24562	25789	26738	26888	26907	26920	26935	26950
		26963	27092#	27243#										

. ABS. 071164 000

ERRORS DETECTED: 0

CBQEAC.BIN, CBQEAC.LST/CRF/SOL/NL: TOC=CBQEAC.P11
RUN-TIME: 89 158 34 SECONDS
RUN-TIME RATIO: 521/282=1.8
CORE USED: 42K (83 PAGES)