

DUP11

OFFLINE SDLC RECEIVER
MD-11-DZDPC-B

EP-DZDPC-B-DL-A
COPYRIGHT © 75-77
FICHE 1 OF 1

AUG 1977
digital
MADE IN USA

This microfiche card contains a grid of frames, each containing technical data. The frames are arranged in approximately 12 rows and 8 columns. The data in the frames includes:

- Hexadecimal addresses and values, such as 00000000, 00000001, 00000002, etc.
- ASCII characters and strings, such as '00000000', '00000001', '00000002', etc.
- Binary data represented as strings of 0s and 1s.
- Control characters and symbols, such as '00000000', '00000001', '00000002', etc.

B01

EJF1CZDP88SEQ

00010000

770712

PDP10 411

RHDR10ZDPCBSEQ

00010000

770712

45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100

1.0 ABSTRACT

THE FUNCTION OF THE DUPII DIAGNOSTICS IS TO VERIFY THAT THE OPTION OPERATES ACCORDING TO SPECIFICATIONS. THE DIAGNOSTICS VERIFY THAT THERE ARE NO MALFUNCTIONS AND THAT ALL OPERATIONS OF THE DUPII ARE CORRECT IN ITS ENVIRONMENT. PARAMETERS MAY BE SET TO ALERT DIAGNOSTICS AS TO THE DUPII CONFIGURATION BY ANSWERING THE PARAMETER DIALOG (LOAD ADDRESS=200, START ADDRESS=1). ALL QUESTIONS SHOULD BE ANSWERED AND THEN EACH DIAGNOSTIC WILL "OVERLAY" THESE PARAMETERS WHICH ARE STORED IN THE "STATUS TABLE" (SEE SECTION 8.4). THE ALTERNATIVE TO THE PARAMETER DIALOG IS DEFAULT PARAMETERS (SEE SECTION 8.5).

THE DIAGNOSTICS WILL RUN UP TO EIGHT CONSECUTIVELY ADDRESSED AND CONSECUTIVELY VECTORED DUPII'S IN A CHAIN MODE, I.E. RUNNING THE DIAGNOSTIC COMPLETELY FOR ONE DEVICE BEFORE STARTING THE NEXT.

DZDPC TESTS ALL RECEIVER SDLC FUNCTIONS IN MAINTENANCE INTERNAL MODE, THAT IS, CLOCKING OF THE DEVICE IS DONE BY THE PROGRAM. THE DEVICE IS SET UP, A SPECIFIC NUMBER OF HALF-CLOCKS ARE DONE, AND A TEST IS MADE FOR A SIGNIFICANT EVENT.

IN CHECKING DATA, THE SOFTWARE EMULATES THE HARDWARE AND USES THE PROCESSOR CARRY BIT AFTER A ROTATE TO PROVIDE AN INPUT TO THE RECEIVER VIA THE MAINTENANCE INPUT DATA BIT. THE PROGRAM CAN THEN LOAD THE RECEIVER DATA BUFFER SERIALY,WITHOUT USING THE TRANSMITTER.

THE RECEIVER BCC IS CHECKED USING THE CRC.CCITT POLYNOMIAL IN THE SAME WAY AS DATA, WITH ONE EXCEPTION--THE BCC IS CALCULATED FIRST BY THE PROGRAM AND THEN COMPARED TO THE RECEIVER OUTPUT.

DZDPC CHECKS ALL MODEM CONTROL AND INTERRUPT LOGIC DEPENDING ON THE PARAMETER INFORMATION SUPPLIED THROUGH THE OVERLAY MAP. IN ADDITION, ALL EIA GATES, WITH THE EXCEPTION OF THE DATA GATES, ARE CHECKED.

CURRENTLY THERE ARE THREE OFF-LINE DIAGNOSTICS THAT ARE TO BE RUN IN SEQUENCE TO ENSURE THAT IF AN ERROR SHOULD OCCUR IT WILL BE DETECTED AT AN EARLY STAGE AND ESTABLISH THAT DIAGNOSIS OF THE ERROR WILL BE IMMEDIATE TO DISCOVERING THE PROBLEM.

NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

E01

SEG 0003

DZOPCB MACY11 27(1006) 18-MAY-77 00:03 PAGE 4
DZOPCB.P11 13-MAY-77 15:26

101
102

103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200

THE THREE DIAGNOSTICS ARE:

1. DZDPB (REV) BASIC AND OFFLINE TRANSMITTER TESTS
2. DZDPC (REV) OFFLINE RECEIVER AND MODEM CONTROL AND INTERRUPT TESTS
3. DZDPD (REV) OFFLINE SDLC AND DECMODE DATA AND FUNCTION TESTS

NOTE: THERE IS A FOURTH MAINDEC, TAPE DZDPE (REV) WHICH IS A QUICK-VERIFY TAPE THAT REQUIRES ANSWERING A DIALOG. ITS FUNCTION IS TO ENABLE THE OPERATOR TO QUICKLY DETERMINE IF THERE IS A PROBLEM WITH THE DEVICE. SEE THE DOCUMENTATION IN THAT LISTING FOR MORE INFORMATION.

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
162
163
164
165
166

2.0 REQUIREMENTS

2.1 EQUIPMENT

ANY PDP11 FAMILY CPU (WITH MINIMUM 8K MEMORY)
 ASR 33 (OR EQUIVALENT)
 DUP11

2.2 STORAGE

PROGRAM WILL USE ALL 8K OF MEMORY EXCEPT WHERE ABS AND BOOTSTRAP LOADER RESIDE. LOCATION 1500 THRU 1560 ARE ESPECIALLY TO BE NOTED AND LEFT UNTOUCHED BY THE OPERATOR AFTER THE DUP11 PARAMETER DIALOG HAS BEEN EXECUTED OR AFTER THE DEFAULT SETUP HAS BEEN DONE.

3.0 LOADING PROCEDURE

3.1 METHOD

ALL PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE ABSOLUTE LOADER. NOTE: IF THE DIAGNOSTICS ARE ON A MEDIA SUCH AS DISK, MAGTAPE, DECTAPE, OR CASSETTE FOLLOW INSTRUCTIONS FOR THE MONITOR WHICH HAS BEEN PROVIDED ON THAT SPECIFIC MEDIA.

ABSOLUTE LOADER STARTING ADDRESS = **500

MEMORY	SIZE
	(*)=
8K	37
12K	57
16K	77
20K	117
24K	137
28K	157

3.1.1 PLACE ADDRESS OF ABS LOADER INTO SWITCH REGISTER. (ALSO PLACE 'HALT' SW UP)

3.1.2 DEPRESS 'LOAD ADDRESS' KEY ON CONSOLE AND RELEASE.

3.1.3 DEPRESS 'START KEY' ON CONSOLE AND RELEASE (PROGRAM SHOULD NOW BE LOADING INTO CPU)

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
217
218
219
220

4.0 STARTING PROCEEDURE

- A. SET SWITCH REGISTER TO 000200
- B. DEPRESS 'LOAD ADDRESS' KEY AND RELEASE
- C. SET SWR TO ZERO FOR DEFAULT PARAMETERS ESTABLISHED IN THE TAPE (SEE SECTION 8.5.3 FOR FULL EXPLANATION OF DEFAULT PARAMETERS) OR LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS PREVIOUSLY SET UP BY THE DUPI1 PARAMETER DIALOG OR A PREVIOUSLY RUN DUPI1 DIAGNOSTIC. SET SWR=1 TO GO THROUGH THE PARAMETER DIALOG. (IT IS NOT NECESSARY TO INPUT NEW PARAMETERS FOR EACH TAPE.) (SECTION 7.2, 8.4 AND 8.5 MAY BE HELPFUL)
- D. DEPRESS 'START KEY' AND RELEASE. THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME (IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO THE FOLLOWING:

'EXAMPLE'

'MAP OF DUPI1 STATUS'

1500	160050	CSR OF FIRST DUPI1
1502	000300	VECTOR OF FIRST DUPI1
1504	140026	STATUS AND SYNC FOR FIRST DUPI1
1506	160060	CSR OF SECOND DUPI1
1510	000310	VECTOR OF SECOND DUPI1
1512	140026	STATUS AND SYNC FOR SECOND DUPI1

THE ABOVE IS ONLY AN EXAMPLE! THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADDRESS 1500 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE USER. FOR INFORMATION ON THE STATUS TABLE SEE SECTION 8.4 FOR HELP.

IT IS POSSIBLE FOR THE OPERATOR TO MANUALLY CHANGE (TOGGLE IN) THE INFORMATION IN THE MAP TO SUIT A SPECIFIC CONFIGURATION OF DEVICES, BUT THE RESPONSIBILITY FOR VERIFYING THAT INFORMATION RESTS WITH THE OPERATOR.

THE PROGRAM WILL TYPE 'R' AND PROCEED TO RUN THE DIAGNOSTIC

4.1 CONTROL SWITCH SETTINGS

- SW 15 SET: HALT ON ERROR
- SW 14 SET: LOOP ON CURRENT TEST
- SW 13 SET: INHIBIT ERROR PRINT OUT
- SW 12 SET: INHIBIT TYPE OUT/BELL ON ERROR.
- SW 11 SET: INHIBIT ITERATIONS. (QUICK PASS)
- SW 10 SET: ESCAPE TO NEXT TEST ON ERROR
- SW 09 SET: LOOP WITH CURRENT DATA
- SW 08 SET: CATCH ERROR AND LOOP ON IT
- SW 07 SET: USE PREVIOUS STATUS TABLE.

I01

DZDPCB MACY11 27(1006) 18-MAY-77 00:03 PAGE 8
DZDPCB.P11 13-MAY-77 15:26

SEG 0007

223
224

SW 06 SET: RESERVED
SW 05 SET: RESERVED

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
269
270
271
272
273
274
275
276
277
278
279
280

SW 04 SET: RESERVED
SW 03 SET: SELECT DUP11'S DESIRED ACTIVE
SW 02 SET: LOCK ON SELECTED TEST
SW 01 SET: RESTART PROGRAM AT SELECTED TEST
SW 00 SET: ENTER PARAMETERS USING MANUAL DIALOG

SWITCHES 8 THROUGH 15 ARE DYNAMIC AND SHOULD BE USED AS NEEDED IN THE DIAGNOSTIC. SWITCHES 0 THROUGH 3 ARE STATIC (ONLY ARE OPERABLE WHEN THE MONITOR PORTION OF THE TAPE IS RUNNING) AND SHOULD BE SET UP PRIOR TO STARTING OR RESTARTING THE DIAGNOSTIC.

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 03 RESELECT DUP11'S DESIRED ACTIVE. PLEASE NOTE THAT A MESSAGE IS TYPED OUT FOR SETTING THE SWITCH REGISTER EQUAL TO DUP11'S ACTIVE. THIS MEANS IF THE SYSTEM HAS THREE DUPIIS B11S 00, 01, 02 WILL BE SET IN LOC 'DUPACTV' FROM THE SWITCH REGISTER. USING THIS SWITCH(SW03) ALTERS THAT LOCATION. THEREFORE, IF THREE DUPIIS ARE IN THE SYSTEM ***DO NOT*** SET SWITCHES GREATER THAN SW 02 IN THE UP POSITION. THIS WOULD BE A FATAL ERROR. DO NOT SELECT MORE ACTIVE DUPIIS THAN HAS BEEN GIVEN INFORMATION ABOUT IN THE PARAMETER PROGRAM.

AS EXPLAINED IN SECTION 1.0, DEVICES SHOULD BE CONSECUTIVELY ADDRESSED, AND CAN BE SELECTED OR DESELECTED USING THIS SWITCH.

- METHOD: A. LOAD ADDRESS 200
- B. START WITH SW 03=1
- C. PROGRAM WILL TYPE MESSAGE
- D. SET THE BINARY NUMBER OF DUPIIS DESIRED ACTIVE. EXAMPLE: 1=1 DUP11; 3=2 DUP11; 7=3 DUP11; 17=4 DUP11 37=5 DUP11 ETC. PRESS CONTINUE.
- E. NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 11/05)
- F. SET WITH ANY OTHER SWITCH SETTINGS DESIRED. PRESS CONTINUE.

SW 01 RESTART PROGRAM AT SELECTED TEST. IT IS STRONGLY SUGGESTED THAT AT LEAST ONE PASS HAS BEEN MADE BEFORE TRYING TO SELECT A TEST THAT IS NOT IN THE ORDER OF SEQUENCE. THE REASON FOR THIS IS THAT THE PROGRAM HAS TO CLEAR AREAS AND SET UP PARAMETERS IN THE MONITOR PORTION OF THE PROGRAM. IT IS POSSIBLE TO LD200, AND RAISE SW01, THEN START, PROVIDED PARAMETERS HAVE BEEN PREVIOUSLY SET UP AS DESCRIBED IN SECTION 4.0. ALSO, WHEN A TEST IS SELECTED, ALWAYS START AT THE VERY BEGINNING OF THAT TEST.

K01

DZDPCB MACY11 27(1006) 18-MAY-77 00:03 PAGE 10
DZDPCB.P11 13-MAY-77 15:26

SEG 0009

281
292

SW 09 LOOP ON CURRENT DATA. THIS SWITCH WILL ONLY WORK IF
CALL 'SCOPI' IS IN THAT TEST. THE REASON IS THAT MOST

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
325
326
327
328
329

TESTS DEAL WITH BLOCKS OF DIFFERENT DATA TO BE SENT OR RECEIVED ALL AT ONCE, THUS KNOWN AS BLOCK DATA--ONE PATTERN CAN'T BE SINGLED OUT. (SEE SECTION 4.1.3.B.1)

4.1.3 SWITCH REGISTER PRIORITIES

A) ERROR SWITCHES

1. SW 12 DELETE PRINT OUT/BELL ON ERROR.
2. SW 13 DELETE ERROR PRINTOUT.
3. SW 15 HALT ON THE ERROR
4. SW 08 GOTO BEGINNING OF THE TEST(ON ERROR).
5. SW 10 GOTO NEXT TEST(ON ERROR).

B) SCOPE SWITCHES

1. SW 09 - (IF ENABLED BY 'SCOPI') ON AN ERROR. IF AN ASTERISK '*' IS PRINTED IN FRONT OF THE TEST NUMBER (EX. *TEST NO. 10), SW09 IS INCORPORATED IN THAT TEST AND THEREFORE SW09 IS USUALLY THE BEST SWITCH FOR THE SCOPE LOOP (SW14=0, SW10=0, SW09=1, SW08=0).

IF SW09 IS NOT ENABLED AND THERE IS A *HARD* ERROR (CONSTANT ERROR) SW08 IS BEST. (SW14=0, SW10=0, SW09=0, SW08=1).

FOR INTERMITTENT ERRORS, SW14=1 WILL LOOP ON TEST REGARDLESS OF ERROR OR NO ERROR. (SW14=1, SW10=0, SW09=0, SW08=1,0)

2. SW 14 - LOOP ON TEST. WILL LOOP ON TEST UNTIL SWITCH IS LOWERED.
3. SW 11 - INHIBIT ITERATIONS (QUICK PASS). ALLOWS ONLY ONE PASS THROUGH A TEST.

4.2 STARTING ADDRESS

STARTING ADDRESS IS AT 000200. THERE ARE NO OTHER STARTING ADDRESSES FOR THE DUP11 DIAGNOSTICS.

NOTE: IF ADDRESS 000042 IS NON-ZERO THE PROGRAM ASSUMES IT IS UNDER ACT11 OR XXDP CONTROL AND WILL ACT ACCORDINGLY. AFTER *ALL* AVAILABLE DUP11'S ARE TESTED THE PROGRAM WILL RETURN TO 'XXDP' OR 'ACT-11'.

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
382

5.0 OPERATING PROCEDURE

WHEN THE PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION FOUR WILL BE PRINTED AND PROGRAM WILL BEGIN RUNNING THE DIAGNOSTIC.

5.1 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

1. HALT ON ERROR (VIA SW 15=1) WHENEVER AN ERROR OCCURS.
2. CLEAR SW 15.
3. SET SW 14: (LOOP ON THIS TEST)
4. SET SW 13: (INHIBIT ERROR PRINT OUT)

THE TEST NUMBER AND PC WILL BE TYPED OUT AND POSSIBLY AN ERROR MESSAGE (THIS DEPENDS ON THE TEST), TO GIVE THE OPERATOR AN IDEA AS TO THE SOURCE OF THE PROBLEM. IF IT IS NECESSARY TO KNOW MORE INFORMATION CONCERNING THE ERROR REPORT, LOOK IN THE LISTING FOR THAT TEST NUMBER WHICH WAS TYPED OUT AND THEN NOTE THE PC OF THE ERROR REPORT. IN THIS WAY THE EXACT FUNCTIONING OF THE TEST CAN BE INTERPRETED SINCE THE ERROR PC IS THE HLT+2 LOCATION.

IN SOME TESTS, THERE IS A SUBROUTINE CALL THROUGH A REGISTER (E.G., JSR R1 FLAG). THE SUBROUTINE DOES THE DATA CHECKING FOR THE TEST AND WILL REPORT AN ERROR IF ONE OCCURS. THIS MEANS THAT THE FAILING TEST COULD BE IN ONE PART OF THE LISTING WHILE THE SUBROUTINE THAT FOUND THE ERROR IS IN ANOTHER PART. TO DETERMINE THE PC OF THE FAILING TEST, CHECK THE REGISTER USED BY THE SUBROUTINE. IT WILL CONTAIN THE RETURN ADDRESS OF THE FAILING TEST.

6.0 ERRORS

AS DESCRIBED PREVIOUSLY THERE WILL ALWAYS BE A TEST NUMBER AND PC TYPED OUT AT THE TIME OF AN ERROR (PROVIDING SW 13=0 AND SW 12=0). IN MOST CASES ADDITIONAL INFORMATION WILL BE SUPPLIED TO THE ERROR MESSAGE WHICH IS TO GIVE THE OPERATOR AN INDICATION OF THE ERROR.

6.1 ERROR RECOVERY

IF FOR SOME REASON THE DUPI1 SHOULD 'HANG THE BUS' (GAIN CONTROL OF BUS SO THAT CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT OR POWER DOWN/UP IS NECESSARY FOR OPERATOR TO REGAIN CONTROL OF CPU. IF THIS SHOULD HAPPEN LOOK IN LOCATION 'TSTNO' FOR THE NUMBER OF THE TEST THAT WAS RUNNING AT THE TIME OF THE CATASTROPHIC ERROR. THIS GIVES THE OPERATOR SOME IDEA AS TO WHAT THE DUPI1 WAS DOING AT THE TIME OF THE ERROR.

383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422

7.0 RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4 (PLEASE). STATUS TABLE SHOULD BE VERIFIED REGARDLESS OF HOW THE PROGRAM WAS STARTED. ALSO, IT IS IMPORTANT TO USE THE LISTING ALONG WITH THE INFORMATION PRINTED ON THE TTY TO COMPLETELY ISOLATE PROBLEMS.

7.2 OPERATING RESTRICTIONS

DUPII "PARAMETER DIALOG" MUST BE RUN ONLY ONCE PRIOR TO THE FIRST RUNNING OF ANY DUPII DIAGNOSTIC IF "DEFAULT PARAMETERS" ARE NOT USED. IF ONLY DUPII DIAGNOSTICS WERE LOADED AFTER DUPII PARAMETER SETUP AND IF CORE MEMORY HAS NOT BEEN CHANGED, I.E. USE OF DIAGNOSTICS OTHER THAN DUPII DIAGNOSTICS, AND IF THERE WERE NO DUPII CONFIGURATION CHANGES, THE DUPII PARAMETER SETUP NEED NEVER BE RUN AGAIN. HOWEVER, IF ANY OF THE ABOVE HAVE BEEN VIOLATED THE DUPII PARAMETER SETUP MUST BE RUN AGAIN BEFORE RUNNING THE DIAGNOSTICS. UNDER NORMAL OPERATING CONDITIONS IT SHOULD NOT BE NECESSARY TO INPUT NEW PARAMETERS TO SUBSEQUENT DIAGNOSTICS, UNLESS A CHANGE IS REQUIRED.

NOTE: AN ALTERNATIVE TO THE ABOVE IS ATTEMPTING THE DEFAULT PARAMETERS WHEN THE PROGRAM IS INITIALLY STARTED WITH SWR=0.

7.3 HARDWARE CONFIGURATION RESTRICTIONS FOR THE PURPOSE OF RUNNING MULTIPLE DUPII'S IN CHAIN MODE.

1. CSR ADDRESSES MUST BE CONSECUTIVE.
2. VECTORS ARE CONSECUTIVE IF PARAMETER PROGRAM IS USED.
3. ALL JUMPERS ARE ASSUMED TO BE AS SETUP IN PARAMETER DIALOG.
4. PRIORITY LEVEL MUST BE THE SAME FOR ALL DEVICES.

423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

ALL DUPII DEVICE DIAGNOSTICS WILL GIVE AN 'END PASS' MESSAGE (PROVIDING NO ERRORS AND SW12=0) WITHIN 4 MINS. THIS IS ASSUMING SW11=1 (DELETE ITERATIONS) IS SET TO GIVE THE FASTEST POSSIBLE EXECUTION. THE ACTUAL EXECUTION TIME DEPENDS GREATLY ON THE PDI1 CPU CONFIGURATION.

8.2 PASS COMPLETE

NOTE: *EVERY* TIME THE PROGRAM IS STARTED, THE TESTS WILL RUN AS IF SW11 (DELETE ITERATIONS) WAS UP (=1). THIS IS TO VERIFY NO *HARD* ERRORS AS SOON AS POSSIBLE. THEREFORE THE FIRST PASS--EACH TIME PROGRAM IS STARTED--WILL BE A 'QUICK PASS' UNTIL ALL DUPII'S IN SYSTEM ARE TESTED. WHEN THE DIAGNOSTIC HAS COMPLETED A PASS WITH THE NORMAL ITERATION COUNT (ICOUNT=50), THE FOLLOWING IS AN EXAMPLE OF THE PRINT OUT TO BE EXPECTED.

END PASS DZDPCB CSR:160050 VEC:300 PASSES:000001 ERRORS:000000

NOTE: THE NUMBERS FOR CSR AND VEC ARE NOT NECESSARILY THE VALUES FOR THE DEVICE. THEY ARE ONLY FOR THIS EXAMPLE.

8.3 KEY LOCATIONS

RETURN CONTAINS THE ADDRESS WHERE PROGRAM WILL RETURN WHEN ITERATION COUNT IS REACHED OR IF LOOP ON TEST IS ASSERTED.

NEXT CONTAINS THE ADDRESS OF THE NEXT TEST TO BE PERFORMED.

TSTNO CONTAINS THE NUMBER OF THE TEST NOW BEING PERFORMED.

RUN THE BIT IN 'RUN' ALWAYS POINTS ONE PAST THE DUPII CURRENTLY BEING TESTED. EXAMPLE: (RUN) /00000000100000 MEANS THAT DUPII NO.05 IS THE DUPII NOW RUNNING.

DUPCRO0-DUPCRO7 (1500)-(1560) THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST UP TO 8 (DECIMAL) DUPIIS SEQUENTIALY. THEY CONTAIN THE CSR, VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DUPII.

DUPACTV EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DUPII WILL BE TESTED IN TURN. EXAMPLE: (DUPACTV) /000000000011111 MEANS

C02

DZDPCB MACY11 27(1006) 18-MAY-77 00:03 PAGE 15
DZDPCB.P11 13-MAY-77 15:26

SEQ 0014

479
480

THAT DUP11 NO. 00,01,02,03,04 WILL BE TESTED.
EXAMPLE: (DUPACTV) /000000000010001 MEANS

481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536

THAT DUPI1 NO. 00,04 WILL BE TESTED.

RXCSR CONTAINS THE RECEIVER CSR OF THE CURRENT DUPI1 UNDER TEST.

8.4 MORE ON THAT 'STATUS TABLE' (1500-1560)

'MAP OF DUPI1 STATUS'

1500	160050
1502	000300
1504	140000

THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF JP TO 8 DUPI1'S IN THE SYSTEM (THESE WILL FOLLOW UNDER THIS TABLE).
EXPLANATION:

1500 160050 THIS IS THE SYSTEM CONTROL REGISTER FOR THE 1ST DUPI1 IN THE SYSTEM.

1502 000300 THIS IS VECTOR 'A' FOR THE FIRST DUPI1 IN THE SYSTEM.

1504 140026 THIS REPRESENTS SYNC AND SOFTWARE STATUS FOR THE FIRST DUP.

THE BITS ARE AS FOLLOWS:

- BIT 15 SET: OPTIONAL CLEAR JUMPER IN
- BIT 14 SET: TURNAROUND CONNECTOR ON
- BIT 13 SET:
- BIT 12 SET:
- BIT 11 SET:
- BIT 10 SET:
- BIT 09 SET:
- BIT 08 SET:
- BIT 07-00 SYNC CHARACTER FOR DECMODE TESTS.

THE ABOVE IS REPEATED FOR EACH DUPI1 IN THE SYSTEM. THE TABLE IS FILLED BY DEFAULT PARAMETERS OR BY THE MANUAL PARAMETER INPUT AS DESCRIBED PREVIOUSLY. ALSO, IF DESIRED BY THE USER - THE LOCATIONS MAY BE ALTERED BY HAND (TOGGLED IN) TO SUIT THE SPECIFIC CONFIGURATION, THUS MAKING EACH DEVICE MAP DIFFERENT. IT IS THE RESPONSIBILITY OF THE OPERATOR TO VERIFY THE DATA IN THE MAP.

8.5 METHOD OF DEVELOPING DEFAULT PARAMETERS

8.5.1 DEFAULT PARAMETER ASSUMPTIONS

TOO MUCH HARDWARE WOULD HAVE TO BE ANALYZED TO SIZE THE THE PARAMETERS. THE PROGRAM MUST ASSUME THE VARIATIONS. THE

E02

DZDPCB MACY11 27(1006) 18-MAY-77 00:03 PAGE 17
DZDPCB.P11 13-MAY-77 15:26

SEG 0016

537
536

RESULT IF NOT TO YOUR SPECIFIC CONFIGURATION, MAY BE ALTERED
BY HAND (TOGGLE IN) AS DESIRED. IN THIS WAY 95% OF THE

539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565

PARAMETER SETUP WAS DONE BY THE PROGRAM AND 5% BY YOU.
THEREFORE:

1) ALL JUMPERS ARE ASSUMED TO BE IN THE FOLLOWING
CONFIGURATION.

	IN	OUT
W1=SECONDARY REC ENABLE	X	
W2=SEC REC DISABLE		X
W3=CLEAR OPTION	X	
W4=SEC TX ENABLE	X	
W5=DSC A CONTROL		X
W6=A+B DS CONTROL	X	
W7=BUS GRANT CONTROL	X	

2) THE H325 TURN AROUND CONNECTOR IS ASSUMED TO BE ON.

3) THE MANUFACTURING OPTION CSR OF 160050 AND VECTOR OF 770
ARE USED.

4) THE BR LEVEL IS ASSUMED TO BE 5.

IN ALL ADJUSTMENTS PLEASE REFER TO SECTION 8.4 FOR GREATER
DETAIL.

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
603
604
605
606
607
608
609
610

DOCUMENT

COPYRIGHT 1977
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

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
659

1 MAINDEC-11-DZDPC-B /<377>/DUP-11 OFFLINE SCLC RECEIVER, MODEM CC
COPYRIGHT(C) 1975,1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

1260 ***** TEST 1 *****
TEST TO PROVE THE INTERACTION OF RECEIVER ENABLE
WITH RECEIVER ACTIVE AND RECEIVE START
OF MESSAGE IN PRIMARY MODE.
1308 ***** TEST 2 *****
TEST TO PROVE THE INTERACTION OF RECEIVER ENABLE
WITH RECEIVER ACTIVE AND RECEIVE
START OF MESSAGE IN SECONDARY MODE.
1379 ***** TEST 3 *****
TEST TO PROVE THE INTERACTION OF REOM
WITH DONE IN PRIMARY MODE
1426 ***** TEST 4 *****
TEST TO PROVE THE INTERACTION OF REOM
1428 WITH DONE IN PRIMARY MODE
USING A COMMON ZERO BIT IN FLAGS.
1479 ***** TEST 5 *****
TEST TO PROVE THE INTERACTION OF REOM
WITH DONE IN SECONDARY MODE. TEST FOR REOM
AT THE WRONG ADDRESS, THEN AT THE CORRECT
SECONDARY ADDRESS.
1554 ***** TEST 6 *****
TEST TO PROVE THE INTERACTION OF ABORT
WITH A DONE AND RX ERROR IN PRIMARY MODE
1598 ***** TEST 7 *****
TEST TO PROVE THE INTERACTION OF ABORT IN SECONDARY MODE
TEST FOR ABORT AT THE WRONG SECONDARY ADDRESS, THEN TEST
AT THE CORRECT SECONDARY ADDRESS
1662 ***** TEST 10 *****
DATA OVERRUN TEST IN PRIMARY MODE. TEST TO
PROVE OVERRUN ERROR AND RX ERROR WILL OCCUR
1701 ***** TEST 11 *****
DATA OVERRUN TEST IN SECONDARY MODE. TEST TO PROVE
THAT OVERRUN DOES NOT OCCUR IF THIS STATION IS NOT
SELECTED. THEN SELECT THIS SECONDARY STATION AND
PROVE OVERRUN ERROR AND RX ERROR WILL OCCUR

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

1756 ***** TEST 12 *****
TEST OF SPECIFIC DATA PATTERNS THRU
THE RECEIVER IN PRIMARY MODE

1816 ***** TEST 13 *****
TEST TO PROVE THAT THE DEVICE WILL
WORK WITH ALL POSSIBLE SECONDARY
STATION ADDRESSES.

1878 ***** TEST 14 *****
TEST OF SPECIFIC CHARACTER DATA PATTERNS
USING BCC CHECK IN PRIMARY MODE.

1997 ***** TEST 15 *****

2060 ***** TEST 16 *****
THIS TEST PROVES THE INTERACTION OF DTR
WITH RING, DSR
AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.

2166 ***** TEST 17 *****
THIS TEST PROVES THE INTERACTION OF RTS
WITH CTS, CARDET
AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.

2272 ***** TEST 20 *****
THIS TEST PROVES THE INTERACTION OF STD
WITH STD, SRD
AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.

2382 ***** TEST 21 *****
THIS TEST PROVES THE INTERACTION OF DTR!RTS!STD
WITH RING, DSR, CTS, CARDET, STD, SRD
AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.

2491 ***** TEST 22 *****

2492 TEST THAT SETTING TRANSMIT INTERRUPT
ENABLE AND TRANSMITTER DONE PRODUCE
AN INTERRUPT ON THE TRANSMITTER VECTOR.

711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763

2525 ***** TEST 23 *****
TEST TO VERIFY THAT A TRANSMITTER DONE
INTERRUPT WILL ONLY OCCUR ONCE IF THE
TXCSR AND TXDBUF ARE *NOT* READ OR WRITTEN.

2577 ***** TEST 24 *****
TEST THAT SETTING DATA SET INTERRUPT
ENABLE AND RECEIVING A DATA SET
CHANGE 1 OR DATA SET CHANGE 2
PRODUCES AN INTERRUPT TO THE
RECEIVER VECTOR

2647 ***** TEST 25 *****
TEST THAT SETTING RECEIVER INTERRUPT
ENABLE AND RECEIVER DONE CAUSES AN
INTERRUPT TO THE RECEIVER VECTOR

2707 ***** TEST 26 *****
TEST TO VERIFY THAT A RECEIVER DONE
INTERRUPT WILL ONLY OCCUR ONCE IF THE
RXCSR AND RXDBUF ARE NOT READ OR WRITTEN

2790 ***** TEST 27 *****
TEST TO VERIFY THAT INTERRUPT VECTOR "A"
OCCURS BEFORE INTERRUPT VECTOR "B" EVEN
WHEN VECTOR "B" IS ENABLED BEFORE
VECTOR "A"

2863 ***** TEST 30 *****
TEST TO VERIFY THAT SERVICING THE
TXDONE BIT RE-ARMS THE INTERRUPT
LOGIC IF INTERRUPT ENABLE IS SET.

2934 ***** TEST 31 *****
TEST TO VERIFY THAT SERVICING THE
RXDONE BIT RE-ARMS THE INTERRUPT
LOGIC IF INTERRUPT ENABLE IS SET.

3017 ***** TEST 32 *****
TEST TO PROVE AN INTERRUPT REQUEST
IS GENERATED WHEN AN ABORT IS RECEIVED.

3075 ***** TEST 33 *****
THIS TEST IS AN AID FOR DEBUGGING CRC
ERRORS. A CHARACTER IS LOADED INTO THE
DUP AND PUSHED OUT BIT BY BIT WHILE
ALLOWING THE OPERATOR TO MONITOR THE CRC
CHARACTER AS IT IS GENERATED. THE DATA CHARACTER
CAN ALSO BE CHANGED BY THE OPERATOR.
PUT SW09=1 TO LOCK ON BITS. TO CONTINUE HIT
ANY KEY ON THE TTY. AFTER 16 TIMES PUT DOWN SW09 TO LEAVE

INTRODUCTION TO DUP11 DIAGNOSTIC

```

764      ;*MAINDEC-11-DZDPC-B /<377>/DUP-11 OFFLINE SDLC RECEIVER, MODEM CONTROL AND INTERRUPT TE
765      ;*COPYRIGHT(C) 1975,1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
766      ;*-----
767
768      ; STARTING PROCEDURE
769      ; LOAD PROGRAM
770      ; LOAD ADDRESS 000200
771      ; PRESS START
772      ; PROGRAM WILL TYPE "MAINDEC-11-DZDPC-B /<377>/DUP-11 OFFLINE SDLC RECEIVER, MODE
773      ; PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
774      ; AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
775      ; AND THEN RESUME TESTING
776
777
778      ; SWITCH REGISTER OPTIONS
779      ; -----
780
781      100000      SW15=100000      ;=1, HALT ON ERROR
782      040000      SW14=40000      ;=1, LOOP ON CURRENT TEST
783      020000      SW13=20000      ;=1, INHIBIT ERROR TYPEOUT
784      010000      SW12=10000      ;=1, DELETE TYPEOUT/BELL ON ERROR.
785      004000      SW11=4000      ;=1, INHIBIT ITERATIONS
786      002000      SW10=2000      ;=1, ESCAPE TO NEXT TEST ON ERROR
787      001000      SW09=1000      ;=1, LOOP WITH CURRENT DATA
788      000400      SW08=400      ;=1, LOOP ON ERROR
789      000200      SW07=200
790      000100      SW06=100
791      000040      SW05=40
792      000020      SW04=20
793      000010      SW03=10
794
795      000004      SW02=4
796      000002      SW01=2
797      000001      SW00=1

```

```

; SELECT DUP'S DESIRED ACTIVE
; NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT
; LOCK ON TEST SELECT
; RESTART PROGRAM AT SELECTED TEST
; ENTER PARAMETERS

```



```

798
799
800      ;REGISTER DEFINITIONS
801      ;-----
802
803      000000      R0=%0      ;GENERAL REGISTER
804      000001      R1=%1      ;GENERAL REGISTER
805      000002      R2=%2      ;GENERAL REGISTER
806      000003      R3=%3      ;GENERAL REGISTER
807      000004      R4=%4      ;GENERAL REGISTER
808      000005      R5=%5      ;GENERAL REGISTER
809      000006      SP=%6      ;PROCESSOR STACK POINTER
810      000007      PC=%7      ;PROGRAM COUNTER
811
812      ;LOCATION EQUIVALENCIES
813      ;-----
814
815      177776      PS=177776    ;PROCESSOR STATUS WORD
816      001150      STACK=1150  ;START OF PROCESSOR STACK
817
818      ;INSTRUCTION DEFINITIONS
819      ;-----
820
821      005746      PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD
822      005726      POP1SP=5726  ;INCREMENT PROCESSOR STACK 1 WORD
823      010046      PUSHRO=10046  ;SAVE R0 ON STACK
824      012600      POPRO=12600   ;RESTORE R0 FROM STACK
825      024646      PUSH2SP=24646 ;DECREMENT STACK TWICE
826      022626      POP2SP=22626  ;INCREMENT STACK TWICE
827      .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
828
829
830      100000      BIT15=100000
831      040000      BIT14=40000
832      020000      BIT13=20000
833      010000      BIT12=10000
834      004000      BIT11=4000
835      002000      BIT10=2000
836      001000      BIT9=1000
837      000400      BIT8=400
838      000200      BIT7=200
839      000100      BIT6=100
840      000040      BIT5=40
841      000020      BIT4=20
842      000010      BIT3=10
843      000004      BIT2=4
844      000002      BIT1=2
845      000001      BIT0=1
846
847

```

TRAPCATCHER FOR UNEXPECTED INTERRUPTS

```

848
849
850 ;*****
851 ;TRAPCATCHER FOR ILLEGAL INTERRUPTS
852 ;THE STANDARD "TRAP CATCHER" IS PLACED
853 ;BETWEEN ADDRESS 0 TO ADDRESS 776.
854 ;IT LOOKS LIKE "PC+2 HALT".
855 ;*****
856
857      000000      .=0
858      ;STANDARD INTERRUPT VECTORS
859      ;-----
860
861      000024      .=24
862      000024      005050      .PFMIL      ;POWER FAIL HANDLER
863      000026      000340      340          ;SERVICE AT LEVEL 7
864      000030      004350      .HLT          ;ERROR HANDLER
865      000032      000340      340          ;SERVICE AT LEVEL 7
866      000034      004316      .TRPSRV     ;GENERAL HANDLER DISPATCH SERVICE
867      000036      000340      340          ;SERVICE AT LEVEL 7
868
869      000040      .=40
870      000042      000000      0          ;SAVE FOR ACT-11 OR DDP2
871      000044      000000      0          ;RETURN ADDRESS IF UNDER ACT-11 OR DDP2
872      000046      003104      0          ;SAVE FOR ACT-11 OR DDP2
873      000052      000052      $ENDAD     ;FOR USE WITH ACT-11 OR DDP2
874      000052      000000      0          ;ACT-11 PROGRAM CHARACTERISTICS
875
876      000174      .=174
877      000174      000000      DISPRG:0   ;SOFTWARE DISPLAY REGISTER
878      000176      000000      SWREG:0      ;SOFTWARE SWITCH REGISTER
879      000200      000200      .=200
880      000200      000137      001562      JMP .START ;GO TO START OF PROGRAM
881
882
883      001000      .=1000
884      001000      005377      040515      047111 MTITLE: .ASCIZ <377><12>/MAINDEC-11-DZDPC-B /<377>/DUP-11 OFF-LINE SDLC RECEIVER, MODEM
885      001200      001200      .=1200
886      ;SWR AND LIGHTS
887      ;-----
888
889      001200      177570      DISPLAY:    177570      ;11/45 CONSOLE LIGHTS
890      001202      177570      SWR:        177570      ;INDIRECT POINTER TO SWITCH REGISTER
891
892      ;INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
893      ;-----
894
895      001204      177560      TKCSR:    177560      ;TELETYPE KEYBOARD CONTROL REGISTER
896      001206      177562      TKDBR:    177562      ;TELETYPE KEYBOARD DATA BUFFER
897      001210      177564      TPCSR:    177564      ;TELEPRINTER CONTROL REGISTER
898      001212      177566      TPDBR:    177566      ;TELEPRINTER DATA BUFFER
899
900      ;PROGRAM CONTROL PARAMETERS
901      ;-----
902

```

DZDPCB MACY11 27(1006) 18-MAY-77 00:03 PAGE 26
 DZDPCB.P11 13-MAY-77 15:26

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

903	001214	000000	RETURN:	0	; SCOPE ADDRESS FOR LOOP ON TEST
904	001216	000000	NEXT:	0	; ADDRESS OF NEXT TEST TO BE EXECUTED
905	001220	000000	LOCK:	0	; ADDRESS FOR LOCK ON CURRENT DATA
906	001222	000001	ICOUNT:	1	; NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
907	001224	000000	LPCNT:	0	; NUMBER OF ITERATIONS COMPLETED
908	001226	000000	TSTNO:	0	; NUMBER OF TEST IN PROGRESS
909	001230	000000	PASCNT:	0	; NUMBER OF PASSES COMPLETED
910	001232	000000	ERRCNT:	0	; TOTAL NUMBER OF ERRORS
911	001234	000000	LSTERR:	0	; PC OF LAST ERROR CALL
912					
913					
914					
915					
916	001236	000000	TEMP1:	0	; TEMPORARY STORAGE
917	001240	000000	TEMP2:	0	; TEMPORARY STORAGE
918	001242	000000	TEMP3:	0	; TEMPORARY STORAGE
919	001244	000000	TEMP4:	0	; TEMPORARY STORAGE
920	001246	000000	TEMP5:	0	; TEMPORARY STORAGE
921	001250	000000	SAVR0:	0	; R0 STORAGE
922	001252	000000	SAVR1:	0	; R1 STORAGE
923	001254	000000	SAVR2:	0	; R2 STORAGE
924	001256	000000	SAVR3:	0	; R3 STORAGE
925	001260	000000	SAVR4:	0	; R4 STORAGE
926	001262	000000	SAVR5:	0	; R5 STORAGE
927	001264	000000	SAVSP:	0	; STACK POINTER STORAGE
928	001266	000000	SAVPC:	0	; PROGRAM COUNTER STORAGE
929					
930	001270	000000	SAVR0A:	0	; R0 STORAGE
931	001272	000000	SAVR1A:	0	; R1 STORAGE
932	001274	000000	SAVR2A:	0	; R2 STORAGE
933	001276	000000	SAVR3A:	0	; R3 STORAGE
934	001300	000000	SAVR4A:	0	; R4 STORAGE
935	001302	000000	SAVR5A:	0	; R5 STORAGE
936	001304	000000	SAVSPA:	0	; STACK POINTER STORAGE
937	001306	000000	SAVPCA:	0	; PROGRAM COUNTER STORAGE
938					
939	001310	000001	DUPACTV:	.BLKB 1	; DUPI1'S SELECTED ACTIVE.
940	001311	000001	DUPNUM:	.BLKB 1	; OCTAL NUMBER OF DUPI1'S.
941	001312	000001	SAVACT:	.BLKB 1	; ORIGINAL ACTV. DEVICES.
942	001313	000001	SAVNUM:	.BLKB 1	; WORKABLE NUMBER.
943	001314	000001	RUN:	.BLKB 1	; POINTER ONE PAST RUNNING DEVICE.
944		001316	.EVEN		
945	001316	001500	CREAM:	DUP.MAP	; TABLE POINTER.

946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
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
:000
:000

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

100000
040000
010000
002000
001000
000400

100000
001000
010000

100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000020
000010

000000
014000
010000
004000

100000
040000
020000
010000

:CONTROL REGISTER DEFINITIONS

:RXCSR BIT DEFINITIONS
DSCA=BIT15 ;DATA SET CHANGE A
RING=BIT14 ;RING
CTS=BIT13 ;CLR TO SEND
CARDET=BIT12 ;CARRIER DETECT
REACT=BIT11 ;REC ACTIVE
SRD=BIT10 ;SEC REC DATA
DSR=BIT9 ;DATA SET RDY
STPSYN=BIT8 ;STRIP SYNC
RXDONE=BIT7 ;REC DONE
RINTEN=BIT6 ;REC INTR ENABLE
DSINTE=BIT5 ;DSC INTR ENABLE
RCVEN=BIT4 ;REC ENABLE
STD=BIT3 ;SEC XMIT DATA
RTS=BIT2 ;REQ TO SEND
DTR=BIT1 ;DATA TERM RDY
DSCB=BIT0 ;DATA SET CHANGE B
:RXDBUF BIT DEFINITIONS
RXDERR=BIT15 ;REC DATA ERROR
OVRUN=BIT14 ;OVERRUN ERROR
CRCERR=BIT12 ;CRC ERROR
RABORT=BIT10 ;REC ABORT
REOM=BIT9 ;REC END OF MESSAGE
RSOM=BIT8 ;REC START OF MESSAGE
:PARCSR BIT DEFINITIONS
DECMOD=BIT15 ;DEC MODE (DDCMP)
CRCEN=BIT9 ;CRC ENABLE
PRISEC=BIT12 ;PRI/SEC SELECT
:TXCSR BIT DEFINITIONS
TXDLAT=BIT15 ;TX DATA LATE
MTDATA=BIT14 ;MAINT DATA OUT
CLK=BIT13 ;CLK
MMODEB=BIT12 ;MAINT MODE B
MMODEA=BIT11 ;MAINT MODE A
BITW=BIT10 ;BIT WINDOW INPUT
TXACT=BIT9 ;TX ACTIVE
MRESET=BIT8 ;MASTER RESET
TXDONE=BIT7 ;XMIT DONE
TXINTE=BIT6 ;XMIT DONE INTR ENABLE
SEND=BIT4 ;SEND
HDXEN=BIT3 ;HDX/FDX
:TXCSR WRD DEFINITIONS
USER=0 ;USER MODE
MMODE=14000 ;MAINT INT MODE
MEXT=10000 ;MAINT EXT MODE
SYSTST=4000 ;SYSTEM TEST MODE
:TXDBUF BIT DEFINITIONS

RCRC7T=BIT15
RCRCIN=BIT14
TCRC7T=BIT13
TCRCIN=BIT12

DZDPCB MACY11 27(1006) 18-MAY-77
DZDPCB.P11 13-MAY-77 15:26

00:03 PAGE 28
PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

1002 004000
1003 002000
1004 001000
1005 003400
1006
1007
1008
1009 001320 000000
1010 001322 000001
1011 001323 000001
1012 001324 000000
1013 001326 000000
1014 001330 000000
1015 001332 000000
1016 001334 000001
1017 001336 000001
1018
1019

TIMER=BIT11 ;MAINTENANCE TIMER
TABORT=BIT10 ;TRANSMIT ABORT
TEOM=BIT9 ;TRANSMIT END OF MESSAGE
TSOM=BIT8 ;TRANSMIT START OF MESSAGE

:MISC. PROGRAM DEFINITIONS

PRIRTY: .WORD 0
TCNFLG: .BLKB 1
OPCLRJ: .BLKB 1
DATA: .WORD 0
SHIFTS: .WORD 0000
MIND: .WORD 0000
FLAG: .WORD 0
STJMFL: .BLKW 1
SRJMFL: .BLKW 1

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

1020
1021
1022
1023
1024
1025
1026
1027
1028
1029
1030
1031
1032
1033
1034
1035
1036
1037
1038
1039
1040
1041
1042
1043
1044
1045
1046
1047
1048
1049
1050
1051
1052
1053
1054
1055
1056
1057
1058
1059
1060
1061
1062
1063
1064
1065
1066

001340 000
001341 000
001342 000
001343 000

000000

;PROGRAM CONTROL FLAGS

INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG
ERRFLG: .BYTE 0 ;ERROR OCCURED FLAG
LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG
QV.FLG: .BYTE 0 ;QUICK VERIFY FLAG.
;ON FIRST PASS OF EACH DUPI1 ITERATIONS
;WILL BE SUPPRESSED

.EVEN
\$Y=0

;DEFINITIONS FOR TRAP SUBROUTINE CALLS
;POINTERS TO SUBROUTINES CAN BE FOUND
;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS

::*****

.TRPTAB:
SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER
;SCOPE
SCOPI=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
;SCOPI
TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE
;TYPE
INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE
;INSTR
INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER
;INSTER
PARAM=TRAP+5 ;CALL TO NUMERICAL DATA INFLT ROUTINE
;PARAM
SAVOS=TRAP+6 ;CALL TO REGISTER SAVE ROUTINE
;SAVOS
RESOS=TRAP+7 ;CALL TO REGISTER RESTORE ROUTINE
;RESOS
CONVRT=TRAP+10 ;CALL TO DATA OUTPUT ROUTINE
;CONVRT
CNVRT=TRAP+11 ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR-LF.
;CNVRT
PKCLK=TRAP+12 ;CALL TO CLOCK ROUTINE
;PKCLK
SETFLG=TRAP+13 ;CALL TO TELETYPE INPUT ROUTINE
;SETFLG

::*****

```

1067                                     ;DUPI1 VECTOR AND REGISTER INDIRECT POINTERS
1068
1069 001374 000000 DUPRVC: 0 ;POINTER TO DUPI1 RECEIVER INTERRUPT VECTOR
1070 001376 000000 DUPRPS: 0 ;POINTER TO DUPI1 RECEIVER INTERRUPT SERVICE PS
1071 001400 000000 DUPTVC: 0 ;POINTER TO DUPI1 TRANSMITTER INTERRUPT VECTOR
1072 001402 000000 DUPTPS: 0 ;POINTER TO DUPI1 TRANSMITTER INTERRUPT SERVICE PS
1073 001404 000000 RXCSR: 0 ;POINTER TO DUPI1 RECEIVER STATUS REGISTER
1074 001406 000000 RXDBUF: 0 ;POINTER TO DUPI1 RECEIVER DATA BUFFER
1075 001410 000000 PARCSR: 0 ;POINTER TO DUPI1 PARAMETER STATUS REGISTER
1076 001412 000000 TXCSR: 0 ;POINTER TO DUPI1 TRANSMITTER STATUS REGISTER
1077 001414 000000 TXDBUF: 0 ;POINTER TO DUPI1 TRANSMITTER DATA BUFFER
1078 001416 000000 DUPSEC: 0 ;POINTER TO DUPI1 SECONDARY REGISTER SELECT REGISTER
1079 001420 000000 HUPPSR: 0 ;POINTER TO PARAMETER STATUS HIGH BYTE
1080 001422 000000 HUPRBF: 0 ;POINTER TO RECEIVER BUFFER HIGH BYTE
1081 001424 000000 HUPCR: 0 ;POINTER TO RECEIVER CONTROL REG HIGH BYTE
1082 001426 000000 HUPTBF: 0 ;POINTER TO TRANSMITTER BUFFER HIGH BYTE
1083 001430 000000 HUPTCR: 0 ;POINTER TO TRANSMITTER CONTROL REG HIGH BYTE
1084
1085                                     ;DUPI1 CONTROL INDICATORS FOR CURRENT DUPI1 UNDER TEST
1086 -----
1087
1088
1089 001432 000 MASK.A: .BYTE 000 ;LAST CHAR TO TEST AND PARITY MASK
1090
1091 001433 010 CLK.A: .BYTE 8. ;NUMBER OF CLOCKS NEEDED FOR ONE CHAR
1092
1093 001434 000000 _00.00: 000000 ;PARAMETERS
1094

```

;DUPI1 STATUS TABLE AND ADDRESS ASSIGNMENTS
;-----

1095					
1096					
1097					
1098		001500	.=1500		
1099	001500		DUP.MAP:		
1100	001500	000001	DUPCR0:	.BLKW 1	;CONTROL STATUS REGISTER FOR DUPI1 NUMBER 0
1101	001502	000001	DUPTR0:	.BLKW 1	;VECTOR "A" FOR DUPI1 NUMBER 0
1102	001504	000001	DUP0.A:	.BLKW 1	;PARAMETER FOR DUPI1 NUMBER 0
1103					
1104	001506	000001	DUPCR1:	.BLKW 1	;CONTROL STATUS REGISTER FOR DUPI1 NUMBER 1
1105	001510	000001	DUPTR1:	.BLKW 1	;VECTOR "A" FOR DUPI1 NUMBER 1
1106	001512	000001	DUP1.A:	.BLKW 1	;PARAMETER FOR DUPI1 NUMBER 1
1107					
1108	001514	000001	DUPCR2:	.BLKW 1	;CONTROL STATUS REGISTER FOR DUPI1 NUMBER 2
1109	001516	000001	DUPTR2:	.BLKW 1	;VECTOR "A" FOR DUPI1 NUMBER 2
1110	001520	000001	DUP2.A:	.BLKW 1	;PARAMETER FOR DUPI1 NUMBER 2
1111					
1112	001522	000001	DUPCR3:	.BLKW 1	;CONTROL STATUS REGISTER FOR DUPI1 NUMBER 3
1113	001524	000001	DUPTR3:	.BLKW 1	;VECTOR "A" FOR DUPI1 NUMBER 3
1114	001526	000001	DUP3.A:	.BLKW 1	;PARAMETER FOR DUPI1 NUMBER 3
1115					
1116	001530	000001	DUPCR4:	.BLKW 1	;CONTROL STATUS REGISTER FOR DUPI1 NUMBER 4
1117	001532	000001	DUPTR4:	.BLKW 1	;VECTOR "A" FOR DUPI1 NUMBER 4
1118	001534	000001	DUP4.A:	.BLKW 1	;PARAMETER FOR DUPI1 NUMBER 4
1119					
1120	001536	000001	DUPCR5:	.BLKW 1	;CONTROL STATUS REGISTER FOR DUPI1 NUMBER 5
1121	001540	000001	DUPTR5:	.BLKW 1	;VECTOR "A" FOR DUPI1 NUMBER 5
1122	001542	000001	DUP5.A:	.BLKW 1	;PARAMETER FOR DUPI1 NUMBER 5
1123					
1124	001544	000001	DUPCR6:	.BLKW 1	;CONTROL STATUS REGISTER FOR DUPI1 NUMBER 6
1125	001546	000001	DUPTR6:	.BLKW 1	;VECTOR "A" FOR DUPI1 NUMBER 6
1126	001550	000001	DUP6.A:	.BLKW 1	;PARAMETER FOR DUPI1 NUMBER 6
1127					
1128	001552	000001	DUPCR7:	.BLKW 1	;CONTROL STATUS REGISTER FOR DUPI1 NUMBER 7
1129	001554	000001	DUPTR7:	.BLKW 1	;VECTOR "A" FOR DUPI1 NUMBER 7
1130	001556	000001	DUP7.A:	.BLKW 1	;PARAMETER FOR DUPI1 NUMBER 7
1131					
1132	001560	000000	DUP.END:	000000	
1133					
1134					
1135					
1136					
1137					

	15	14	13	12	11	10	09	08	07	06	05	04	03	02	01	00
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
I	C	O	N	T	R	O	L	I	R	E	G	I	S	T	E	R
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I
I	A	B	C	D	E	F	G	H	I	I	I	I	I	I	I	I
I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I	I

DEFINITIONS

- A- OPTIONAL CLEAR JUMPER IN=1
- B- TURNAROUND CONNECTOR ON=1
- C-
- D-

```

1157
1158 ;PROGRAM INITIALIZATION
1159 ;LOCK OUT INTERRUPTS
1160 ;SET UP PROCESSOR STACK
1161 ;SET UP POWER FAIL VECTOR
1162 ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1163 ;TYPE TITLE MESSAGE
1164
1165 001562 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
1166 001570 012706 001150 MOV #STACK,SP ;SET UP STACK
1167 001574 012737 005050 000024 MOV #.PFAIL,2#24 ;SET UP POWER FAIL VECTOR
1168 001602 113737 001311 001313 MOV# DUPNUM,SAVNUM ;SAVE NUMBER OF DEVICES IN SYSTEM
1169 001610 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
1170 001614 105037 001341 CLRB ERRFLG ;CLEAR ERROR FLAG
1171 001620 105037 001343 CLRB QV.FLG ;ZERO QUICK VERIFY FLAG
1172 001624 012737 001500 001316 MOV #DUP.MAP,CREAM ;GET MAP POINTER.
1173 001632 112737 000001 001314 MOV# #1,RUN ;POINT POINTER TO FIRST DEVICE.
1174 001640 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
1175 001644 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
1176 001650 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
1177 001656 012737 001562 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
1178 ;TESTING STARTS
1179 001664 013746 000006 MOV 2#6,-(SP) ;SAVE CURRENT VECTORS
1180 001670 013746 000004 MOV 2#4,-(SP) ;
1181 001674 012737 001710 000004 MOV #12$,2#4 ;SETUP FOR TIMEOUT
1182 001702 005777 177274 TST 2$SWR ;REFERENCE HARDWARE SWITCH REG
1183 001706 000407 BR 13$ ;BR IF IT EXISTS
1184 001710 012737 000176 001202 12$: MOV #SWREG,SWR ;POINT TO SOFT SWR
1185 001716 012737 000174 001200 MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
1186 001724 022626 CMP (SP)+,(SP)+ ;ADJUST STACK
1187 001726 012637 000004 13$: MOV (SP)+,2#4 ;RESTORE VECTORS
1188 001732 012637 000006 MOV (SP)+,2#6 ;
1189 001736 105737 001340 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
1190 001742 001401 BEQ 11$
1191 001744 000410 BR 6$
1192 001746 022737 003104 000042 11$: CMP #SENDAD,2#42 ;IF ACT-11 AUTO MODE.
1193 001754 001404 BEQ 6$ ;DON'T TYPE ID
1194 001756 104402 001000 TYPE #MTITLE ;TYPE TITLE MESSAGE
1195 001762 105137 001340 COMB #INIFLG ;IF NOT SET FLAG AND DO
1196 001766 105777 177210 6$: TSTB 2$SWR ;BIT7=1??
1197 001772 100002 BPL 10$
1198 001774 000137 002520 JMP 1$
1199 002000 10$:
1200 002000 032777 000001 177174 BIT #SW00,2$SWR ;ENTER PARAMETERS
1201 002006 001002 BNE .+6 ;YES
1202 002010 000137 002360 JMP 21$ ;NO
1203 002014 105137 001332 COMB FLAG
1204 002020 112737 000001 001340 MOV# #1,INIFLG ;SET TO MANUAL ENTRY
1205 002026 012700 001500 MOV #DUP.MAP,RO ;CLR MAP
1206 002032 005020 68$: CLR (RO)+
1207 002034 020027 001560 CMP RO,#DUP.END ;DONE WITH MAP?
1208 002040 001374 BNE 68$ ;BR IF NO
1209 002042 104403 INSTR ;OUTPUT MESSAGE $ GET INPJT STRING
1210 002044 005473 MCSR ;MESSAGE
1211 002046 104405 PARAM ;CONVERT STRING
1212 002050 160000 160000 ;LOW LIMIT

```

1213	002052	175500			175500	:HIGH LIMIT
1214	002054	001500			DUPCRO	:STORE AT THIS LOCATION
1215	002056	001			.BYTE 1	:MASK
1216	002057	001			.BYTE 1	:HOW MANY TIMES + 2
1217	002060	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1218	002062	005512			MVEC	:MESSAGE
1219	002064	104405			PARAM	:CONVERT STRING
1220	002066	000300			300	:LOW LIMIT
1221	002070	000770			770	:HIGH LIMIT
1222	002072	001502			DUPTRO	:STORE AT THIS LOCATION
1223	002074	001			.BYTE 1	:MASK
1224	002075	001			.BYTE 1	:HOW MANY TIMES + 2
1225	002076	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1226	002100	005702			MPAR	:MESSAGE
1227	002102	104405			PARAM	:CONVERT STRING
1228	002104	000004			4	:LOW LIMIT
1229	002106	000007			7	:HIGH LIMIT
1230	002110	001240			TEMP2	:STORE AT THIS LOCATION
1231	002112	000			.BYTE 0	:MASK
1232	002113	001			.BYTE 1	:HOW MANY TIMES + 2
1233	002114	013737	001240	001320	MOV	TEMP2,PRIRTY :SAVE PRIORITY
1234	002122	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1235	002124	005647			MTOTAL	:MESSAGE
1236	002126	104405			PARAM	:CONVERT STRING
1237	002130	000001			1	:LOW LIMIT
1238	002132	000010			8	:HIGH LIMIT
1239	002134	001236			TEMP1	:STORE AT THIS LOCATION
1240	002136	000			.BYTE 0	:MASK
1241	002137	001			.BYTE 1	:HOW MANY TIMES + 2
1242	002140	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1243	002142	005525			MJMPR	:MESSAGE
1244	002144	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1245	002146	001323			OPCLRJ	:THIS FLAG
1246	002150	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1247	002152	005600			MTCN	:MESSAGE
1248	002154	104413			SETFLG	:SET FLAG BASED JPON INPUT STRING
1249	002156	001322			TCNFLG	:THIS FLAG
1250	002160	105737	001322		TSTB	TCNFLG
1251	002164	001410			BEQ	71\$
1252	002166	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1253	002170	005726			MSTJM	:MESSAGE
1254	002172	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1255	002174	001334			STJMFL	:THIS FLAG
1256	002176	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1257	002200	005761			MSRJM	:MESSAGE
1258	002202	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1259	002204	001336			SRJMFL	:THIS FLAG
1260	002206	105737	001323		TSTB	OPCLRJ
1261	002212	001403			BEQ	69\$
1262	002214	052737	100000	001504	BIS	#BIT15,DUPO.A
1263	002222	105737	001322		TSTB	TCNFLG
1264	002226	001403			BEQ	70\$
1265	002230	052737	040000	001504	BIS	#BIT14,DUPO.A
1266	002236	112737	000001	001312	MOVB	#1,SAVACT
1267	002244	113737	001236	001311	MOVB	TEMP1,DUPNUM
1268	002252	113737	001236	001313	MOVB	TEMP1,SAVNUM

DZDPCB MACY11 27(1006) 18-MAY-77 00:03 PAGE 35
DZDPCB.P11 13-MAY-77 15:26 PROGRAM INITIALIZATION AND START UP.

1269	002260	005337	001236	65\$:	DEC	TEMP1			
1270	002264	001404			BEQ	64\$			
1271	002266	000261			SEC				
1272	002270	106137	001312		ROLB	SAVACT			
1273	002274	000771			BR	65\$			
1274	002276	113737	001312	001240	64\$:	MOVB	SAVACT,TEMP2	;# OF TIMES	
1275	002304	113737	001312	001310		MOVB	SAVACT,DUPACTV		
1276	002312	000241			CLC				
1277	002314	106037	001240		RORB	TEMP2			
1278	002320	012700	001500		MOV	#DUPCR0,R0			
1279	002324	012701	001506		MOV	#DUPCR1,R1			
1280	002330	000241			CLC				
1281	002332	106037	001240		67\$:	RORB	TEMP2		
1282	002336	103051			BCC	66\$			
1283	002340	012011			MOV	(R0)+(R1)			
1284	002342	062721	000010		ADD	#10,(R1)+	;CSR		
1285	002346	012011			MOV	(R0)+(R1)			
1286	002350	062721	000010		ADC	#10,(R1)+	;VECTOR		
1287	002354	012021			MOV	(R0)+(R1)+	;PARAMETERS		
1288	002356	000764			BR	67\$			
1289	002360	012700	001500		21\$:	MOV	#DUP.MAP,R0	;SETUP TO CLEAR MAP	
1290	002364	005020			20\$:	CLR	(R0)+	;CLEAR	
1291	002366	020027	001560			CMP	R0,#DUP.END	;CHECK FOR FINISH	
1292	002372	001374				BNE	20\$;BR IF MORE TO GO	
1293	002374	012700	001500			MOV	#DUP.MAP,R0	;SETUP TO DEFAULT	
1294	002400	012710	160050			MOV	#160050,(R0)	;LOAD CSR	
1295	002404	012760	000770	000002		MOV	#770,2(R0)	;LOAD VECTOR	
1296	002412	012760	140026	000004		MOV	#140026,4(R0)	;LOAD PARAMETERS AND SYNC	
1297	002420	112737	000005	001320		MOVB	#5,PRIORITY	;LOAD PRIORITY	
1298	002426	012700	000001			MOV	#1,R0	;SAVE CORE THIS WAY	
1299	002432	110037	001310			MOVB	R0,DUPACTV	;PRESET PROGRAM CONTROLS	
1300	002436	110037	001311			MOVB	R0,DUPNUM	;DITTO	
1301	002442	110037	001312			MOVB	R0,SAVACT	;DITTO	
1302	002446	110037	001313			MOVB	R0,SAVNUM	;DITTO	
1303	002452	110037	001322			MOVB	R0,TCNFLAG	;DITTO	
1304	002456	110037	001323			MOVB	R0,OPCLRJ	;DITTO	
1305	002462				66\$:				
1306	002462	104402	006014		16\$:	TYPE	XHEAD	;TYPE HEADER	
1307	002466	012737	001500	001236		MOV	#DUP.MAP,TEMP1	;SET POINTER	
1308	002474	017737	176536	001240	5\$:	MOV	#TEMP1,TEMP2	;SET DATA	
1309	002502	001406				BEQ	1\$;ALL DONE WITH DATA	
1310	002504	104410				CONVRT			
1311	002506	006042				XSTATQ			
1312	002510	062737	000002	001236		ADD	#2,TEMP1	;UPDATE POINTER	
1313	002516	000766				BR	5\$		
1314	002520	032777	000001	176454	1\$:	BIT	#SW00,#SWR		
1315	002526	001405				BEQ	7\$		
1316	002530	005737	001332			TST	FLAG		
1317	002534	001002				BNE	7\$		
1318	002536	000137	002000			JMP	10\$		
1319	002542	005037	001332		7\$:	CLR	FLAG		
1320	002546	005737	000042			TST	#42	;IS PROGRAM RUNNING UNDER MONITOR	
1321	002552	001030				BNE	3\$;BR IF YES	
1322	002554	032777	000010	176420		BIT	#SW03,#SWR	;SELECT SPECIFIC DEVICES"	
1323	002562	001424				BEQ	3\$;BR IF NO.	
1324	002564	104402	005413			TYPE	,MNEW	;TYPE THE MESSAGE.	

```

1325 002570 005000          CLR      RO
1326 002572 000000          HALT
1327 002574 127737 176402 001312  CMPB    @SWR, SAVACT
1328 002602 101404          BLOS   2$
1329 002604 104402 005254    TYPE    ,MERR3
1330 002610 000000          HALT
1331 002612 000776          BR      -2
1332 002614 117737 176362 001310 2$:  MOVB   @SWR, DUPACTV
1333 002622 113700 001310    MOVB   DUPACTV, RO
1334 002626 042700 177400    BIC    #1C<377>, RO
1335 002632 000000          HALT
1336 002634 012700 000300    3$:  MOV    #300, RO
1337 002640 012701 000302    MOV    #302, R1
1338 002644 010120    4$:  MOV    R1 (RO)+
1339 002646 005021          CLR    (R1)+
1340 002650 022021          CMP    (RO)+, (R1)+
1341 002652 022700 001000    CMP    #1000, RO
1342 002656 001372          BNE   4$
1343
1344          ;TEST START AND RESTART
1345          :-----
1346
1347 002660 012737 000340 177776 .BEGIN: MOV    #340, PS
1348 002666 012706 001150    MOV    #STACK, SP
1349 002672 005737 000042    TST   @#42
1350 002676 001023          BNE   2$
1351 002700 C32777 000004 176274    BIT   #BIT2, @SWR
1352 002706 001411          BEQ   1$
1353 002710 104402 005312    TYPE    ,MLOCK
1354 002714 012737 000240 003174    MOV    #NOP, TTST
1355 002722 012737 000240 003176    MOV    #NOP, TTST+2
1356 002730 000406          BR    2$
1357 002732 013737 003306 003174 1$:  MOV    BRW, TTST
1358 002740 013737 003310 003176    MOV    BRX, TTST+2
1359 002746 012737 006224 001214 2$:  MOV    #CYCLE, RETURN
1360 002754 104402 005202    TYPE    MR
1361 002760 000177 176230    JMP   @RETURN

```

```

;ZERO DATA LIGHTS
;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
;IS THE NUMBER VALID?
;BR IF NUMBER IS OK.
;TELL USER OF INVALID NUMBER.
;STOP EVERY THING.
;RESTART THE PROGRAM AGAIN.
;GET NEW DEVICE PATTERN
;SHOW THE USER WHAT HE SELECTED.
;USE ONLY LOW BYTE.
;CONTINUE DYNAMIC SWITCHES.
;PREPARE TO CLEAR THE FLOATING
;VECTOR AREA. 300-776
;START PUTTING "PC+2 - HALT"
;IN VECTOR AREA.
;POP POINTERS
;ALL DONE??
;BR IF NO.

;LOCK OUT INTERRUPTS
;SET UP STACK
;IS PROGRAM UNDER MONITOR CONTROL
;BR IF YES
;CHECK FOR LOCK ON TEST
;BR IF NO LOCK DESIRED.
;TYPE LOCK SELECTED.
;ADJUST SCOPE ROUTINE.
;SET UP TO LOCK
;CONTINUE ALONG.
;PREPARE NORMAL SCOPE ROUTINE
;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
;START AT "CYCLE" FIND WHICH DEVICE TO TEST
;TYPE R
;START TESTING

```

```

1362 ;END OF PASS
1363 ;TYPE NAME OF TEST
1364 ;UPDATE PASS COUNT
1365 ;CHECK FOR EXIT TO ACT-11
1366 ;RESTART TEST
1367
1368 002764 005037 001234 .EOP: CLR LSTERR ;CLEAR LAST ERROR PC
1369 002770 105037 001341 CLR ERRFLG ;CLEAR ERROR FLAG
1370 002774 005237 001230 INC PASCNT ;UPDATE PASS COUNT
1371 003000 013777 001230 176172 MOV PASCNT,DISP ;DISPLAY PASS COUNT
1372 003006 104402 005157 TYPE ,MEPASS ;TYPE END PASS
1373 003012 104402 005341 TYPE ,MCSRX ;TYPE CSR
1374 003016 104411 003130 CNVRT ,XCSR ;SHOW IT
1375 003022 104402 005347 TYPE ,MVECX ;TYPE VECTOR
1376 003026 104411 003136 CNVRT ,XVEC ;SHOW IT
1377 003032 104402 005355 TYPE ,MPASSX ;TYPE PASSES
1378 003036 104411 003144 CNVRT ,XPASS ;SHOW IT
1379 003042 104402 005366 TYPE ,MERRX ;TYPE ERRORS
1380 003046 104411 003152 CNVRT ,XERR ;SHOW IT
1381 003052 105337 001313 DECB SAVNUM ;ARE ALL DEVICES TESTED?
1382 003056 001017 BNE RESTR ;BR IF NO.
1383 003060 112737 000377 001343 MOVB #377,QV,FLG ;SET THE QUICK VERIFY FLAG.
1384 003066 113737 001311 001313 MOVB DUPNUM,SAVNUM ;RESTORE THE COUNT
1385 003074 013701 000042 MOV #42,R1 ;CHECK FOR ACT-11 OR DDP
1386 003100 001406 BEQ RESTR ;IF NOT, CONTINUE TESTING
1387 003102 000005 RESET ;STOP THE SHOW--CLEAR THE WCRLC
1388
1389 003104 $ENDAD: JSR PC,(R1)
1390 003106 NOP
1391 003110 NOP
1392 003112 NOP
1393 003114 NOP
1394 003116 012737 006224 001214 RESTR: MOV #CYCLE,RETURN
1395 003124 000137 006224 JMP CYCLE
1396 003130 000001 XCSR: 1
1397 003132 006 002 .BYTE 6,2
1398 003134 001404 RXCSR
1399 003136 000001 XVEC: 1
1400 003140 003 002 .BYTE 3,2
1401 003142 001374 DUPRVC
1402 003144 000001 XPASS: 1
1403 003146 006 002 .BYTE 6,2
1404 003150 001230 PASCNT
1405 003152 000001 XERR: 1
1406 003154 006 002 .BYTE 6,2
1407 003156 001232 ERRCNT
1408
1409 ;SCOPE LOOP AND INTERATION HANDLER
1410
1411 003160 005037 001234 .SCOPE: CLR LSTERR ;CLEAR LAST ERROR PC
1412 003164 010016 MOV RO,(SP) ;SAVE RO ON STACK
1413 003166 032777 040000 176006 BIT #BIT14,DISP ;LOOP ON TEST?
1414 003174 001407 TTST: BEQ 1$ ;BR IF NO (IF LOCK SW01 = 1;THIS LOCATION = 240)
1415 003176 000437 BR 3$ ;GO TO 3$ (DITTO)
1416 003200 105777 176000 TSTB #TKCSR ;KYBD DONE?
1417 003204 100034 BPL 3$ ;BR IF NO (LOCK: HIT A KEY ON TTY TO GO TO NEXT TEST)

```

```

1418 003206 017700 175774      MOV      @TKDBR,R0      ;CLR DONE BIT
1419 003212 000415              BR        2$           ;CONTINUE
1420 003214 032777 004000 175760 1$:  BIT      #SW11,@SWR    ;DELETE ITERATION (QUICK PASS)?
1421 003222 001011              BNE      2$           ;BR IF YES
1422 003224 105737 001343      TSTB    QV.FLG        ;HAS FIRST PASS BEEN COMPLETED?
1423 003230 001406              BEQ     2$           ;BR IF QUICK VERIFY
1424 003232 005237 001224      INC     LPCNT          ;UPDATE ITERATION COUNTER
1425 003236 023737 001224 001222  CMP     LPCNT,ICCJNT   ;ALL ITERATIONS DONE?
1426 003244 001014              BNE     3$           ;BR IF NOT YET
1427 003246 105037 001341      CLRB   ERRFLG         ;PREPARE FOR NEW TEST
1428 003252 005037 001224      CLR     LPCNT          ;START ICOUNT AT ZERO
1429 003256 005037 001220      CLR     LOCK
1430 003262 012737 000050 001222  MOV     #50,ICOUNT     ;RESET ITERATIONS
1431 003270 013737 001216 001214  MOV     NEXT,RETURN    ;GET NEXT TEST
1432 003276 011600              MOV     (SP),R0        ;POP R0 OFF STACK
1433 003300 022626              POP2SP                ;FAKE AN RTI
1434 003302 000177 175706      JMP     @RETURN        ;GO DO THE TEST
1435 003306 001407      BRW:    1407
1436 003310 000437      BRX:    437
1437
1438                                ;CHECK FOR FREEZE ON CURRENT DATA
1439                                ;-----
1440
1441 003312 032777 001000 175662 .SCOP1: BIT      #SW09,@SWR    ;IS SW09=1(SET)?
1442 003320 001405              BEQ     1$           ;BR IF NOT SET.
1443 003322 005737 001220      TST    LOCK
1444 003326 001402              BEQ     1$
1445 003330 013716 001220      MOV     LOCK,(SP)     ;GOTO THE ADDRESS IN LOCK.
1446 003334 000002              1$:  RTI              ;GO BACK.
1447
1448                                ;TELETYPE OUTPUT ROUTINE
1449                                ;-----
1450
1451 003336 010546              .TYPE: MOV     R5,-(SP)  ;SAVE R5 ON THE STACK.
1452 003340 017605 000002      MOV     @2(SP),R5     ;GET ADDRESS OF MESSAGE.
1453 003344 062766 000002 000002  ADD     #2,2(SP)      ;POP OVER ADDRESS.
1454 003352 032777 010000 175622 1$:  BIT      #SW12,@SWR    ;INHIBIT ALL PRINT OUT??
1455 003360 001012              BNE     3$           ;BR IF NO PRINT OUT WANTED (SW12=1,
1456 003362 105715              TSTB   (R5)          ;IS NUMBER MINUS? (MSB=1(BIT7))
1457 003364 100002              BPL     2$           ;BR IF NUMBER IS PLUS
1458 003366 104402 005136      TYPE   MCRLF         ;TYPE A CR/LF!
1459 003372 105777 175612 2$:  TSTB   @TPCSR        ;TTY READY?
1460 003376 100375              BPL     2$           ;BR IF NO.
1461 003400 112577 175606      MOVB   (R5)+,@TPDBR  ;PRINT CURRENT CHAR.
1462 003404 001362              BNE     1$           ;IF NOT ZERO KEEP PRINTING!
1463 003406 012605 3$:  MOV     (SP)+,R5     ;END OF OUTPUT. RESTORE R5
1464 003410 000002              RTI              ;GO HOME
1465
1466                                ;-----
1467 003412 010346              .INSTR: MOV     R3,-(SP) ;SAVE R3 ON STACK
1468 003414 010446              MOV     R4,-(SP)     ;SAVE R4 ON STACK
1469 003416 017637 000004 003434  MOV     @4(SP),MSG    ;
1470 003424 062766 000002 000004  ADD     #2,4(SP)
1471 003432 104402              .INST1: TYPE
1472 003434 000000      .MSG:  0
1473 003436 012704 006160      MOV     #INBUF,R4

```

DZDPCB MACY11 27(1006) 18-MAY-77 00:03 PAGE 39
 NZDPCB.P11 13-MAY-77 15:26 END OF PASS ROUTINE

1474	003442	012703	000007		MOV	#7,R3		
1475	003446	105272	175532	1\$:	TSTB	@TKCSR		
1476	003452	100375			BPL	1\$		
1477	003454	117714	175526		MOVB	@TKOBR,(R4)		
1478	003460	142714	000200		BICB	#200,(R4)		
1479	003464	122427	000015		CMPB	(R4)+,#15		
1480	003470	001417			BEQ	INSTR2		
1481	003472	105777	175512	2\$:	TSTB	@TPCSR		
1482	003476	100375			BPL	2\$		
1483	003500	017777	175502	175504	MOV	@TKOBR,@TPOBR		
1484	003506	005303			DEC	R3		
1485	003510	001356			BNE	1\$		
1486	003512	012604			MOV	(SP)+,R4		
1487	003514	012603			MOV	(SP)+,R3		
1488	003516	010346		.INSTE:	MOV	R3,-(SP)		
1489	003520	010446			MOV	R4,-(SP)		
1490	003522	104402	005132		TYPE	,MQM		
1491	003526	000741			BR	.INST1		
1492	003530	012604		INSTR2:	MOV	(SP)+,R4	;RESTORE R4	
1493	003532	012603			MOV	(SP)+,R3	;RESTORE R3	
1494	003534	007002			RTI			
1495								
1496								
1497								
1498								
1499	003536	010546		.PARAM:	MOV	R5,-(SP)		
1500	003540	010446			MOV	R4,-(SP)		
1501	003542	016605	000004		MOV	4(SP),R5		
1502	003546	012537	003726		MOV	(R5)+,LOLIM		
1503	003552	012537	003730		MOV	(R5)+,HILIM		
1504	003556	012537	003732		MOV	(R5)+,DEVADR		
1505	003562	112537	003734		MOVB	(R5)+,LOBITS		
1506	003566	112537	003735		MOVB	(R5)+,ADRCNT		
1507	003572	010566	000004		MOV	R5,4(SP)		
1508	003576	005005		PARAM1:	CLR	R5		
1509	003600	012704	006160		MOV	#INBUF,R4		
1510	003604	122714	000015		CMPB	#15,(R4)		
1511	003610	001420			BEQ	PARERR		
1512	003612	121427	000060	1\$:	CMPB	(R4),#60		
1513	003616	002415			BLT	PARERR		
1514	003620	121427	000067		CMPB	(R4),#67		
1515	003624	003012			BGT	PARERR		
1516	003626	142714	000060		BICB	#60,(R4)		
1517	003632	152405			BISB	(R4)+,R5		
1518	003634	122714	000015		CMPB	#15,(R4)		
1519	003640	001406			BEQ	LIMITS		
1520	003642	006205			ASL	R5		
1521	003644	006305			ASL	R5		
1522	003646	006305			ASL	R5		
1523	003650	000760			BR	1\$		
1524	003652	104404		PARERR:	INSTR			
1525	003654	000750			BR	PARAM1		
1526								
1527								
1528								
1529								

;CONVERT ASCII STRING TO OCTAL

 ;TEST TO SEE IF NUMBER IS WITHIN LIMITS

```

1530 003656 020537 003730 LIMITS: CMP R5,HILIM
1531 003662 101373 BHI PARERR
1532 003664 020537 003726 CMP R5,LOLIM
1533 003670 103770 BLO PARERR
1534 003672 133705 003734 BITB LOBITS,R5
1535 003676 001365 BNE PARERR
1536
1537 ;STORE NUMBER AT SPECIFIED ADDRESS
1538
1539 003700 013704 003732 1$: MOV DEVAOR,R4
1540 003704 010524 MOV R5,(R4)+
1541 003706 062705 000002 ADD #2,R5
1542 003712 105337 003735 DECB ADRCNT
1543 003716 001372 BNE 1$
1544 003720 012604 MOV (SP)+,R4
1545 003722 012605 MOV (SP)+,R5
1546 003724 000002 RTI
1547 003726 000000 LOLIM: 0
1548 003730 000000 HILIM: 0
1549 003732 000000 DEVAOR: 0
1550 003734 000000 LOBITS: 0
1551 003735 ADRCNT=LOBITS+1
1552
1553 ;SAVE PC OF TEST THAT FAILED AND R0-R5
1554 -----
1555
1556 003736 016637 000004 001266 .SAVOS: MOV 4(SP),SAVPC ;SAVE R7 (PC)
1557
1558 ;SAVE R0-R5
1559
1560 003744 010537 001262 SVOS: MOV R5,SAVR5 ;SAVE R5
1561 003750 010437 001260 MOV R4,SAVR4 ;SAVE R4
1562 003754 010337 001256 MOV R3,SAVR3 ;SAVE R3
1563 003760 010237 001254 MOV R2,SAVR2 ;SAVE R2
1564 003764 010137 001252 MOV R1,SAVR1 ;SAVE R1
1565 003770 010037 001250 MOV R0,SAVR0 ;SAVE R0
1566 003774 000002 RTI ;LEAVE.
1567
1568 ;RESTORE R0-R5
1569
1570 003776 013700 001250 .RESOS: MOV SAVR0,R0 ;RESTORE R0
1571 004002 013701 001252 MOV SAVR1,R1 ;RESTORE R1
1572 004006 013702 001254 MOV SAVR2,R2 ;RESTORE R2
1573 004012 013703 001256 MOV SAVR3,R3 ;RESTORE R3
1574 004016 013704 001260 MOV SAVR4,R4 ;RESTORE R4
1575 004022 013705 001262 MOV SAVR5,R5 ;RESTORE R5
1576 004026 000002 RTI ;LEAVE
1577
1578
1579 ;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
1580 -----
1581
1582 004030 104402 005136 .CONVR: TYPE MCRLF
1583 004034 010046 .CNVRT: MOV R0,-(SP)
1584 004036 010146 MOV R1,-(SP)
1585 004040 010346 MOV R3,-(SP)

```

1586	004042	010446			MOV	R4, -(SP)
1587	004044	010546			MOV	R5, -(SP)
1588	004046	017601	000012		MOV	2(12(SP), R1
1589	004052	062766	000002	000012	ADD	#2, 12(SP)
1590	004060	012137	004234		MOV	(R1)+, WRDCNT
1591	004064	112137	004236	1\$:	MOVB	(R1)+, CHRCNT
1592	004070	112137	004237		MOVB	(R1)+, SPACNT
1593	004074	013137	004240		MOV	2(R1)+, BINWRD
1594	004100	013704	004240	2\$:	MOV	BINWRD, R4
1595	004104	113705	004236		MOVB	CHRCNT, R5
1596	004110	012700	006054		MOV	#TEMP, R0
1597	004114	010403		3\$:	MOV	R4, R3
1598	004116	042703	177770		BIC	#177770, R3
1599	004122	062703	000060		ADD	#060, R3
1600	004126	110320			MOVB	R3, (R0)+
1601	004130	000241			CLC	
1602	004132	006004			ROR	R4
1603	004134	000241			CLC	
1604	004136	006004			ROR	R4
1605	004140	000241			CLC	
1606	004142	006004			ROR	R4
1607	004144	005305			DEC	R5
1608	004146	001362			BNE	3\$
1609	004150	012703	006116		MOV	#MDATA, R3
1610	004154	114023		4\$:	MOVB	-(R0), (R3)+
1611	004156	105337	004236		DECB	CHRCNT
1612	004162	001374			BNE	4\$
1613	004164	105737	004237		TSTB	SPACNT
1614	004170	001405			BEQ	6\$
1615	004172	112723	000040	5\$:	MOVB	#040, (R3)+
1616	004176	105337	004237		DECB	SPACNT
1617	004202	001373			BNE	5\$
1618	004204	105011		6\$:	CLRB	(R3)
1619	004206	104402	006116		TYPE	, MDATA
1620	004212	005337	004234		DEC	WRDCNT
1621	004216	001322			BNE	1\$
1622	004220	012605			MOV	(SP)+, R5
1623	004222	012604			MOV	(SP)+, R4
1624	004224	012603			MOV	(SP)+, R3
1625	004226	012601			MOV	(SP)+, R1
1626	004230	012600			MOV	(SP)+, R0
1627	004232	000002			RTI	
1628	004234	000000			WRDCNT:	0
1629	004236	000000			CHRCNT:	0
1630		004237			SPACNT=	CHRCNT+1
1631	004240	000000			BINWRD:	0
1632						
1633						
1634						
1635						
1636						
1637						
1638						
1639	004242	017605	000000		.SETFLG:	MOV 2(SP), R5
1640	004246	042737	000040	006160	BIC	#40, INBUF
1641	004254	122737	000116	006160	CMPB	#'N', INBUF ; IS IT "N" ?

```

;COMPARE THE FIRST CHARACTER IN THE TELETYPE INPUT
;BUFFER TO THE CHARACTERS "N" AND "Y".
;IF THE CHARACTER IS "N" CLEAR THE FLAG
;IF THE CHARACTER IS "Y" SET THE FLAG

```

```

1642 004262 001002      BNE      1$
1643 004264 105015      CLAB     (R5) ;000
1644 004266 000406      BR       2$
1645 004270 122737 000131 006160 1$:  CMPB    #'Y,INBUF ;IS IT "Y" ?
1646 004276 001005      BNE      3$
1647 004300 112715 177777      MOVB    #-1,(R5) ;377
1648 004304 062716 000002      2$:  ADD     #2,(SP)
1649 004310 000002      RTI
1650 004312 104404      3$:  INSTER ;RETRY
1651 004314 000752      BR      .SETFLG
1652
1653
1654 ;TRAP DISPATCH SERVICE
1655 ;ARGUMENT OF TRAP IS EXTRACTED
1656 ;AND USED AS OFFSET TO OBTAIN POINTER
1657 ;TO SELECTED SUBROUTINE
1658
1659 004316 011646      .TRPSR: MOV     (SP),-(SP) ;GET PC OF RETURN
1660 004320 162716 000002      SUB     #2,(SP) ;=PC OF TRAP
1661 004324 017616 000000      MOV     @2(SP),-SP ;GET TRAP
1662 004330 006316      TRPOK: ASL     (SP) ;MULTIPLY TRAP ARG BY 2
1663 004332 042716 177001      BIC     #177001,(SP) ;CLEAR UNWANTED BITS
1664 004336 062716 001344      ADD     #.TRPTAB,(SP) ;POINTER TO SUBROUTINE ADDRESS
1665 004342 017616 000000      MOV     @2(SP),-SP ;SUBROUTINE ADDRESS
1666 004346 000136      JMP     @2(SP)+ ;GO TO SUBROUTINE
1667
1668 ;ERROR HANDLER
1669 -----
1670
1671 004350 032777 010000 174624 .HLT:  BIT     #SW12,@SWR ;BELL ON ERROR?
1672 004356 001406      BEQ     XBX ;BR IF NO BELL
1673 004360 105777 174624      TSTB   @TPCSR ;TTY READY.
1674 004364 100003      BPL     XBX ;DON'T WAIT IF TTY NOT READY.
1675 004366 112777 000207 174616      MOVB   #207,@TPDBR ;PUSH A BELL AT THE TTY.
1676 004374 032777 020000 174600 XBX:  BIT     #SW13,@SWR ;DELETE ERROR PRINT OUT?
1677 004402 001105      BNE     HALTS ;BR IF NO PRINT OUT WANTED.
1678 004404 021637 001234      CMP     (SP),LSTERR ;WAS THIS ERROR FOUND LAST TIME?
1679 004410 001404      BEQ     1$ ;BR IF YES
1680 004412 011637 001234      MOV     (SP),LSTERR ;RECORD BEING HERE
1681 004416 105037 001341      CLAB   ERRFLG ;PREPARE HEADER
1682 004422 104406      1$:  SAVOS ;SAVE ALL PROC REGISTERS
1683 004424 011605      MOV     (SP),R5 ;GET THE PC OF ERROR
1684 004426 162705 000002      SUB     #2,R5 ;GET ADDRESS OF TRAP CALL
1685 004432 011504      MOV     (R5),R4 ;GET HLT INSTRUCTION
1686 004434 006304      ASL    R4 ;MULT BY TWO
1687 004436 061504      ADD     (R5),R4 ;DOUBLE IT
1688 004440 006304      ASL    R4 ;MULT AGAIN
1689 004442 042704 177001      BIC     #177001,R4 ;CLEAR JUNK
1690 004446 062704 023236      ADD     #.ERRTAB,R4 ;GET POINTER
1691 004452 012437 004566      MOV     (R4)+,ERRMSG ;GET ERROR MESSAGE
1692 004456 012437 004600      MOV     (R4)+,DATAHD ;GET DATA HEADREER
1693 004462 011437 004612      MOV     (R4),DATABP ;GET DATA TABLE
1694 004466 105737 001341      TSTB   ERRFLG ;TYPE HEADREER
1695 004472 001403      BEQ     TYPMSG ;BR IF YES
1696 004474 005737 004612      TST    DATABP ;DOES DATA TABLE EXIST?
1697 004500 001040      BNE     TYPDAT ;BR IF YES.

```

1698	004502	104402	005136		TYPMSG: TYPE	,MCRLF	
1699	004506	104402	005136		TYPE	,MCRLF	
1700	004512	005737	001220		TST	LOCK	
1701	004516	001402			BEQ	1\$	
1702	004520	104402	005411		TYPE	,MASTEK	
1703	004524	104402	005377		1\$: TYPE	,MTSTN	
1704	004530	104411	005000		CONVRT	,XTSTN	:SHOW IT
1705	004534	104402	005466		TYPE	,MERRPC	:TYPE PC.
1706	004540	104411	004772		CONVRT	,ERTABO	:SHOW IT
1707	004544	104402	005136		TYPE	,MCRLF	:GIVE A CR/LF
1708	004550	112737	177777	001341	MOVB	1-1,ERRFLG	:NO MORE HEADER UNLESS NO DATA TABLE.
1709	004556	005737	004566		TST	ERRMSG	:IS THERE AN ERROR MESSAGE?
1710	004562	001402			BEQ	WRKO.FM	:BR IF NO.
1711	004564	104402			TYPE		:TYPE
1712	004566	000000			ERRMSG: C		: ERROR MESSAGE
1713	004570				WRKO.FM:		
1714	004570	005737	004600		TST	DATAHD	:DATA HEADER?
1715	004574	001402			BEG	TYPDAT	:BR IF NO
1716	004576	104402			TYPE		:TYPE
1717	004600	000000			DATAHD: 0		: DATA HEADER
1718	004602	005737	004612		TYPDAT: TST	DATABP	:DATA TABLE?
1719	004606	001402			BEQ	RESREG	:BR IF NO.
1720	004610	104410			CONVRT		:SHOW
1721	004612	000000			DATABP: 0		: DATA TABLE
1722	004614	104407			RESREG: RESOS		:RESTORE PROC REGISTERS
1723	004616	022737	003104	000042	HALTS: CMP	#SENDAD,2#42	:IF ACT-11 AUTO MODE--HALT!!
1724	004624	001403			BEQ	1\$	
1725	004626	005777	174350		TST	2\$SWR	:HALT ON ERROR?
1726	004632	100035			BPL	EXITER	:BR IF NO HALT ON ERROR
1727	004634	010046			1\$: PUSHRO		:SAVE RO
1728	004636	016600	000002		MOV	2(SP),RO	:SHOW ERROR PC IN DATA LIGHTS
1729	004642	013746	000004		MOV	4,-(SP)	:SAVE OLD TRAP
1730	004646	013746	000006		MOV	6,-(SP)	
1731	004652	012737	004710	000004	MOV	#22\$,4	:FORCE HALT IF TIME-OUT
1732	004660	012737	000340	000006	MOV	#340,6	:WHEN REFERENCING TXCSR
1733	004666	042777	014000	174516	BIC	#SYSST!MEXT,2TXCSR	
1734	004674	000000			HALT		:HALT
1735	004676	012637	000006		MOV	(SP)+,6	:RESTORE TRAP
1736	004702	012637	000004		MOV	(SP)+,4	
1737	004706	000406			BR	33\$	
1738	004710	000000			22\$: HALT		:HALT
1739	004712	022626			CMP	(SP)+,(SP)+	:POP STACK
1740	004714	012637	000006		MOV	(SP)+,6	:RESTORE TRAP
1741	004720	012637	000004		MOV	(SP)+,4	
1742	004724	012600			33\$: POPRO		:GET RO
1743	004726	005237	001232		EXITER: INC	ERRCNT	:UPDATE ERROR COUNT
1744	004732	032777	000400	174242	BIT	#SW08,2\$SWR	:GOTO TOP OF TEST"
1745	004740	001007			BNE	1\$:BR IF YES
1746	004742	032777	002000	174232	BIT	#SW10,2\$SWR	:GOTO NEXT TEST"
1747	004750	001407			BEQ	2\$:BR IF NO
1748	004752	013737	001216	001214	MOV	NEXT,RETURN	:SET FOR NEXT TEST
1749	004760	012706	001150		1\$: MOV	#STACK,SP	:RESET SP
1750	004764	000177	174224		JMP	2\$RETURN	:GOTO SPECIFIED TEST
1751	004770	000002			2\$: RTI		:RETURN
1752	004772	000001			ER*ABC: 1		
1753	004774	006	002		.BYTE	6,2	

```

1754 004776 001266          SAVPC
1755 005000 000001          XTSTN: 1
1756 005002 003 002          .BYTE 3,2
1757 005004 001226          .TSTNO
1758 005006 017600 000000    .PKCLK: MOV 2(SP),R0 ;GET THE # OF TICKS TO POKE
1759 005012 062716 000002    ADD 2,(SP) ;POP OVER THE #
1760 005016
1761 005016 052777 020000 174366    1$: BIS #CLK,@TXCSR ;POKE CLOCK UP
1762 005024 005300          DEC R0 ;ARE WE DONE?
1763 005026 001405          BEQ 2$ ;YES-GO TO 2$
1764 005030 042777 020000 174354    BIC #CLK,@TXCSR ;POKE CLOCK DOWN
1765 005036 005300          DEC R0 ;ARE WE DONE?
1766 005040 001366          BNE 1$ ;NO-REPEAT
1767 005042 000002          2$: RTI ;RETURN
1768
1769
1770          ;WAIT ROUTINE
1771 005044 000240          SMALL: NCF ;STALL
1772 005046 000207          RTS PC ;RETURN
1773
1774          ;POWER FAIL ROUTINE
1775
1776 005050 012737 005060 000024    .PFAIL: MOV #PWRUP,24 ;LOAD PFAIL VECTOR FOR POWER UP
1777 005056 000000          HALT
1778 005060 000005          PWRUP: RESET ;WAIT TTY TO COME UP
1779 005062 012706 001150          MOV #STACK,SP ;REINIT STACK POINTER
1780 005066 012737 005050 000024    MOV #.PFAIL,24 ;LOAD PFAIL VECTOR FOR POWER DOWN
1781 005074 104402          TYPE
1782 005076 005141          MPOWER
1783 005100 000177 174110          JMP @RETURN
1784          ;CLRVEC,ROUTINE TO FILL COMMUNICATION VECTOR AREA WITH .+2,HALT
1785
1786 005.04 012702 000300          CLRVEC: MOV #300,R2 ;R2 COMM VECTOR AREA ADRS
1787 005110 012701 000302          MOV #302,R1 ;INIT R1 WITH ADRS CF HALT
1788 005114 010122          1$: MOV R1,(R2)+ ;MOV .+2 TO PC
1789 005116 005022          CLR (R2)+ ;MOV HALT TO PC
1790 005120 022121          CMP (R1)+,(R1)+ ;INC TO NEXT VECTOR AREA
1791 005122 022701 000776          CMP #776,R1 ;END OF VECTOR AREA
1792 005126 001372          BNE 1$ ;NO
1793 005130 000207          RTS PC ;RETURN
1794
1795
1796
1797 005132 020040 000077          MQM: .ASCIZ / ?/
(2) 005136 005015 000          MCRLF: .ASCIZ <15><12>
(2) 005141 377 053520 020122    MPOWER: .ASCIZ <377>/PWR FAILED. /
(2) 005157 015 042777 042116    MEPASS: .ASCIZ <15><377>/END PASS DZDPCB /
(2) 005202 051377 000          MR: .ASCIZ <377>/R/
(2) 005205 377 051120 043517    MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT.
(2) 005254 044777 051516 043125    MERR3: .ASCIZ <377>/INSUFFICIENT DATA! /
(2) 005300 052377 051505 020124    MTSTPC: .ASCIZ <377>/TEST PC-/
(2) 005312 046377 041517 020113    MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST/
(2) 005341 103 051123 020072    MCSRX: .ASCIZ /CSR: /
(2) 005347 126 041505 020072    MVECX: .ASCIZ /VEC: /
(2) 005355 120 051501 042523    MPASSX: .ASCIZ /PASSES: /
(2) 005366 051105 047522 051522    MERRX: .ASCIZ /ERRORS: /

```

```

(2) 005377      124 051505 020124 MTSTN: .ASCIZ /TEST NO: /
(2) 005411      052      000 MASTEK: .ASCIZ /*/
(2) 005413      377 042523 020124 MNEW: .ASCIZ <377>/SET SWITCH REG TO DUP11'S DESIRED ACTIVE./
(2) 005466 041520 020072      000 MERRPC: .ASCIZ /PC: /
(2) 005473      377 042522 020103 MCSR: .ASCIZ <377>/REC CSR ADRS /
(2) 005512 053377 041505 040440 MVEC: .ASCIZ <377>/VEC ADRS /
(2) 005525      377 051511 052040 MJMPR: .ASCIZ <377>/IS THE OPTIONAL CLR JMPR IN? (Y OR N) /
(2) 005600 044777 020123 044124 MTCN: .ASCIZ <377>/IS THE H325 CONNECTOR ON? (Y OR N) /
(2) 005647      377 020043 043117 MTOTAL: .ASCIZ <377>/# OF DUP'S (IN OCTAL) /
(2) 005702 050377 044522 051117 MPAR: .ASCIZ <377>/PRIORITY (4 TO 7) /
(2) 005726 051777 041505 052040 MSTJM: .ASCIZ <377>/SEC TX JMPR IN? (Y OR N) /
(2) 005761      377 042523 020103 MSRJM: .ASCIZ <377>/SEC RX JMPR IN? (Y OR N) /
(2) 006014 046777 050101 047440 XHEAD: .ASCIZ <377>/MAP OF DUP11 STATUS/<377>
(2)          .EVEN
(2) 006042 000002          XSTATQ: 2
1798 006044      006      003      .BYTE 6.3
1799 006046 001236          TEMP1
1800 006050      006      002      .BYTE 6.2
1801 006052 001240          TEMP2
1802          .EVEN
1803
1804 006054 000000          TEMP: 0
1805          .=. +40
1806 006116 000000          MDATA: 0
1807          .=. +40
1808 006160 000000          INBUF: 0
1809          .=. +40
1810 006222 000001          TRF.PC: .BLKW 1
:6:1

```

```

1812
1813
1814
1815
1816
1817
1818
1819
1820
1821 006224 105737 001310      CYCLE:  TSTB  DUPACTV  :ARE ANY DUPI1'S TO BE TESTED?
1822 006230 001004          BNE      1$      :BR IF OK.
1823 006232 104402 005205      TYPE    ,MERR2  :NO DUPI1'S SELECTED!!
1824 006236 000000          HALT                    :STOP THE SHOW.
1825 006240 000776          BR      -2        :DISQUALIFY CONT. SW.
1826 006242 133737 001314 001310 1$:  BITB  RUN,DUPACTV :IS THIS ONE "ACTIVE"
1827 006250 001020          BNE      2$      :BR IF GOOD ONE FOUND.
1828 006252 000241          CLC                    :CLEAR PROC. CARRY BIT.
1829 006254 106137 001314      ROLB  RUN        :UPDATE POINTER
1830 006260 105537 001314      ADCB  RJN        :CATCH CARRY FROM RUN
1831 006264 062737 000006 001316  ADD    #6,CREAM   :UPDATE ADDRESS POINTER.
1832 006272 022737 001560 001316  CMP    #DUP.END,CREAM
1833 006300 001360          BNE      1$      :KEEP GOING; NOT ALL TESTED FOR.
1834 006302 012737 001500 001316  MOV    #DUP.MAP,CREAM :RESET ADDRESS POINTER.
1835 006310 000754          BR      1$      :KEEP LOOKING FOR ACTIVE DUPI1
1836 006312 000241          CLC                    :CLEAR PROC. CARRY.
1837 006314 106137 001314      ROLB  RUN        :UPDATE POINTER.
1838 006320 105537 001314      ADCB  RUN        :CATCH CARRY.
1839 006324 013700 001316      MOV    CREAM,RO   :GET ADDRESS POINTER.
1840 006330 062737 000006 001316  ADD    #6,CREAM   :UPDATE.
1841 006336 022737 001560 001316  CMP    #DUP.END,CREAM
1842
1843 006344 001003          BNE      3$      :ALL DONE?
1844 006346 012737 001500 001316  MOV    #DUP.MAP,CREAM :BR IF NO.
1845 006354 012037 001404          MOV    (RO)+,RXCSR  :RESTORE POINTER.
1846 006360 012037 001374          MOV    (RO)+,DUPRVC :LOAD SYSTEM CTRL. REG
1847 006364 012037 001434          MOV    (RO)+,LOO.OO :LOAD VECTOR
1848 006370 012700 000002          MOV    #2,RO       :GET PARAMETERS
1849 006374 013737 001404 001424  MOV    RXCSR,HUPRCR :SAVE CORE THIS WAY!
1850 006402 005237 001424          INC    HUPRCR      :GET CONTROL REG HIGH BYTE
1851 006406 013737 001424 001406  MOV    HUPRCR,RXDBUF :GOT IT
1852 006414 005237 001406          INC    RXDBUF      :GET RX CONTROL REG BUFFER
1853 006420 013737 001406 001416  MOV    RXDBUF,DUPSEC :GOT IT
1854 006426 013737 001406 001410  MOV    RXDBUF,PARCSR :GOT SECONDARY REG SELECT REG
1855 006434 013737 001406 001422  MOV    RXDBUF,HUPRBF :GOT PARAMETER STATUS REGISTER
1856 006442 005237 001422          INC    HUPRBF      :GET RX BUFFER HIGH BYTE
1857 006446 013737 001422 001420  MOV    HUPRBF,HUPPSR :GOT IT
1858 006454 013737 001420 001412  MOV    HUPPSR,TXCSR  :GOT PAR STATUS REG HIGH BYTE
1859 006462 005237 001412          INC    TXCSR       :GET TX CONTROL REGISTER
1860 006466 013737 001412 001430  MOV    TXCSR,HUPTCR  :GOT IT
1861 006474 005237 001430          INC    HUPTCR      :GET TX CONTROL REG HIGH BYTE
1862 006500 013737 001430 001414  MOV    HUPTCR,TXDBUF :GOT IT
1863 006506 005237 001414          INC    TXDBUF      :BET TX BUFFER
1864 006512 013737 001414 001426  MOV    TXDBUF,HUPTBF :GOT IT
1865 006520 005237 001426          INC    HUPTBF      :GET TX BUFFER HIGH BYTE
1866
1867 006524 013737 001374 001376  MOV    DJPRVC,DUPRPS :GOT IT
:RX VECTOR

```

```

1868 006532 060037 001376          ADD    RO,DUPRPS      ;RX PRIORITY LEVEL
1869 006536 013737 001376 001400      MOV    DUPRPS,DUPTVC
1870 006544 060037 001400          ADD    RO,DUPTVC     ;TX VECTOR
1871 006550 013737 001400 001402      MOV    DUPTVC,DUPTPS
1872 006556 060037 001402          ADD    RO,DUPTPS     ;TX PRIORITY LEVEL
1873
1874
1875 006562 012700 001434          MOV    #L00.00,RO    ;LOAD STAU5 00-00
1876 006566 012701 001432          MOV    #MASK.A,R1   ;PREPARE MASK.
1877 006572 012702 001433          MOV    #CLK.A,R2    ;PREPARE CLOCKS
1878 006576 004737 006742          JSR    PC,FIX.00    ;GO AND CALCULATE CONFIGURATION.
1879 006602 065737 000042          TST   #42
1880 006606 001050          BNE   4$
1881 006610 032777 000002 172364      BIT   #SW01,#SWR    ;IF SW01=1,GET STARTING TEST #
1882 006616 001444          BEQ   4$
1883 006620 104402 005136          7$:   TYPE ,MCRLF
1884 006624 104403          INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1885 006626 005377          MTSTN ;MESSAGE
1886 006630 104405          PARAM ;CONVERT STRING
1887 006632 000001          I     ;LOW LIMIT
1888 006634 001000          I000 ;HIGH LIMIT
1889 006636 001226          TSTNO ;STORE AT THIS LOCATION
1890 006640 000          .BYTE 0 ;MASK
1891 006641 001          .BYTE 1 ;HOW MANY TIMES + 2
1892 006642 012700 007356          MOV    #TST1,RO
1893 006646 022710 012737          5$:   CMP    #12737,(RO)
1894 006652 001017          BNE   6$
1895 006654 023760 001226 000002      CMP    TSTNO,2(RO)
1896 006662 001013          BNE   6$
1897 006664 022760 001226 000004      CMP    #TSTNO.4(RO)
1898 006672 001007          BNE   6$
1899 006674 010037 001214          MOV    RO,RETURN    ;SAVE PC
1900 006700 104402 005136          TYPE ,MCRLF
1901 006704 104402 005202          TYPE ,MR
1902 006710 000412          BR    8$
1903 006712 005720          6$:   TST   (RO)+
1904 006714 020027 021606      CMP    RO,#TLAST+10
1905 006720 001352          BNE   5$
1906 006722 104402 005132          TYPE ,MQM
1907 006726 000734          BR    7$
1908
1909 006730 012737 007356 001214 4$:   MOV    #TST1,RETURN ;PREPARE RETURN ADDRESS
1910 006736 000177 172252 8$:   JMP    @RETURN      ;GO START TESTING.
1911
1912 006742 011003          FIX.00: MOV   (RO),R3 ;GET PARAMETERS.
1913 006744 000207          5$:   RTS   PC          ;
    
```



```

1914
1915
1916
1917
1918 ;ROUTINE TO DETERMINE IF BIT STUFFING IS REQUIRED
1919 -----
1920
1921 006746 012737 000010 001326 STUFCK: MOV #8, SHIFTS
1922 006754 013700 001324 MOV DATA, R0
1923 006760 012701 000200 MOV #200, R1
1924 006764 006000 1$: ROR R0
1925 006766 005537 001330 ADC MIND
1926 006772 022737 000005 001330 CMP #5, MIND
1927 007000 003004 BGT 2$
1928 007002 005037 001330 CLR MIND
1929 007006 005237 001326 INC SHIFTS
1930 007012 006001 2$: ROR R1
1931 007014 103363 BCC 1$
1932 007016 000207 RTS PC
1933
1934
1935 ;THIS ROUTINE POKES THE RECEIVER BIT WINDOW
1936 ;BASED ON THE INFORMATION IN DATA AND SHIFTS
1937 -----
1938
1939 007020 042777 002000 172364 RPOKE: BIC #BITW, @TXCSR
1940 007026 006037 001324 ROR DATA
1941 007032 103003 BCC 1$
1942 007034 052777 002000 172350 BIS #BITW, @TXCSR
1943 007042 000240 1$: NOP
1944 007044 104412 000002 PKCLK 2
1945 007050 005337 001326 DEC SHIFTS
1946 007054 001361 BNE RPOKE
1947 007056 000201 RTS R1
1948
1949
1950 007060 012137 001244 RFLG: MOV (R1)+, TEMP4 ;GET THE # OF FLAGS
1951 007064 005037 001246 1$: CLR TEMPS ;CLEAR ONES COUNT
1952 007070 042777 002000 172314 BIC #BITW, @TXCSR ;SET FIRST BIT
1953 007076 104412 000002 PKCLK 2 ;PUSH OUT THE BIT
1954 007102 052777 002000 172302 2$: BIS #BITW, @TXCSR ;LOAD THE BIT
1955 007110 104412 000002 PKCLK 2 ;PUSH OUT THE BIT
1956 007114 005237 001246 INC TEMPS ;INCREMENT 1'S COUNTER
1957 007120 022737 000006 001246 CMP #6, TEMPS ;DID WE PUSH OUT 6 ONES
1958 007126 001365 BNE 2$ ;NO-GO BACK
1959 007130 042777 002000 172254 BIC #BITW, @TXCSR ;SET THE LAST BIT
1960 007136 104412 000002 PKCLK 2 ;PUSH OUT THE BIT
1961 007142 005337 001244 DEC TEMP4 ;ARE WE DONE WITH FLAGS?
1962 007146 001346 BNE 1$ ;BR IF NO
1963 007150 000201 RTS R1
1964
1965
1966 007152 012137 001244 SFLG: MOV (R1)+, TEMP4 ;GET THE # OF FLAGS
1967 007156 005037 001246 1$: CLR TEMPS ;CLEAR ONES COUNT
1968 007162 052777 002000 172222 2$: BIS #BITW, @TXCSR ;LOAD THE BIT
1969 007170 104412 000002 PKCLK 2 ;PUSH OUT THE BIT

```

```

1970 007174 005237 001246      INC      TEMPS      ; INCREMENT I'S COUNTER
1971 007200 022737 000006 001246      CMP      #6, TEMPS ; DID WE PUSH OUT 6 ONES
1972 007206 001365                BNE                ; NO-GO BACK
1973 007210 042777 002000 172174      BIC      #BITW, @TXCSR ; SET THE LAST BIT
1974 007216 104412 000002                PKCLK      2        ; PUSH OUT THE BIT
1975 007222 005337 001244                DEC      TEMP4      ; ARE WE DONE WITH FLAGS?
1976 007226 001353                BNE      1$        ; BR IF NO
1977 007230 000201                RTS      R1
1978
1979 007232 012577 172136      SETVEC: MOV      (R5)+, @DUPRVC
1980 007236 012577 172136                MOV      (R5)+, @DUPTVC
1981 007242 112577 172130                MOVB     (R5)+, @DUPRPS
1982 007246 112577 172130                MOVB     (R5)+, @DUPTPS
1983 007252 000205                RTS      R5
1984 007254                NO. ATRAP:
1985 007254 104025                HLT      25
1986 007256 000002                RTI
1987
1988 007260                NO. BTRAP:
1989 007260 104026                HLT      26
1990 007262 000002                RTI
1991
1992                ; THIS ROUTINE PICKS UP THE ADDRESS OF
1993                ; THE JUMPER TABLE AND LOADS R5 WITH
1994                ; THE CORRECT DATA BASED ON THE STATE
1995                ; OF THE JUMPER AND CONNECTOR FLAGS.
1996                ;-----
1997
1998 007264 012100      JUMPER: MOV      (R1)+, R0      ; GET THE TABLE ADDRESS
1999 007266 105737 001322      TSTB     TCNFLAG      ; TEST THE TURN AROUND CONNECTOR FLAG
2000 007272 001406                BEQ      2$          ; BRANCH IF CONNECTOR IS MISSING
2001 007274 105737 001323      TSTB     OPCLRJ      ; TEST CLEAR JUMPER FLAG
2002 007300 001403                BEQ      2$          ; BRANCH IF JUMPER IS MISSING
2003 007302 011005                MOV      (R0), R5    ; MOVE THE DATA TO R5, BOTH JUMPER
2004                ; AND CONNECTOR ARE THERE
2005 007304 000137 007330                JMP      5$
2006 007310 022020      2$:      CMP      (R0)+, (R0)+    ; POP POINTER
2007 007312 105737 001323      TSTB     OPCLRJ      ; TEST CLEAR JUMPER FLAG
2008 007316 001403                BEQ      3$          ; BRANCH IF MISSING
2009 007320 011005                MOV      (R0), R5    ; MOVE DATA- JUMPER IN, CONNECTOR OFF
2010 007322 000137 007330                JMP      5$
2011 007326 012005      3$:      MOV      (R0)+, R5    ; NO CONNECTOR OR JUMPER
2012 007330 000201      5$:      RTS      R1        ; RETURN
2013
2014 007332 012100      OJUMPER: MOV     (R1)+, R0    ; GET THE POINTER ADDRESS
2015 007334 105737 001322      TSTB     TCNFLAG      ; CHECK FOR TURNAROUND CONNECTOR
2016 007340 001403                BEQ      4$          ; BR IF MISSING
2017 007342 011005                MOV      (R0), R5    ; MOVE THE INFO TO R5
2018 007344 000137 007354                JMP      6$
2019 007350 022020      4$:      CMP      (R0)+, (R0)+    ; POP POINTER
2020 007352 011005                MOV      (R0), R5    ; LOAD DATA TO R5
2021 007354 000201      6$:      RTS      R1        ; RETURN
2022
    
```

2023
2024
2025
2026
2027
2028
2029
2030
2031
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

***** TEST 1 *****
*TEST TO PROVE THE INTERACTION OF RECEIVER ENABLE
*WITH RECEIVER ACTIVE AND RECEIVE START
*OF MESSAGE IN PRIMARY MODE.

*
TEST 1
*

```
*****
TEST1:  MOV  #1, @TSTNO
        MOV  #TST2, NEXT
        BIS  #MRESET, @TXCSR ; RESET THE DEVICE
        JSR  PC, SMALL ; WAIT FOR RESET TO FINISH
        MOV  RXCSR, R3 ; GET THE RECEIVER CONTROL REGISTER
        BIS  #RCEN, @PARCSR ; TURN OFF CRC
        BIS  #RCVEN, (R3) ; TURN ON THE RECEIVER
        BIS  #MMODE, @TXCSR ; ENTER M/MODE
        JSR  R1, RFLG ; PUSH OUT A FLAG
        BIT  #REACT, (R3) ; CHECK RECEIVER ACTIVE
        BEQ  1$ ; BRANCH IF BIT IS OFF
        HLT  6 ; BIT WAS SET - SHOULD BE CLEAR
1$:     CLR  DATA ; SETUP FOR CHAR
        MOV  #9, SHIFTS ; POKE CLOCK SETUP
        JSR  R1, @POKE ; PUSH DATA INTO RECEIVER
        BIT  #RXDONE, (R3) ; TEST DONE
        BNE  2$ ; BR IF SET
        MOV  #RXDONE, R5 ; ERROR MSG SETUP
        MOV  (R3), R4 ; SET UP FOUND
        HLT  3 ; RXDONE FAILED TO SET
2$:     BIT  #RSOM, @RXDBUF ; TEST FOR START OF MSG
        BNE  10$ ; BR IF OK
        MOV  #RSOM, R5 ; SET EXPECTED
        MOV  RXDBUF, R3 ; SET UP ERROR MSG
        MOV  (R3), R4 ; SET FOUND
        HLT  3 ; RSOM FAILED TO SET
10$:   MOV  RXCSR, R3
        BIT  #REACT, (R3)
        BNE  11$
        HLT  14
11$:   SCOPE
*****
```

***** TEST 2 *****
*TEST TO PROVE THE INTERACTION OF RECEIVER ENABLE
*WITH RECEIVER ACTIVE AND RECEIVE
*START OF MESSAGE IN SECONDARY MODE.

*

TEST OF REC ENABLE, RXACT, AND RSOM IN SECONDARY MCDE

```

2079 ; TEST 2
2080 ;
2081 ;*****
2082 ;*****
2083 007544 012737 000002 001226 tst2:  MOV  #2, @TSTNO
2084 007552 012737 010034 001216      MOV  #TST3, NEXT
2085 007560 052777 000400 171624      BIS  #MRESÉT, @TXCSR ;RESET THE DEVICE
2086 007566 004737 005044          JSR  PC, SMALL ;WAIT FOR RESET TO FINISH
2087 007572 013703 001404          MOV  RXCSR, R3 ;GRt THE RECEIVER CONTROL REGISTER
2088 007576 052777 001000 171604      BIS  #CRCEN, @PARCSR ;TURN OFF CRC
2089 007604 052713 000020          BIS  #RCVEN, (R3) ;TURN ON THE RECEIVER
2090 007610 052777 010000 171572  BIS  #PRISEC, @PARCSR ;ENTER SECONDARY MODE
2091 007616 052777 014000 171566      BIS  #MMODE, @TXCSR ;ENTER M/MODE
2092 007624 004137 007060          JSR  R1, RFLG ;PUSH OUT A FLAG
2093 007630 000001          1
2094 007632 032713 004000          BIT  #RECACT, (R3) ;CHECK RECEIVER ACTIVE
2095
2096 007636 001403          BEG  3$ ;BR IF OFF
2097 007640 005005          CLR  R5 ;SET EXPECTED
2098 007642 011304          MOV  (R3), R4 ;SET FOUND
2099 007644 104003          HLT  3 ;BIT WAS SET AND SHOULD BE CLEARED
2100 ;WE ARE IN SECONDARY MODE
2101 007646 032713 000200 3$:  BIT  #RXDONE, (R3) ;TEST DONE
2102 007652 001401          BEQ  4$ ;BR IF CLEAR
2103 007654 104003          HLT  3 ;DONE IS SET - SHOULD NOT BE BECAUSE
2104 ;WE ARE IN SECONDARY MODE AND HAVE NOT PUSHED DATA
2105 007656 005037 001324 4$:  CLR  DATA ;CLEAR DATA CHAR
2106 007662 012737 000011 001326  MOV  #9, SHIFTS ;LOAD THE # OF BITS TO PUSH
2107 007670 004137 007020          JSR  R1, RPOKE ;PUSH DATA TO RECEIVER
2108 007674 032713 004000          BIT  #RECACT, (R3) ;TEST ACTIVE
2109 007700 001004          BNE  5$ ;BR IF CLEAR
2110 007702 012705 004000          MOV  #RECACT, R5 ;SET EXPECTED
2111 007706 011304          MOV  (R3), R4 ;SET FOUND
2112 007710 104003          HLT  3 ;ACTIVE FAILED TO SET
2113 007712 032713 000200 5$:  BIT  #RXDONE, (R3) ;TEST DONE
2114 007716 001404          BEQ  6$ ;BR IF NOT SET - THE CHAR WAS
2115 ;THE SECONDARY ADRS
2116 007720 012705 000200          MOV  #RXDONE, R5 ;SET EXPECTED
2117 007724 011304          MOV  (R3), R4 ;SET FOUND
2118 007726 104003          HLT  3 ;DONE WAS SET AND SHOULDN'T BE - THE FIRST
2119 ;DATA WAS SECONDARY ADRS, NOT DATA TO THE BUFFER
2120 007730 032777 000400 171450 6$:  BIT  #RSOM, @RXDBUF ;CHECK START OF MSG
2121 007736 001401          BEQ  7$ ;BR IF NOT SET
2122 007740 104020          HLT  20 ;START OF MSG SHOULD BE CLEAR
2123 007742 112737 000252 001324 7$:  MOVB #252, DATA ;LOAD DATA CHAR
2124 007750 012737 000010 001326  MOV  #8, SHIFTS ;LOAD CLOCK COUNT
2125 007756 004137 007020          JSR  R1, RPOKE ;PUSH OUT DATA
2126 007762 104412 000002          PKCLK 2 ;POKE ANOTHER FULL CLOCK
2127 007766 032713 000200          BIT  #RXDONE, (R3) ;TEST DONE
2128 007772 001011          BNE  10$ ;BR IF SET
2129 007774 012705 000200          MOV  #RXDONE, R5 ;SET EXPECTED
2130 010000 011304          MOV  (R3), R4 ;SET FOUND
2131 010002 104003          HLT  3 ;DONE FAILED TO SET AFTER PUSHING
2132 ;OUT A CHAR IN SEC. MODE
2133 010004 032777 000400 171374      BIT  #RSOM, @RXDBUF
2134 010012 001001          BNE  10$

```

DZDPCB MACY11 27(1006) 18-MAY-77 00:03 PAGE 52
DZDPCB.P11 13-MAY-77 15:26

TEST OF REC ENABLE, RXACT, AND R50M IN SECONDARY MODE

2135 010014 104021
2136 010016 013703 001404
2137 010022 032713 004000
2138 010026 001001
2139 010030 104014
2140 010032 104400

10\$: HLT 21
MOV RXCSR, R3
BIT #REACT, (R3)
BNE 11\$
HLT 14
11\$: SCOPE

***** TEST 3 *****
*TEST TO PROVE THE INTERACTION OF REOM
*WITH DONE IN PRIMARY MODE

*
* TEST 3
*

2153 010034 012737 000003 001226
2154 010042 012737 010262 001216
2155 010050 052777 000400 171334
2156 010056 004737 005044
2157 010062 052777 000020 171314
2158 010070 052777 001000 171312
2159 010076 005037 001236
2160 010102 052777 014000 171302
2161 010110 004137 007060
2162 010114 000001
2163 010116 012737 000125 001324
2164 010124 012737 000010 001326
2165 010132 004137 007020
2166 010136 042777 002000 171246
2167 010144 104412 000002
2168 010150 032777 000200 171226
2169 010156 001001
2170 010160 104000
2171 010162 105777 171220
2172 010166 052777 002000 171216
2173 010174 104412 000002
2174 010200 005237 001236
2175 010204 022737 000006 001236
2176 010212 001365
2177 010214 042777 002000 171170
2178 010222 104412 000004
2179 010226 017737 171152 001240
2180 010234 032777 001000 171144
2181 010242 001001
2182 010244 104001
2183 010246 032737 000200 001240
2184 010254 001001
2185 010256 104002
2186
2187 010260 104400
2188
2189
2190

TST3: MOV #3, #TSTNO
MOV #TST4, NEXT
BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
BIS #RCVEN, @RXCSR ; TURN ON RECEIVER
BIS #RCEN, @PARCSR ; TURN OFF CRC
CLR TEMP1 ; BIT COUNTER
BIS #MMODE, @TXCSR ; ENTER MAINT MODE
JSR R1, RFLG ; PUSH OUT A FLAG
1 ; ONE FLAG
MOV #125, DATA ; CLEAR OUT CHAR
MOV #8, SHIFTS ; LOAD THE # OF CLOCKS
JSR R1, APOKE ; PUSH OUT THE DATA
BIC #BITW, @TXCSR ; CLEAR THE DATA WINDOW
PKCLK 2 ; PUSH OUT THE BIT
BIT #RXDONE, @RXCSR ; CHECK FOR DONE
BNE 5\$; BR IF SET
HLT ; DONE BIT ERROR
5\$: TSTB @RXDBUF ; READ THE BUFFER
64\$: BIS #BITW, @TXCSR ; PUT A.1 IN THE WINDOW
PKCLK 2 ; PUSH IT OUT
INC TEMP1 ; INC THE # TO DO
CMP #6, TEMP1 ; CHECK FOR FINISH
BNE 64\$; BR IF MORE TO GO
BIC #BITW, @TXCSR ; CLEAR THE WINDOW
PKCLK, 4 ; PUSH OUT 2 BINTS
MOV @RXCSR, TEMP2 ; GET THE CSR
BIT #REOM, @RXDBUF ; TEST FOR END OF MSG
BNE 1\$; BR IF SET
HLT 1 ; BIT FAILED TO SET
1\$: BIT #RXDONE, TEMP2 ; TEST DONE
BNE 4\$; BR IF SET
HLT 2 ; DONE FAILED TO SET
4\$: SCOPE ; SCOPE THIS TEST

***** TEST 4 *****
*TEST TO PROVE THE INTERACTION OF REOM

```

2191 ;*WITH DONE IN PRIMARY MODE
2192 ;*USING A COMMON ZERO BIT IN FLAGS.
2193
2194 :*****
2195 :*****
2196 *
2197 TEST 4
2198 *
2199 :*****
2200 :*****
2201 TST4: MOV #4, @TSTNO
2202 MOV @TSTS_NEXT
2203 BIS #RESET, @TXCSR ;RESET THE DEVICE
2204 JSR PC_SMALL ;WAIT FOR RESET TO FINISH
2205 BIS #RCVEN, @RXCSR ;TURN ON RECEIVER
2206 BIS #CRCEN, @PARCSR ;TURN OFF CRC
2207 CLR TEMP1 ;BIT COUNTER
2208 BIS #MMODE, @TXCSR ;ENTER MAINT MODE
2209 JSR R1, RFLG ;PUSH OUT A FLAG
2210 1 ;ONE FLAG
2211 JSR R1, SFLG ;PUSH OUT SPECIAL FLAGS
2212 4 ;THE # TO DO
2213 MOV #125, DATA ;CLEAR OUT CHAR
2214 MOV #8, SHIFTS ;LOAD THE # OF CLOCKS
2215 JSR R1, RPOKE ;PUSH OUT THE DATA
2216 BIC #BITW, @TXCSR ;CLEAR THE DATA WINDOW
2217 PKCLK 2 ;PUSH OUT THE BIT
2218 BIT #RXDONE, @RXCSR ;CHECK FOR DONE
2219 BNE SS ;BR IF SET
2220 HLT ;DONE BIT ERROR
2221 MOV @RXDBUF, TEMP ;GET DATA
2222 CMPB #125, TEMP ;CHECK IT
2223 BEQ 64$ ;BR IF A MATCH
2224 HLT 10 ;DATA COMPARE ERROR
2225 BIS #BITW, @TXCSR ;PUT A 1 IN THE WINDOW
2226 PKCLK 2 ;PUSH IT OUT
2227 INC TEMP1 ;INC THE # TO DO
2228 CMP #6, TEMP1 ;CHECK FOR FINISH
2229 BNE 64$ ;BR IF MORE TO GO
2230 BIC #BITW, @TXCSR ;CLEAR THE WINDOW
2231 PKCLK, 4 ;PUSH OUT 2 BINTS
2232 MOV @RXCSR, TEMP2 ;GET THE CSR
2233 BIT #REOM, @RXDBUF ;TEST FOR END OF MSG
2234 BNE 1$ ;BR IF SET
2235 HLT 1 ;BIT FAILED TO SET
2236 BIT #RXDONE, TEMP2 ;TEST DONE
2237 BNE 4$ ;BR IF SET
2238 HLT 2 ;DONE FAILED TO SET
2239
2240 4$: SCOPE ;SCOPE THIS TEST
2241
2242 :***** TEST 5 *****
2243 ;*TEST TO PROVE THE INTERACTION OF REOM
2244 ;*WITH DONE IN SECONDARY MODE. TEST FOR REOM
2245 ;*AT THE WRONG ADDRESS, THEN AT THE CORRECT
2246 ;*SECONDARY ADDRESS.
  
```

TEST OF REOM AND DONE IN SECONDARY MODE

```

2247
2248
2249
2250
2251
2252
2253
2254 010532 012737 000005 001226
2255 010540 012737 011140 001216
2256 010546 052777 000400 170636
2257 010554 004737 005044
2258 010560 052777 000020 170616
2259 010566 052777 001000 170614
2260 010574 005037 001236
2261 010600 052777 010000 170602
2262 010606 052777 014000 170576
2263 010614 004137 007060
2264 010620 000001
2265 010622 012737 000001 001324
2266 010630 012737 000010 001326
2267 010636 004137 007020
2268 010642 042777 002000 170542
2269 010650 104412 000002
2270 010654 032777 000200 170522
2271 010662 001401
2272 010664 104000
2273 010666 105777 170514
2274 010672 052777 002000 170512
2275 010700 104412 000002
2276 010704 005237 001236
2277 010710 022737 000006 001236
2278 010716 001365
2279 010720 042777 002000 170464
2280 010726 104412 000004
2281 010732 032777 001000 170446
2282 010740 001401
2283 010742 104004
2284
2285 010744 004137 007060
2286 010750 000001
2287 010752 005037 001324
2288 010756 012737 000010 001326
2289 010764 004137 007020
2290 010770 012737 000252 001324
2291 010776 012737 000010 001326
2292 011004 004137 007020
2293 011010 042777 002000 170374
2294 011016 104412 000002
2295 011022 032777 000200 170354
2296 011030 001001
2297 011032 104002
2298 011034 005037 001236
2299 011040 105777 170342
2300 011044 052777 002000 170340
2301 011052 104412 000002
2302 011056 005237 001236

```

```

*****
:*****
:*****
:*****
:*****
*****
TESTS:  MOV    #5, @TSTNO
        MOV    @TST6, NEXT
        BIS    #MRESET, @TXCSR ; RESET THE DEVICE
        JSR    PC, SMALL      ; WAIT FOR RESET TO FINISH
        BIS    #RCVEN, @RXCSR ; TURN ON RECEIVER
        BIS    #CRCEN, @PARCSR ; TURN OFF CRC
        CLR    TEMP1         ; BIT COUNTER
        BIS    #PRISEC, @PARCSR ; ENTER SECONDARY MODE
        BIS    #MMODE, @TXCSR ; ENTER MAINT MODE
        JSR    R1, RFLG      ; PUSH OUT A FLAG
        1                    ; ONE FLAG
        MOV    #1, DATA     ; LOAD A CHAR
        MOV    #8, SHIFTS    ; LOAD THE # OF CLOCKS
        JSR    R1, RPOKE     ; PUSH OUT THE DATA
        BIC    #BITW, @TXCSR ; CLEAR THE DATA WINDOW
        PKCLK 2              ; PUSH OUT THE BIT
        BIT    #RXDONE, @RXCSR ; CHECK FOR DONE
        BEQ    $$            ; BR IF CLEAR
        HLT
        5$: TSTB  @RXDBUF    ; DONE BIT ERROR
        64$: BIS  #BITW, @TXCSR ; READ THE BUFFER
        PKCLK 2              ; PUT A 1 IN THE WINDOW
        INC   TEMP1          ; PUSH IT OUT
        CMP   #6, TEMP1     ; INC THE # TO DO
        BNE  64$            ; CHECK FOR FINISH
        BIC  #BITW, @TXCSR  ; BR IF MORE TO GO
        PKCLK 4              ; CLEAR THE WINDOW
        BIT  #REOM, @RXDBUF ; PUSH OUT 2 BINTS
        BEQ  2$             ; TEST REC END OF MSG
        HLT  4              ; BR IF NOT SET
        2$: JSR  R1, RFLG    ; BIT IS SET AND SHOULDN'T
        1                    ; BE - THIS IS SECONDARY MODE
        CLR  DATA           ; OUTPUT A FLAG
        MOV  #8, SHIFTS     ; CLEAR DATA
        JSR  R1, RPOKE      ; LOAD THE # OF CLOCKS
        MOV  #252, DATA    ; PUSHOUT SEC. ADRS
        MOV  #8, SHIFTS    ; LOAD DATA
        JSR  R1, RPOKE     ; LOAD # OF CLOCKS
        BIC  #BITW, @TXCSR ; PUSHOUT DATA
        PKCLK 2            ; CLEAR OUT DATA WINDOW
        BIT  #RXDONE, @RXCSR ; PUSH OUT A BIT
        BNE  6$            ; CHECK DONE
        HLT  2             ; BR IF SET
        CLR  TEMP1         ; DONE FAILED TO SET
        TSTB @RXDBUF       ; CLEAR TO KEEP TRACK OF THE BITS
        BIS  #BITW, @TXCSR ; READ THE BUFFER
        PKCLK 2            ; SET THE WINDOW BIT
        INC  TEMP1          ; PUSH IT OUT
        65$:
        65$:
        INC  TEMP1          ; CHECK TO SEE IF

```

```

2303 011062 022737 000006 001236      CMP      #6,TEMP1      ;6 ARE DONE YET
2304 011070 001365      BNE      65$           ;BR IF NO
2305 011072 042777 002000 170312      BIC      #BITW,@TXCSR  ;CLEAR THE WINDOW
2306 011100 104412 000004      PKCLK,   4            ;PUSH 2 BITS
2307 011104 017737 170274 006054      MOV      @RXCSR,TEMP  ;READ THE CSR
2308 011112 032777 001000 170266      BIT      #REOM,@RXDBUF ;TEST END OF MSG
2309 011120 001001      BNE      3$           ;BR IF SET
2310 011122 104001      HLT      1            ;EOM FAILED TO SET IN SEC. MODE
2311 011124 032737 000200 001240 3$:      BIT      #RXDONE,TEMP2 ;TEST DONE
2312 011132 001001      BNE      4$           ;BR IF SET
2313 011134 104002      HLT      2            ;BIT FAILED TO SET
2314
2315 011136 104400      4$:      SCOPE              ;SCOPE THIS TEST
2316
2317      ;***** TEST 6 *****
2318      ;*TEST TO PROVE THE INTERACTION OF ABORT
2319      ;*WITH A DONE AND RX ERROR IN PRIMARY MODE
2320
2321      ;*****
2322      ;*****
2323      ;*****
2324      ;*****
2325      ;*****
2326      ;*****
2327      ;*****
2328 011140 012737 000006 001226  TST6:  MOV      #6,@TSTNO
2329 011146 012737 001352 001216      MOV      #TST7,NEXT
2330 011154 052777 000400 170230      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2331 011162 004737 005044      JSR      PC,SMALL     ;WAIT FOR RESET TO FINISH
2332 011166 052777 001000 170214      BIS      #RCEN,@PARCSR ;TURN OFF CRC
2333 011174 052777 000020 170202      BIS      #RCVEN,@RXCSR ;TURN ON RECEIVER
2334 011202 052777 014000 170202      BIS      #MMODE,@TXCSR ;ENTER MAINT MODE
2335 011210 004137 007060      JSR      R1,RFLG     ;PUSH OUT A FLAG
2336 011214 000001      1            ;ONE FLAG
2337 011216 005037 001324      CLR      DATA      ;CLEAR DATA
2338 011222 012737 000010 001326      MOV      #8,SHIFTS  ;LOAD CLOCKS
2339 011230 004137 007020      JSR      R1,RPOKE   ;PUSH OUT DATA CHAR
2340 011234 052777 002000 170150      BIS      #BITW,@TXCSR  ;SET THE WINDOW
2341 011242 104412 000002      PKCLK,   2            ;PUSH OUT A BIT
2342 011246 105777 170134      TSTB    @RXDBUF     ;READ A CHAR
2343 011252 112737 000177 001324      MOVB    #177,DATA   ;LOAD A SECOND CHAR
2344 011260 012737 000010 001326      MOV      #8,SHIFTS  ;LOAD CLOCKS
2345 011266 004137 007020      JSR      R1,RPOKE   ;PUSH OUT THE DATA
2346 011272 017737 170106 001236      MOV      @RXCSR,TEMP1 ;READ THE CSR
2347 011300 017700 170102      MOV      @RXDBUF,R0  ;SAVE THE BUFFER
2348 011304 032700 002000      BIT      #RABORT,R0  ;TEST ABORT BIT
2349 011310 001001      BNE      1$         ;BR IF SET
2350 011312 104005      HLT      5          ;ABORT BIT FAILED TO SET
2351 011314 032737 004000 001236 1$:      BIT      #REACT,TEMP1 ;TEST REC ACT
2352 011322 001401      BEQ     2$         ;BR IF CLEAR
2353 011324 104006      HLT      6          ;REC ACT. FAILED TO CLEAR
2354 011326 032737 000200 001236 2$:      BIT      #RXDONE,TEMP1 ;TEST DONE
2355 011334 001001      BNE      3$         ;BR IF SET
2356 011336 104002      HLT      2          ;DONE FAILED TO SET
2357 011340 032700 100000 3$:      BIT      #RXDERR,R0  ;TEST ERROR BIT
2358 011344 001001      BNE     12$        ;BR IF SET

```


TEST OF ABORT,DONE,AND RXERR IN PRIMARY MODE

```

2359 011346 104007 HLT 7 ;ERROR BIT FAILED TO SET AFTER ABORT
2360 011350 104400 12$: SCOPE ;SCOPE THIS TEST
2361 ;***** TEST 7 *****
2362 ;*TEST TO PROVE THE INTERACTION OF ABORT IN SECONDARY MODE
2363 ;*TEST FOR ABORT AT THE WRONG SECONDARY ADDRESS, THEN TEST
2364 ;*AT THE CORRECT SECONDARY ADDRESS
2365 ;*****
2366 ;*****
2367 ;*
2368 ; TEST 7
2369 ;*
2370 ;*****
2371 ;*****
2372 011352 012737 000007 001226 TST7: MOV #7, @TSTNO
2373 011360 012737 011710 001216 MOV #TST10, NEXT
2374 011366 052777 000400 170016 BIS #MRESET, @TXCSR ;RESET THE DEVICE
2375 011374 004737 005044 JSR PC, SMALL ;WAIT FOR RESET TO FINISH
2376 011400 052777 001000 170002 BIS #RCEN, @PARCSR ;TURN OFF CRC
2377 011406 052777 000020 167770 BIS #RCVEN, @RXCSR ;TURN ON RECEIVER
2378 011414 052777 010000 167766 BIS #PRISEC, @PARCSR ;ENTER SECONDARY MODE
2379 011422 052777 014000 167762 BIS #MMODE, @TXCSR ;ENTER MAINT MODE
2380 011430 004137 007060 JSR R1, RFLG ;PUSH OUT A FLAG
2381 011434 000001 1 JSR ;ONE FLAG
2382 011436 012737 000001 001324 MOV #1, DATA ;LOAD WRONG ADRS
2383 011444 012737 000010 001326 MOV #8, SHIFTS ;LOAD CLOCKS
2384 011452 004137 007020 JSR R1, RPOKE ;PUSH OUT DATA CHAR
2385 011456 052777 002000 167726 BIS #BITW, @TXCSR ;SET THE WINDOW
2386 011464 104412 000002 PKCLK, 2 ;PUSH OUT A BIT
2387 011470 105777 167712 TSTB @RXDBUF ;READ A CHAR
2388 011474 112737 000177 001324 MOVEB #177, DATA ;LOAD A SECOND CHAR
2389 011502 012737 000010 001326 MOV #8, SHIFTS ;LOAD CLOCKS
2390 011510 004137 007020 JSR R1, RPOKE ;PUSH OUT THE DATA
2391 011514 032777 002000 167664 BIT #RABORT, @RXDBUF ;TEST FOR ABORT
2392 011522 001401 BEQ 5$ ;BR IF NOT SET
2393 011524 104011 HLT 11 ;ABORT IS SET AND SHOULDN'T
2394 ;BE - THIS IS A SECONDARY STATION
2395 011526 004137 007060 5$: JSR R1, RFLG ;PUSH OUT A FLAG
2396 011532 000001 1 ;ONE FLAG
2397 011534 005037 001324 CLR DATA ;LOAD ADRS
2398 011540 012737 000010 001326 MOV #8, SHIFTS ;LOAD CLOCKS
2399 011546 004137 007020 JSR R1, RPOKE ;PUSH OUT THE ADRS
2400 011552 012737 000252 001324 MOV #252, DATA ;LOAD A CHAR
2401 011560 012737 000010 001326 MOV #8, SHIFTS ;LOAD CLOCKS
2402 011566 004137 007020 JSR R1, RPOKE ;PUSH OUT THE CHAR
2403 011572 052777 002000 167612 BIS #BITW, @TXCSR ;SET THE WINDOW
2404 011600 104412 000002 PKCLK, 2 ;PUSH OUT A BIT
2405 011604 105777 167576 TSTB @RXDBUF ;READ THE CHAR
2406 011610 112737 000177 001324 MOVEB #177, DATA ;LOAD THE ABORT
2407 011616 012737 000010 001326 MOV #8, SHIFTS ;LOAD CLOCKS
2408 011624 004137 007020 JSR R1, RPOKE ;PUSH OUT THE ABORT
2409 011630 017737 167550 001240 MOV @RXCSR, TEMP2 ;READ THE CSR
2410 011636 017700 167544 MOV @RXDBUF, RO ;SAVE THE BUFFER
2411 011642 032700 002000 BIT #RABORT, RO ;TEST THE ABORT BIT
2412 011646 001001 BNE 6$ ;BR IF SET
2413 011650 104005 HLT 5 ;BIT IS CLEAR - SHOULD BE SET
2414 011652 032737 004000 001240 6$: BIT #REACT, TEMP2 ;TEST ACTIVE

```

```

2415 011660 001401 BEQ 7$ ;BR IF CLEAR
2416 011662 104006 HLT 6 ;BIT FAILED TO CLEAR AFTER ABORT
2417 011664 032737 000200 001240 7$: BIT #RXDONE,TEMP2 ;TEST DONE
2418 011672 001001 BNE 10$ ;BR IF SET
2419 011674 104002 HLT 2 ;DONE FAILED TO SET
2420 011676 032700 100000 10$: BIT #RXDERR,RO ;TEST FOR ERROR
2421 011702 001001 BNE 12$ ;BR IF SET
2422 011704 104007 HLT 7 ;ERROR WASN'T SET AFTER ABORT
2423
2424 011706 104400 12$: SCOPE ;SCOPE THIS TEST
2425 ;***** TEST 10 *****
2426 ;*DATA OVERRUN TEST IN PRIMARY MODE. TEST TO
2427 ;*PROVE OVERRUN ERROR AND RX ERROR WILL OCCUR
2428 ;*****
2429
2430 ;*****
2431 ;*
2432 ;* TEST 10
2433 ;*
2434 ;*****
2435 ;*****
2436 011710 012737 000010 001226 TST10: MOV #10,#TSTNO
2437 011716 012737 012062 001216 MOV #TST11,NEXT
2438 011724 052777 000400 167460 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2439 011732 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2440 011736 013703 001404 MOV RXCSR,R3 ;LOAD THE CONTROL REGISTER
2441 011742 052777 001000 167440 BIS #CRCE#,@PARCSR ;TURN OFF CRC
2442 011750 052713 000020 BIS #RCVEN,(R3) ;TURN ON THE RECEIVER
2443 011754 052777 014000 167430 BIS #MMODE,@TXCSR ;ENTER MAINTENANCE MODE
2444 011762 004137 007060 JSR R1,RFLG ;PUSH OUT A FLAG
2445 011766 000001 1 MOV #1,ONE FLAG
2446 011770 012737 000252 001324 MOV #252,DATA ;LOAD CHAR TO BE OUTPUT
2447 011776 012737 000010 001326 MOV #8,SHIFTS ;CLOCK SETUP
2448 012004 004137 007020 JSR R1,RPOKE ;PUSH DATA INTO RECEIVER
2449 012010 012737 000070 001324 MOV #70,DATA ;LOAD A SECOND CHARACTER
2450 012016 012737 000011 001326 MOV #9,SHIFTS ;SETUP #OF CLOCKS
2451 012024 004137 007020 JSR R1,RPOKE ;PUSH THE SECOND CHARACTER INTO RX
2452 012030 017700 167352 MOV @RXDBUF,RO ;SAVE BUFFER
2453 012034 042700 037777 BIC #C<RXDERR!OVRUN>,RO ;CLEAR JUNK
2454 012040 022700 140000 CMP #RXDERR!OVRUN,RO ;CHECK TO SEE IF BOTH ARE SET
2455 012044 001401 BEQ 2$ ;BR IF OK
2456 012046 104013 HLT 13 ;THEY DIDN'T MATCH
2457
2458 012050 032713 004000 2$: BIT #REACT,(R3) ;TEST REC. ACT
2459 012054 001001 BNE 3$ ;BR IF STILL SET
2460 012056 104014 HLT 14 ;REC. ACTIVE CLEARED AND SHOULD BE SET
2461
2462 012060 104400 3$: SCOPE ;SCOPE THIS TEST
2463
2464 ;***** TEST 11 *****
2465 ;*DATA OVERRUN TEST IN SECONDARY MODE. TEST TO PROVE
2466 ;*THAT OVERRUN DOES NOT OCCUR IF THIS STATION IS NOT
2467 ;*SELECTED. THEN SELECT THIS SECONDARY STATION AND
2468 ;*PROVE OVERRUN ERROR AND RX ERROR WILL OCCUR
2469 ;*****
2470

```

2471
2472
2473
2474
2475
2476
2477
2478
2479
2480
2481
2482
2483
2484
2485
2486
2487
2488
2489
2490
2491
2492
2493
2494
2495
2496
2497
2498
2499
2500
2501
2502
2503
2504
2505
2506
2507
2508
2509
2510
2511
2512
2513
2514
2515
2516
2517
2518
2519
2520
2521
2522
2523
2524
2525
2526

```

:*****
: TEST 11
:*****
:*****
TST11: MOV #11, @TSTNO
MOV #TST12, NEXT
BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
MOV RXCSR, R3 ; LOAD THE CONTROL REGISTER
BIS #CRCEN, @PARCSR ; TURN OFF CRC
BIS #RCVEN, (R3) ; TURN ON THE RECEIVER
BIS #PRISEC, @PARCSR ; ENTER SECONDARY MODE
BIS #MMODE, @TXCSR ; ENTER MAINTENANCE MODE
JSR R1, RFLG ; PUSH OUT A FLAG
1 ; ONE FLAG
MOV #252, DATA ; LOAD AN INCORRECT SEC. STA ADRS
MOV #25, SHIFTS ; PUSH OUT 3 CHARS
JSR R1, RPOKE ; THRU THE RECEIVER
BIT #OVRUN, @RXDBUF ; TEST FOR OVERRUN
BEQ 4$ ; BR IF NOT SET

2494 012200 104015 HLT 15 ; OVERRUN IS SET AND SHOULDN'T BE
; THIS IS A SECONDARY STATION WHICH
; DID NOT GET ITS ADDRESS
4$: JSR R1, RFLG ; PUSH OUT A FLAG
1 ; ONE FLAG
CLR DATA ; GET CORRECT SEC. STATION ADRS
MOV #8, SHIFTS ; MOVE THE #OF CLOCKS TO PUSH
JSR R1, RPOKE ; PUSH OUT THE SEC. ADRS
MOV #252, DATA ; PUSH OUT
MOV #8, SHIFTS ; TWO
JSR R1, RPOKE ; DATA
MOV #70, DATA ; CHARACTERS
MOV #9, SHIFTS ; TO FORCE
JSR R1, RPOKE ; AN OVERRUN ERROR
MOV @RXDBUF, RO ; SAVE
BIC #1<RXDERR!OVRUN>, RO ; CLEAR JUNK
CMP #RXDERR!OVRUN, RO ; ARE THE ERROR BITS THERE
BEQ 6$ ; BR IF YES
HLT 13 ; ERROR BITS MISSING

2514 012306 032713 004000 6$: BIT #REACT, (R3) ; TEST ACTIVE
2515 012312 001001 3$: BNE 3$ ; BR IF SET
2516 012314 104005 HLT 5 ; ACTIVE DROPPED OUT
2517 012316 104400 3$: SCOPE ; SCOPE THIS TEST

:***** TEST 12 *****
: *TEST OF SPECIFIC DATA PATTERNS THRU
: *THE RECEIVER IN PRIMARY MODE
:*****
:*****
: TEST 12
:

```

```

2527      ::*****
2528      ::*****
2529      012320 012737 000012 001226 1ST12:  MOV    #12, #TSTNO
2530      012326 012737 012570 001216      MOV    #TST13, NEXT
2531      012334 052777 000400 167050      BIS    #MRESET, #TXCSR ; RESET THE DEVICE
2532      012342 004737 005044      JSR    PC, SMALL ; WAIT FOR RESET TO FINISH
2533      012346 012703 012556      MOV    #TBLA, R3 ; LOAD THE TABLE POINTER
2534      012352 012704 012556      MOV    #TBLA, R4 ; DITTO
2535      012356 012702 000004      MOV    #4, R2 ; LOAD THE # OF CHARS TO DO
2536      012362 052777 001000 167020      BIS    #CRCEN, #PARCSR
2537      012370 005077 167020      CLR    #TXDBUF ; RESET TXDONE
2538      012374 052777 014000 167010      BIS    #MMODE, #TXCSR ; ENTER M/MODE
2539      012402 052777 000020 166774      BIS    #RCVEN, #RXCSR ; TURN ON THE RECEIVER
2540      012410 052777 000020 166774      BIS    #SEND, #TXCSR ; TURN ON THE TRANSMITTER
2541      012416 052777 000400 166770      BIS    #TSOM, #TXDBUF ; START A FLAG
2542      012424 104412 000010      PKCLK #8 ; START A FLAG
2543      012430 012377 166760      MOV    (R3)+, #TXDBUF ; LOAD FIRST DATA CHAR
2544      012434 005302      DEC    R2 ; LOWER THE # TO DO
2545      012436 104412 000020      PKCLK #16 ; PUSH OUT
2546      012442 012377 166746 1$:  MOV    (R3)+, #TXDBUF ; LOAD DATA
2547      012446 005302      DEC    R2 ; LOWER THE # TO DO
2548      012450 104412 000020      PKCLK #16 ; PUSH OUT DATA
2549      012454 105777 166724      TSTB  #RXCSR ; CHECK FOR RX DONE
2550      012460 100401      BMI   #2$ ; BR IF DONE SET
2551      012462 104000      HLT   ; RX DONE FAILED TO SET
2552      012464 017737 166716 001324 2$:  MOV    #RXDBUF, DATA ; GET THE BUFFER
2553      012472 121437 001324      CMPB  (R4), DATA ; CHECK IT
2554      012476 001401      BEQ   #3$ ; BR IF OK
2555      012500 104000      HLT   ; DATA COMPARE ERROR
2556      012502 005724 3$:  TST   (R4)+
2557      012504 005702      TST   R2 ; CHECK FOR FINISH
2558      012506 001355      BNE   #1$ ; BR IF MORE TO GO
2559      012510 104412 000020      PKCLK #16 ; PUSH OUT LAST CHAR
2560      012514 105777 166664      TSTB  #RXCSR ; CHECK FOR DONE
2561      012520 100401      BMI   #4$ ; BR IF DONE SET
2562      012522 104000      HLT   ; DONE FAILED TO SET
2563      012524 017737 166656 001324 4$:  MOV    #RXDBUF, DATA ; GET THE BUFFER
2564      012532 121437 001324      CMPB  (R4), DATA ; CHECK IT
2565      012536 001401      BEQ   #5$ ; BR IF A MATCH
2566      012540 104000      HLT   ; DATA COMPARE ERROR
2567      012542 5$:  HLT
2568      012542 052777 000400 166642      BIS    #MRESET, #TXCSR ; RESET THE DEVICE
2569      012550 004737 005044      JSR    PC, SMALL ; WAIT FOR RESET TO FINISH
2570      012554 104400      SCOPE
2571
2572      :DATA TABLE
2573      012556 000125 1BLA:  .WORD 125
2574      012560 000252      .WORD 252
2575      012562 000000      .WORD 000
2576      012564 000377      .WORD 377
2577      012566 000000      .WORD 000
2578
2579      :***** TEST 13 *****
2580      :*TEST TO PROVE THAT THE DEVICE WILL
2581      :*WORK WITH ALL POSSIBLE SECONDARY
2582      :*STATION ADDRESSES.

```

```

2583 ;:*****
2584 ;:*****
2585 ;:*****
2586 ;:*****
2587 ;:*****
2588 ;:*****
2589 ;:*****
2590 ;:*****
2591 012570 012737 000013 001226 TST13: MOV #13, @TSTNO
2592 012576 012737 013064 001216 MOV #TST14, NEXT
2593 012604 012737 012632 001220 MOV #20$, LOCK ; SW09 SETUP
2594 012612 005005 CLR R5 ; CLEAR SEC ADRS HOLD
2595 012614 005037 001236 CLR TEMP1 ; CLEAR TEMP STORAGE
2596 012620 005037 001240 CLR TEMP2 ; DITTO
2597 012624 012737 011000 001236 MOV #CRCEN!PRISEC, TEMP1 ; LOAD MODE AND NO CRC
2598 012632 20$:
2599 012632 052777 000400 166552 BIS #MRESET, @TXCSR ; RESET THE DEVICE
2600 012640 004737 005044 JSR PC, SMALL ; WAIT FOR RESET TO FINISH
2601 012644 013777 001236 166536 MOV TEMP1, @PARCSR ; LOAD MODE AND NO CRC AND SEC STATION
2602 012652 052777 000020 166524 1$: BIS #RCVEN, @RXCSR ; TURN ON RECEIVER
2603 012660 052777 004020 166524 BIS #SYSTST!SEND, @TXCSR ; TURN ON TRANSMITTER AND CLOCK
2604 012666 105777 166520 2$: TSTB @TXCSR ; WAIT FOR
2605 012672 100375 BPL 2$ ; DONE
2606 012674 012777 000400 166512 MOV #TSOM, @TXDBUF ; LOAD START OF MSG
2607 012702 105777 166504 3$: TSTB @TXCSR ; WAIT FOR
2608 012706 100375 BPL 3$ ; DONE AGAIN
2609 012710 013777 001240 166476 MOV TEMP2, @TXDBUF ; LOAD SEC STATION ADRS AND
2610 012716 105777 166470 4$: TSTB @TXCSR ; WAIT FOR
2611 012722 100375 BPL 4$ ; DONE AGAIN
2612 012724 012777 000252 166462 MOV #252, @TXDBUF ; NOW LOAD DATA
2613 012732 032777 004000 166454 64$: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
2614 012740 001374 BNE 64$ ; BR IF SET
2615 012742 032777 004000 166444 65$: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
2616 012750 001774 BEQ 65$ ; BR IF CLEAR
2617 012752 105777 166426 5$: TSTB @RXCSR ; TEST FOR
2618 012756 100375 BPL 5$ ; RX DONE
2619 012760 005777 166420 TST @RXCSR ; TEST FOR ERROR
2620 012764 100001 BPL 6$ ; BR IF NO ERROR
2621 012766 104014 HLT 14 ; ERROR FOUND!
2622 012770 017704 166412 63$: MOV @RXDBUF, R4 ; GET THE BUFFER
2623 012774 032704 000400 BIT #RSOM, R4 ; CHECK FOR START OF MSG
2624 013000 001001 BNE 7$ ; BR IF SET
2625 013002 104000 HLT ; START OF MSG FAILED TO SET
2626 013004 122704 000252 7$: CMPB #252, R4 ; CHECK FOR DATA
2627 013010 001401 BEQ 10$ ; BR IF A MATCH
2628 013012 104012 HLT 12 ; FAILED TO RECEIVE DATA AS A SEC STATION
2629 013014 104401 10$: SCOP1 ; SW09=1?
2630 013016 052777 000400 166366 BIS #MRESET, @TXCSR ; RESET THE DEVICE
2631 013024 004737 005044 JSR PC, SMALL ; WAIT FOR RESET TO FINISH
2632 013030 105205 INCB R5 ; GET NEXT SEC ADRS
2633 013032 001413 BEQ 11$ ; BR IF ALL DONE
2634 013034 110537 001236 MOVB R5, TEMP1 ; LOAD THE ADRS FOR PAR CSR
2635 013040 110537 001240 MOVB R5, TEMP2 ; DITTO FOR TXD BUF
2636 013044 052737 011000 001236 BIS #CRCEN!PRISEC, TEMP1 ; LOAD FOR PAR CSR
2637 013052 013777 001236 166330 MOV TEMP1, @PARCSR ; DO IT
2638 013060 000674 BP 1$ ; BR TO DO OVER

```

2639 013062 104400
2640
2641
2642
2643
2644
2645
2646
2647
2648
2649
2650
2651
2652 013064 012737 000014 001226
2653 013072 012737 013602 001216
2654 013100 012704 013410
2655 013104 012703 000004
2656 013110 012701 013410
2657 013114 012702 013410
2658 013120 005037 001324
2659 013124 012737 102010 013574
2660 013132 012737 177777 013600
2661 013140 013737 013600 013162
2662 013146 011437 013160
2663 013152 004537 013422
2664 013156 000010
2665 013160 000001
2666 013162 000001
2667 013164 052777 000400 166220
2668 013172 004737 005044
2669 013176 052777 014000 166206
2670 013204 052777 000020 166172
2671 013212 052777 000020 166172
2672 013220 052777 000400 166166
2673 013226 104412 000012
2674 013232 105777 166154
2675 013236 100401
2676 013240 104000
2677 013242
2678 013242 011177 166146
2679 013246 104412 000020
2680 013252 052777 001000 166134
2681 013260 104412 000020
2682 013264 105777 166114
2683 013270 100401
2684 013272 104000
2685 013274 017737 166106 001324
2686 013302 121237 001324
2687 013306 001401
2688 013310 104000
2689 013312 104412 000020
2690 013316 105777 166062
2691 013322 100401
2692 013324 104000
2693 013326 017737 166054 001324
2694 013334 104412 000032

```
11$: SCOPE ;SCOPE THIS TEST
;***** TEST 14 *****
; *TEST OF SPECIFIC CHARACTER DATA PATTERNS
; *USING BCC CHECK IN PRIMARY MODE.
;*****
;*****
; TEST 14
;*****
;*****
TST14: MOV #14, @TSTNO ;GET THE TABLE POINTER
MOV #TST15, NEXT ;GET THE # TO DO
MOV @TBLB, R4 ;GET THE POINTER
MOV #4, R3 ;DITTO
MOV @TBLB, R1 ;CLR DATA HOLD
MOV @TBLB, R2 ;CLR DATA HOLD
CLR DATA ;LOAD THE POLYNOMIAL FOR CRC
10$: MOV #CRC, CCITT, XPOLY ;PRESET BCC FOR SCLC
MOV #-1, CALBCC ;MOVE BCC
MOV CALBCC, 12$ ;MOVE DATA
MOV (R4), 11$ ;GO CALCULATE BCC
JSR R5, SIMBCC ;BASED
B. ;ON THESE
11$: .BLKW 1 ;PARAMETERS
12$: .BLKW 1 ;RESET THE DEVICE
BIS #MRESET, @TXCSR ;WAIT FOR RESET TO FINISH
JSR PC, SMALL ;ENTER MAINT MODE
BIS #MMODE, @TXCSR ;TURN ON THE RECEIVER
BIS #RCVEN, @RXCSR ;TURN ON TRANSMITTER
BIS #SEND, @TXCSR ;TURN ON START OF MSG
BIS #TSOM, @TXDBUF ;PUSH OUT 4 BITS
PKCLK .10. ;CHECK FOR DONE
TSTB @TXCSR ;BR IF SET
BMI 15 ;DONE FAILED TO SET
HLT
13$: MOV (R1), @TXDBUF ;LOAD DATA
PKCLK .16. ;PUSH OUT 8 BITS
BIS #TEOM, @TXDBUF ;SET END OF MSG
PKCLK .16. ;PUSH OUT 8 MORE BITS
TSTB @RXCSR ;CHECK FOR DONE
BMI 25 ;BR IF SET
HLT ;DONE FAILED TO SET
2$: MOV @RXDBUF, DATA ;READ THE BUFFER
CMPB (R2), DATA ;CHECK THE DATA
BEQ 3$ ;BR IF A MATCH
HLT ;DATA COMPARE ERROR
3$: PKCLK .16. ;PUSH OUT 8 MORE BITS
TSTB @RXCSR ;CHECK FOR DONE
BMI 4$ ;BR IF SET
HLT ;DONE FAILED TO SET ON FIRST OCTET OF BCC
4$: MOV @RXDBUF, DATA ;READ THE BUFFER
PKCLK .26. ;FINISH POKING BITS
```

2695	013340	105777	166040		TSTB	BRXCSR	:CHECK FOR DONE
2696	013344	100401			BMI	5\$:BR IF SET
2697	013346	104000			HLT		:DONE FAILED TO SET ON SECOND OCTET OF BCC
2698	013350	117737	166032	001325	5\$:	MOV B	:GET IT
2699	013356	005137	013600		COM	CALBCC	:INVERT SOFTWARE
2700	013362	023737	013600	001324	CMP	CALBCC,DATA	:COMPARE
2701	013370	001401			BEQ	6\$:BR IF A MATCH
2702	013372	104000			HLT		:BCC COMPARE FAILURE
2703							:USE CRC DEBUGGING ROUTINE TO
2704							:REPAIR CRC.SEE FRONT OF LISTING
2705	013374	005721			6\$:	TST	:POP POINTER
2706	013376	005722				TST	:DITTO
2707	013400	005724				TST	:DITTO
2708	013402	005303				DEC	:DECREMENT THE # TO DO
2709	013404	001247				BNE	:BR IF MORE TOGO
2710	013406	104400				SCOPE	:SCOPE THIS TEST
2711							
2712							
2713	013410	000252					
2714	013412	000000					
2715	013414	000125					
2716	013416	000377					
2717	013420	000000					
2718							
2719							
2720	013422	010046					
2721	013424	010146					
2722	013426	010246					
2723	013430	012537	001236				
2724	013434	012537	001240				
2725	013440	012537	001242				
2726	013444	005037	013576				
2727	013450	013700	001242				
2728	013454	006037	001240				
2729	013460	005500					
2730	013462	032700	000001				
2731	013466	001402					
2732	013470	005137	013576				
2733	013474	013700	013574				
2734	013500	005100					
2735	013502	040037	013576				
2736	013506	000241					
2737	013510	006037	001242				
2738	013514	013700	013576				
2739	013520	013701	001242				
2740	013524	010102					
2741	013526	040100					
2742	013530	043702	013576				
2743	013534	050200					
2744	013536	043737	013574	001242			
2745	013544	050037	001242				
2746	013550	005337	001236				
2747	013554	001333					
2748	013556	013737	001242	013600			
2749	013564	012602					
2750	013566	012601					

```

:DATA TABLE
†BLB: .WORD 252
      .WORD 0
      .WORD 125
      .WORD 377
      .WORD 0
  
```

```

SIMBCC: MOV R0,-(SP)
        MOV R1,-(SP)
        MOV R2,-(SP)
        MOV (R5)+,TEMP1
        MOV (R5)+,TEMP2
        MOV (R5)+,TEMP3
1$: CLR BCCFBK
     MOV TEMP3,R0
     ROR TEMP2
     ADC R0
     BIT #BIT0,R0
     BEQ 2$
     COM BCCFBK
2$: MOV XPOLY,R0
     COM R0
     BIC R0,BCCFBK
     CLC
     ROR TEMP3
     MOV BCCFBK,R0
     MOV TEMP3,R1
     MOV R1,R2
     BIC R1,R0
     BIC BCCFBK,R2
     BIS R2,R0
     BIC XPOLY,TEMP3
     BIS R0,TEMP3
     DEC TEMP1
     BNE 1$
     MOV TEMP3,CALBCC
     MOV (SP)+,R2
     MOV (SP)+,R1
  
```

2751 013570 012600
2752 013572 000205
2753 013574 000000
2754 013576 000000
2755 013600 000000
2756 120001
2757 102010

MOV (SP)+,R0
RTS R5
XPOLY: 0
BCCFBK: 0
CALBCC: 0
CRC16=120001
CRC.CCITT=102010

***** TEST 15 *****
; THIS TEST PROVES THAT THE DEVICE WILL
; OUTPUT SIXTEEN SPACES IN FRONT OF A
; FLAG IF STARTING FROM THE IDLE STATE.

; TEST 15

2771 *****
2772 013602 012737 000015 001226
2773 013610 012737 014110 001216
2774 013616 052777 000400 165566
2775 013624 004737 005044
2776 013630 005037 001236
2777 013634 005037 001240
2778 013640 005037 001242
2779 013644 005077 165544
2780 013650 052777 014000 165534
2781 013656 052777 001000 165524
2782 013664 052777 000020 165520
2783 013672 052777 000020 165504
2784 013700 052777 001400 165506
2785 013706 104412 000002
2786 013712 005237 001236 1\$:
2787 013716 022737 000021 001236
2788 013724 001407
2789 013726 104412 000002
2790 013732 032777 040000 165452
2791 013740 001764
2792 013742 104000
2793 013744 104412 000002 2\$:
2794 013750 032777 040000 165434
2795 013756 001401
2796 013760 104000
2797 013762 012777 000125 165424 3\$:
2798 013770 104412 000002 4\$:
2799 013774 032777 040000 165410
2800 014002 001001
2801 014004 104000
2802 014006 005237 001240 5\$:
2803 014012 022737 000006 001240
2804 014020 001363
2805 014022 104412 000002
2806 014026 032777 040000 165356

1ST15: MOV #15, @TSTNO
MOV #TST16, NEXT
BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
CLR TEMP1 ; CLEAR FOR SOFTWARE STORAGE
CLR TEMP2
CLR TEMP3
CLR @TXDBUF ; RESET TXDONE
BIS #MMODE, @TXCSR ; ENTER MAINT MODE
BIS #CRCEN, @PARCSR ; TURN OFF CRC
BIS #SEND, @TXCSR ; TURN ON TRANSMITTER
BIS #RCVEN, @RXCSR ; TURN ON RECEIVER
BIS #TEOM!T\$OM, @TXDBUF ; START SPECIAL SEQUENCE
PKCLK 2 ; SYNC UP DUP
1\$: INC TEMP1 ; UPDATE COUNT
CMP #17., TEMP1 ; CHECK FOR FINISH
BEQ 2\$; BR IF YES
PKCLK 2 ; POKE CLOCK
BIT #MTDATA, @TXCSR ; CHECK THE DATA
BEQ 1\$; BR IF OK
HLT ; DATA SHOULD BE A 0, WAS A 1
2\$: PKCLK 2 ; POKE CLOCK
BIT #MTDATA, @TXCSR ; CHECK THE DATA
BEQ 3\$; BR IF OK
HLT ; DATA SHOULD BE 0, WAS A ONE
3\$: MOV #125, @TXDBUF ; LOAD DATA CHAR
4\$: PKCLK 2 ; POKE CLOCK
BIT #MTDATA, @TXCSR ; CHECK THE BIT
BNE 5\$; BR IF OK
HLT ; DATA SHOULD BE 1, WAS 0
5\$: INC TEMP2 ; UPDATE FOR FINISH
CMP #6, TEMP2 ; CHECK FOR DONE
BNE 4\$; BR IF MORE TO GO
PKCLK 2 ; POKE CLOCK
BIT #MTDATA, @TXCSR ; CHECK THE DATA


```

2807 014034 001401 BEQ 6$ ;BR IF OK
2808 014036 104000 HLT ;DATA SHOULD BE 0, WAS A 1
2809 014040 104412 000006 165340 6$: PKCLK 6 ;START OUT DATA CHAR
2810 014044 052777 001000 BIS #TEOM,@TXCSR ;TURN OFF TRANSMITTER
2811 014052 104412 000016 PKCLK 14 ;FINISH
2812 014056 105777 165322 TSTB @RXCSR ;CHECK RECEIVER
2813 014062 100401 BMI 7$ ;BR IF OK
2814 014064 104000 HLT ;RECEIVER FAILED TO ACCEPT SPECIAL CHAR
2815 014066 117737 165314 001242 7$: MOVB @RXDBUF,TEMP3 ;GET THE CHAR
2816 014074 122737 000125 001242 CMPB #125,TEMP3 ;CHECK IT
2817 014102 001401 BEQ 10$
2818 014104 104000 HLT ;DATA FAILED TO MATCH AFTER
2819 ;SPECIAL SPACES CHARACTER
2820 014106 104400 10$: SCOPE ;SCOPE THIS TEST
2821
2822
2823
2824 ;***** TEST 16 *****
2825 ;*THIS TEST PROVES THE INTERACTION OF DTR
2826 ;*WITH RING,DSR
2827 ;*AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
2828 ;*SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
2829 ;*THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.
2830 ;:*****

```

```

:*****
:
: TEST 16
:
:*****

```

```

2831
2832
2833
2834
2835
2836 ;*****
2837 014110 012737 000016 001226 TST16: MOV #16,@TSTNO
2838 014116 012737 014556 001216 MOV #TST17,NEXT
2839 014124 012737 014204 001220 MOV #15,LOCK
2840 014132 105737 001322 TSTB TCNFLG
2841 014136 001002 BNE .+6
2842 014140 000137 014544 10$: JMP 6$
2843 014144 005077 165234 CLR @RXCSR ;CLEAR THE REGISTER
2844 014150 004137 007332 JSR R1,OJUMPER ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
2845 014154 014546 7$ ;AND OPTIONAL JUMPER ARE USED
2846 ;AND LOADS R5 (EXPECTED) ACCORDINGLY.
2847 014156 052777 000400 165226 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2848 014164 004737 005044 JSR PC_SMALL ;WAIT FOR RESET TO FINISH
2849 014170 013703 001404 MOV RXCSR,R3 ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
2850 014174 005013 CLR (R3) ;CLEAR OUT EXTRA BITS
2851 014176 052777 010000 165206 BIS #MEXT,@TXCSR ;ENTER EXTERNAL MAINT. MODE
2852 014204 052713 000002 1$: BIS #DTR,(R3) ;TURN ON DTR
2853 014210 012737 000110 014240 MOV #110,68$ ;LOAD THE NUMBER
2854 014216 032777 004000 165170 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2855 014224 001374 BNE 66$ ;BR IF SET
2856 014226 032777 004000 165160 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2857 014234 001774 BEQ 67$ ;BR IF CLEAR
2858 014236 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2859 014240 000110 68$: 110 ;OF TIMES TO REPEAT
2860 014242 001365 BNE 66$ ;BR IF MORE TO GO
2861 014244 011304 MOV (R3),R4 ;GET THE BITS FROM THE RXCSR
2862 014246 020504 CMP R5,R4 ;R5=GOOD R4=?

```

2863	014250	001423				BEQ	2\$;BRANCH IF THEY MATCH
2864	014252	104003				HLT	3		;NO MATCH - SHOW OPR.
2865	014254	104401				SCOP1			
2866	014256	012737	000005	014306		MOV	#5,73\$;LOAD THE NUMBER
2867	014264	032777	004000	165122	71\$:	BIT	#TIMER,@TXDBUF		;CHECK THE TIMER BIT
2868	014272	001374				BNE	71\$;BR IF SET
2869	014274	032777	004000	165112	72\$:	BIT	#TIMER,@TXDBUF		;CHECK THE BIT
2870	014302	001774				BEQ	72\$;BR IF CLEAR
2871	014304	005327				DEC	(PC)+		;DECREMENT THE NUMBER
2872	014306	000005			73\$:	5			;OF TIMES TO REPEAT
2873	014310	001365				BNE	71\$;BR IF MORE TO GO
2874	014312	032713	040000			BIT	#RING,(R3)		
2875	014316	001000				BNE	2\$		
2876	014320	012737	014332	001220	2\$:	MOV	#3\$,LOCK		;SW09 SETUP
2877	014326	042705	041C02			BIC	#RING!DSR!DTR,R5		;CLEAR OUT UNWANTED BITS
2878	014332	005013			3\$:	CLR	(R3)		;CLEAR OUT THE REGISTER
2879	014334	012737	000005	014364		MOV	#5,78\$;LOAD THE NUMBER
2880	014342	032777	004000	165044	76\$:	BIT	#TIMER,@TXDBUF		;CHECK THE TIMER BIT
2881	014350	001374				BNE	76\$;BR IF SET
2882	014352	032777	004000	165034	77\$:	BIT	#TIMER,@TXDBUF		;CHECK THE BIT
2883	014360	001774				BEQ	77\$;BR IF CLEAR
2884	014362	005327				DEC	(PC)+		;DECREMENT THE NUMBER
2885	014364	000005			78\$:	5			;OF TIMES TO REPEAT
2886	014366	001365				BNE	76\$;BR IF MORE TO GO
2887	014370	011304				MOV	(R3),R4		;READ BACK THE REGISTER
2888	014372	020504				CMP	R5,R4		;R5=GOOD R4=?
2889	014374	001402				BEQ	4\$;BRANCH IF ONLY THE DSC BITS ARE SET
2890	014376	104003				HLT	3		;NO-GO TELL OPR
2891	014400	104401				SCOP1			
2892	014402	012737	014410	001220	4\$:	MOV	#5\$,LOCK		;SW09 SETUP
2893	014410	052713	000002		5\$:	BIS	#DTR,(R3)		;TURN ON DTR
2894	014414	012737	000005	014444		MOV	#5,83\$;LOAD THE NUMBER
2895	014422	032777	004000	164764	81\$:	BIT	#TIMER,@TXDBUF		;CHECK THE TIMER BIT
2896	014430	001374				BNE	81\$;BR IF SET
2897	014432	032777	004000	164754	82\$:	BIT	#TIMER,@TXDBUF		;CHECK THE BIT
2898	014440	001774				BEQ	82\$;BR IF CLEAR
2899	014442	005327				DEC	(PC)+		;DECREMENT THE NUMBER
2900	014444	000005			83\$:	5			;OF TIMES TO REPEAT
2901	014446	001365				BNE	81\$;BR IF MORE TO GO
2902	014450	005005				CLR	R5		;CLEAR OUT EXPECTED
2903	014452	005013				CLR	(R3)		;CLEAR OUT THE REGISTER
2904	014454	052777	000400	164730		BIS	#MRESET,@TXCSR		;RESET THE DEVICE
2905	014462	004737	005044			JSR	PC,SMALL		;WAIT FOR RESET TO FINISH
2906	014466	052777	010000	164716		BIS	#MEXT,@TXCSR		;TURN ON EXTERNAL MODE
2907	014474	012737	000005	014524		MOV	#5,88\$;LOAD THE NUMBER
2908	014502	032777	004000	164704	86\$:	BIT	#TIMER,@TXDBUF		;CHECK THE TIMER BIT
2909	014510	001374				BNE	86\$;BR IF SET
2910	014512	032777	004000	164674	87\$:	BIT	#TIMER,@TXDBUF		;CHECK THE BIT
2911	014520	001774				BEQ	87\$;BR IF CLEAR
2912	014522	005327				DEC	(PC)+		;DECREMENT THE NUMBER
2913	014524	000005			88\$:	5			;OF TIMES TO REPEAT
2914	014526	001365				BNE	86\$;BR IF MORE TO GO
2915	014530	005713				TST	(R3)		;STRIP DSCA & DSCB FROM CSR
2916	014532	011304				MOV	(R3),R4		;GET THE REGISTER
2917	014534	020504				CMP	R5,R4		;R5=GOOD,R4=?
2918	014536	001402				BEQ	6\$;BR IF OK

2919 014540 104003
 2920 014542 104401
 2921
 2922 014544 104400
 2923 014546 141003
 2924 014550 141001
 2925 014552 001002
 2926 014554 000000
 2927
 2928
 2929
 2930
 2931
 2932
 2933
 2934
 2935
 2936
 2937
 2938
 2939
 2940
 2941
 2942
 2943 014556 012737 000017 001226
 2944 014564 012737 015224 001216
 2945 014572 012737 014652 001220
 2946 014600 105737 001322
 2947 014604 001002
 2948 014606 000137 015212
 2949 014612 005077 164566
 2950 014616 004137 007332
 2951 014622 015214
 2952
 2953 014624 052777 000400 164560
 2954 014632 004737 005044
 2955 014636 013703 001404
 2956 014642 005013
 2957 014644 052777 010000 164540
 2958 014652 052713 000004 15:
 2959 014656 012737 000110 014706
 2960 014664 032777 004000 164522 66\$:
 2961 014672 001374
 2962 014674 032777 004000 164512 67\$:
 2963 014702 001774
 2964 014704 005327
 2965 014706 000110 68\$:
 2966 014710 001365
 2967 014712 011304
 2968 014714 020504
 2969 014716 001423
 2970 014720 104003
 2971 014722 104401
 2972 014724 012737 000005 014754
 2973 014732 032777 004000 164454 71\$:
 2974 014740 001374

HLT 3 ;REPORT THE ERROR
 SCOP1 ;SW09=1?
 6\$: SCOPE ;SCOPE THE WHOLE TEST
 7\$: .WORD 141003
 .WORD 141001
 .WORD 1002
 .WORD 0

```

:***** TEST 17 *****
:*THIS TEST PROVES THE INTERACTION OF RTS
:*WITH CTS,CARDET
:*AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
:*SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
:*THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.
:*****
    
```

```

:*****
: TEST 17
:*****
:*****
↑ST17: MOV #17,@TSTNO
MOV #TST20,NEXT
MOV #15,LOCK
TSTB TCNPLG
BNE .+6
10$: JMP 6$
CLR @RXCSR ;CLEAR THE REGISTER
JSR R1,0JUMPER ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
7$ ;AND OPTIONAL JUMPER ARE USED
;AND LOADS R5 (EXPECTED) ACCORDINGLY.
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
MOV RXCSR,R3 ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
CLR (R3) ;CLEAR OUT EXTRA BITS
BIS #NEXT,@TXCSR ;ENTER EXTERNAL MAINT. MODE
15: BIS #RTS,(R3) ;TURN ON RTS
MOV #110,68$ ;LOAD THE NUMBER
66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
BNE 66$ ;BR IF SET
67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
BEQ 67$ ;BR IF CLEAR
DEC (PC)+ ;DECREMENT THE NUMBER
68$: 110 ;OF TIMES TO REPEAT
BNE 66$ ;BR IF MORE TO GO
MOV (R3),R4 ;GET THE BITS FROM THE RXCSR
CMP R5,R4 ;R5=GOOD R4=?
BEQ 2$ ;BRANCH IF THEY MATCH
HLT 3 ;NO MATCH - SHOW OPR.
SCOP1
MOV #5,73$ ;LOAD THE NUMBER
71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
BNE 71$ ;BR IF SET
    
```

2975	014742	032777	004000	164444	72\$:	BIT	#TIMER,@TXDBUF	:CHECK THE BIT
2976	014750	001774				BEQ	72\$:BR IF CLEAR
2977	014752	005327				DEC	(PC)+	:DECREMENT THE NUMBER
2978	014754	000005			73\$:	S		:OF TIMES TO REPEAT
2979	014756	001365				BNE	71\$:BR IF MORE TO GO
2980	014760	032713	040000			BIT	#RING,(R3)	
2981	014764	001000				BNE	2\$	
2982	014766	012737	015000	001220	2\$:	MOV	#3\$ LOCK	:SW09 SETUP
2983	014774	042705	030004			BIC	#CTS!CARDET!RTS,R5	:CLEAR OUT UNWANTED BITS
2984	015000	005013			3\$:	CLR	(R3)	:CLEAR OUT THE REGISTER
2985	015002	012737	000005	015032		MOV	#5,78\$:LOAD THE NUMBER
2986	015010	032777	004000	164376	76\$:	BIT	#TIMER,@TXDBUF	:CHECK THE TIMER BIT
2987	015016	001374				BNE	76\$:BR IF SET
2988	015020	032777	004000	164366	77\$:	BIT	#TIMER,@TXDBUF	:CHECK THE BIT
2989	015026	001774				BEQ	77\$:BR IF CLEAR
2990	015030	005327				DEC	(PC)+	:DECREMENT THE NUMBER
2991	015032	000005			78\$:	S		:OF TIMES TO REPEAT
2992	015034	001365				BNE	76\$:BR IF MORE TO GO
2993	015036	011304				MOV	(R3),R4	:READ BACK THE REGISTER
2994	015040	020504				CMP	R5,R4	:R5=GOOD,R4=?
2995	015042	001402				BEQ	4\$:BRANCH IF ONLY THE DSC BITS ARE SET
2996	015044	104003				HLT	3	:NO-GO TELL OPR
2997	015046	104401				SCOPI		
2998	015050	012737	015056	001220	4\$:	MOV	#5\$ LOCK	:SW09 SETUP
2999	015056	052713	000004		5\$:	BIS	#RTS,(R3)	:TURN ON RTS
3000	015062	012737	000005	015112		MOV	#5,83\$:LOAD THE NUMBER
3001	015070	032777	004000	164316	81\$:	BIT	#TIMER,@TXDBUF	:CHECK THE TIMER BIT
3002	015076	001374				BNE	81\$:BR IF SET
3003	015100	032777	004000	164306	82\$:	BIT	#TIMER,@TXDBUF	:CHECK THE BIT
3004	015106	001774				BEQ	82\$:BR IF CLEAR
3005	015110	005327				DEC	(PC)+	:DECREMENT THE NUMBER
3006	015112	000005			83\$:	S		:OF TIMES TO REPEAT
3007	015114	001365				BNE	81\$:BR IF MORE TO GO
3008	015116	005005				CLR	R5	:CLEAR OUT EXPECTED
3009	015120	005013				CLR	(R3)	:CLEAR OUT THE REGISTER
3010	015122	052777	000400	164262		BIS	#MRESET,@TXCSR	:RESET THE DEVICE
3011	015130	004737	005044			JSR	PC,SMALL	:WAIT FOR RESET TO FINISH
3012	015134	052777	010000	164250		BIS	#NEXT,@TXCSR	:TURN ON EXTERNAL MODE
3013	015142	012737	000005	015172		MOV	#5,88\$:LOAD THE NUMBER
3014	015150	032777	004000	164236	86\$:	BIT	#TIMER,@TXDBUF	:CHECK THE TIMER BIT
3015	015156	001374				BNE	86\$:BR IF SET
3016	015160	032777	004000	164226	87\$:	BIT	#TIMER,@TXDBUF	:CHECK THE BIT
3017	015166	001774				BEQ	87\$:BR IF CLEAR
3018	015170	005327				DEC	(PC)+	:DECREMENT THE NUMBER
3019	015172	000005			98\$:	S		:OF TIMES TO REPEAT
3020	015174	001365				BNE	86\$:BR IF MORE TO GO
3021	015176	005713				TST	(R3)	:STRIP DSCA & DSCB FROM CSR
3022	015200	011304				MOV	(R3),R4	:GET THE REGISTER
3023	015202	020504				CMP	R5,R4	:R5=GOOD,R4=?
3024	015204	001402				BEQ	6\$:BR IF OK
3025	015206	104003				HLT	3	:REPORT THE ERROR
3026	015210	104401				SCOPI		:SW09=1?
3027								
3028	015212	104400			6\$:	SCOPE		:SCOPE THE WHOLE TEST
3029	015214	130005			7\$:	.WORD	130005	
3030	015216	130001				.WORD	130001	

3031 015220 000004
 3032 015222 000000
 3033
 3034
 3035
 3036
 3037
 3038
 3039
 3040
 3041
 3042
 3043
 3044
 3045
 3046
 3047
 3048
 3049 015224 012737 000020 001226
 3050 015232 012737 015706 001216
 3051 015240 012737 015334 001220
 3052 015246 105737 001322
 3053 015252 001002
 3054 015254 000137 015674
 3055 015260 105737 001334
 3056 015264 001773
 3057 015266 105737 001336
 3058 015272 001770
 3059 015274 005077 164104
 3060 015300 004137 007332
 3061 015304 015676
 3062
 3063 015306 052777 000400 164076
 3064 015314 004737 005044
 3065 015320 013703 001404
 3066 015324 005013
 3067 015326 052777 010000 164056
 3068 015334 052713 000010
 3069 015340 012737 000110 015370
 3070 015346 032777 004000 164040
 3071 015354 001374
 3072 015356 032777 004000 164030
 3073 015364 001774
 3074 015366 005327
 3075 015370 000110
 3076 015372 001365
 3077 015374 011304
 3078 015376 020504
 3079 015400 001423
 3080 015402 104003
 3081 015404 104401
 3082 015406 012737 000005 015436
 3083 015414 032777 004000 163772
 3084 015422 001374
 3085 015424 032777 004000 163762
 3086 015432 001774

.WORD 4
.WORD 0

***** TEST 20 *****
 ; THIS TEST PROVES THE INTERACTION OF STD
 ; WITH STD,SRD
 ; AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
 ; SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
 ; THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.

 ; TEST 20
 ; *****

```

ST20:  MOV    #20,@TSTNO
        MOV    #TST21,NEXT
        MOV    #15,LOCK
        TSTB   TCNFLG
        BNE   .+6
10$:   JMP    6$
        TSTB   STJMFL
        BEQ   10$
        TSTB   SRJMFL
        BEQ   10$
        CLR    @RXCSR          ;CLEAR THE REGISTER
        JSR    R1,0JUMPER     ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
                                ;AND OPTIONAL JUMPER ARE USED
                                ;AND LOADS R5 (EXPECTED) ACCORDINGLY.
        BIS    #MRESET,@TXCSR ;RESET THE DEVICE
        JSR    PC,SMALL       ;WAIT FOR RESET TO FINISH
        MOV    RXCSR,R3      ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
        CLR    (R3)          ;CLEAR OUT EXTRA BITS
        BIS    #NEXT,@TXCSR  ;ENTER EXTERNAL MAINT. MODE
1$:   BIS    #STD,(R3)       ;TURN ON STD
        MOV    #110,68$     ;LOAD THE NUMBER
66$:  BIT    #TIMER,@TXDBUF  ;CHECK THE TIMER BIT
        BNE   66$           ;BR IF SET
67$:  BIT    #TIMER,@TXDBUF  ;CHECK THE BIT
        BEQ   67$           ;BR IF CLEAR
        DEC   (PC)+         ;DECREMENT THE NUMBER
68$:  110                   ;OF TIMES TO REPEAT
        BNE   66$           ;BR IF MORE TO GO
        MOV   (R3),R4       ;GET THE BITS FROM THE RXCSR
        CMP   R5,R4         ;R5=GOOD R4=?
        BEQ   2$            ;BRANCH IF THEY MATCH
        HLT   3             ;NO MATCH - SHOW OPR.
        SCOP1
        MOV    #5,73$       ;LOAD THE NUMBER
71$:  BIT    #TIMER,@TXDBUF  ;CHECK THE TIMER BIT
        BNE   71$           ;BR IF SET
72$:  BIT    #TIMER,@TXDBUF  ;CHECK THE BIT
        BEQ   72$           ;BR IF CLEAR
  
```

MODEM CONTROL BITS STD AND SRD INTERACTION TEST

3087	015434	005327				DEC	(PC)+		; DECREMENT THE NUMBER
3088	015436	000005			73\$:	5			; OF TIMES TO REPEAT
3089	015440	001365				BNE	71\$; BR IF MORE TO GO
3090	015442	032713	040000			BIT	#RING, (R3)		
3091	015446	001000				BNE	2\$		
3092	015450	012737	015462	001220	2\$:	MOV	#3\$, LOCK		; SW09 SETUP
3093	015456	042705	002010			BIC	#STD, SRD, R5		; CLEAR OUT UNWANTED BITS
3094	015462	005013			3\$:	CLR	(R3)		; CLEAR OUT THE REGISTER
3095	015464	012737	000005	015514		MOV	#5, 78\$; LOAD THE NUMBER
3096	015472	032777	004000	163714	76\$:	BIT	#TIMER, @TXDBUF		; CHECK THE TIMER BIT
3097	015500	001374				BNE	76\$; BR IF SET
3098	015502	032777	004000	163704	77\$:	BIT	#TIMER, @TXDBUF		; CHECK THE BIT
3099	015510	001774				BEQ	77\$; BR IF CLEAR
3100	015512	005327				DEC	(PC)+		; DECREMENT THE NUMBER
3101	015514	000005			78\$:	5			; OF TIMES TO REPEAT
3102	015516	001365				BNE	76\$; BR IF MORE TO GO
3103	015520	011304				MOV	(R3), R4		; READ BACK THE REGISTER
3104	015522	020504				CMP	R5, R4		; R5=GOOD, R4=?
3105	015524	001402				BEQ	4\$; BRANCH IF ONLY THE DSC BITS ARE SET
3106	015526	104003				HLT	3		; NO-GO TELL OPR
3107	015530	104401				SCOPE1			
3108	015532	012737	015540	001220	4\$:	MOV	#5\$, LOCK		; SW09 SETUP
3109	015540	052713	000010		5\$:	BIS	#STD, (R3)		; TURN ON STD
3110	015544	012737	000005	015574		MOV	#5, 83\$; LOAD THE NUMBER
3111	015552	032777	004000	163634	81\$:	BIT	#TIMER, @TXDBUF		; CHECK THE TIMER BIT
3112	015560	001374				BNE	81\$; BR IF SET
3113	015562	032777	004000	163624	82\$:	BIT	#TIMER, @TXDBUF		; CHECK THE BIT
3114	015570	001774				BEQ	82\$; BR IF CLEAR
3115	015572	005327				DEC	(PC)+		; DECREMENT THE NUMBER
3116	015574	000005			83\$:	5			; OF TIMES TO REPEAT
3117	015576	001365				BNE	81\$; BR IF MORE TO GO
3118	015600	005005				CLR	R5		; CLEAR OUT EXPECTED
3119	015602	005013				CLR	(R3)		; CLEAR OUT THE REGISTER
3120	015604	052777	000400	163600		BIS	#MRESET, @TXCSR		; RESET THE DEVICE
3121	015612	004737	005044			JSR	PC, SMALL		; WAIT FOR RESET TO FINISH
3122	015616	052777	010000	163566		BIS	#MEXT, @TXCSR		; TURN ON EXTERNAL MODE
3123	015624	012737	000005	015654		MOV	#5, 88\$; LOAD THE NUMBER
3124	015632	032777	004000	163554	86\$:	BIT	#TIMER, @TXDBUF		; CHECK THE TIMER BIT
3125	015640	001374				BNE	86\$; BR IF SET
3126	015642	032777	004000	163544	87\$:	BIT	#TIMER, @TXDBUF		; CHECK THE BIT
3127	015650	001774				BEQ	87\$; BR IF CLEAR
3128	015652	005327				DEC	(PC)+		; DECREMENT THE NUMBER
3129	015654	000005			88\$:	5			; OF TIMES TO REPEAT
3130	015656	001365				BNE	86\$; BR IF MORE TO GO
3131	015660	005713				TST	(R3)		; STRIP DSCA & DSCB FROM CSR
3132	015662	011304				MOV	(R3), R4		; GET THE REGISTER
3133	015664	020504				CMP	R5, R4		; R5=GOOD, R4=?
3134	015666	001402				BEQ	6\$; BR IF OK
3135	015670	104003				HLT	3		; REPORT THE ERROR
3136	015672	104401				SCOPE1			; SW09=1?
3137									
3138	015674	104403			6\$:	SCOPE			; SCOPE THE WHOLE TEST
3139	015676	002011			7\$:	.WORD	2011		
3140	015700	002001				.WORD	2001		
3141	015702	000010				.WORD	10		
3142	015704	000000				.WORD	0		

```

***** TEST 21 *****
*THIS TEST PROVES THE INTERACTION OF DTR!RTS!STD
*WITH RING DSR CTS CARJET STD SRD
*AND DATA SET CHANGE ONE AND DATA SET CHANGE TWO.
*SET THE BIT AND VERIFY THE OTHER BITS ARE SET. CLEAR
*THE BIT AND VERIFY CLEAR. REPEAT FOR MRESET.
*****

```

```

*****
: TEST 21
*****
*****

```

```

3143
3144
3145
3146
3147
3148
3149
3150
3151
3152
3153
3154
3155
3156
3157
3158
3159 015706 012737 000021 001226
3160 015714 012737 016370 001216
3161 015722 012737 016016 001220
3162 015730 105737 001322
3163 015734 001002
3164 015736 000137 016356
3165 015742 105737 001334
3166 015746 001773
3167 015750 105737 001336
3168 015754 001770
3169 015756 005077 163422
3170 015762 004137 007332
3171 015766 016360
3172
3173 015770 052777 000400 163414
3174 015776 004737 005044
3175 016002 013703 001404
3176 016006 005013
3177 016010 052777 010000 163374
3178 016016 052713 000016
3179 016022 012737 000110 016052
3180 016030 032777 004000 163356
3181 016036 001374
3182 016040 032777 004000 163346
3183 016046 001774
3184 016050 005327
3185 016052 000110
3186 016054 001365
3187 016056 011304
3188 016060 020504
3189 016062 001423
3190 016064 104003
3191 016066 104401
3192 016070 012737 000005 016120
3193 016076 032777 004000 163310
3194 016104 001374
3195 016106 032777 004000 163300
3196 016114 001774
3197 016116 005327
3198 016120 000005

```

```

TST21: MOV #21, @TSTNO
MOV #TST22, NEXT
MOV #1$, LOCK
TSTB TCNPLG
BNE .+6
10$: JMP 6$
TSTB STJMF
BEQ 10$
TSTB SRJMF
BEQ 10$
CLR @RXCSR ; CLEAR THE REGISTER
JSR R1, OJUMPER ; THIS CALL DETERMINES IF TURNAROUND CONNECTOR
; AND OPTIONAL JUMPER ARE USED
; AND LOADS R5 (EXPECTED) ACCORDINGLY.
BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
MOV @RXCSR, R3 ; LOAD THE RECEIVER CONTROL REGISTER TO R3.
CLR (R3) ; CLEAR OUT EXTRA BITS
BIS #NEXT, @TXCSR ; ENTER EXTERNAL MAINT. MODE
BIS #DTR!RTS!STD, (R3) ; TURN ON DTR!RTS!STD
MOV #110, 68$ ; LOAD THE NUMBER
66$: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
BNE 66$ ; BR IF SET
67$: BIT #TIMER, @TXDBUF ; CHECK THE BIT
BEQ 67$ ; BR IF CLEAR
DEC (PC)+ ; DECREMENT THE NUMBER
68$: BNE 66$ ; OF TIMES TO REPEAT
MOV (R3), R4 ; GET THE BITS FROM THE RXCSR
CMP R5, R4 ; R5=GOOD R4=?
BEQ 2$ ; BRANCH IF THEY MATCH
HLT 3 ; NO MATCH - SHOW JPR.
MOV #5, 73$ ; LOAD THE NUMBER
71$: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
BNE 71$ ; BR IF SET
72$: BIT #TIMER, @TXDBUF ; CHECK THE BIT
BEQ 72$ ; BR IF CLEAR
DEC (PC)+ ; DECREMENT THE NUMBER
73$: S ; OF TIMES TO REPEAT

```

ALL MODEM CONTROL BITS INTERACTION TEST

```

3199 016122 001365          BNE 71$          ;BR IF MORE TO GO
3200 016124 032713 040000  BIT  #RING, (R3)
3201 016130 001000          BNE 2$
3202 016132 012737 016144 001220 2$: MOV #3$, LOCK          ;SW09 SETUP
3203 016140 042705 073016          BIC #RING!CTS!CARDET!SRD!DSR!STD!RTS!DTR, R5          ;CLEAR OUT UNWANTED BITS
3204 016144 005013          CLR (R3)          ;CLEAR OUT THE REGISTER
3205 016146 012737 000005 016176          MOV #5, 78$          ;LOAD THE NUMBER
3206 016154 032777 004000 163232 76$: BIT #TIMER, @TXDBUF          ;CHECK THE TIMER BIT
3207 016162 001374          BNE 76$          ;BR IF SET
3208 016164 032777 004000 163222 77$: BIT #TIMER, @TXDBUF          ;CHECK THE BIT
3209 016172 001774          BEQ 77$          ;BR IF CLEAR
3210 016174 005327          DEC (PC)+          ;DECREMENT THE NUMBER
3211 016176 000005          5          ;OF TIMES TO REPEAT
3212 016200 001365          BNE 76$          ;BR IF MORE TO GO
3213 016202 011304          MOV (R3), R4          ;READ BACK THE REGISTER
3214 016204 020504          CMP R5, R4          ;R5=GOOD, R4=?
3215 016206 001402          BEQ 4$          ;BRANCH IF ONLY THE DSC BITS ARE SET
3216 016210 104003          HLT 3          ;NO-GO TELL OPR
3217 016212 104401          SCOPE1
3218 016214 012737 016222 001220 4$: MOV #5$, LOCK          ;SW09 SETUP
3219 016222 052713 000016          BIS #DTR!RTS!STD, (R3)          ;TURN ON DTR!RTS!STD
3220 016226 012737 000005 016256          MOV #5, 83$          ;LOAD THE NUMBER
3221 016234 032777 004000 163152 81$: BIT #TIMER, @TXDBUF          ;CHECK THE TIMER BIT
3222 016242 001374          BNE 81$          ;BR IF SET
3223 016244 032777 004000 163142 82$: BIT #TIMER, @TXDBUF          ;CHECK THE BIT
3224 016252 001774          BEQ 82$          ;BR IF CLEAR
3225 016254 005327          DEC (PC)+          ;DECREMENT THE NUMBER
3226 016256 000005          5          ;OF TIMES TO REPEAT
3227 016260 001365          BNE 81$          ;BR IF MORE TO GO
3228 016262 005005          CLR R5          ;CLEAR OUT EXPECTED
3229 016264 005013          CLR (R3)          ;CLEAR OUT THE REGISTER
3230 016266 052777 000400 163116          BIS #MRESET, @TXCSR          ;RESET THE DEVICE
3231 016274 004737 005044          JSR PC, SMALL          ;WAIT FOR RESET TO FINISH
3232 016300 052777 010000 163104          BIS #MEXT, @TXCSR          ;TURN ON EXTERNAL MODE
3233 016306 012737 000005 016336          MOV #5, 86$          ;LOAD THE NUMBER
3234 016314 032777 004000 163072 86$: BIT #TIMER, @TXDBUF          ;CHECK THE TIMER BIT
3235 016322 001374          BNE 86$          ;BR IF SET
3236 016324 032777 004000 163062 87$: BIT #TIMER, @TXDBUF          ;CHECK THE BIT
3237 016332 001774          BEQ 87$          ;BR IF CLEAR
3238 016334 005327          DEC (PC)+          ;DECREMENT THE NUMBER
3239 016336 000005          5          ;OF TIMES TO REPEAT
3240 016340 001365          BNE 86$          ;BR IF MORE TO GO
3241 016342 005713          TST (R3)          ;STRIP DSCA & DSCB FROM CSR
3242 016344 011304          MOV (R3), R4          ;GET THE REGISTER
3243 016346 020504          CMP R5, R4          ;R5=GOOD, R4=?
3244 016350 001402          BEQ 1$          ;BR IF OK
3245 016352 104003          HLT 3          ;REPORT THE ERROR
3246 016354 104401          SCOPE1          ;SW09=1?
3247
3248 016356 104400          6$: SCOPE          ;SCOPE THE WHOLE TEST
3249 016360 173017          7$: .WORD 173017
3250 016362 173001          .WORD 173001
3251 016364 001016          .WORD 1016
3252 016366 000000          .WORD 0
3253
3254

```


TRANSMITTER DONE BIT INTERRUPT TEST

```

3255
3256
3257
3258
3259
3260
3261
3262
3263
3264
3265
3266 016370 012737 000022 001226
3267 016376 012737 016506 001216
3268 016404 012706 001150
3269 016410 000005
3270 016412 012737 000340 177776
3271 016420 052777 000400 162764
3272 016426 004737 005044
3273 016432 004537 007232
3274 016436 007254
3275 016440 016474
3276 016442 340 340
3277 016444 052777 004000 162740
3278 016452 005037 177776
3279 016456 052777 000100 162726
3280 016464 000240
3281 016466 000240
3282 016470 000240
3283 016472 104027
3284 016474 012706 001150
3285 016500 005077 162706
3286 016504 104400
3287
3288
3289
3290
3291
3292
3293
3294
3295
3296
3297
3298
3299
3300 016506 012737 000023 001226
3301 016514 012737 016700 001216
3302 016522 012737 000340 177776
3303 016530 052777 000400 162654
3304 016536 004737 005044
3305 016542 004537 007232
3306 016546 007254
3307 016550 016650
3308 016552 340 340
3309 016554 052777 004000 162630
3310 016562 052777 000100 162622

```

```

;*TEST THAT SETTING TRANSMIT INTERRUPT
;*ENABLE AND TRANSMITTER DONE PRODUCE
;*AN INTERRUPT ON THE TRANSMITTER VECTOR.
:*****

```

```

:*****
:
: TEST 22
:
:*****
:*****

```

```

†ST22: MOV #22, @TSTNO
MOV #TST23, NEXT
MOV #STACK, SP
RESET
MOV #340, PS ;LOCK OUT INTERRUPTS
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
JSR RS, SETVEC ;SET UP THE VECTORS
NO. ATRAP ;VECTOR "A"
IS ;VECTOR B
.BYTE 340, 340 ;PRIORITY
BIS #SYSTST, @TXCSR
CLR PS ;ZERO CPU PRIORITY
BIS #TXINTE, @TXCSR ;TURN ON TXINT ENABLE
NOP ;STALL
NOP ;DITTO
NOP ;DITTO
HLT 27 ;DUP FAILED TO INTERRUPT
IS: MOV #STACK, SP ;RESET THE STACK
CLR @TXCSR ;DISABLE DUPII
SCOPE ;SCOPE THIS TEST

```

```

:***** TEST 23 *****
;*TEST TO VERIFY THAT A TRANSMITTER DONE
;*INTERRUPT WILL ONLY OCCUR ONCE IF THE
;*TXCSR AND TXDBUF ARE *NOT* READ OR WRITTEN.
:*****

```

```

:*****
:
: TEST 23
:
:*****
:*****

```

```

†ST23: MOV #23, @TSTNO
MOV #TST24, NEXT
MOV #340, PS ;LOCK OUT INTERRUPTS
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
JSR RS, SETVEC ;SETUP FOR INTERRUPTS
NO. ATRAP ;RECEIVER
IS ;TRANSMITTER
.BYTE 340, 340 ;LEVEL
BIS #SYSTST, @TXCSR ;TURN ON CLOCK
BIS #TXINTE, @TXCSR ;TURN ON INT. ENABLE

```

```

3311 016570 005037 177776          CLR      FS          ;LOWER PROCESSOR STATUS
3312 016574 000240                NOP          ;STALL
3313 016576 000240                NOP          ;DITTO
3314 016600 000240                NOP          ;DITTO
3315 016602 104027                HLT      27        ;DUP FAILED TO INTERRUPT
3316 016604 000411                BR       5$        ;LEAVE TEST
3317 016606 005037 177776          CLR      PS          ;LOWER PROCESSOR STATUS
3318 016612 000240                NOP          ;STALL
3319 016614 000240                NOP          ;DITTO
3320 016616 000240                NOP          ;DITTO
3321 016620 105777 162566          4$: TSTB   @TXCSR    ;CHECK THE DONE BIT
3322 016624 100401                BMI     5$        ;BR IF SET
3323 016626 104024                HLT     24        ;DONE IS CLEARED AND SHOULDN'T BE
3324 016630 012737 000340 177776 5$: MOV     #340,PS    ;RAISE PROCESSOR STATUS
3325 016636 005077 162550          CLR     @TXCSR    ;CLEAR OUT DUP
3326 016642 012706 001150          MOV     #STACK,SP ;RESET STACK
3327 016646 104400                SCOPE        ;SCOPE THIS TEST
3328
3329 016650 012716 016606          1$: MOV     #2$, (SP) ;SET UP SECOND PART OF TEST
3330 016654 004537 007232          JSR     R5,SETVEC ;SETUP FOR SECOND INTERRUPT TRY
3331 016660 007254                NO.ATRAP      ;RECEIVER
3332 016662 016670                3$          ;TRANSMITTER
3333 016664 340 340                .BYTE 340,340 ;LEVEL
3334 016666 000002                RTI          ;RETURN
3335 016670
3336 016670 104026                3$: HLT     26        ;REPORT THE FACT YOU GOT HERE
3337 016672 012716 016620          MOV     #4$, (SP) ;SETUP FOR END OF TEST
3338 016676 000002                RTI          ;RETURN
3339
3340
3341
3342
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354 016700 012737 000024 001226  TEST 24 *****
3355 016706 012737 017216 001216  ;*TEST THAT SETTING DATA SET INTERRUPT
3356 016714 105737 001322          ;*ENABLE AND RECEIVING A DATA SET
3357 016720 001002                ;*CHANGE 1 OR DATA SET CHANGE 2
3358 016722 000137 017204          ;*PRODUCES AN INTERRUPT TO THE
3359 016726 012737 016774 001220  ;*RECEIVER VECTOR
3360 016734 012737 000340 177776  ;*****
3361 016742 052777 000400 162442  ;*****
3362 016750 004737 005044          ;*****
3363 016754 052777 010000 162430  ;*****
3364 016762 004537 007232          ;*****
3365 016766 017054                ;*****
3366 016770 007260                ;*****

*****
:
: TEST 24
:
: *****
:*****
TST24: MOV     #24,@TSTNO
MOV     #TST25,NEXT
TSTB   TCNFLG
BNE    .+6
JMP    5$
MOV     #1$,LOCK          ;SW09 SETUP
MOV     #340,PS          ;LOCK OUT INTERRUPTS
BIS    #MRESET,@TXCSR   ;RESET THE DEVICE
JSR    PC,SMALL         ;WAIT FOR RESET TO FINISH
BIS    #NEXT,@TXCSR     ;ENTER MAINT EXTERNAL MODE
JSR    R5,SETVEC        ;SET UP VECTORS
2$          ;RECEIVER
NO.BTRAP ;TRANSMITTER

```

RECEIVER DATA SET CHANGE BITS INTERRUPT TEST

```

3367 016772 340 340 .BYTE 340,340 ;PRORITY AT 7
3368 016774 005037 177776 15: CLR PS ;LOWER PS
3369 017000 052777 000040 162376 BIS #DSINTE,@RXCSR ;TURN ON INT. ENABLE
3370 017006 052777 000004 162370 BIS #RTS,@RXCSR ;TURN ON INT. BIT
3371 017014 012737 000005 017044 MOV #5,68$ ;LOAD THE NUMBER
3372 017022 032777 004000 162364 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3373 017030 001374 BNE 66$ ;BR IF SET
3374 017032 032777 004000 162354 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3375 017040 001774 BEQ 67$ ;BR IF CLEAR
3376 017042 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3377 017044 000005 68$: 5 ;OF TIMES TO REPEAT
3378 017046 001365 BNE 66$ ;BR IF MORE TO GO
3379 017050 104022 HLT 22 ;FAILED TO INTERRUPT
3380 017052 104401 SCOPE1 ;SW09=1
3381 017054 012706 001150 2$: MOV #STACK,SP ;RESET THE STACK
3382 017060 005077 162320 CLR @RXCSR ;CLEAR OUT RECEIVER CONTRL REGISTER.
3383 017064 012737 017104 001220 MOV #3$,LOCK ;SW09 SETUP
3384 017072 004537 007232 JSR R5,SETVEC ;SET THE VECTORS
3385 017076 017172 4$ ;RECEIVER
3386 017100 007260 NO.BTRAP ;TRANSMITTERS
3387 017102 340 340 .BYTE 340,340 ;PRIOR @7
3388 017104 012737 000005 001236 3$: MOV #5,TEMP1 ;LOAD TEMP1
3389 017112 052777 000040 162264 BIS #DSINTE,@RXCSR ;TURN ON INT. ENABLE
3390 017120 005037 177776 CLR PS ;LOWER CPU STATUS
3391 017124 052777 000002 162252 BIS #DTR,@RXCSR ;PUSH OUT INT. BITS
3392 017132 012737 000156 017162 MOV #110,73$ ;LOAD THE NUMBER
3393 017140 032777 004000 162246 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3394 017146 001374 BNE 71$ ;BR IF SET
3395 017150 032777 004000 162236 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3396 017156 001774 BEQ 72$ ;BR IF CLEAR
3397 017160 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3398 017162 000156 73$: 110. ;OF TIMES TO REPEAT
3399 017164 001365 BNE 71$ ;BR IF MORE TO GO
3400 017166 104022 HLT 22 ;FAILED TO INTERRUPT
3401 017170 104401 SCOPE1 ;BIT09=1?
3402 017172 032777 100000 162204 4$: BIT #BIT15,@RXCSR ;CHECK FOR INTERRUPT FROM DSC1
3403 017200 001001 BNE 5$ ;IT CAME FROM DSC1
3404 017202 104023 HLT 23 ;BIT15 IS CLEARED - INTERRUPTED
3405 017204 005077 162174 5$: CLR @RXCSR ;CLEAR OUT RECEIVER CONTRL REG
3406 017210 012706 001150 MOV #STACK,SP ;RESET THE STACK
3407 017214 104400 SCOPE ;SCOPE THIS TEST
3410 ;***** TEST 25 *****
3411 ;*TEST THAT SETTING RECEIVER INTERRUPT
3412 ;*ENABLE AND RECEIVER DONE CAUSES AN
3413 ;*INTERRUPT TO THE RECEIVER VECTOR
3414 ;*****
3415
3416 ;*****
3417 ;*
3418 ;* TEST 25
3419 ;*
3420 ;*****
3421 ;*****
3422 017216 012737 000025 001226 1$T25: MOV #25,@TSTNO

```

```

3423 017224 012737 017476 001216 MOV #TST26,NEXT ;LOCK OUT INTERRUPTS
3424 017232 012737 000340 177776 MOV #340,PS ;RESET THE DEVICE
3425 017240 052777 000400 162144 BIS #MRESET,@TXCSR ;WAIT FOR RESET TO FINISH
3426 017246 004737 005044 JSR PC,SMALL ;ENTER SYSTEM TEST MODE
3427 017252 052777 004000 162132 BIS #SYSTST,@TXCSR ;SET UP VECTORS
3428 017260 004537 007232 JSR R5,SETVEC ;RECEIVER
3429 017264 017460 3$ ;TRANSMITTER
3430 017266 007260 NO.BTRAP ;PRIORITY AT 7
3431 017270 340 340 .BYTE 340,340 ;LOWER PS
3432 017272 005037 177776 CLR PS ;TURN ON RECEIVER
3433 017276 052777 000020 162100 BIS #RCVEN,@RXCSR ;TURN ON INT. ENABLE
3434 017304 052777 000100 162072 BIS #RINTEN,@RXCSR ;TURN ON TRANSMITTER
3435 017312 052777 000020 162072 BIS #SEND,@TXCSR
3436 017320 1$:
3437 017320 012737 000005 017350 MOV #5,68$ ;LOAD THE NUMBER
3438 017326 032777 004000 162060 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3439 017334 001374 BNE 66$ ;BR IF SET
3440 017336 032777 004000 162050 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3441 017344 001774 BEQ 67$ ;BR IF CLEAR
3442 017346 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3443 017350 000005 68$: 5 ;OF TIMES TO REPEAT
3444 017352 001365 BNE 66$ ;BR IF MORE TO GO
3445 017354 032777 000200 162030 BIT #TXDONE,@TXCSR ;TEST TXDONE
3446 017362 001001 BNE 2$ ;BR IF SET
3447 017364 104024 HLT 2$ ;TXDONE FAILED TO SET
3448 017366 2$:
3449 017366 012777 000400 162020 MOV #400,@TXDBUF ;LOAD TX BUFFER
3450 017374 105777 162012 TSTB @TXCSR ;CHECK FOR
3451 017400 100375 BPL -4 ;DONE
3452 017402 005077 162006 CLR @TXDBUF ;CLEAR TX BUFFER
3453 017406 105777 162000 TSTB @TXCSR ;AND CHECK
3454 017412 100375 BPL -4 ;FOR DONE
3455 017414 012777 001000 161772 MOV #1000,@TXDBUF ;LOAD END OF MSG
3456 017422 012737 000050 017452 MOV #40,73$ ;LOAD THE NUMBER
3457 017430 032777 004000 161756 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3458 017436 001374 BNE 71$ ;BR IF SET
3459 017440 032777 004000 161746 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3460 017446 001774 BEQ 72$ ;BR IF CLEAR
3461 017450 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3462 017452 000050 73$: 40. ;OF TIMES TO REPEAT
3463 017454 001365 BNE 71$ ;BR IF MORE TO GO
3464 017456 104022 HLT 22 ;RECEIVER FAILED TO INTERRUPT
3465 017460 012706 001150 3$: MOV #STACK,SP ;RESET STACK
3466 017464 005077 161714 CLR @RXCSR ;CLEAR OUT REGISTER
3467 017470 005077 161716 CLR @TXCSR ;DITTO
3468 017474 104400 SCOPE ;SCOPE THIS TEST
3469
3470 ;***** TEST 26 *****
3471 ;*TEST TO VERIFY THAT A RECEIVER DONE
3472 ;*INTERRUPT WILL ONLY OCCUR ONCE IF THE
3473 ;*RXCSR AND RXDBUF ARE NOT READ OR WRITTEN
3474 ;:*****
3475
3476 ;:*****
3477 ;*
3478 ; TEST 26

```

```

3479
3480
3481
3482 017476 012737 000026 001226
3483 017504 012737 020056 001216
3484 017512 012737 000340 177776
3485 017520 052777 000400 161664
3486 017526 004737 005044
3487 017532 052777 004000 161652
3488 017540 004537 007232
3489 017544 020026
3490 017546 007260
3491 017550 340
3492 017552 052777 000020 161624
3493 017560 052777 000100 161616
3494 017566 052777 000020 161616
3495 017574
3496 017574 012737 000005 017624
3497 017602 032777 004000 161604
3498 017610 001374
3499 017612 032777 004000 161574
3500 017620 001774
3501 017622 005327
3502 017624 000005
3503 017626 001365
3504 017630 032777 000200 161554
3505 017636 001001
3506 017640 104024
3507 017642 012777 000400 161544
3508 017650 105777 161536
3509 017654 100375
3510 017656 005377 161532
3511 017662 105777 161524
3512 017666 100375
3513 017670 012777 001000 161516
3514 017676 005037 177776
3515 017702
3516 017702 012737 000050 017732
3517 017710 032777 004000 161476
3518 017716 001374
3519 017720 032777 004000 161466
3520 017726 001774
3521 017730 005327
3522 017732 000050
3523 017734 001365
3524 017736 104022
3525 017740 000420
3526 017742
3527 017742 032777 004000 161444
3528 017750 001374
3529 017752 032777 004000 161434
3530 017760 001774
3531 017762 012737 000340 177776
3532 017770 033777 000200 161406
3533 017776 001001
3534 020000 104024

```

```

:*****
:*****
:*****
TST26: MOV #26, #TSTNO
MOV #TST27, NEXT
MOV #340, PS
BIS #MRESET, #TXCSR
JSR PC, SMALL
BIS #SYSTST, #TXCSR
JSR RS, SETVEC
BS
NO. BTRAP
.BYTE 340, 340
BIS #RCVEN, #RXCSR
BIS #RINTEN, #RXCSR
BIS #SEND, #TXCSR
1$: MOV #5, 68$
66$: BIT #TIMER, #TXDBUF
BNE 66$
67$: BIT #TIMER, #TXDBUF
BEQ 67$
DEC (PC)+
68$: 5
BNE 66$
BIT #TXDONE, #TXCSR
BNE 2$
HLT 24
2$: MOV #TSOM, #TXDBUF
TSTB #TXCSR
BPL -4
CLR #TXDBUF
TSTB #TXCSR
BPL -4
MOV #TEOM, #TXDBUF
CLR PS
10$: MOV #40, 73$
71$: BIT #TIMER, #TXDBUF
BNE 71$
72$: BIT #TIMER, #TXDBUF
BEQ 72$
DEC (PC)+
73$: 40
BNE 71$
HLT 22
BR 5$
3$:
74$: BIT #TIMER, #TXDBUF
BNE 74$
75$: BIT #TIMER, #TXDBUF
BEQ 75$
MOV #340, PS
BIT #RXDONE, #RXCSR
BNE 5$
HLT 24
; LOCK OUT INTERRUPTS
; RESET THE DEVICE
; WAIT FOR RESET TO FINISH
; ENTER SYSTST MODE
; SETUP VECTORS
; RECEIVER VECTOR
; TRANSMITTER VECTOR
; LEVEL
; TURN ON RECEIVER
; TURN ON INT. ENABLE
; TURN ON TRANSMITTER
; LOAD THE NUMBER
; CHECK THE TIMER BIT
; BR IF SET
; CHECK THE BIT
; BR IF CLEAR
; DECREMENT THE NUMBER
; OF TIMES TO REPEAT
; BR IF MORE TO GO
; TEST TXDONE
; WAIT
; TXDONE FAILED TO SET
; LOAD TX BUFFER
; CHECK DONE
; AND THEN
; LOAD BUFFER
; AND CHECK
; DONE AGAIN, THEN
; SET END OF MSG
; LOWER PS
; LOAD THE NUMBER
; CHECK THE TIMER BIT
; BR IF SET
; CHECK THE BIT
; BR IF CLEAR
; DECREMENT THE NUMBER
; OF TIMES TO REPEAT
; BR IF MORE TO GO
; RECEIVER FAILED TO INTERRUPT
; LEAVE
; CHECK THE TIMER BIT
; BR IF SET
; CHECK THE TIMER BIT
; BR IF CLEAR
; RAISE PS
; TEST RXDONE
; BR IF SET
; RXDONE IS NOT SET AND SHOULD BE

```

ONLY ONE INTERRUPT PER RXDONE TEST

```

3535 020002 012737 000340 177776 5$: MOV #340,PS ;LOCKOUT INTERRUPTS
3536 020010 005077 161370 CLR @RXCSR ;CLEAR OUT DEVICE
3537 020014 005077 161366 CLR @RXDBUF ;DITTO
3538 020020 012706 001150 MOV #STACK,SP ;RESET THE STACK
3539 020024 104400 SCOPE ;SCOPE THIS TEST
3540 020026 6$:
3541 020026 012716 017742 MOV #3$, (SP) ;2ND PART SETUP
3542 020032 004537 007232 JSR RS,SETVEC ;SETUP FOR SECOND INTERRUPT TRY
3543 020036 020046 7$ ;RECEIVER VECTOR
3544 020040 007260 NO.BTRAP ;TRANSMITTER VECTOR
3545 020042 340 340 .BYTE 340,340 ;LEVEL
3546 020044 000002 RTI ;RETURN
3547 020046 7$:
3548 020046 104023 HLT 23 ;REPORT THE FACT YOU GOT HERE
3549 ;YOU TOOK A SECOND INTERRUPT AND SHOULDN'T HAVE
3550 020050 012716 020002 MOV #5$, (SP) ;SETUP TO LEAVE TEST
3551 020054 000002 RTI ;LEAVE
3552
3553 ;***** TEST 27 *****
3554 ;*TEST TO VERIFY THAT INTERRUPT VECTOR "A"
3555 ;*OCCURS BEFORE INTERRUPT VECTOR "B" EVEN
3556 ;*WHEN VECTOR "B" IS ENABLED BEFORE
3557 ;*VECTOR "A"
3558 ;*****
3559
3560 ;*****
3561 ;*
3562 ;TEST 27
3563 ;*
3564 ;*****
3565 ;*****
3566 020056 012737 000027 001226 TST27: MOV #27,@TSTNO
3567 020064 012737 020416 001216 MOV #TST30,NEXT
3568 020072 012737 000340 177776 MOV #340,PS ;SET PRIORITY TO 7
3569 020100 052777 000400 161304 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3570 020106 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3571 020112 004537 007232 JSR RS,SETVEC ;SET UP THE VECTORS
3572 020116 020376 4$ ;RECEIVER VECTOR
3573 020120 007260 NO.BTRAP ;TRANSMITTER VECTOR
3574 020122 340 340 .BYTE 340,340 ;LEVEL
3575 020124 052777 004000 161260 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
3576 020132 052777 000020 161244 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
3577 020140 052777 000100 161236 BIS #RINTEN,@RXCSR ;TURN ON REC. DONE INT. ENABLE
3578 020146 052777 001000 161234 BIS #CRCEN,@PARCSR ;TURN OFF CRC
3579 020154 052777 000020 161230 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
3580 020162 1$:
3581 020162 012737 000005 020212 MOV #5,68$ ;LOAD THE NUMBER
3582 020170 032777 004000 161216 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3583 020176 001374 66$ ;BR IF SET
3584 020200 032777 004000 161206 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3585 020206 001774 67$ ;BR IF CLEAR
3586 020210 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3587 020212 000005 68$: 5 ;OF TIMES TO REPEAT
3588 020214 001365 BNE 66$ ;BR IF MORE TO GO
3589 020216 032777 000200 161166 BIT #TXDONE,@TXCSR ;TEST TXDONE
3590 020224 001001 BNE 2$ ;BR IF SET

```

DUAL VECTORING---RECEIVER BEFORE TRANSMITTER---TEST

```

3591 020226 104024          HLT      24          ;TXDONE FAILED TO SET
3592 020230                2$:      MOV      #400,@TXDBUF ;LOAD TX BUFFER
3593 020230 012777 000400 161156  TSTB   @TXCSR    ;CHECK FOR
3594 020236 105777 161150          BPL     -4        ;DONE
3595 020242 100375          CLR     @TXDBUF  ;LOAD THE BUFFER
3596 020244 005077 161144  TSTB   @TXCSR    ;AND CHECK
3597 020250 105777 161136  BPL     -4        ;DONE AGAIN
3598 020254 100375          MOV     #1000,@TXDBUF ;LOAD TEOM
3599 020256 012777 001000 161130  MOV     #100.,73$ ;LOAD THE NUMBER
3600 020264 012737 000144 020314  BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3601 020272 032777 004000 161114 71$:  BNE     71$      ;BR IF SET
3602 020300 001374          BIT     #TIMER,@TXDBUF ;CHECK THE BIT
3603 020302 032777 004000 161104 72$:  BEQ     72$      ;BR IF CLEAR
3604 020310 001774          DEC     (PC)+    ;DECREMENT THE NUMBER
3605 020312 005327          73$:  100.      ;OF TIMES TO REPEAT
3606 020314 000144          BNE     71$      ;BR IF MORE TO GO
3607 020316 001365          TSTB   @RXCSR    ;CHECK DONE
3608 020320 105777 161060          SMI     5$      ;BR IF SET
3609 020324 100401          HLT     24      ;DONE FAILED TO SET
3610 020326 104024          BIT     #TXDONE,@TXCSR ;TEST TXDONE
3611 020330 032777 000200 161054 5$:  BNE     3$      ;BR IF SET
3612 020336 001001          HLT     24      ;TXDONE NOT SET SHOULD BE
3613 020340 104024          BIS     #TXINTE,@TXCSR ;TURN ON TRANSMITTER INT. ENABLE
3614 020342 052777 000100 161042 3$:  CLR     PS      ;LOWER PROCESSOR STATUS
3615 020350 005037 177776          BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3616 020354 032777 004000 161032 74$:  BNE     74$      ;BR IF SET
3617 020362 001374          BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3618 020364 032777 004000 161022 75$:  BEQ     75$      ;BR IF CLEAR
3619 020372 001774          HLT     27      ;DUP FAILED TO INTERRUPT
3620 020374 104027          MOV     #STACK,SP ;RESET THE STACK
3621 020376 012706 001150 4$:      BIS     #MRESET,@TXCSR ;RESET THE DEVICE
3622 020402 052777 000400 161002  JSR     PC,SMALL ;WAIT FOR RESET TO FINISH
3623 020410 004737 005044          JSR     PC,SMALL ;SCOPE THIS TEST
3624 020414 104400          SCOPE

```

```

***** TEST 30 *****
; *TEST TO VERIFY THAT SERVICING THE
; *TXDONE BIT RE-ARMS THE INTERRUPT
; *LOGIC IF INTERRUPT ENABLE IS SET.
*****

```

```

; *****
; *
; * TEST 30
; *
; *****
; *****

```

```

3638 020416 012737 000030 001226 3T30: MOV     #30,@TSTNO
3639 020424 012737 020724 001216  MOV     #TST31,NEXT
3640 020432 012737 000340 177776  MOV     #340,PS ;LOCK OUT INTERRUPTS
3641 020440 052777 000400 160744  BIS     #MRESET,@TXCSR ;RESET THE DEVICE
3642 020446 004737 005044          JSR     PC,SMALL ;WAIT FOR RESET TO FINISH
3643 020452 004537 007232          JSR     RS,SETVEC ;INTERRUPT VECTOR SETUP
3644 020456 007254          NO.ATRAP ;RECEIVER VECTOR
3645 020460 020636          4$      ;TRANSMITTER VECTOR
3646 020462          34C     34C    ;LEVEL

```

TXDONE BIT RE-ARM INTERRUPT TEST

```

3647 020464 052777 004120 160720     BIS      #SYSTST!SEND!TXINTE,@TXCSR ;TURN ON TRANSMITTER, CLOCK
3648                                     ;AND INTERRUPTS
3649 020472 052777 000400 160714     BIS      @TSOM,@TXDBUF ;LOAD START OF MSG
3650 020500 005037 177776             CLR      PS ;LOWER PROCESSOR STATUS
3651 020504 012737 000005 020534     MOV      #5,58$ ;LOAD THE NUMBER
3652 020512 032777 004000 160674 66$:  BIT      @TIMER,@TXDBUF ;CHECK THE TIMER BIT
3653 020520 001374             BNE     66$ ;BR IF SET
3654 020522 032777 004000 160664 67$:  BIT      @TIMER,@TXDBUF ;CHECK THE BIT
3655 020530 001774             BEQ     67$ ;BR IF CLEAR
3656 020532 005327             DEC     (PC)+ ;DECREMENT THE NUMBER
3657 020534 000005 68$:          S ;OF TIMES TO REPEAT
3658 020536 001365             BNE     66$ ;BR IF MORE TO GO
3659 020540 104027             HLT     27 ;DUP FAILED TO INTERRUPT THE FIRST TIME
3660 020542 000427             BR      3$ ;LEAVE THE TEST
3661 020544 005037 177776 1$:      CLR      PS ;LOWER PROCESSOR STATUS
3662 020550 012737 000005 020600     MOV      #5,73$ ;LOAD THE NUMBER
3663 020556 032777 004000 160630 71$:  BIT      @TIMER,@TXDBUF ;CHECK THE TIMER BIT
3664 020564 001374             BNE     71$ ;BR IF SET
3665 020566 032777 004000 160620 72$:  BIT      @TIMER,@TXDBUF ;CHECK THE BIT
3666 020574 001774             BEQ     72$ ;BR IF CLEAR
3667 020576 005327             DEC     (PC)+ ;DECREMENT THE NUMBER
3668 020600 000005 73$:          S ;OF TIMES TO REPEAT
3669 020602 001365             BNE     71$ ;BR IF MORE TO GO
3670 020604 104027             HLT     27 ;DUP FAILED TO INTERRUPT AFTER SERVICING DONE
3671 020606 000405             BR      3$ ;LEAVE
3672 020610 2$:
3673 020610 005077 160576     CLR      @TXCSR ;SHUT DOWN THE DUP
3674 020614 012716 020622     MOV      #3$, (SP) ;SETUP TO END TEST
3675 020620 000002             RTI ;RETURN
3676 020622 012737 000340 177776 3$:  MOV      #340,PS ;RAISE PROCESSOR STATUS
3677 020630 012706 001150     MOV      @STACK,SP ;RESET STACK
3678 020634 104400             SCOPE ;SCOPE THIS TEST
3679 020636 032777 000200 160546 4$:  BIT      @TXDONE,@TXCSR ;CLEAR DONE AND RE-ARM INTERRUPT
3680 020644 005077 160544     CLR      @TXDBUF ;LOAD BUFFER
3681 020650 012737 000005 020700     MOV      #5,78$ ;LOAD THE NUMBER
3682 020656 032777 004000 160530 76$:  BIT      @TIMER,@TXDBUF ;CHECK THE TIMER BIT
3683 020664 001374             BNE     76$ ;BR IF SET
3684 020666 032777 004000 160520 77$:  BIT      @TIMER,@TXDBUF ;CHECK THE BIT
3685 020674 001774             BEQ     77$ ;BR IF CLEAR
3686 020676 005327             DEC     (PC)+ ;DECREMENT THE NUMBER
3687 020700 000005 78$:          S ;OF TIMES TO REPEAT
3688 020702 001365             BNE     76$ ;BR IF MORE TO GO
3689 020704 012716 020544     MOV      #1$, (SP) ;SETUP TO FINISH TEST
3690 020710 004537 007232     JSR     R5,SETVEC ;SETUP VECTORS FOR NEXT PART OF TEST
3691 020714 007254             NO.ATRAP ;RECEIVER VECTOR
3692 020716 020610 2$:          ;TRANSMITTER VECTOR
3693 020720 340 340             .BYTE 340,340 ;LEVEL
3694 020722 000002             RTI ;RETURN

```

```

:***** TEST 31 *****
:*TEST TO VERIFY THAT SERVICING THE
:*RXDONE BIT RE-ARMS THE INTERRUPT
:*LOGIC IF INTERRUPT ENABLE IS SET.
:*****

```

```

3695
3696
3697
3698
3699
3700
3701
3702

```



```

3703      :*****
3704      :*
3705      : TEST 31
3706      :*
3707      :*****
3708      :*****
3709      020724 012737 000031 001226 †ST31:  MOV    #31,@TSTNO
3710      020732 012737 021316 001216      MOV    @TST32,NEXT
3711      020740 012737 000340 177776      MOV    #340,PS
3712      020746 052777 000400 160436      BIS    @MRESET,@TXCSR ;LOCK OUT INTERRUPTS
3713      020754 004737 005044      JSR    PC,SMALL ;RESET THE DEVICE
3714      020760 052777 004000 160424      BIS    @SYSTST,@TXCSR ;WAIT FOR RESET TO FINISH
3715      020766 004537 007232      JSR    RS,SETVEC ;ENTER SYSTST MODE
3716      020772 021234      JSR    4$ ;SETUP VECTORS
3717      020774 007260      NO.BTRAP ;RECEIVER VECTOR
3718      020776      340      340      .BYTE 340,340 ;TRANSMITTER VECTOR
3719      021000 052777 000020 160376      BIS    @RCVEN,@RXCSR ;LEVEL
3720      021006 052777 000100 160370      BIS    @RINTEN,@RXCSR ;TURN ON RECEIVER
3721      021014 052777 000020 160370      BIS    @SEND,@TXCSR ;TURN ON INT. ENABLE
3722      021022      1$:      MOV    #5,68$ ;TURN ON TRANSMITTER
3723      021022 012737 000005 021052      MOV    #5,68$ ;LOAD THE NUMBER
3724      021030 032777 004000 160356 66$:      BIT    @TIMER,@TXDBUF ;CHECK THE TIMER BIT
3725      021036 001374      BNE    66$ ;BR IF SET
3726      021040 032777 004000 160346 67$:      BIT    @TIMER,@TXDBUF ;CHECK THE BIT
3727      021046 001774      BEQ    67$ ;BR IF CLEAR
3728      021050 005327      DEC    (PC)+ ;DECREMENT THE NUMBER
3729      021052 000005      68$:      5 ;OF TIMES TO REPEAT
3730      021054 001365      BNE    66$ ;BR IF MORE TO GO
3731      021056 032777 000200 160326      BIT    @TXDONE,@TXCSR ;TEST TXDONE
3732      021064 001001      BNE    2$ ;BR IF SET
3733      021066 104024      HLT    24 ;TXDONE FAILED TO SET
3734      021070 005037 177776      CLR    PS ;LOWER PROCESSOR STATUS
3735      021074 012777 000400 160312 2$:      MOV    #400,@TXDBUF ;LOAD TX BUFFER
3736      021102 105777 160304      TSTB  @TXCSR
3737      021106 100375      BPL    -4
3738      021110 005077 160300      CLR    @TXDBUF
3739      021114 105777 160272      TSTB  @TXCSR
3740      021120 100375      BPL    -4
3741      021122 012777 001000 160264 7$:      MOV    #1000,@TXDBUF
3742      021130      7$:      MOV    #40,73$ ;LOAD THE NUMBER
3743      021130 012737 000050 021160      MOV    #40,73$
3744      021136 032777 004000 160250 71$:      BIT    @TIMER,@TXDBUF ;CHECK THE TIMER BIT
3745      021144 001374      BNE    71$ ;BR IF SET
3746      021146 032777 004000 160240 72$:      BIT    @TIMER,@TXDBUF ;CHECK THE BIT
3747      021154 001774      BEQ    72$ ;BR IF CLEAR
3748      021156 005327      DEC    (PC)+ ;DECREMENT THE NUMBER
3749      021160 000050      73$:      40 ;OF TIMES TO REPEAT
3750      021162 001365      BNE    71$ ;BR IF MORE TO GO
3751      021164 104022      HLT    22 ;RECEIVER FAILED TO INTERRUPT
3752      021166 000445      BR    6$ ;GET OUT OF TEST
3753      021170 005037 177776      CLR    PS ;LOWER STATUS
3754      021174 012737 000005 021224 3$:      MOV    #5,78$ ;LOAD THE NUMBER
3755      021202 032777 004000 160204 76$:      BIT    @TIMER,@TXDBUF ;CHECK THE TIMER BIT
3756      021210 001374      BNE    76$ ;BR IF SET
3757      021212 032777 004000 160174 77$:      BIT    @TIMER,@TXDBUF ;CHECK THE BIT
3758      021220 001774      BEQ    77$ ;BR IF CLEAR

```

RXDONE BIT RE-ARM INTERRUPT TEST

```

3759 021222 005327          DEC      (PC)+      ;DECREMENT THE NUMBER
3760 021224 000005      78$:      S              ;OF TIMES TO REPEAT
3761 021226 001365          BNE      76$          ;BR IF MORE TO GO
3762 021230 104022          HLT      22          ;DUP FAILED TO INTERRUPT AFTER SERVICING DONE
3763 021232 000423          BR       6$          ;LEAVE IT
3764
3765 021234 032777 000200 160142 4$:      BIT      #RXDONE, @RXCSR ;SERVICE THE DONE BIT
3766 021242 012716 021170      MOV      #3$, (SP)    ;SETUP FOR 2ND PART OF TEST
3767 021246 004537 007232      JSR      R5, SETVEC  ;SETUP NEW VECTORS
3768 021252 021262          SS              ;RECEIVER VECTOR
3769 021254 007260          NO.BTRAP        ;TRANSMITTER VECTOR
3770 021256          340      .BYTE 340, 340 ;LEVEL
3771 021260 000002          RTI             ;GO FINISH TEST
3772 021262
3773 021262 052777 000400 160122 5$:      BIS      #MRESET, @TXCSR ;RESET THE DEVICE
3774 021270 004737 005044      JSR      PC, SMALL   ;WAIT FOR RESET TO FINISH
3775 021274 012716 021302      MOV      #6$, (SP)  ;SETUP TO FINISH TEST
3776 021300 000002          RTI             ;LEAVE
3777 021302 012737 000340 177776 6$:      MOV      #340, PS    ;RAISE PROCESSOR STATUS
3778 021310 012706 001150      MOV      #STACK, SP ;RESET THE STACK
3779 021314 104400          SCOPE          ;SCOPE THIS TEST
3780          ;***** TEST 32 *****
3781          ;*TEST TO PROVE AN INTERRUPT REQUEST
3782          ;*IS GENERATED WHEN AN ABORT IS RECEIVED.
3783          ;*****
3784
3785          ;*****
3786          ;*
3787          ;TEST 32
3788          ;*
3789          ;*****
3790          ;*****
3791 021316 012737 000032 001226 †TST32: MOV      #32, @TSTNO
3792 021324 012737 021576 001216      MOV      #TST33, NEXT
3793 021332 012737 000340 177776      MOV      #340, PS    ;LOCK OUT INTERRUPTS
3794 021340 052777 000400 160044      BIS      #MRESET, @TXCSR ;RESET THE DEVICE
3795 021346 004737 005044      JSR      PC, SMALL   ;WAIT FOR RESET TO FINISH
3796 021352 052777 004000 160032      BIS      #SYSTST, @TXCSR ;ENTER SYS TST MODE
3797 021360 004537 007232      JSR      R5, SETVEC  ;SET UP VECTORS
3798 021364 021550          2$              ;RECEIVER
3799 021366 007260          NO.BTRAP        ;TRANSMITTER
3800 021370          340      .BYTE 340, 340 ;LEVEL
3801 021372 005037 177776      CLR      PS         ;LOWER PROCESSOR STATUS
3802 021376 052777 000120 160000      BIS      #RCVEN!RINTEN, @RXCSR ;TURN ON RECEIVER AND INTERRUPT ENABLE
3803 021404 052777 000020 160000      BIS      #SEND, @TXCSR ;TURN ON TRANSMITTER
3804 021412 012737 000005 021442      MOV      #5, 68$    ;LOAD THE NUMBER
3805 021420 032777 004000 157766 66$:      BIT      #TIMER, @TXDBUF ;CHECK THE TIMER BIT
3806 021426 001374          BNE      66$        ;BR IF SET
3807 021430 032777 004000 157756 67$:      BIT      #TIMER, @TXDBUF ;CHECK THE BIT
3808 021436 001774          BEQ      67$        ;BR IF CLEAR
3809 021440 005327          DEC      (PC)+      ;DECREMENT THE NUMBER
3810 021442 000005      68$:      S              ;OF TIMES TO REPEAT
3811 021444 001365          BNE      66$        ;BR IF MORE TO GO
3812 021446 105777 157740      TSTB     @TXCSR     ;TEST DONE
3813 021452 100401          BMI      1$         ;BR IF SET
3814 021454 104024          HLT      24         ;DONE FAILED TO SET

```

```

3815 021456 012777 000400 157730 1$: MOV #T50M,@TXDBUF ;LOAD TX BUFFER
3816 021464 105777 157722 TSTB @TXCSR ;CHECK FOR DONE
3817 021470 100375 BPL -4 ;BR IF NOT SET
3818 021472 005077 157716 CLR @TXDBUF ;PUSH OUT A CHARACTER
3819 021476 105777 157710 TSTB @TXCSR ;WAIT FOR DONE
3820 021502 100375 BPL -4 ;BR IF NOT SET
3821 021504 052777 002000 157702 BIS #TABORT,@TXDBUF ;SET ABORT
3822 021512 012737 000113 021542 MOV #75,@73$ ;LOAD THE NUMBER
3823 021520 032777 004000 157666 71$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3824 021526 001374 BNE 71$ ;BR IF SET
3825 021530 032777 004000 157656 72$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3826 021536 001774 BEQ 72$ ;BR IF CLEAR
3827 021540 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3828 021542 000113 73$: 75. ;OF TIMES TO REPEAT
3829 021544 001365 BNE 71$ ;BR IF MORE TO GO
3830 021546 104022 HLT 22 ;RECEIVER FAILED TO INTERRUPT
3831 ;ON AN ABORTED MESSAGE
3832 021550 012737 000340 177776 2$: MOV #340,PS ;SET STATUS TO 7
3833 021556 012706 001150 MOV #STACK,SP ;RESET THE STACK
3834 021562 052777 000400 157622 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3835 021570 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3836 021574 104400 SCOPE
3837
3838 ;***** TEST 33 *****
3839 ;*THIS TEST IS AN AID FOR DEBUGGING CRC
3840 ;*ERRORS. A CHARACTER IS LOADED INTO THE
3841 ;*DUP AND PUSHED OUT BIT BY BIT WHILE
3842 ;*ALLOWING THE OPERATOR TO MONITOR THE CRC
3843 ;*CHARACTER AS IT IS GENERATED. THE DATA CHARACTER
3844 ;*CAN ALSO BE CHANGED BY THE OPERATOR.
3845 ;*PUT SW09=1 TO LOCK ON BITS. TO CONTINUE HIT
3846 ;*ANY KEY ON THE TTY. AFTER 16 TIMES PUT DOWN SW09 TO LEAVE
3847 ;*NOTE: REMEMBER--IN SDC A ONE IS A LOGIC LOW IN THE
3848 ;*CRC GENERATOR.
3849 ;*****
3850 ;*****
3851 ;*
3852 ;* TEST 33
3853 ;*
3854 ;*****
3855 ;*****
3856 021576 012737 000033 001226 ST33: MOV #33,@TSTNO
3857 021604 012737 002764 001216 MOV #.EOP,NEXT
3858 021612 052777 000400 157572 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3859 021620 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3860 021624 012737 102010 013574 MOV #CRC.CCITT,XPOLY ;LOAD THE POLYNOMIAL
3861 021632 012737 000125 022000 MOV #125,3$ ;LOAD DATA TO SOFTWARE BCC-CHANGE CHARACTER HERE
3862 021640 013737 022000 001252 MOV 3$,SAVR1
3863 021646 012737 177777 013600 MOV #-1,CALBCC ;CLEAR FOR SOFTWARE BCC
3864 021654 013737 013600 022002 MOV CALBCC,4$
3865 021662 005037 001242 CLR TEMP3
3866 021666 005037 001244 CLR TEMP4 ;CLEAR BIT COUNTER
3867 021672 005037 001246 CLR TEMP5
3868 021676 005077 157512 CLR @TXDBUF ;RESET TXDONE
3869 021702 052777 014000 157502 BIS #MMODE,@TXCSR ;ENTER MAINT MODE-PROGRAM CLOCKING
3870 021710 052777 000020 157466 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER

```

3871	021716	052777	000020	157466		BIS	#SEND, @TXCSR	;TURN ON TRANSMITTER
3872	021724	012777	000400	157462		MOV	#TSOM, @TXDBUF	
3873	021732	104412	000044			PKCLK	36.	;PUSH OUT 2
3874	021736	013777	022000	157450		MOV	3\$, @TXDBUF	;LOAD DATA
3875	021744	104412	000020			PKCLK	.16.	;PUSH OUT ANOTHER
3876	021750	104412	000002		1\$:	PKCLK	2	;PUSH OUT A BIT
3877	021754	013737	001244	001254		MOV	TEMP4, SAVR2	;SET UP TO TYPE
3878	021762	005237	001242			INC	TEMP3	
3879	021766	005237	001244			INC	TEMP4	;UPDATE BIT COUNTER
3880	021772	004537	013422		2\$:	JSR	R5, SIMBCC	;CALCULATE SOFTWARE BCC BASED ON THESE PARAMETERS
3881	021776	000001				1		;SHIFTS
3882	022000	000000			3\$:	.WORD	0	;DATA
3883	022002	000000			4\$:	.WORD	0	;PREVIOUS BCC
3884	022004	004737	022102			JSR	PC, 5\$;CHECK TO SEE IF WE SHOULD WAIT FOR SCOPING
3885	022010	000241				CLC		;CLEAR FOR NEXT ROTATE
3886	022012	106037	022000			RORB	3\$;SET UP THE NEXT BIT
3887	022016	013737	013600	022002		MOV	CALBCC, 4\$;FOR THE SOFTWARE BCC
3888	022024	022737	000006	001244		CMF	#6, TEMP4	
3889	022032	001002				BNE	.+6	
3890	022034	005077	157354			CLR	@TXDBUF	
3891	022040	022737	000014	001242		CMP	#12., TEMP3	
3892	022046	001003				BNE	12\$	
3893	022050	012777	001000	157336		MOV	#TEOM, @TXDBUF	
3894	022056	022737	000020	001244	12\$:	CMP	#16., TEMP4	;ALL DONE WITH THE CHARACTER?
3895								;INCREASE THE COMPARE NUMBER TO OUTPUT
3896								;CRC TO THE RECEIVER
3897	022064	001331				BNE	1\$;BR IF MORE TO GO
3898	022066	052777	000400	157316		BIS	#MRESET, @TXCSR	;RESET THE DEVICE
3899	022074	004737	005044			JSR	PC, SMALL	;WAIT FOR RESET TO FINISH
3900	022100	104400				SCOPE		;SCOPE THIS TEST
3901								
3902	022102	032777	001000	157072	5\$:	BIT	#SW09, @SWR	;SW09=1?
3903	022110	001432				BEQ	6\$;BR IF NO
3904	022112	013704	013600			MOV	CALBCC, R4	;THE DATA CHARACTER IS ALWAYS
3905	022116	012737	000001	001256		MOV	#1, SAVR3	;FOLLOWED BY A ZERO CHARACTER. THE
3906	022124	000241				CLC		;DATA BIT IN CRC SHOWS WHICH BIT OF
3907	022126	006004			11\$:	ROR	R4	;THE TWO CHARACTERS IS BEING GENERATED.
3908	022130	006137	001256			ROR	SAVR3	
3909	022134	103374				BCC	11\$	
3910	022136	105737	001246			TSTB	TEMP5	
3911	022142	001006				BNE	10\$	
3912	022144	104402	022641			TYPE	.EM17	;TYPE MSG
3913	022150	104402	022670			TYPE	MH1	;TYPE HEADER
3914	022154	105137	001246			COMB	TEMP5	
3915	022160	104410			10\$:	CONVRT		
3916	022162	023456				DTI		
3917	022164	105777	157014		7\$:	TSTB	@TKCSR	;CHECK TTY DONE--GO SCOPE THE CRC GENERATOR
3918								;NOTE: THE LSB IS RIGHT JUSTIFIED ON THE TTY OUTPUT!!
3919	022170	100375				BPL	7\$;BR IF NOT YET
3920	022172	017701	157010			MOV	@TKDBR, R1	;READ THE BUFFER
3921	022176	000207			6\$:	RTS	PC	;RETURN
3922								
3923								

```

3924 022200 051377 047505 020115 EM1: .ASCIZ <377>/REOM BIT /
(1) 022213 377 054122 047504 EM2: .ASCIZ <377>/RXDONE BIT /
(1) 022230 051377 040530 047502 EM3: .ASCIZ <377>/RXABORT BIT /
(1) 022246 051377 040530 052103 EM4: .ASCIZ <377>/RXACTIVE BIT /
(1) 022265 377 054122 042504 EM5: .ASCIZ <377>/RXDERR BIT /
(1) 022302 051377 043505 051511 EM6: .ASCIZ <377>/REGISTER ERROR /
(1) 022323 377 054122 042504 EM7: .ASCIZ <377>/RXDERR OR OVERRUN /
(1) 022347 377 053117 051105 EM10: .ASCIZ <377>/OVERRUN BIT /
(1) 022365 377 040504 040524 EM11: .ASCIZ <377>/DATA COMPARE ERROR /
(1) 022412 042377 052101 020101 EM12: .ASCIZ <377>/DATA COMPARE ERROR IN SECONDARY MODE /
(1) 022461 377 051103 020103 EM13: .ASCIZ <377>/CRC CALCULATION ERROR /
(1) 022511 377 051522 046517 EM14: .ASCIZ <377>/RSOM BIT /
(1) 022524 040506 046111 042105 DH1: .ASCIZ /FAILED TO CLEAR /
(1) 022545 106 044501 042514 DH2: .ASCIZ /FAILED TO SET /
(1) 022564 052777 042516 050130 DH3: .ASCIZ <377>/UNEXPECTED /
(1) 022601 377 054105 042520 DH6: .ASCIZ <377>/EXPECTED FOUND REGISTER /
(1) 022641 377 051103 020103 EM17: .ASCIZ <377>/CRC GENERATOR STATUS /
(1) 022670 042377 052101 020101 MH1: .ASCIZ <377>/DATA CHAR DATA BIT IN CRC GEN. CRC FOR THIS BIT /
(1) 022763 015 052012 040522 EM15: .ASCIZ <15><12>/TRANSMITTER /
(1) 023003 015 051012 041505 EM16: .ASCIZ <15><12>/RECEIVER /
(1) 023020 047504 042516 041040 EM23: .ASCIZ /DONE BIT /
(1) 023033 106 044501 042514 EM22: .ASCIZ /FAILED TO SET /
(1) 023052 052777 042516 050130 EM20: .ASCIZ <377>/UNEXPECTED RECEIVER INTERRUPT /
(1) 023113 377 047125 054105 EM21: .ASCIZ <377>/UNEXPECTED TRANSMITTER INTERRUPT /
(1) 023157 106 044501 042514 DH4: .ASCIZ /FAILED TO INTERRUPT. /
(1) 023204 047111 042524 052522 DH5: .ASCIZ /INTERRUPTED UNEXPECTEDLY. /
(1) 023236 .EVEN

```

```

(1) 023236 .ERRTAB:
(1) 023236 000000 0
(1) 023240 000000 0
(1) 023242 000000 0
(1) 023244 022200 EM1
(1) 023246 022545 DH2 :HALT 1
(1) 023250 000000 0
(1) 023252 022213 EM2
(1) 023254 022545 DH2 :HALT 2
(1) 023256 000000 0
(1) 023260 022302 EM6
(1) 023262 022601 DH6 :HALT 3
(1) 023264 023474 DT6
(1) 023266 022200 EM1
(1) 023270 022524 DH1 :HALT 4
(1) 023272 000000 0
(1) 023274 022230 EM3
(1) 023276 022545 DH2 :HALT 5
(1) 023300 000000 0
(1) 023302 022246 EM4
(1) 023304 022524 DH1 :HALT 6

```

(1)	023306	000000	0	
(1)	023310	022265	EM5	
(1)	023312	022545	DH2	:HALT 7
(1)	023314	000000	0	
(1)	023316	022365	EM11	
(1)	023320	000000	0	:HALT 10
(1)	023322	000000	0	
(1)	023324	022230	EM3	
(1)	023326	022524	DH1	:HALT 11
(1)	023330	000000	0	
(1)	023332	022412	EM12	
(1)	023334	000000	0	:HALT 12
(1)	023336	000000	0	
(1)	023340	022323	EM7	
(1)	023342	022545	DH2	:HALT 13
(1)	023344	000000	0	
(1)	023346	022246	EM4	
(1)	023350	022545	DH2	:HALT 14
(1)	023352	000000	0	
(1)	023354	022347	EM10	
(1)	023356	022524	DH1	:HALT 15
(1)	023360	000000	0	
(1)	023362	022564	DH3	
(1)	023364	022302	EM6	:HALT 16
(1)	023366	000000	0	
(1)	023370	022461	EM13	
(1)	023372	000000	0	:HALT 17
(1)	023374	000000	0	
(1)	023376	022511	EM14	
(1)	023400	022524	DH1	:HALT 20
(1)	023402	000000	0	
(1)	023404	022511	EM14	
(1)	023406	022545	DH2	:HALT 21
(1)	023410	000000	0	
(1)	023412	023003	EM16	
(1)	023414	023157	DH4	:HALT 22
(1)	023416	000000	0	
(1)	023420	023003	EM16	
(1)	023422	023204	DH5	:HALT 23
(1)	023424	000000	0	
(1)	023426	023020	EM23	
(1)	023430	023033	EM22	:HALT 24

```

(1) 023432 000000 0
(1)
(1) 023434 023052 EM20
(1) 023436 000000 0 ;HALT 25
(1) 023440 000000 0
(1)
(1) 023442 023113 EM21
(1) 023444 000000 0 ;HALT 26
(1) 023446 000000 0
(1)
(1) 023450 022763 EM15
(1) 023452 023157 DM4 ;HALT 27
(1) 023454 000000 0
(1)
(1) 023456 000003 DT1: 3
(1) 023460 006 021 .BYTE 6,17.
(1) 023462 001252 SAVR1
(1) 023464 006 017 .BYTE 6,15.
(1) 023466 001254 SAVR2
(1) 023470 006 002 .BYTE 6,2
(1) 023472 001256 SAVR3
(1)
(1) 023474 000003 DT6: 3
(1) 023476 006 004 .BYTE 6,4
(1) 023500 001262 SAVR5
(1) 023502 006 002 .BYTE 6,2
(1) 023504 001260 SAVR4
(1) 023506 006 002 .BYTE 6,2
(1) 023510 001256 SAVR3
(1) 023512 CORMAX:
000001 .END

```


DZDPCB.P11 13-MAY-77 15:26 CROSS REFERENCE TABLE -- USER SYMBOLS

MPASSX	005355	1377	1797#											
MPOWER	005141	1782	1797#											
MQM	005132	1490	1797#	1906										
MR	005202	1360	1797#	1901										
MRESET=	000400	846#	985#	2038	2085	2155	2203	2256	2330	2374	2438	2479	2531	2568
		2599	2630	2667	2774	2847	2904	2953	3010	3063	3120	3173	3230	3271
		3303	3361	3425	3485	3569	3622	3641	3712	3773	3794	3834	3858	3898
MSRJM	005761	1257	1797#											
MSTJM	005726	1253	1797#											
MTCN	005600	1247	1797#											
MTDATA=	040000	979#	2790	2794	2799	2806								
MTITLE	001000	884#	1194											
MTOTAL	005647	1235	1797#											
MTSTN	005377	1703	1797#	1885										
MTSTPC	005300	1797#												
MVEC	005512	1218	1797#											
MVECX	005347	1375	1797#											
NEXT	001216	904#	1431	1748	2037*	2084*	2154*	2202*	2255*	2329*	2373*	2437*	2478*	2530*
		2592*	2653*	2773*	2838*	2944*	3050*	3160*	3267*	3301*	3355*	3423*	3483*	3567*
		3639*	3710*	3792*	3857*									
NO. 3TR	007254	1984#	3274	3306	3331	3644	3691							
NO. BTR	007260	1988#	3366	3386	3430	3490	3544	3573	3717	3769	3799			
OJUMPE	007332	2014#	2844	2950	3060	3170								
OPCLK	001323	1011#	1245	1260	1304*	2001	2007							
OVRRUN=	040000	968#	2453	2454	2491	2509	2510							
PARAM =	104405	1050#	1211	1219	1227	1236	1886							
PARAM1	003576	1508#	1525											
PARBIT=	000000	846#												
PARCSR	001410	1075#	1854*	2041*	2088*	2090*	2158*	2206*	2259*	2261*	2332*	2376*	2378*	2441*
		2482*	2484*	2536*	2601*	2637*	2781*	3578*						
PARERR	003652	1511	1513	1515	1524#	1531	1533	1535						
PASCNT	001230	909#	1169*	1370*	1371	1404								
PERFOR=	000000	846#												
PKCLK =	104412	1060#	1944	1953	1955	1960	1969	1974	2126	2167	2173	2178	2217	2226
		2231	2269	2275	2280	2294	2301	2306	2341	2386	2404	2542	2545	2548
		2559	2673	2679	2681	2689	2694	2785	2789	2793	2796	2805	2809	2811
		3873	3875	3876										
POPPO =	012600	824#	1742											
POP1SP=	005726	822#												
POP2SP=	022626	826#	1433											
PRIPTY	001320	1009#	1233*	1297*										
PRISEC=	010000	976#	2090	2261	2378	2484	2597	2636						
PS =	177776	815#	1165*	1347*	3270*	3278*	3302*	3311*	3317*	3324*	3360*	3368*	3390*	3424*
		3432*	3484*	3514*	3531*	3535*	3568*	3615*	3640*	3650*	3661*	3676*	3711*	3734*
		3753*	3777*	3793*	3801*	3832*								
PUSHRO=	010046	823#	1727											
PUSH1S=	005746	821#												
PUSH2S=	024646	825#												
PWRUP	005060	1776	1778#											
QV.FLG	001343	1027#	1171*	1383*	1422									
RABORT=	002000	970#	2348	2391	2411									
RCRCIN=	040000	999#												
RCRC7T=	100000	998#												
RCVEN =	000020	961#	2042	2089	2157	2205	2258	2333	2377	2442	2483	2539	2602	2670
		2783	3433	3492	3576	3719	3802	3870						
RECACT=	004000	954#	2046	2066	2094	2108	2110	2137	2351	2414	2458	2514		

DZDPCB MACY11 27(1006) 18-MAY-77 00:03 PAGE 92
 DZDPCB.P11 13-MAY-77 15:26 CROSS REFERENCE TABLE -- USER SYMBOLS

REOM = 001000	971#	2180	2233	2281	2308									
RESREG 004614	1719	1722#												
RESTR 003116	1382	1386	1394#											
RESOS = 104407	1054#	1722												
RETURN 001214	903#	1177#	1359*	1361	1394*	1431*	1434	1748*	1750	1783	1899*	1909*	1910	
RFLG 007060	1950#	2044	2092	2161	2209	2263	2285	2335	2380	2395	2444	2486	2497	
RING = 040000	951#	2874	2877	2980	3090	3200	3203							
RINTEN= 000100	959#	3434	3493	3577	3720	3802								
RPOKE 007020	1939#	1946	2052	2107	2125	2165	2215	2267	2289	2292	2339	2345	2384	
	2390	2399	2402	2408	2448	2451	2490	2501	2504	2507				
RSOM = 000400	972#	2058	2060	2120	2133	2623								
RTS = 000004	963#	2958	2983	2999	3178	3203	3213	3370						
RUN 001314	943#	1173*	1826	1829*	1830*	1837*	1838*							
RXCSR 001404	1073#	1398	1845*	1849	2040	2065	2087	2136	2157*	2168	2179	2205*	2218	
	2232	2258*	2270	2295	2307	2333*	2346	2377*	2409	2440	2481	2539*	2549	
	2560	2602*	2617	2619	2670*	2652	2690	2695	2783*	2812	2843*	2849	2949*	
	2955	3059*	3065	3169*	3175	3569*	3370*	3382*	3389*	3391*	3403	3407*	3433*	
	3434*	3466*	3492*	3493*	3532	3536*	3576*	3577*	3608	3719*	3720*	3765	3802*	
	3870*													
RXDBUF 001406	1074#	1851*	1852*	1853	1854	1855	2058	2061	2120	2133	2171	2180	2221	
	2233	2273	2281	2299	2308	2342	2347	2387	2391	2405	2410	2452	2491	
	2508	2552	2563	2622	2685	2693	2698	2815	3537*					
	967#	2357	2420	2453	2454	2509	2510							
RXDERR= 100000	958#	2053	2055	2101	2113	2116	2127	2129	2168	2183	2218	2236	2270	
RXDONE= 000200	2295	2311	2354	2417	3532	3765								
	941#	1266*	1272*	1274	1275	1301*	1327							
SAVACT 001312	942#	1168*	1268*	1302*	1381*	1384*								
SAVNUM 001313	928#	1556*	1754											
SAVPC 001266	937#													
SAVPCA 001306	921#	1565*	1570											
SAVRO 001250	930#													
SAVROA 001270	922#	1564*	1571	3862*	3924									
SAVR1 001252	931#													
SAVR1A 001272	923#	1563*	1572	3877*	3924									
SAVR2 001254	932#													
SAVR2A 001274	924#	1562*	1573	3905*	3908*	3924								
SAVR3 001256	933#													
SAVR3A 001276	925#	1561*	1574	3924										
SAVR4 001260	934#													
SAVR4A 001300	926#	1560*	1575	3924										
SAVR5 001262	935#													
SAVR5A 001302	927#													
SAVSP 001264	936#													
SAVSPA 001304	1052#	1682												
SAVOS = 104406	1040#	2069	2140	2187	2240	2315	2360	2424	2462	2517	2570	2635	2710	
SCOPE = 104400	2820	2922	3028	3138	3248	3286	3327	3409	3468	3539	3624	3678	3779	
	3836	3900												
SCOP1 = 104401	1042#	2629	2865	2891	2920	2971	2997	3026	3081	3107	3136	3191	3217	
	3246	3380	3401											
SEND = 000020	988#	2540	2603	2671	2782	3435	3494	3579	3647	3721	3803	3871		
SETFLG= 104413	1062#	1244	1248	1254	1258									
SETVEC 007232	1979#	3273	3305	3330	3364	3384	3428	3488	3542	3571	3643	3690	3715	
	3767	3797												
SFLG 007152	1966#	2211												
SHIFTS 001326	1013#	1921*	1929*	1945*	2051*	2106*	2124*	2164*	2214*	2266*	2298*	2291*	2338*	
	2344*	2383*	2399*	2398*	2401*	2407*	2447*	2450*	2489*	2500*	2503*	2506*		

CROSS REFERENCE TABLE -- USER SYMBOLS

SIMBCC	013422	2663	2720*	3880										
SMALL	005044	1771*	2039	2086	2156	2204	2257	2331	2375	2439	2480	2532	2569	2600
		2631	2668	2775	2848	2905	2954	3011	3064	3121	3174	3231	3272	3304
		3362	3426	3486	3570	3623	3642	3713	3774	3795	3835	3859	3899	
SPACNT=	004237	1592*	1613	1616*	1630*									
SRO	= 002000	955*	3093	3203										
SRJMF	001336	1017*	1259	3057	3167									
STACK	= 001150	816*	1166	1348	1749	1779	3268	3284	3326	3381	3408	3465	3538	3621
		3677	3778	3833										
STD	= 000010	962*	3068	3093	3109	3178	3203	3219						
STJMF	001334	1016*	1255	3055	3165									
STPSYN=	000400	957*												
STUFCK	006746	1321*												
SVOS	003744	1560*												
SWR	001202	890*	1182	1184*	1196	1200	1314	1322	1327	1332	1351	1413	1420	1441
		1454	1671	1676	1725	1744	1746	1881	3902					
SWREG	000176	878*	1184											
SW00	= 000001	797*	1200	1314										
SW01	= 000002	796*	1881											
SW02	= 000004	795*												
SW03	= 000010	793*	1322											
SW04	= 000020	792*												
SW05	= 000040	791*												
SW06	= 000100	790*												
SW07	= 000200	789*												
SW08	= 000400	788*	1744											
SW09	= 001000	787*	1441	3902										
SW10	= 002000	786*	1746											
SW11	= 004000	785*	1420											
SW12	= 010000	784*	1454	1671										
SW13	= 020000	783*	1676											
SW14	= 040000	782*												
SW15	= 100000	781*												
SYSTST=	004000	994*	1733	2603	3277	3309	3427	3487	3575	3647	3714	3796		
TABORT=	002000	1003*	3821											
TBLA	012556	2533	2534	2573*										
TBLB	013410	2654	2656	2657	2713*									
TCNFLG	001322	1010*	1249	1250	1263	1303*	1999	2015	2840	2946	3052	3162	3356	
TCRCIN=	010000	1001*												
TCRC7T=	020000	1000*												
TEMP	006054	1596	1804*	2221*	2222	2307*								
TEMP1	001236	916*	1239	1267	1268	1269*	1307*	1308	1312*	1799	2159*	2174*	2175	2207*
		2227*	2228	2260*	2276*	2277	2298*	2302*	2303	2346*	2351	2354	2595*	2597*
		2601	2634*	2636*	2637	2723*	2746*	2776*	2786*	2787	3388*			
TEMP2	001240	917*	1230	1233	1274*	1277*	1281*	1308*	1801	2179*	2183	2232*	2236	2311
		2409*	2414	2417	2596*	2609	2635*	2724*	2728*	2777*	2802*	2803		
TEMP3	001242	918*	2725*	2727	2737*	2739	2744*	2745*	2748	2778*	2815*	2816	3865*	3879*
		3891												
TEMP4	001244	919*	1950*	1961*	1966*	1975*	3866*	3877	3879*	3888	3894			
TEMP5	001246	920*	1951*	1956*	1957	1967*	1970*	1971	3867*	3910	3914*			
TEOM	= 001000	1004*	2680	2784	2810	3513	3893							
TIMER	= 004000	1002*	2613	2615	2854	2856	2867	2869	2880	2882	2995	2997	2908	2910
		2960	2962	2973	2975	2986	2988	3001	3003	3014	3016	3070	3072	3083
		3085	3096	3098	3111	3113	3124	3126	3180	3182	3193	3195	3206	3208
		3221	3223	3234	3236	3372	3374	3393	3395	3438	3440	3457	3459	3497
		3499	3517	3519	3527	3529	3582	3584	3601	3603	3616	3618	3652	3654

		3663	3665	3682	3684	3724	3726	3744	3746	3755	3757	3805	3807	3823
		3825												
TKCSR	001204	895#	1416	1475	3917									
TKDBR	001206	896#	1418	1477	1483	3920								
TLAST =	021576	1904	3924#											
TPCSR	001210	897#	1459	1481	1673									
TPDBR	001212	898#	1461*	1483*	1675*									
TRPOK	004330	1662#												
TRP.PC	006222	1810#												
TSON =	000400	1005#	2541	2606	2672	2784	3507	3649	3815	3872				
TSTNO	001226	908#	1176*	1757	1889	1895	1897	2036*	2083*	2153*	2201*	2254*	2329*	2372*
		2436*	2477*	2529*	2591*	2652*	2772*	2837*	2943*	3049*	3159*	3266*	3300*	3354*
		3422*	3482*	3566*	3638*	3709*	3791*	3856*						
TST1	007356	1892	1909	2036#										
TST10	011710	2373	2436#											
TST11	012062	2437	2477#											
TST12	012320	2478	2529#											
TST13	012570	2530	2591#											
TST14	013064	2592	2652#											
TST15	013602	2653	2772#											
TST16	014110	2773	2837#											
TST17	014556	2838	2943#											
TST2	007544	2037	2083#											
TST20	015224	2944	3049#											
TST21	015706	3050	3159#											
TST22	016370	3160	3266#											
TST23	016506	3267	3300#											
TST24	016700	3301	3354#											
TST25	017216	3355	3422#											
TST26	017476	3423	3482#											
TST27	020056	3483	3566#											
TST3	010034	2084	2153#											
TST30	020416	3567	3638#											
TST31	020724	3639	3709#											
TST32	021316	3710	3791#											
TST33	021576	3792	3856#	3924										
TST34 =	***** U	3857												
TST4	010262	2154	2201#											
TST5	010532	2202	2254#											
TST6	011140	2255	2328#											
TST7	011352	2329	2372#											
TTST	003174	1354*	1355*	1357*	1358*	1414#								
TWOSYM=	000000	846#												
TXACT =	001000	984#												
TXCSR	001412	1076#	1733*	1761*	1764*	1858*	1859*	1860	1939*	1942*	1952*	1954*	1959*	1968*
		1973*	2036*	2043*	2085*	2091*	2155*	2160*	2166*	2172*	2177*	2203*	2208*	2216*
		2225*	2230*	2256*	2262*	2268*	2274*	2279*	2293*	2300*	2305*	2330*	2334*	2340*
		2374*	2379*	2385*	2403*	2438*	2443*	2479*	2485*	2531*	2538*	2540*	2568*	2599*
		2603*	2604	2607	2610	2630*	2667*	2669*	2671*	2674	2774*	2780*	2782*	2790
		2794	2799	2806	2810*	2847*	2851*	2904*	2906*	2953*	2957*	3010*	3012*	3063*
		3067*	3120*	3122*	3173*	3177*	3230*	3232*	3271*	3277*	3279*	3285*	3303*	3309*
		3310*	3321	3325*	3361*	3363*	3425*	3427*	3435*	3445	3450	3453	3467*	3485*
		3487*	3494*	3504	3508	3511	3569*	3575*	3579*	3589	3594	3597	3611	3614*
		3622*	3641*	3647*	3673*	3679	3712*	3714*	3721*	3731	3736	3739	3773*	3794*
		3796*	3803*	3812	3816	3819	3834*	3858*	3869*	3871*	3898*			
TXCJF	001414	1077#	1862*	1863*	1864	2537*	2541*	2543*	2546*	2606*	2609*	2612*	2613	2615

CROSS REFERENCE TABLE -- USER SYMBOLS

	2672*	2678*	2680*	2779*	2784*	2797*	2854	2856	2867	2869	2880	2882	2895
	2897	2908	2910	2960	2962	2973	2975	2986	2988	3001	3003	3014	3016
	3070	3072	3083	3085	3096	3098	3111	3113	3124	3126	3180	3182	3193
	3195	3206	3208	3221	3223	3234	3236	3372	3374	3393	3395	3438	3440
	3449*	3452*	3455*	3457	3459	3497	3499	3507*	3510*	3513*	3517	3519	3527
	3529	3582	3584	3593*	3596*	3599*	3601	3603	3616	3618	3649*	3652	3654
	3663	3665	3680*	3682	3684	3724	3726	3735*	3738*	3741*	3744	3746	3755
	3757	3805	3807	3815*	3818*	3821*	3823	3825	3868*	3872*	3874*	3890*	3893*
TXDLAT= 100000	978#												
TXDONE= 000200	986#	3445	3504	3589	3611	3679	3731						
TXINTE= 000100	987#	3279	3310	3614	3647								
TYPDAT 004602	1697	1715	1718#										
TYPE = 104402	1044#	1194	1306	1324	1329	1353	1360	1372	1373	1375	1377	1379	1458
	1471	1490	1582	1619	1698	1699	1702	1703	1705	1707	1711	1716	1791
	1823	1883	1900	1901	1906	3912	3913						
	1695	1698#											
TYPMSG 004502	991#												
JSER = 000000	1590*	1620*	1628#										
WRDCNT 004234	1710	1713#											
WRKO.F 004570	1672	1674	1676#										
XBX 004374	1374	1396#											
XCSR 003130	1380	1405#											
XERR 003152	1306	1797#											
XHEAD 006014	1378	1402#											
XPASS 003144	2659*	2733	2744	2753#	3860*								
XPOLY 013574	1311	1797#											
XSTATQ 006042	1704	1755#											
XTSTN 005000	1376	1399#											
XVEC 003136	1#	2037	2038#	2084	2085#	2154	2155#	2202	2203#	2255	2256#	2329	2330#
SE = 000035	2373	2374#	2437	2438#	2478	2479#	2530	2531#	2592	2593#	2653	2654#	2773
	2774#	2838	2840#	2944	2946#	3050	3052#	3160	3162#	3267	3268#	3301	3302#
	3355	3356#	3423	3424#	3483	3484#	3567	3568#	3639	3640#	3710	3711#	3792
	3793#	3857	3858#										
	872	1192	1388#	1723									
SENCAC 003104	1#	2023	2030	2038#	2071	2077	2085#	2142	2147	2155#	2189	2195	2203#
SN = 000033	2242	2248	2256#	2317	2322	2330#	2361	2366	2374#	2425	2430	2438#	2464
	2471	2479#	2519	2523	2531#	2579	2585	2593#	2641	2646	2654#	2760	2766
	2774#	2823	2831	2840#	2929	2937	2946#	3035	3043	3052#	3145	3153	3162#
	3254	3260	3268#	3288	3294	3302#	3340	3348	3356#	3410	3416	3424#	3470
	3476	3484#	3553	3560	3568#	3626	3632	3640#	3697	3703	3711#	3780	3785
	3793#	3838	3850	3858#	3924#								
SRAYD = 177777	1#	2023#	2024	2027#	2071#	2072	2075#	2142#	2143	2146#	2189#	2190	2194#
	2242#	2243	2247#	2317#	2318	2321#	2361#	2362	2365#	2425#	2426	2428#	2464#
	2465	2469#	2519#	2520	2522#	2579#	2580	2583#	2641#	2642	2644#	2760#	2761
	2764#	2823#	2824	2829#	2929#	2930	2935#	3035#	3036	3041#	3145#	3146	3151#
	3254#	3255	3258#	3288#	3289	3292#	3340#	3341	3346#	3410#	3411	3414#	3470#
	3471	3474#	3553#	3554	3558#	3626#	3627	3630#	3697#	3698	3701#	3780#	3781
	3783#	3838#	3839	3849#									
SY = 000014	1031#	1040	1042#	1044#	1046#	1048#	1050#	1052#	1054#	1056#	1058#	1060#	1062#
	1064#												
.	857#	858	861#	868#	873#	876#	879#	883#	885#	939#	940#	941#	942#
	943#	944#	1010#	1011#	1016#	1017#	1098#	1100#	1101#	1102#	1104#	1105#	1106#
	1108#	1109#	1110#	1112#	1113#	1114#	1116#	1117#	1118#	1120#	1121#	1122#	1124#
	1125#	1126#	1128#	1129#	1130#	1201	1331	1805#	1807#	1809#	1910#	1925	2665#
	2666#	2841	2947	3053	3163	3357	3451	3454	3509	3512	3595	3599	3737
	3740	3817	3820	3889	3924#								

.BEGIN	002660	1347#		
.CNVRT	004034	1059	1583#	
.CONVR	004030	1057	1582#	
.EOP	002764	1368#	3857	
.ERRTA	023236	1690	3924#	
.HLT	004350	864	1671#	
.INSTE	003516	1049	1488#	
.INSTR	003412	1047	1467#	
.INST1	003432	1471#	1491	
.MSG	003434	1469#	1472#	
.PARAM	003536	1051	1499#	
.PFAIL	005050	862	1167	1776# 1780
.PKCLK	005006	1061	1758#	
.RESOS	003776	1055	1570#	
.SAVOS	003736	1053	1556#	
.SCOPE	003160	1041	1411#	
.SCOPI	003312	1043	1441#	
.SETFL	004242	1063	1639#	1651
.START	001562	830	1165#	1177
.TRPSR	004316	866	1659#	
.TRPTA	001344	1039#	1664	
.TYPE	003336	1045	1451#	

SSMALL	18	1769													
SSPACE	7648	2759													
SSUF	7648	1917													
STINTX	7648	3288													
STNT	7648	3626													
STRPDE	18	1040	1042	1044	1046	1048	1050	1052	1054	1056	1058	1060	1062		
STSTN	18	2030	2077	2147	2195	2248	2322	2366	2430	2471	2523	2585	2646	2766	2831
	2937	3043	3153	3260	3294	3348	3416	3476	3560	3632	3703	3785	3850		
STXINT	7648	3254													
SVARIA	18	882													
SWAIT	18	2613	2853	2866	2879	2894	2907	2959	2972	2985	3000	3013	3069	3082	3095
	3110	3123	3179	3192	3205	3220	3233	3371	3392	3436	3456	3495	3515	3527	3580
	3600	3616	3651	3662	3681	3722	3742	3754	3804	3822					
\$XZ	18	2023	2027	2071	2075	2142	2146	2189	2194	2242	2247	2317	2321	2361	2365
	2425	2428	2464	2469	2519	2522	2579	2583	2641	2644	2760	2764	2823	2829	2929
	2935	3035	3041	3145	3151	3254	3258	3288	3292	3340	3346	3410	3414	3470	3474
	3553	3558	3626	3630	3697	3701	3780	3783	3838	3849					

. ABS. 023512 000

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

DSKZ:DZDPCB, DSKZ:DZDPCB, SEQ/SOL/CRF/DOC/NL: TOC=DZDPCB.MAC, DZDPCB.P11
RUN-TIME: 10 15 1 SECONDS
RUN-TIME RATIO: 35/26=1.3
CORE USED: 27K (53 PAGES)

DOCUMENT PAGES: 96