

DUP11

SDLC DATA AND FUNCTION
MD-11-DZDPD-B

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

AUG 1977
digital
MADE IN USA

This microfiche card contains a grid of frames. The first column on the left contains 11 frames with titles such as 'FUNCTION', 'DESCRIPTION', and 'OPERATION'. The subsequent columns contain various data tables, diagrams, and lists of parameters. The text is small and difficult to read due to the high resolution of the microfiche format.

B01

EOF1DZDPCBSEQ

00010000

770712

PDP10 411

31HOR1DZDPCBSEQ

00010000

770712

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 DU11 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 DU11 ARE CORRECT IN ITS ENVIRONMENT. PARAMETERS MAY BE SET TO ALERT DIAGNOSTICS AS TO THE DU11 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 DU11'S IN A CHAIN MODE, I.E., RUNNING THE DIAGNOSTIC COMPLETELY FOR ONE DEVICE BEFORE STARTING THE NEXT.

DZDPD TESTS THE ABILITY OF THE DEVICE TO RUN A LIMITED SOLC PROTOCOL AND LONG DATA PATTERNS. SPECIFIC DATA PATTERNS ARE RUN TO PROVE BIT-STUFF CAPABILITY. THE EIA DATA GATES ARE PROVEN AND THE PRIORITY LOGIC FUNCTIONS ARE CHECKED.

DZDPD TESTS ALL THE FUNCTIONS OF DECMODE AND DOES LONG AND SHORT DATA TESTS INTERNALLY AND OVER THE CABLE, IF ATTACHED. BCC, USING THE CRC16 POLYNOMIAL IS CHECKED, AND THE DEVICE IS TESTED RUNNING A LIMITED DDCMP PROTOCOL.

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.

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 SOLC 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.

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

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)

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
201
202
203
204
205

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 DUP11 PARAMETER DIALOG OR A PREVIOUSLY RUN DUP11 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 DUP11 STATUS'

1500	160050	CSR OF FIRST DUP11
1502	000300	VECTOR OF FIRST DUP11
1504	140026	STATUS AND SYNC FOR FIRST DUP11
1506	160060	CSR OF SECOND DUP11
1510	000310	VECTOR OF SECOND DUP11
1512	140026	STATUS AND SYNC FOR SECOND DUP11

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.

GO1

DZDPD-B MACY11 27(1006) 18-MAY-77 00:04 PAGE 6
DZDPDB.P11 13-MAY-77 15:59

SEQ 0005

206
207

SW 06 SET: RESERVED
SW 05 SET: RESERVED

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

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 DUP11S BITS 00, 01, 02 WILL BE SET IN LOC 'DUPACTV' FROM THE SWITCH REGISTER. USING THIS SWITCH(SW03) ALTERS THAT LOCATION. THEREFORE, IF THREE DUP11S 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 DUP11S 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 DUP11S 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.

I01

DZDPD-B MACY11 27(1006) 18-MAY-77 00:04 PAGE 8
DZDPDB.P11 13-MAY-77 15:58

SEQ 0007

264
265

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

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

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 ENABELED 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'.

313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365

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 DUP11 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 DUP11 WAS DOING AT THE TIME OF THE ERROR.

366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405

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

DUP11 "PARAMETER DIALOG" MUST BE RUN ONLY ONCE PRIOR TO THE FIRST RUNNING OF ANY DUP11 DIAGNOSTIC IF "DEFAULT PARAMETERS" ARE NOT USED. IF ONLY DUP11 DIAGNOSTICS WERE LOADED AFTER DUP11 PARAMETER SETUP, AND IF CORE MEMORY HAS NOT BEEN CHANGED, I.E. USE OF DIAGNOSTICS OTHER THAN DUP11 DIAGNOSTICS, AND IF THERE WERE NO DUP11 CONFIGURATION CHANGES, THE DUP11 PARAMETER SETUP NEED NEVER BE RUN AGAIN. HOWEVER, IF ANY OF THE ABOVE HAVE BEEN VIOLATED THE DUP11 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 DUP11'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.

406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461

8.0 MISCELLANEOUS

8.1 EXECUTION TIME

ALL DUP11 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 PDP11 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 DUP11'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 DZDPDB 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 DUP11 CURRENTLY BEING TESTED. EXAMPLE: (RUN) /0000000001000000 MEANS THAT DUP11 NO.05 IS THE DUP11 NOW RUNNING.

DUPCRO0-DUPCRO7 (1500)-(1560) THESE LOCATIONS CONTAIN THE INFORMATION NEEDED TO TEST UP TO 8 (DECIMAL) DUP11S SEQUENTIALLY. THEY CONTAIN THE CSR, VECTOR AND STATUS CONCERNING THE CONFIGURATION OF EACH DUP11.

DUPACTV EACH BIT SET IN THIS LOCATION INDICATES THAT THE ASSOCIATED DUP11 WILL BE TESTED IN TURN. EXAMPLE: (DUPACTV) /0000000000011111 MEANS

NO1

DZDPD-B MACY11 27(1006) 18-MAY-77 00:04 PAGE 13
DZDPDB.P11 13-MAY-77 15:58

SEQ 0012

462
463

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

464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519

THAT DUP11 NO. 00,04 WILL BE TESTED.

RXCSR CONTAINS THE RECEIVER CSR OF THE CURRENT DUP11 UNDER TEST.

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

'MAP OF DUP11 STATUS'

1500	160050
1502	000300
1504	140000

THE ABOVE INFORMATION WILL BE REPEATED FOR EACH OF UP TO 8 DUP11'S IN THE SYSTEM (THESE WILL FOLLOW UNDER THIS TABLE).
EXPLANATION:

1500 160050 THIS IS THE SYSTEM CONTROL REGISTER FOR THE 1ST DUP11 IN THE SYSTEM.

1502 000300 THIS IS VECTOR 'A' FOR THE FIRST DUP11 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 DUP11 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

C02

DZDPD-B MACY11 27(1006) 18-MAY-77 00:04 PAGE 15
DZDPDB.P11 13-MAY-77 15:58

SEQ 0014

520
521

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

522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548

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.

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

DOCUMENT

COPYRIGHT 1977
DIGITAL EQUIPMENT CORPORATION
MAYNARD, MASS. 01754

594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645

1 MAINDEC-11-DZDPD-B /(<377>)/DUP-11 OFFLINE SOLC AND DEC MODE DATA
COPYRIGHT(C) 1975,1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS.

1207 ***** TEST 1 *****
PRIORITY INTERRUPT TEST.
SET PROCESSOR STATUS TO PRIORITY 7
AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.
1239 ***** TEST 2 *****
PRIORITY INTERRUPT TEST.
SET PROCESSOR STATUS TO PRIORITY 6
AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.
1273 ***** TEST 3 *****
PRIORITY INTERRUPT TEST.
SET PROCESSOR STATUS TO PRIORITY 5
AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.
1307 ***** TEST 4 *****
PRIORITY INTERRUPT TEST.
SET PROCESSOR STATUS TO PRIORITY 4
AND VERIFY THAT THE DUP11 WILL INTERRUPT.
1341 ***** TEST 5 *****
TEST TO PROVE THE HALF-DUPLEX FUNCTION
PROVE THAT THE RECEIVER WILL NOT RECOGNIZE
DATA IF SEND IS ASSERTED.
1416 ***** TEST 6 *****
TEST OF THE DUP RUNNING A BINARY COUNT
PATTERN WITHOUT A CRC CALCULATION
1513 ***** TEST 7 *****
TEST OF THE DUP RUNNING A BINARY COUNT
PATTERN WITH A CRC CALCULATION
1637 ***** TEST 10 *****
TEST OF THE DUP RUNNING A BINARY COUNT
PATTERN WITH A CRC CALCULATION
1763 ***** TEST 11 *****
THIS TEST WILL CHECK FOR ABORT SEQUENCE
OF THE DUP IN A DATA STREAM
1833 ***** TEST 12 *****
THIS TEST PROVES THE RECEIVER WILL STOP
ACCEPTING DATA IF SHUT OFF IN THE MIDDLE
OF A MESSAGE, AND THAT IT WILL NOT
RESTART UNTIL IT RECEIVES A FLAG

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

1961 ***** TEST 13 *****
THIS TEST WILL TRANSMIT CONTIGUOUS ONES CHARACTERS
IN SECONDARY MODE WITH A BCC CHECK.

2057 ***** TEST 14 *****
THIS TEST PROVES THE INTERACTION OF DEC MODE,
TSOM, SYNC, TXACT, TXDONE

2097 ***** TEST 15 *****
THIS TEST PROVES THE INTERACTION OF TEOM,
SEND, TXACT AND TXDONE IN DEC MODE.

2140 ***** TEST 16 *****
THIS TEST PROVES THAT THE DUP WILL NOT
SYNC UP IN LESS THAN TWO SYNCs

2210 ***** TEST 17 *****
THIS TEST PROVES THE RECEIVER WILL STRIP THE FIRST
TWO SYNCs AND WILL PRESENT ALL SUBSEQUENT SYNCs.

2258 ***** TEST 20 *****
THIS TEST PROVES THE DUP11 WILL
IDLE SYNCs. IDLE 64. SYNCs

2312 ***** TEST 21 *****
THIS TEST PROVES THE STRIP SYNC
FUNCTION OF THE RECEIVER. SYNC UP
THE RECEIVER, SEND DATA WITH A SYNC
CHARACTER IMBEDDED AND CHECK FOR
THE SYNC TO BE RECEIVED.

2425 ***** TEST 22 *****
THIS TEST PROVES THAT A BINARY COUNT
PATTERN CAN BE RUN IN DEC MODE
WITHOUT A BCC CALCULATION

2535 ***** TEST 23 *****
THIS TEST PROVES THAT A BINARY COUNT
PATTERN CAN BE RUN IN DEC MODE
WITH A BCC CALCULATION USING
THE CRC16 POLYNOMIAL

2683 ***** TEST 24 *****
TEST TO PROVE THE DEVICE IDLES SYNCs AND
WILL SHIFT OUT DATA AT THE APPROPRIATE TIME

2725 ***** TEST 25 *****
TEST TO PROVE THE DEVICE WILL
WORK WITH EVERY POSSIBLE SYNC CHARACTER

697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716

2788 ***** TEST 26 *****
THIS TEST PROVES THAT THE CRC ERROR BIT FUNCTIONS
CORRECTLY. FORCE AN ERROR AND VERIFY THE BIT.

2864 ***** TEST 27 *****
THIS TEST PROVES THE DEVICE WILL HANDLE THE
DDCMP PROTOCOL. SEND AND RECEIVE SYNCs,
FOLLOWED BY DATA,BCC,DATA AND FINAL BCC.

3018 ***** TEST 30 *****
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

2

DZDPD-B MACY11 27(1006) 18-MAY-77 00:04 PAGE 21
 DZDPDB.P11 13-MAY-77 15:58

SEG 0020

INTRODUCTION TO DUP11 DIAGNOSTIC

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

;*MAINDEC-11-DZDPD-B /<377>/DUP-11 OFFLINE SDLC AND DEC MODE DATA AND FUNCTION TESTS
 ;*COPYRIGHT(C) 1975,1977, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
 ;*

;STARTING PROCEDURE
 ;LOAD PROGRAM
 ;LOAD ADDRESS 000200
 ;PRESS START
 ;PROGRAM WILL TYPE "MAINDEC-11-DZDPD-B /<377>/DUP-11 OFFLINE SDLC AND DEC MODE D
 ;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
 ;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
 ;AND THEN RESUME TESTING

;SWITCH REGISTER OPTIONS

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

SW15=100000
SW14=40000
SW13=20000
SW12=10000
SW11=4000
SW10=2000
SW09=1000
SW08=400
SW07=200
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1

=1,HALT ON ERROR
 =1,LOOP ON CURRENT TEST
 =1,INHIBIT ERROR TYPEOUT
 =1,DELETE TYPEOUT/BELL ON ERROR.
 =1,INHIBIT ITERATIONS
 =1,ESCAPE TO NEXT TEST ON ERROR
 =1,LOOP WITH CURRENT DATA
 =1,LOOP ON ERROR
 ;SELECT DUP'S DESIRED ACTIVE
 ;NOTE:THIS MUST NOT EXCEED ORIGINAL COUNT
 ;LOCK ON TEST SELECT
 ;RESTART PROGRAM AT SELECTED TEST
 ;ENTER PARAMETERS

GENERAL DEFINITIONS AND EQUIVALENCIES

751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800

000000
000001
000002
000003
000004
000005
000006
000007

177776
001150

005746
005726
010046
012600
024646
022626

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

;REGISTER DEFINITIONS

```

;-----
R0=%0      ;GENERAL REGISTER
R1=%1      ;GENERAL REGISTER
R2=%2      ;GENERAL REGISTER
R3=%3      ;GENERAL REGISTER
R4=%4      ;GENERAL REGISTER
R5=%5      ;GENERAL REGISTER
SP=%6      ;PROCESSOR STACK POINTER
PC=%7      ;PROGRAM COUNTER
  
```

;LOCATION EQUIVALENCIES

```

;-----
PS=177776 ;PROCESSOR STATUS WORD
STACK=1150 ;START OF PROCESSOR STACK
  
```

;INSTRUCTION DEFINITIONS

```

;-----
PUSH1SP=5746 ;DECREMENT PROCESSOR STACK 1 WORD
POP1SP=5726  ;INCREMENT PROCESSOR STACK 1 WORD
PUSHRO=10046 ;SAVE R0 ON STACK
POPRO=12600  ;RESTORE R0 FROM STACK
PUSH2SP=24646 ;DECREMENT STACK TWICE
POP2SP=22626 ;INCREMENT STACK TWICE
.EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
  
```

```

BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10
BIT2=4
BIT1=2
BIT0=1
  
```

```

801      ;*****
802      ;-----
803      ; TRAPCATCHER FOR ILLEGAL INTERRUPTS
804      ; THE STANDARD "TRAP CATCHER" IS PLACED
805      ; BETWEEN ADDRESS 0 TO ADDRESS 776.
806      ; IT LOOKS LIKE "PC+2 HALT".
807      ;-----
808      ;*****
809
810      000000      .=0
811      ; STANDARD INTERRUPT VECTORS
812      ;-----
813
814      000024      .=24
815      000024 005050      .PFAIL      ; POWER FAIL HANDLER
816      000026 000340      340          ; SERVICE AT LEVEL 7
817      000030 004350      .HLT          ; ERROR HANDLER
818      000032 000340      340          ; SERVICE AT LEVEL 7
819      000034 004316      .TRPSRV     ; GENERAL HANDLER DISPATCH SERVICE
820      000036 000340      340          ; SERVICE AT LEVEL 7
821
822      000040 000000      .=40
823      000042 000000      0          ; SAVE FOR ACT-11 OR DDP2
824      000044 000000      0          ; RETURN ADDRESS IF UNDER ACT-11 OR DDP2
825      000046 003104      0          ; SAVE FOR ACT-11 OR DDP2
826      000052 000052      SENDAD      ; FOR USE WITH ACT-11 OR DDP2
827      000052 000000      0          ; ACT-11 PROGRAM CHARACTERISTICS
828
829      000174      .=174
830      000174 000000      DISPREG:0  ; SOFTWARE DISPLAY REGISTER
831      000176 000000      SWREG: 0      ; SOFTWARE SWITCH REGISTER
832
833      000200 000137 001562      .=200      JMP      .START      ; GO TO START OF PROGRAM
834
835
836      001000 001000      .=1000
837      001000 005377 040515 047111 MTITLE: .ASCIZ <377><12>/MAINDEC-11-DZDPD-B /<377>/DUP-11 OFFLINE SDLC AND DEC MODE DAT
838      001200      .=1200
839      ; SWR AND LIGHTS
840      ;-----
841
842      001200 177570      DISPLAY: 177570      ; 11/45 CONSOLE LIGHTS
843      001202 177570      SWR: 177570      ; INDIRECT POINTER TO SWITCH REGISTER
844
845      ; INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
846      ;-----
847
848      001204 177560      TKCSR: 177560      ; TELETYPE KEYBOARD CONTROL REGISTER
849      001206 177562      TKDBR: 177562      ; TELETYPE KEYBOARD DATA BUFFER
850      001210 177564      TPCSR: 177564      ; TELEPRINTER CONTROL REGISTER
851      001212 177566      TPDBR: 177566      ; TELEPRINTER DATA BUFFER
852
853      ; PROGRAM CONTROL PARAMETERS
854      ;-----
855

```


PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

856	001214	000000	RETURN:	0	; SCOPE ADDRESS FOR LOOP ON TEST
857	001216	000000	NEXT:	0	; ADDRESS OF NEXT TEST TO BE EXECUTED
858	001220	000000	LOCK:	0	; ADDRESS FOR LOCK ON CURRENT DATA
859	001222	000001	ICOUNT:	1	; NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
860	001224	000000	LPCNT:	0	; NUMBER OF ITERATIONS COMPLETED
861	001226	000000	TSTNO:	0	; NUMBER OF TEST IN PROGRESS
862	001230	000000	PASCNT:	0	; NUMBER OF PASSES COMPLETED
863	001232	000000	ERRCNT:	0	; TOTAL NUMBER OF ERRORS
864	001234	000000	LSTERR:	0	; PC OF LAST ERROR CALL
865					
866			; PROGRAM VARIABLES		
867			; -----		
868					
869	001236	000000	TEMP1:	0	; TEMPORARY STORAGE
870	001240	000000	TEMP2:	0	; TEMPORARY STORAGE
871	001242	000000	TEMP3:	0	; TEMPORARY STORAGE
872	001244	000000	TEMP4:	0	; TEMPORARY STORAGE
873	001246	000000	TEMP5:	0	; TEMPORARY STORAGE
874	001250	000000	SAVR0:	0	; R0 STORAGE
875	001252	000000	SAVR1:	0	; R1 STORAGE
876	001254	000000	SAVR2:	0	; R2 STORAGE
877	001256	000000	SAVR3:	0	; R3 STORAGE
878	001260	000000	SAVR4:	0	; R4 STORAGE
879	001262	000000	SAVR5:	0	; R5 STORAGE
880	001264	000000	SAVSP:	0	; STACK POINTER STORAGE
881	001266	000000	SAVPC:	0	; PROGRAM COUNTER STORAGE
882					
883	001270	000000	SAVR0A:	0	; R0 STORAGE
884	001272	000000	SAVR1A:	0	; R1 STORAGE
885	001274	000000	SAVR2A:	0	; R2 STORAGE
886	001276	000000	SAVR3A:	0	; R3 STORAGE
887	001300	000000	SAVR4A:	0	; R4 STORAGE
888	001302	000000	SAVR5A:	0	; R5 STORAGE
889	001304	000000	SAVSPA:	0	; STACK POINTER STORAGE
890	001306	000000	SAVPCA:	0	; PROGRAM COUNTER STORAGE
891					
892	001310	000001	DUPACTV:	.BLKB 1	; DUP11'S SELECTED ACTIVE.
893	001311	000001	DUPNUM:	.BLKB 1	; OCTAL NUMBER OF DUP11'S.
894	001312	000001	SAVACT:	.BLKB 1	; ORIGINAL ACTV. DEVICES.
895	001313	000001	SAVNUM:	.BLKB 1	; WORKABLE NUMBER.
896	001314	000001	RUN:	.BLKB 1	; POINTER ONE PAST RUNNING DEVICE.
897		001316	.EVEN		
898	001316	001500	CREAM:	DUP.MAP	; TABLE POINTER.

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```
899  
900  
901  
902  
903  
904  
905  
906  
907  
908  
909  
910  
911  
912  
913  
914  
915  
916  
917  
918  
919  
920  
921  
922  
923  
924  
925  
926  
927  
928  
929  
930  
931  
932  
933  
934  
935  
936  
937  
938  
939  
940  
941  
942  
943  
944  
945  
946  
947  
948  
949  
950  
951  
952  
953  
954
```

;CONTROL REGISTER DEFINITIONS

;RXCSR BIT DEFINITIONS

100000	DSCA=BIT15	;DATA SET CHANGE A
040000	RING=BIT14	;RING
020000	CTS=BIT13	;CLR TO SEND
010000	CARDET=BIT12	;CARRIER DETECT
004000	REACT=BIT11	;REC ACTIVE
002000	SRD=BIT10	;SEC REC DATA
001000	DSR=BIT9	;DATA SET RDY
000400	STP^YN=BIT8	;STRIP SYNC
000200	RXDONE=BIT7	;REC DONE
000100	RINTEN=BIT6	;REC INTR ENABLE
000040	DSINTE=BIT5	;DSC INTR ENABLE
000020	RCVEN=BIT4	;REC ENABLE
000010	STD=BIT3	;SEC XMIT DATA
000004	RTS=BIT2	;REQ TO SEND
000002	DTR=BIT1	;DATA TERM RDY
000001	DSCB=BIT0	;DATA SET CHANGE B

;RXDBUF BIT DEFINITIONS

100000	RXDERR=BIT15	;REC DATA ERROR
040000	OVRUN=BIT14	;OVERRUN ERROR
010000	CRCERR=BIT12	;CRC ERROR
002000	RABORT=BIT10	;REC ABORT
001000	REOM=BIT9	;REC END OF MESSAGE
000400	RSOM=BIT8	;REC START OF MESSAGE

;PARCSR BIT DEFINITIONS

100000	DECMOD=BIT15	;DEC MODE (DDCMP)
001000	CRCEN=BIT9	;CRC ENABLE
010000	PRISEC=BIT12	;PRI/SEC SELECT

;TXCSR BIT DEFINITIONS

100000	TXDLAT=BIT15	;TX DATA LATE
040000	MTDATA=BIT14	;MAINT DATA OUT
020000	CLK=BIT13	;CLK
010000	MMODEB=BIT12	;MAINT MODE B
004000	MMODEA=BIT11	;MAINT MODE A
002000	BITW=BIT10	;BIT WINDOW INPUT
001000	TXACT=BIT9	;TX ACTIVE
000400	MRESET=BIT8	;MASTER RESET
000200	TXDONE=BIT7	;XMIT DONE
000100	TXINTE=BIT6	;XMIT DONE INTR ENABLE
000020	SEND=BIT4	;SEND
000010	HDXEN=BIT3	;HDX/FDX

;TXCSR WRD DEFINITIONS

000000	USER=0	;USER MODE
014000	MMODE=14000	;MAINT INT MODE
010000	MEXT=10000	;MAINT EXT MODE
004000	SYSTST=4000	;SYSTEM TEST MODE

;TXDBUF BIT DEFINITIONS

100000	RCRC7T=BIT15	
040000	RCRCIN=BIT14	
020000	TCRC7T=BIT13	
010000	TCRCIN=BIT12	

PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

955		004000
956		002000
957		001000
958		000400
959		
960		
961		
962	001320	000000
963	001322	000001
964	001323	000001
965	001324	000000
966	001326	000000
967	001330	000000
968	001332	000000
969	001334	000001
970	001336	000001
971		
972		

```
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 0
MIND: .WORD 0
FLAG: .WORD 0
STJMFL: .BLKW 1
SRJMFL: .BLKW 1
```

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

001340 000
001341 000
001342 000
001343 000

000000

PROGRAM CONTROL FLAGS

INIFLG: .BYTE 0
ERRFLG: .BYTE 0
LOKFLG: .BYTE 0
QV.FLG: .BYTE 0

PROGRAM INITIALIZATION FLAG
ERROR OCCURED FLAG
LOCK ON CURRENT TEST FLAG
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
SCOP1=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER
.SCOP1
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 INPUT 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

```

1020                                     ;DUP11 VECTOR AND REGISTER INDIRECT POINTERS
1021
1022 001374 000000 DUPRVC: 0 ; POINTER TO DUP11 RECEIVER INTERRUPT VECTOR
1023 001376 000000 DUPRPS: 0 ; POINTER TO DUP11 RECEIVER INTERRUPT SERVICE PS
1024 001400 000000 DUPTVC: 0 ; POINTER TO DUP11 TRANSMITTER INTERRUPT VECTOR
1025 001402 000000 DUPTPS: 0 ; POINTER TO DUP11 TRANSMITTER INTERRUPT SERVICE PS
1026 001404 000000 RXCSR: 0 ; POINTER TO DUP11 RECEIVER STATUS REGISTER
1027 001406 000000 RXDBUF: 0 ; POINTER TO DUP11 RECEIVER DATA BUFFER
1028 001410 000000 PARCSR: 0 ; POINTER TO DUP11 PARAMETER STATUS REGISTER
1029 001412 000000 TXCSR: 0 ; POINTER TO DUP11 TRANSMITTER STATUS REGISTER
1030 001414 000000 TXDBUF: 0 ; POINTER TO DUP11 TRANSMITTER DATA BUFFER
1031 001416 000000 DUPSEC: 0 ; POINTER TO DUP11 SECONDARY REGISTER SELECT REGISTER
1032 001420 000000 HUPPSR: 0 ; POINTER TO PARAMETER STATUS HIGH BYTE
1033 001422 000000 HUPRBF: 0 ; POINTER TO RECEIVER BUFFER HIGH BYTE
1034 001424 000000 HUPRCR: 0 ; POINTER TO RECEIVER CONTROL REG HIGH BYTE
1035 001426 000000 HUPTBF: 0 ; POINTER TO TRANSMITTER BUFFER HIGH BYTE
1036 001430 000000 HUPTCR: 0 ; POINTER TO TRANSMITTER CONTROL REG HIGH BYTE
1037
1038
1039                                     ;DUP11 CONTROL INDICATORS FOR CURRENT DUP11 UNDER TEST
1040                                     ;-----
1041
1042 001432 000 MASK.A: .BYTE 000 ; LAST CHAR TO TEST AND PARITY MASK
1043
1044 001433 010 CLK.A: .BYTE 8. ; NUMBER OF CLOCKS NEEDED FOR ONE CHAR
1045
1046 001434 000000 LOO.OO: 000000 ; PARAMETERS
1047

```

;DUP11 STATUS TABLE AND ADDRESS ASSIGNMENTS

```

1048
1049
1050
1051      001500      001500
1052      001500      000001      . =1500
1053      001500      000001      DUP.MAP:
1054      001502      000001      DUPCR0: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 0
1055      001504      000001      DUPTR0: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 0
1056      001504      000001      DUPO.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 0
1057      001506      000001      DUPCR1: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 1
1058      001510      000001      DUPTR1: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 1
1059      001512      000001      DUP1.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 1
1060
1061      001514      000001      DUPCR2: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 2
1062      001516      000001      DUPTR2: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 2
1063      001520      000001      DUP2.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 2
1064
1065      001522      000001      DUPCR3: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 3
1066      001524      000001      DUPTR3: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 3
1067      001526      000001      DUP3.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 3
1068
1069      001530      000001      DUPCR4: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 4
1070      001532      000001      DUPTR4: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 4
1071      001534      000001      DUP4.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 4
1072
1073      001536      000001      DUPCR5: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 5
1074      001540      000001      DUPTR5: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 5
1075      001542      000001      DUP5.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 5
1076
1077      001544      000001      DUPCR6: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 6
1078      001546      000001      DUPTR6: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 6
1079      001550      000001      DUP6.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 6
1080
1081      001552      000001      DUPCR7: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 7
1082      001554      000001      DUPTR7: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 7
1083      001556      000001      DUP7.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 7
1084
1085      001560      000000      DUP.END:      000000
1086
1087
1088
1089
1090

```

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	C	O	N	T	R	O	L	I	R	E	G	I	I	S	T
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

DEFINITIONS

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

```

1110
1111 ;PROGRAM INITIALIZATION
1112 ;LOCK OUT INTERRUPTS
1113 ;SET UP PROCESSOR STACK
1114 ;SET UP POWER FAIL VECTOR
1115 ;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
1116 ;TYPE TITLE MESSAGE
1117
1118 001562 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
1119 001570 012706 001150 MOV #STACK,SP ;SET UP STACK
1120 001574 012737 005050 000024 MOV #.PFAIL,2#24 ;SET UP POWER FAIL VECTOR
1121 001602 113737 001311 001313 MOV# DUPNUM,SAVNUM ;SAVE NUMBER OF DEVICES IN SYSTEM
1122 001610 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
1123 001614 105037 001341 CLRB ERRFLG ;CLEAR ERROR FLAG
1124 001620 105037 001343 CLRB QV.FLG ;ZERO QUICK VERIFY FLAG
1125 001624 012737 001500 001316 MOV #DUP.MAP,CREAM ;GET MAP POINTER.
1126 001632 112737 000001 001314 MOV# #1,RUN ;POINT POINTER TO FIRST DEVICE.
1127 001640 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
1128 001644 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
1129 001650 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
1130 001656 012737 001562 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
1131 ;TESTING STARTS
1132 001664 013746 000006 MOV 2#6,-(SP) ;SAVE CURRENT VECTORS
1133 001670 013746 000004 MOV 2#4,-(SP)
1134 001674 012737 001710 000004 MOV #12$,2#4 ;SETUP FOR TIMEOUT
1135 001702 005777 177274 TST 2$SWR ;REFERENCE HARDWARE SWITCH REG
1136 001706 000407 BR 13$ ;BR IF IT EXISTS
1137 001710 012737 000176 001202 12$: MOV #SWREG,SWR ;POINT TO SOFT SWR
1138 001716 012737 000174 001200 MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
1139 001724 022626 CMP (SP)+,(SP)+ ;ADJUST STACK
1140 001726 012637 000004 13$: MOV (SP)+,2#4 ;RESTORE VECTORS
1141 001732 012637 000006 MOV (SP)+,2#6
1142 001736 105737 001340 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
1143 001742 001401 BEQ 11$
1144 001744 000410 BR 6$
1145 001746 022737 003104 000042 11$: CMP #SENDAD,2#42 ;IF ACT-11 AUTO MODE,
1146 001754 001404 BEQ 6$ ;DON'T TYPE ID
1147 001756 104402 001000 TYPE #MTITLE ;TYPE TITLE MESSAGE
1148 001762 105137 001340 COMB INIFLG ;IF NOT SET FLAG AND DO
1149 001766 105777 177210 6$: TSTB 2$SWR ;BIT7=1??
1150 001772 100002 BPL 10$
1151 001774 000137 002520 JMP 1$
1152 002000 10$:
1153 002000 032777 000001 177174 BIT #SW00,2$SWR ;ENTER PARAMETERS
1154 002006 001002 BNE .+6 ;YES
1155 002010 000137 002360 JMP 21$ ;NO
1156 002014 105137 001332 COMB FLAG
1157 002020 112737 000001 001340 MOV# #1,INIFLG ;SET TO MANUAL ENTRY
1158 002026 012700 001500 MOV #DUP.MAP,RO ;CLR MAP
1159 002032 005020 68$: CLR (RO)+
1160 002034 020027 001560 CMP RO,#DUP.END ;DONE WITH MAP?
1161 002040 001374 BNE 68$ ;BR IF NO
1162 002042 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1163 002044 005473 MCSR ;MESSAGE
1164 002046 104405 PARAM ;CONVERT STRING
1165 002050 160000 160000 ;LOW LIMIT
    
```


PROGRAM INITIALIZATION AND START UP.

1166	002052	175500			175500	:HIGH LIMIT
1167	002054	001500			DUPCRO	:STORE AT THIS LOCATION
1168	002056	001			.BYTE 1	:MASK
1169	002057	001			.BYTE 1	:HOW MANY TIMES + 2
1170	002060	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1171	002062	005512			MVEC	:MESSAGE
1172	002064	104405			PARAM	:CONVERT STRING
1173	002066	000300			300	:LOW LIMIT
1174	002070	000770			770	:HIGH LIMIT
1175	002072	001502			DUPTRO	:STORE AT THIS LOCATION
1176	002074	001			.BYTE 1	:MASK
1177	002075	001			.BYTE 1	:HOW MANY TIMES + 2
1178	002076	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1179	002100	005702			MPAR	:MESSAGE
1180	002102	104405			PARAM	:CONVERT STRING
1181	002104	000004			4	:LOW LIMIT
1182	002106	000007			7	:HIGH LIMIT
1183	002110	001240			TEMP2	:STORE AT THIS LOCATION
1184	002112	000			.BYTE 0	:MASK
1185	002113	001			.BYTE 1	:HOW MANY TIMES + 2
1186	002114	013737	001240	001320	MOV	TEMP2,PRIRTY :SAVE PRIORITY
1187	002122	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1188	002124	005647			MTOTAL	:MESSAGE
1189	002126	104405			PARAM	:CONVERT STRING
1190	002130	000001			1	:LOW LIMIT
1191	002132	000010			8.	:HIGH LIMIT
1192	002134	001236			TEMP1	:STORE AT THIS LOCATION
1193	002136	000			.BYTE 0	:MASK
1194	002137	001			.BYTE 1	:HOW MANY TIMES + 2
1195	002140	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1196	002142	005525			MJMPR	:MESSAGE
1197	002144	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1198	002146	001323			OPCLRJ	:THIS FLAG
1199	002150	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1200	002152	005600			MTCN	:MESSAGE
1201	002154	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1202	002156	001322			TCNFLG	:THIS FLAG
1203	002160	105737	001322		TSTB	TCNFLG
1204	002164	001410			BEQ	71\$
1205	002166	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1206	002170	005726			MSTJM	:MESSAGE
1207	002172	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1208	002174	001334			STJMFL	:THIS FLAG
1209	002176	104403			INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1210	002200	005761			MSRJM	:MESSAGE
1211	002202	104413			SETFLG	:SET FLAG BASED UPON INPUT STRING
1212	002204	001336			SRJMFL	:THIS FLAG
1213	002206	105737	001323		TSTB	OPCLRJ
1214	002212	001403			BEQ	69\$
1215	002214	052737	100000	001504	BIS	#BIT15,DUPO.A
1216	002222	105737	001322		TSTB	TCNFLG
1217	002226	001403			BEQ	70\$
1218	002230	052737	040000	001504	BIS	#BIT14,DUPO.A
1219	002236	112737	000001	001312	MOVB	#1,SAVACT
1220	002244	113737	001236	001311	MOVB	TEMP1,DUPNUM
1221	002252	113737	001236	001313	MOVB	TEMP1,SAVNUM

1222	002260	005337	001236		65\$:	DEC	TEMP1		
1223	002264	001404				BEQ	64\$		
1224	002266	000261				SEC			
1225	002270	106137	001312			ROLB	SAVACT		
1226	002274	000771				BR	65\$		
1227	002276	113737	001312	001240	64\$:	MOVB	SAVACT,TEMP2	;# OF TIMES	
1228	002304	113737	001312	001310		MOVB	SAVACT,DUPACTV		
1229	002312	000241				CLC			
1230	002314	106037	001240			RORB	TEMP2		
1231	002320	012700	001500			MOV	#DUPCRO,RO		
1232	002324	012701	001506			MOV	#DUPCRI,RI		
1233	002330	000241			67\$:	CLC			
1234	002332	106037	001240			RORB	TEMP2		
1235	002336	103051				BCC	66\$		
1236	002340	012011				MOV	(RO)+,(R1)		
1237	002342	062721	000010			ADD	#10,(R1)+	;CSR	
1238	002346	012011				MOV	(RO)+,(R1)		
1239	002350	062721	000010			ADD	#10,(R1)+	;VECTOR	
1240	002354	012021				MOV	(RO)+,(R1)+	;PARAMETERS	
1241	002356	000764				BR	67\$		
1242	002360	012700	001500		21\$:	MOV	#DUP.MAP,RO	;SETUP TO CLEAR MAP	
1243	002364	005020			20\$:	CLR	(RO)+	;CLEAR	
1244	002366	020027	001560			CMP	RO,#DUP.END	;CHECK FOR FINISH	
1245	002372	001374				BNE	20\$;BR IF MORE TO GO	
1246	002374	012700	001500			MOV	#DUP.MAP,RO	;SETUP TO DEFAULT	
1247	002400	012710	160050			MOV	#160050,(RO)	;LOAD CSR	
1248	002404	012760	000770	000002		MOV	#770,2(RO)	;LOAD VECTOR	
1249	002412	012760	140026	000004		MOV	#140026,4(RO)	;LOAD PARAMETERS AND SYNC	
1250	002420	112737	000005	001320		MOVB	#5,PRIATY	;LOAD PRIORITY	
1251	002426	012700	000001			MOV	#1,RO	;SAVE CORE THIS WAY	
1252	002432	110037	001310			MOVB	RO,DUPACTV	;PRESET PROGRAM CONTROLS	
1253	002436	110037	001311			MOVB	RO,DUPNUM	;DITTO	
1254	002442	110037	001312			MOVB	RO,SAVACT	;DITTO	
1255	002446	110037	001313			MOVB	RO,SAVNUM	;DITTO	
1256	002452	110037	001322			MOVB	RO,TCNFLAG	;DITTO	
1257	002456	110037	001323			MOVB	RO,OPCLRJ	;DITTO	
1258	002462				66\$:				
1259	002462	104402	006014		16\$:	TYPE	,XHEAD	;TYPE HEADER	
1260	002466	012737	001500	001236		MOV	#DUP.MAP,TEMP1	;SET POINTER	
1261	002474	017737	176536	001240	5\$:	MOV	TEMP1,TEMP2	;SET DATA	
1262	002502	001406				BEQ	1\$;ALL DONE WITH DATA	
1263	002504	104410				CONVRT			
1264	002506	006042				XSTATQ			
1265	002510	062737	000002	001236		ADD	#2,TEMP1	;UPDATE POINTER	
1266	002516	000766				BR	5\$		
1267	002520	032777	000001	176454	1\$:	BIT	#SW00,DSWR		
1268	002526	001405				BEQ	7\$		
1269	002530	005737	001332			TST	FLAG		
1270	002534	001002				BNE	7\$		
1271	002536	000137	002000			JMP	10\$		
1272	002542	005037	001332		7\$:	CLR	FLAG		
1273	002546	005737	000042			TST	#42	;IS PROGRAM RUNNING UNDER MONITOR	
1274	002552	001030				BNE	3\$;BR IF YES	
1275	002554	032777	000010	176420		BIT	#SW03,DSWR	;SELECT SPECIFIC DEVICES??	
1276	002562	001424				BEQ	3\$;BR IF NO.	
1277	002564	104402	005413			TYPE	,MNEW	;TYPE THE MESSAGE.	

1278	002570	005000			CLR	RO				:ZERO DATA LIGHTS
1279	002572	000000			HALT					:WAIT FOR USER TO TELL WHAT DEVICES TO RUN
1280	002574	127737	176402	001312	CMPB	ASWR, SAVACT				:IS THE NUMBER VALID?
1281	002602	101404			BLOS	2\$:BR IF NUMBER IS OK.
1282	002604	104402	005254		TYPE	,MERR3				:TELL USER OF INVALID NUMBER.
1283	002610	000000			HALT					:STOP EVERY THING.
1284	002612	000776			BR	-2				:RESTART THE PROGRAM AGAIN.
1285	002614	117737	176362	001310	2\$:	MOV B	ASWR, DUPACTV			:GET NEW DEVICE PATTERN
1286	002622	113700	001310		MOV B	DUPACTV, RO				:SHOW THE USER WHAT HE SELECTED.
1287	002626	042700	177400		BIC	#1C<377>, RO				:USE ONLY LOW BYTE.
1288	002632	000000			HALT					:CONTINUE DYNAMIC SWITCHES.
1289	002634	012700	000300		3\$:	MOV	#300, RO			:PREPARE TO CLEAR THE FLOATING
1290	002640	012701	000302		MOV	#302, R1				:VECTOR AREA. 300-776
1291	002644	010120			4\$:	MOV	R1, (RO)+			:START PUTTING "PC+2 - HALT"
1292	002646	005021			CLR	(R1)+				:IN VECTOR AREA.
1293	002650	022021			CMP	(RO)+, (R1)+				:POP POINTERS
1294	002652	022700	001000		CMP	#1000, RO				:ALL DONE??
1295	002656	001372			BNE	4\$:BR IF NO.
1296										
1297										
1298										
1299										
										:TEST START AND RESTART

1300	002660	012737	000340	177776	.BEGIN:	MOV	#340, PS			:LOCK OUT INTERRUPTS
1301	002666	012706	001150		MOV	#STACK, SP				:SET UP STACK
1302	002672	005737	000042		TST	#42				:IS PROGRAM UNDER MONITOR CONTROL
1303	002676	001023			BNE	2\$:BR IF YES
1304	002700	032777	000004	176274	BIT	#BIT2, ASWR				:CHECK FOR LOCK ON TEST
1305	002706	001411			BEQ	1\$:BR IF NO LOCK DESIRED.
1306	002710	104402	005312		TYPE	, MLOCK				:TYPE LOCK SELECTED.
1307	002714	012737	000240	003174	MOV	#NOP, TTST				:ADJUST SCOPE ROUTINE.
1308	002722	012737	000240	003176	MOV	#NOP, TTST+2				:SET UP TO LOCK
1309	002730	000406			BR	2\$:CONTINUE ALONG.
1310	002732	013737	003306	003174	1\$:	MOV	BRW, TTST			:PREPARE NORMAL SCOPE ROUTINE
1311	002740	013737	003310	003176	MOV	BRX, TTST+2				:LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1312	002746	012737	006224	001214	2\$:	MOV	#CYCLE, RETURN			:START AT "CYCLE" FIND WHICH DEVICE TO TEST
1313	002754	104402	005202		TYPE	MR				:TYPE R
1314	002760	000177	176230		JMP	RETURN				:START TESTING

```

1315 ;END OF PASS
1316 ;TYPE NAME OF TEST
1317 ;UPDATE PASS COUNT
1318 ;CHECK FOR EXIT TO ACT-11
1319 ;RESTART TEST
1320
1321 002764 005037 001234 .EOP: CLR LSTERR ;CLEAR LAST ERROR PC
1322 002770 105037 001341 CLR ERRFLG ;CLEAR ERROR FLAG
1323 002774 005237 001230 INC PASCNT ;UPDATE PASS COUNT
1324 003000 013777 001230 176172 MOV PASCNT, @DISPLAY ;DISPLAY PASS COUNT
1325 003006 104402 005157 TYPE ,MEPASS ;TYPE END PASS
1326 003012 104402 005341 TYPE ,MCSRX ;TYPE CSR
1327 003016 104411 003130 CNVRT ,XCSR ;SHOW IT
1328 003022 104402 005347 TYPE ,MVECX ;TYPE VECTOR
1329 003026 104411 003136 CNVRT ,XVEC ;SHOW IT
1330 003032 104402 005355 TYPE ,MPASSX ;TYPE PASSES
1331 003036 104411 003144 NVRT ,XPASS ;SHOW IT
1332 003042 104402 005366 TYPE ,MERRX ;TYPE ERRORS
1333 003046 104411 003152 CNVRT ,XERR ;SHOW IT
1334 003052 105337 001313 DECB SAVNUM ;ARE ALL DEVICES TESTED?
1335 003056 001017 BNE RESTR ;BR IF NO.
1336 003060 112737 000377 001343 MOVB #377, QV.FLG ;SET THE QUICK VERIFY FLAG.
1337 003066 113737 001311 001313 MOVB DUPNUM, SAVNUM ;RESTORE THE COUNT
1338 003074 013701 000042 MOV @#42, R1 ;CHECK FOR ACT-11 OR DDP
1339 003100 001406 BEQ RESTR ;IF NOT, CONTINUE TESTING
1340 003102 000005 RESET ;STOP THE SHOW--CLEAR THE WORLD
1341 003104
1342 003104 004711 $ENDAD: JSR PC, (R1)
1343 003106 000240 NOP
1344 003110 000240 NOP
1345 003112 000240 NOP
1346 003114 000240 NOP
1347 003116 012737 006224 001214 RESTR: MOV #CYCLE, RETURN
1348 003124 000137 006224 JMP CYCLE
1349 003130 000001 XCSR: 1
1350 003132 006 002 .BYTE 6,2
1351 003134 001404 RXCSR
1352 003136 000001 XVEC: 1
1353 003140 003 002 .BYTE 3,2
1354 003142 001374 DUPRVC
1355 003144 000001 XPASS: 1
1356 003146 006 002 .BYTE 6,2
1357 003150 001230 PASCNT
1358 003152 000001 XERR: 1
1359 003154 006 002 .BYTE 6,2
1360 003156 001232 ERRCNT
1361
1362 ;SCOPE LOOP AND INTERATION HANDLER
1363
1364 003160 005037 001234 .SCOPE: CLR LSTERR ;CLEAR LAST ERROR PC
1365 003164 010016 MOV RO, (SP) ;SAVE RO ON STACK
1366 003166 032777 040000 176006 BIT #BIT14, @SWR ;LOOP ON TEST?
1367 003174 001407 TTST: BEQ 1$ ;BR IF NO (IF LOCK SW01 = 1; THIS LOCATION = 240)
1368 003176 000437 BR 3$ ;GO TO 3$ (DITTO)
1369 003200 105777 176000 TSTB @TKCSR ;KYBD DONE?
1370 003204 100034 BPL 3$ ;BR IF NO (LOCK: HIT A KEY ON TTY TO GO TO NEXT TEST)

```

```

1371 003206 017700 175774      MOV      @TKDBR,RO      ;CLR DONE BIT
1372 003212 000415              BR      2$             ;CONTINUE
1373 003214 032777 004000 175760 1$:  BIT      #SW11,@SWR    ;DELETE ITERATION (QUICK PASS)?
1374 003222 001011              BNE     2$             ;BR IF YES
1375 003224 105737 001343      TSTB   QV.FLG         ;HAS FIRST PASS BEEN COMPLETED?
1376 003230 001406              BEQ     2$             ;BR IF QUICK VERIFY
1377 003232 005237 001224              INC     LPCNT          ;UPDATE ITERATION COUNTER
1378 003236 023737 001224 001222  CMP     LPCNT,ICOUNT   ;ALL ITERATIONS DONE?
1379 003244 001014              BNE     3$             ;BR IF NOT YET
1380 003246 105037 001341      CLRB   ERRFLG        ;PREPARE FOR NEW TEST
1381 003252 005037 001224      CLR    LPCNT         ;START ICOUNT AT ZERO
1382 003256 005037 001220      CLR    LOCK
1383 003262 012737 000050 001222  MOV     #50,ICOUNT    ;RESET ITERATIONS
1384 003270 013737 001216 001214  MOV     NEXT,RETURN   ;GET NEXT TEST
1385 003276 011600      3$:  MOV     (SP),RO      ;POP RO OFF STACK
1386 003300 022626      POP2SP ;FAKE AN RTI
1387 003302 000177 175706      JMP     @RETURN       ;GO DO THE TEST
1388 003306 001407      BRW:   1407
1389 003310 000437      BRX:   437
1390
1391                                ;CHECK FOR FREEZE ON CURRENT DATA
1392                                -----
1393
1394 003312 032777 001000 175662 .SCOP1: BIT      #SW09,@SWR    ;IS SW09=1(SET)?
1395 003320 001405              BEQ     1$             ;BR IF NOT SET.
1396 003322 005737 001220      TST    LOCK
1397 003326 001402              BEQ     1$
1398 003330 013716 001220      MOV     LOCK,(SP)    ;GOTO THE ADDRESS IN LOCK.
1399 003334 000002      1$:  RTI              ;GO BACK.
1400
1401                                ;TELETYPE OUTPUT ROUTINE
1402                                -----
1403
1404 003336 010546      .TYPE: MOV     R5,-(SP)  ;SAVE R5 ON THE STACK.
1405 003340 017605 000002      MOV     @2(SP),R5    ;GET ADDRESS OF MESSAGE.
1406 003344 062766 000002 000002  ADD     #2,2(SP)     ;POP OVER ADDRESS.
1407 003352 032777 010000 175612 1$:  BIT      #SW12,@SWR  ;INHIBIT ALL PRINT OUT??
1408 003360 001012              BNE     3$             ;BR IF NO PRINT OUT WANTED (SW12=1)
1409 003362 105715              TSTB   (R5)          ;IS NUMBER MINUS? (MSB=1(BIT7))
1410 003364 100002              BPL     2$             ;BR IF NUMBER IS PLUS
1411 003366 104402 005136      TYPE   MCRLF        ;TYPE A CR/LF!
1412 003372 105777 175612      2$:  TSTB   @TPCSR     ;TTY READY?
1413 003376 100375              BPL     2$             ;BR IF NO.
1414 003400 112577 175606      MOVB   (R5)+,@TPDBR ;PRINT CURRENT CHAR.
1415 003404 001362              BNE     1$             ;IF NOT ZERO KEEP PRINTING!
1416 003406 012605      3$:  MOV     (SP)+,R5    ;END OF OUTPUT. RESTORE R5
1417 003410 000002      RTI              ;GO HOME
1418
1419                                -----
1420 003412 010346      .INSTR: MOV     R3,-(SP) ;SAVE R3 ON STACK
1421 003414 010446      MOV     R4,-(SP)    ;SAVE R4 ON STACK
1422 003416 017637 000004 003434  MOV     @4(SP),MSG
1423 003424 062766 000002 000004  ADD     #2,4(SP)
1424 003432 104402      .INST1: TYPE
1425 003434 000000      .MSG:   0
1426 003436 012704 006160      MOV     #INBUF,R4

```

```

1427 003442 012703 000007      MOV      #7,R3
1428 003446 105777 175532      1$: TSTB   @TKCSR
1429 003452 100375          BPL      1$
1430 003454 117714 175526      MOVB    @TKDBR,(R4)
1431 003460 142714 000200      BICB    #200,(R4)
1432 003464 122427 000015      CMPB    (R4),#15
1433 003470 001417          BEQ     INSTR2
1434 003472 105777 175512      2$: TSTB   @TPCSR
1435 003476 100375          BPL      2$
1436 003500 017777 175502 175504      MOV     @TKDBR,@TPDBR
1437 003506 005303          DEC     R3
1438 003510 001356          BNE     1$
1439 003512 012604          MOV     (SP)+,R4
1440 003514 012603          MOV     (SP)+,R3
1441 003516 010346      .INSTE: MOV     R3,-(SP)
1442 003520 010446          MOV     R4,-(SP)
1443 003522 104402 005132          TYPE   'MQM
1444 003526 000741          BR      INSTR1
1445 003530 012604      INSTR2: MOV     (SP)+,R4      ;RESTORE R4
1446 003532 012603          MOV     (SP)+,R3      ;RESTORE R3
1447 003534 000002          RTI
1448
1449
1450
1451
1452 003536 010546      .PARAM: MOV     R5,-(SP)
1453 003540 010446          MOV     R4,-(SP)
1454 003542 016605 000004          MOV     4(SP),R5
1455 003546 012537 003726          MOV     (R5)+,LOLIM
1456 003552 012537 003730          MOV     (R5)+,HILIM
1457 003556 012537 003732          MOV     (R5)+,DEVADR
1458 003562 112537 003734          MOVB    (R5)+,LOBITS
1459 003566 112537 003735          MOVB    (R5)+,ADRCNT
1460 003572 010566 000004          MOV     R5,4(SP)
1461 003576 005005      PARAM1: CLR     R5
1462 003600 012704 006160          MOV     #INBUF,R4
1463 003604 122714 000015          CMPB    #15,(R4)
1464 003610 001420          BEQ     PARERR
1465 003612 121427 000060      1$: CMPB    (R4),#60
1466 003616 002415          BLT     PARERR
1467 003620 121427 000067          CMPB    (R4),#67
1468 003624 003012          BGT     PARERR
1469 003626 142714 000060          BICB    #60,(R4)
1470 003632 152405          BISB    (R4)+,R5
1471 003634 122714 000015          CMPB    #15,(R4)
1472 003640 001406          BEQ     LIMITS
1473 003642 006305          ASL     R5
1474 003644 006305          ASL     R5
1475 003646 006305          ASL     R5
1476 003650 000760          BR      1$
1477 003652 104404      PARERR: INSTER
1478 003654 000750          BR      PARAM1
1479
1480
1481
1482
;CONVERT ASCII STRING TO OCTAL
;-----
;TEST TO SEE IF NUMBER IS WITHIN LIMITS
;-----

```

1483	003656	020537	003730	LIMITS:	CMP	R5,HILIM	
1484	003662	101373			BHI	PARERR	
1485	003664	020537	003726		CMP	R5,LOLIM	
1486	003670	103770			BLO	PARERR	
1487	003672	133705	003734		BITB	LOBITS,R5	
1488	003676	001365			BNE	PARERR	
1489							
1490							
1491							
1492	003700	013704	003732				
1493	003704	010524		1\$:	MOV	DEVADR,R4	
1494	003706	062705	000002		MOV	R5,(R4)+	
1495	003712	105337	003735		ADD	#2,R5	
1496	003716	001372			DECB	ADRCNT	
1497	003720	012604			BNE	1\$	
1498	003722	012605			MOV	(SP)+,R4	
1499	003724	000002			MOV	(SP)+,R5	
1500	003726	000000			RTI		
1501	003730	000000		LOLIM:	0		
1502	003732	000000		HILIM:	0		
1503	003734	000000		DEVADR:	0		
1504		003735		LOBITS:	0		
1505				ADRCNT=	LOBITS+1		
1506							
1507							
1508							
1509	003736	016637	000004	001266	.SAV05:	MOV	4(SP),SAVPC ;SAVE R7 (PC)
1510							
1511							
1512							
1513	003744	010537	001262		SV05:	MOV	R5,SAVR5 ;SAVE R5
1514	003750	010437	001260			MOV	R4,SAVR4 ;SAVE R4
1515	003754	010337	001256			MOV	R3,SAVR3 ;SAVE R3
1516	003760	010237	001254			MOV	R2,SAVR2 ;SAVE R2
1517	003764	010137	001252			MOV	R1,SAVR1 ;SAVE R1
1518	003770	010037	001250			MOV	R0,SAVR0 ;SAVE R0
1519	003774	000002				RTI	;LEAVE.
1520							
1521							
1522							
1523	003776	013700	001250		.RES05:	MOV	SAVR0,R0 ;RESTORE R0
1524	004002	013701	001252			MOV	SAVR1,R1 ;RESTORE R1
1525	004006	013702	001254			MOV	SAVR2,R2 ;RESTORE R2
1526	004012	013703	001256			MOV	SAVR3,R3 ;RESTORE R3
1527	004016	013704	001260			MOV	SAVR4,R4 ;RESTORE R4
1528	004022	013705	001262			MOV	SAVR5,R5 ;RESTORE R5
1529	004026	000002				RTI	;LEAVE
1530							
1531							
1532							
1533							
1534							
1535	004030	104402	005136		.CONVR:	TYPE	MCRLF
1536	004034	010046			.CNVRT:	MOV	R0,-(SP)
1537	004036	010146				MOV	R1,-(SP)
1538	004040	010346				MOV	R3,-(SP)

1539	004042	010446			MOV	R4,-(SP)
1540	004044	010546			MOV	R5,-(SP)
1541	004046	017601	000012		MOV	2(12(SP),R1
1542	004052	062766	000002	000012	ADD	#2,12(SP)
1543	004060	012137	004234		MOV	(R1)+,WRDCNT
1544	004064	112137	004236	15:	MOVB	(R1)+,CHRCNT
1545	004070	112137	004237		MOVB	(R1)+,SPACNT
1546	004074	013137	004240		MOV	2(R1)+,BINWRD
1547	004100	013704	004240	25:	MOV	BINWRD,R4
1548	004104	113705	004236		MOVB	CHRCNT,R5
1549	004110	012700	006054		MOV	#TEMP,R0
1550	004114	010403		35:	MOV	R4,R3
1551	004116	042703	177770		BIC	#177770,R3
1552	004122	062703	000060		ADD	#060,R3
1553	004126	110320			MOVB	R3,(R0)+
1554	004130	000241			CLC	
1555	004132	006004			ROR	R4
1556	004134	000241			CLC	
1557	004136	006004			ROR	R4
1558	004140	000241			CLC	
1559	004142	006004			ROR	R4
1560	004144	005305			DEC	R5
1561	004146	001362			BNE	35
1562	004150	012703	006116		MOV	#MDATA,R3
1563	004154	114023		45:	MOVB	-(R0),(R3)+
1564	004156	105337	004236		DECB	CHRCNT
1565	004162	001374			BNE	45
1566	004164	105737	004237		TSTB	SPACNT
1567	004170	001405			BEQ	65
1568	004172	112723	000040	55:	MOVB	#040,(R3)+
1569	004176	105337	004237		DECB	SPACNT
1570	004202	001373			BNE	55
1571	004204	105013		65:	CLRB	(R3)
1572	004206	104402	006116		TYPE	,MDATA
1573	004212	005337	004234		DEC	WRDCNT
1574	004216	001322			BNE	15
1575	004220	012505			MOV	(SP)+,R5
1576	004222	012604			MOV	(SP)+,R4
1577	004224	012603			MOV	(SP)+,R3
1578	004226	012601			MOV	(SP)+,R1
1579	004230	012600			MOV	(SP)+,R0
1580	004232	000002			RTI	
1581	004234	000000			WRDCNT: 0	
1582	004236	000000			CHRCNT: 0	
1583		004237			SPACNT=CHRCNT+1	
1584	004240	000000			BINWRD: 0	
1585						
1586						
1587						
1588						
1589						
1590						
1591						
1592	004242	017605	000000		.SETFLG:MOV	2(SP),R5
1593	004246	042737	000040	006160	BIC	#40,INBUF
1594	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

```



```

1595 004262 001002      BNE      1$
1596 004264 105015      CLRB    (R5)      ;000
1597 004266 000406      BR      2$
1598 004270 122737 000131 006160 1$:  CMPB    #'Y,INBUF      ;IS IT "Y" ?
1599 004276 001005      BNE      3$
1600 004300 112715 177777      MOVVB   #-1,(R5)      ;377
1601 004304 062716 000002      2$:  ADD     #2,(SP)
1602 004310 000002      RTI
1603 004312 104404      3$:  INSTER ;RETRY
1604 004314 000752      BR      .SETFLG

1605
1606
1607      ;TRAP DISPATCH SERVICE
1608      ;ARGUMENT OF TRAP IS EXTRACTED
1609      ;AND USED AS OFFSET TO OBTAIN POINTER
1610      ;TO SELECTED SUBROUTINE
1611
1612 004316 011646      .TRPSR: MOV     (SP),-(SP)      ;GET PC OF RETURN
1613 004320 162716 000002      SUB     #2,(SP)        ;=PC OF TRAP
1614 004324 017616 000000      MOV     @2(SP),(SP)    ;GET TRP
1615 004330 006316      TRPOK: ASL     (SP)      ;MULTIPLY TRAP ARG BY 2
1616 004332 042716 177001      BIC     #177001,(SP)   ;CLEAR UNWANTED BITS
1617 004336 062716 001344      ADD     #.TRPTAB,(SP)  ;POINTER TO SUBROUTINE ADDRESS
1618 004342 017616 000000      MOV     @2(SP),(SP)    ;SUBROUTINE ADDRESS
1619 004346 000136      JMP     @2(SP)+        ;GO TO SUBROUTINE
1620
1621      ;ERROR HANDLER
1622      ;-----
1623
1624 004350 032777 010000 174624 .HLT:  BIT     #SW12,@SWR    ;BELL ON ERROR?
1625 004356 001406      BEQ     XBX           ;BR IF NO BELL
1626 004360 105777 174624      TSTB   @TPCSR        ;TTY READY.
1627 004364 100003      BPL     XBX           ;DON'T WAIT IF TTY NOT READY.
1628 004366 112777 000207 174616      MOVVB  #207,@TPDBR    ;PUSH A BELL AT THE TTY.
1629 004374 032777 020000 174600 XBX:  BIT     #SW13,@SWR    ;DELETE ERROR PRINT OUT?
1630 004402 001105      BNE     HALTS         ;BR IF NO PRINT OUT WANTED.
1631 004404 021637 001234      CMP     (SP),LSTERR   ;WAS THIS ERROR FOUND LAST TIME?
1632 004410 001404      BEQ     1$           ;BR IF YES
1633 004412 011637 001234      MOV     (SP),LSTERR   ;RECORD BEING HERE
1634 004416 105037 001341      CLRB   ERRFLG        ;PREPARE HEADER
1635 004422 104406      1$:  SAVOS   ;SAVE ALL PROC REGISTERS
1636 004424 011605      MOV     (SP),R5       ;GET THE PC OF ERROR
1637 004426 162705 000002      SUB     #2,R5         ;GET ADDRESS OF TRAP CALL
1638 004432 011504      MOV     (R5),R4       ;GET HLT INSTRUCTION
1639 004434 006304      ASL     R4            ;MULT BY TWO
1640 004436 061504      ADD     (R5),R4       ;DOUBLE IT
1641 004440 006304      ASL     R4            ;MULT AGAIN
1642 004442 042704 177001      BIC     #177001,R4    ;CLEAR JUNK
1643 004446 062704 023274      ADD     #.ERRTAB,R4   ;GET POINTER
1644 004452 012437 004566      MOV     (R4)+,ERRMSG  ;GET ERROR MESSAGE
1645 004456 012437 004600      MOV     (R4)+,DATAHD  ;GET DATA HEADRER
1646 004462 011437 004612      MOV     (R4),DATABP   ;GET DATA TABLE
1647 004466 105737 001341      TSTB   ERRFLG        ;TYPE HEADREER
1648 004472 001403      BEQ     TYPMSG        ;BR IF YES
1649 004474 005737 004612      TST    DATABP        ;DOES DATA TABLE EXIST?
1650 004500 001040      BNE     TYPDAT        ;BR IF YES.
    
```

1651	004502	104402	005136		TYPMSG:	TYPE	,MCRLF		
1652	004506	104402	005136			TYPE	,MCRLF		
1653	004512	005737	001220			TST	LOCK		
1654	004516	001402				BEG	1\$		
1655	004520	104402	005411			TYPE	,MASTEK		
1656	004524	104402	005377		1\$:	TYPE	,MTSTN		
1657	004530	104411	005000			CNVRT	,XTSTN		;SHOW IT
1658	004534	104402	005466			TYPE	,MERRPC		;TYPE PC.
1659	004540	104411	004772			CNVRT	,ERTABO		;SHOW IT
1660	004544	104402	005136			TYPE	,MCRLF		;GIVE A CR/LF
1661	004550	112737	177777	001341		MOV8	1-1,ERRFLG		;NO MORE HEADER UNLESS NO DATA TABLE.
1662	004556	005737	004566			TST	ERRMSG		;IS THERE AN ERROR MESSAGE?
1663	004562	001402				BEG	WRKO.FM		;BR IF NO.
1664	004564	104402				TYPE			;TYPE
1665	004566	000000			ERRMSG:	0			;ERROR MESSAGE
1666	004570				WRKO.FM:				
1667	004570	005737	004600			TST	DATAHD		;DATA HEADER?
1668	004574	001402				BEG	TYPDAT		;BR IF NO
1669	004576	104402				TYPE			;TYPE
1670	004600	000000			DATAHD:	0			;DATA HEADER
1671	004602	005737	004612		TYPDAT:	TST	DATABP		;DATA TABLE?
1672	004606	001402				BEG	RESREG		;BR IF NO.
1673	004610	104410				CNVRT			;SHOW
1674	004612	000000			DATABP:	0			;DATA TABLE
1675	004614	104407			RESREG:	RESOS			;RESTORE PROC REGISTERS
1676	004616	022737	003104	000042	HALTS:	CMP	#SENDAD,2#42		;IF ACT-11 AUTO MODE--HALT!!
1677	004624	001403				BEG	1\$		
1678	004626	005777	174350			TST	2SWR		;HALT ON ERROR?
1679	004632	100035				BPL	EXITER		;BR IF NO HALT ON ERROR
1680	004634	010046			1\$:	PUSHRO			;SAVE RO
1681	004636	016600	000002			MOV	2(SP),RO		;SHOW ERROR PC IN DATA LIGHTS
1682	004642	013746	000004			MOV	4,-(SP)		;SAVE OLD TRAP
1683	004646	013746	000006			MOV	6,-(SP)		
1684	004652	012737	004710	000004		MOV	#22\$,4		;FORCE HALT IF TIME-OUT
1685	004660	012737	000340	000006		MOV	#340,6		;WHEN REFERENCING TXCSR
1686	004666	042777	014000	174516		BIC	#SYSST!MEXT,2TXCSR		
1687	004674	000000				HALT			;HALT
1688	004676	012637	000006			MOV	(SP)+,6		;RESTORE TRAP
1689	004702	012637	000004			MOV	(SP)+,4		
1690	004706	000406				BR	33\$		
1691	004710	000000			22\$:	HALT			;HALT
1692	004712	022626				CMP	(SP)+,(SP)+		;POP STACK
1693	004714	012637	000006			MOV	(SP)+,6		;RESTORE TRAP
1694	004720	012637	000004			MOV	(SP)+,4		
1695	004724	012600			33\$:	POPRO			;GET RO
1696	004726	005237	001232		EXITER:	INC	ERRCNT		;UPDATE ERROR COUNT
1697	004732	032777	000400	174242		BIT	#SW08,2SWR		;GOTO TOP OF TEST?
1698	004740	001007				BNE	1\$;BR IF YES
1699	004742	032777	002000	174232		BIT	#SW10,2SWR		;GOTO NEXT TEST?
1700	004750	001407				BEG	2\$;BR IF NO
1701	004752	013737	001216	001214		MOV	NEXT,RETURN		;SET FOR NEXT TEST
1702	004760	012706	001150		1\$:	MOV	#STACK,SP		;RESET SP
1703	004764	000177	174224			JMP	2RETURN		;GOTO SPECIFIED TEST
1704	004770	000002			2\$:	RTI			;RETURN
1705	004772	000001			ERTABO:	1			
1706	004774	006	002			.BYTE	6,2		

```

1707 004776 001266
1708 005000 000001
1709 005002 003 002
1710 005004 001226
1711 005006 017600 000000
1712 005012 062716 000002
1713 005016
1714 005016 052777 020000 174366
1715 005024 005300
1716 005026 001405
1717 005030 042777 020000 174354
1718 005036 005300
1719 005040 001366
1720 005042 000002
1721
1722
1723
1724 005044 000240
1725 005046 000207
1726
1727
1728
1729 005050 012737 005060 000024
1730 005056 000000
1731 005060 000005
1732 005062 012706 001150
1733 005066 012737 005050 000024
1734 005074 104402
1735 005076 005141
1736 005100 000177 174110
1737
1738
1739 005104 012702 000300
1740 005110 012701 000302
1741 005114 010122
1742 005116 005022
1743 005120 022121
1744 005122 022701 000776
1745 005126 001372
1746 005130 000207
1747
1748
1749
1750 005132 020040 000077
(2) 005136 005015 000
(2) 005141 377 053520 020122
(2) 005157 015 042777 042116
(2) 005202 051377 000
(2) 005205 377 051120 043517
(2) 005254 044777 051516 043125
(2) 005300 052377 051505 020124
(2) 005312 046377 041517 020113
(2) 005341 103 051123 020072
(2) 005347 126 041505 020072
(2) 005355 120 051501 042523
(2) 005366 051105 047522 051522

```

SAVPC
XTSTN: 1
.BYTE 3,2
TSTNO
.FKCLK: MOV @ (SP), R0 ;GET THE # OF TICKS TO POKE
ADD #2, (SP) ;POP OVER THE #
1\$: BIS #CLK, @TXCSR ;POKE CLOCK UP
DEC R0 ;ARE WE DONE?
BEQ 2\$;YES-GO TO 2\$
BIC #CLK, @TXCSR ;POKE CLOCK DOWN
DEC R0 ;ARE WE DONE?
BNE 1\$;NO-REPEAT
2\$: RTI ;RETURN
;WAIT ROUTINE
SMALL: NOP ;STALL
RTS PC ;RETURN
;POWER FAIL ROUTINE
.PFAIL: MOV #PWRUP, 24 ;LOAD PFAIL VECTOR FOR POWER UP
HALT
PWRUP: RESET ;WAIT TTY TO COME UP
MOV #STACK, SP ;REINIT STACK POINTER
MOV #.PFAIL, 24 ;LOAD PFAIL VECTOR FOR POWER DOWN
TYPE
MPOWER
JMP @RETURN
;CLRVEC, ROUTINE TO FILL COMMUNICATION VECTOR AREA WITH .+2, HALT
CLRVEC: MOV #300, R2 ;R2 COMM VECTOR AREA ADRS
MOV #302, R1 ;INIT R1 WITH ADRS OF HALT
1\$: MOV R1, (R2)+ ;MOV .+2 TO PC
CLR (R2)+ ;MOV HALT TO PC
CMP (R1)+, (R1)+ ;INC TO NEXT VECTOR AREA
CMP #776, R1 ;END OF VECTOR AREA
BNE 1\$;NO
RTS PC ;RETURN
MQM: .ASCIZ / ?/
MCRLF: .ASCIZ <15><12>
MPOWER: .ASCIZ <377>/PWR FAILED. /
MEPASS: .ASCIZ <15><377>/END PASS DZDPDE /
MR: .ASCIZ <377>/R/
MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT. /
MERR3: .ASCIZ <377>/INSUFFICIENT DATA! /
MTSTPC: .ASCIZ <377>/TEST PC- /
MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST /
MCSRX: .ASCIZ /CSR: /
MVECX: .ASCIZ /VEC: /
MPASSX: .ASCIZ /PASSES: /
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	01520	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	
1751	006044	006	003		.BYTE	6,3
1752	006046	001236			TEMP1	
1753	006050	006	002		.BYTE	6,2
1754	006052	001240			TEMP2	
1755					.EVEN	
1756						
1757	006054	000000			TEMP: 0	
1758		006116			.+.40	
1759	006116	000000			MDATA: 0	
1760		006160			.+.40	
1761	006160	000000			INBUF: 0	
1762		006222			.+.40	
1763	006222	000001			TRP.PC: .BLKW 1	
1764						

```

1765
1766
1767
1768
1769
1770
1771
1772
1773
1774 006224 105737 001310      CYCLE: TSTB      DUPACTV      ;ARE ANY DUP11'S TO BE TESTED?
1775 006230 001004                BNE          1$          ;BR IF OK.
1776 006232 104402 005205      TYPE        ,MERR2      ;NO DUP11'S SELECTED!!
1777 006236 000000                HALT                ;STOP THE SHOW.
1778 006240 000776                BR          -2          ;DISQUALIFY CONT. SW.
1779 006242 133737 001314 001310 1$: BITB      RUN,DUPACTV ;IS THIS ONE "ACTIVE"
1780 006250 001020                BNE          2$          ;BR IF GOOD ONE FOUND.
1781 006252 000241                CLC                ;CLEAR PROC. CARRY BIT.
1782 006254 106137 001314      ROLB      RUN          ;UPDATE POINTER
1783 006260 105537 001314      ADCB      RUN          ;CATCH CARRY FROM RUN
1784 006264 062737 000006 001316  ADD      #6,CREAM      ;UPDATE ADDRESS POINTER.
1785 006272 022737 001560 001316  CMP      #DUP.END,CREAM
1786 006300 001360                BNE          1$          ;KEEP GOING: NOT ALL TESTED FOR.
1787 006302 012737 001500 001316  MOV      #DUP.MAP,CREAM ;RESET ADDRESS POINTER.
1788 006310 000754                BR          1$          ;KEEP LOOKING FOR ACTIVE DUP11
1789 006312 000241                CLC                ;CLEAR PROC. CARRY.
1790 006314 106137 001314      ROLB      RUN          ;UPDATE POINTER.
1791 006320 105537 001314      ADCB      RUN          ;CATCH CARRY.
1792 006324 013700 001316      MOV      CREAM,RO      ;GET ADDRESS POINTER.
1793 006330 062737 000006 001316  ADD      #6,CREAM      ;UPDATE.
1794 006336 022737 001560 001316  CMP      #DUP.END,CREAM
1795
1796 006344 001003                BNE          3$          ;ALL DONE?
1797 006346 012737 001500 001316  MOV      #DUP.MAP,CREAM ;BR IF NO.
1798 006354 012037 001404                MOV      (RO)+,RXCSR    ;RESTORE POINTER.
1799 006360 012037 001374                MOV      (RO)+,DUPRVC   ;LOAD SYSTEM CTRL. REG
1800 006364 012037 001434                MOV      (RO)+,LOO.00   ;LOAD VECTOR
1801 006370 012700 000002                MOV      #2,RO          ;GET PARAMETERS
1802 006374 013737 001404 001424  MOV      RXCSR,HUPRCR   ;SAVE CORE THIS WAY!
1803 006402 005237 001424                INC      HUPRCR         ;GET CONTROL REG HIGH BYTE
1804 006406 013737 001424 001406  MOV      HUPRCR,RXDBUF  ;GOT IT
1805 006414 005237 001406                INC      RXDBUF         ;GET RX CONTROL REG BUFFER
1806 006420 013737 001406 001416  MOV      RXDBUF,DUPSEC  ;GOT IT
1807 006426 013737 001406 001410  MOV      RXDBUF,PARCSR  ;GOT SECONDARY REG SELECT REG
1808 006434 013737 001406 001422  MOV      RXDBUF,HUPRBF  ;GOT PARAMETER STATUS REGISTER
1809 006442 005237 001422                INC      HUPRBF         ;GET RX BUFFER HIGH BYTE
1810 006446 013737 001422 001420  MOV      HUPRBF,HUPPSR  ;GOT IT
1811 006454 013737 001420 001412  MOV      HUPPSR,TXCSR   ;GOT PAR STATUS REG HIGH BYTE
1812 006462 005237 001412                INC      TXCSR          ;GOT TX CONTROL REGISTER
1813 006466 013737 001412 001430  MOV      TXCSR,HUPTCR   ;GOT IT
1814 006474 005237 001430                INC      HUPTCR         ;GET TX CONTROL REG HIGH BYTE
1815 006500 013737 001430 001414  MOV      HUPTCR,TXDBUF  ;GOT IT
1816 006506 005237 001414                INC      TXDBUF         ;BET TX BUFFER
1817 006512 013737 001414 001426  MOV      TXDBUF,HUPTBF  ;GOT IT
1818 006520 005237 001426                INC      HUPTBF         ;GET TX BUFFER HIGH BYTE
1819
1820 006524 013737 001374 001376  MOV      DUPRVC,DUPRPS  ;GOT IT
                                ;RX VECTOR
    
```

1821	006532	060037	001376			ADD	RO, DUPRPS	;RX PRIORITY LEVEL
1822	006536	013737	001376	001400		MOV	DUPRPS, DUPTVC	
1823	006544	060037	001400			ADD	RO, DUPTVC	;TX VECTOR
1824	006550	013737	001400	001402		MOV	DUPTVC, DUPTPS	
1825	006556	060037	001402			ADD	RO, DUPTPS	;TX PRIORITY LEVEL
1826								
1827								
1828	006562	012700	001434			MOV	#L00.00, RO	;LOAD STAU5 00-00
1829	006566	012701	001432			MOV	#MASK.A, R1	;PREPARE MASK.
1830	006572	012702	001433			MOV	#CLK.A, R2	;PREPARE CLOCKS
1831	006576	004737	006742			JSR	PC, FIX.00	;GO AND CALCULATE CONFIGURATION.
1832	006602	005737	000042			TST	#42	
1833	006606	001050				BNE	4\$	
1834	006610	032777	000002	172364		BIT	#SW01, #SWR	;IF SW01=1, GET STARTING TEST #
1835	006616	001444				BEQ	4\$	
1836	006620	104402	005136		7\$:	TYPE	, MCRLF	
1837	006624	104403				INSTR	;OUTPUT MESSAGE & GET INPUT STRING	
1838	006626	005377				MTSTN	;MESSAGE	
1839	006630	104405				PARAM	;CONVERT STRING	
1840	006632	000001				1	;LOW LIMIT	
1841	006634	001000				1000	;HIGH LIMIT	
1842	006636	001226				TSTNO	;STORE AT THIS LOCATION	
1843	006640	000				0	;MASK	
1844	006641	001				1	;HOW MANY TIMES + 2	
1845	006642	012700	007160			MOV	#TST1, RO	
1846	006646	022710	012737		5\$:	CMP	#12737, (RO)	
1847	006652	001017				BNE	6\$	
1848	006654	023760	001226	000002		CMP	TSTNO, 2(RO)	
1849	006662	001013				BNE	6\$	
1850	006664	022760	001226	000004		CMP	#TSTNO, 4(RO)	
1851	006672	001007				BNE	6\$	
1852	006674	010037	001214			MOV	RO, RETURN	;SAVE PC
1853	006700	104402	005136			TYPE	, MCRLF	
1854	006704	104402	005202			TYPE	, MR	
1855	006710	000412				BR	8\$	
1856	006712	005720			6\$:	TST	(RO)+	
1857	006714	020027	021464			CMP	RO, #TLAST+10	
1858	006720	001352				BNE	5\$	
1859	006722	104402	005132			TYPE	, MQM	
1860	006726	000734				BR	7\$	
1861								
1862	006730	012737	007160	001214	4\$:	MOV	#TST1, RETURN	;PREPARE RETURN ADDRESS
1863	006736	000177	172252		8\$:	JMP	3RETURN	;GO START TESTING.
1864								
1865	006742	011003			FIX.00:	MOV	(RO), R3	;GET PARAMETERS.
1866	006744	000207			5\$:	RTS	PC	;

1867					
1868					
1869					
1870	006746	012577	172422	SETVEC:	MOV (R5)+,JDUPRVC
1871	006752	012577	172422		MOV (R5)+,JDUPTVC
1872	006756	112577	172414		MOVB (R5)+,JDUPRPS
1873	006762	112577	172414		MOVB (R5)+,JDUPTPS
1874	006766	000205			RTS R5
1875	006770			NO.ATRAP:	
1876	006770	104012			HLT 12
1877	006772	000002			RTI
1878					
1879	006774			NO.BTRAP:	
1880	006774	104013			HLT 13
1881	006776	000002			RTI
1882					
1883	007000	010046		SIMBCC:	MOV R0,-(SP)
1884	007002	010146			MOV R1,-(SP)
1885	007004	010246			MOV R2,-(SP)
1886	007006	012537	001236		MOV (R5)+,TEMP1
1887	007012	012537	001240		MOV (R5)+,TEMP2
1888	007016	012537	001242		MOV (R5)+,TEMP3
1889	007022	005037	007154	1\$:	CLR BCCFBK
1890	007026	013700	001242		MOV TEMP3,R0
1891	007032	006037	001240		ROR TEMP2
1892	007036	005500			ROR R0
1893	007040	032700	000001		BIT #BIT0,R0
1894	007044	001402			BEQ 2\$
1895	007046	005137	007154		COM BCCFBK
1896	007052	013700	007152	2\$:	MOV XPOLY,R0
1897	007056	005100			COM R0
1898	007060	040037	007154		BIC R0,BCCFBK
1899	007064	000241			CLC
1900	007066	006037	001242		ROR TEMP3
1901	007072	013700	007154		MOV BCCFBK,R0
1902	007076	013701	001242		MOV TEMP3,R1
1903	007102	010102			MOV R1,R2
1904	007104	040100			BIC R1,R0
1905	007106	043702	007154		BIC BCCFBK,R2
1906	007112	050200			BIS R2,R0
1907	007114	043737	007152 001242		BIC XPOLY,TEMP3
1908	007122	050037	001242		BIS R0,TEMP3
1909	007126	005337	001236		DEC TEMP1
1910	007132	001333			BNE 1\$
1911	007134	013737	001242 007156		MOV TEMP3,CALBCC
1912	007142	012602			MOV (SP)+,R2
1913	007144	012601			MOV (SP)+,R1
1914	007146	012600			MOV (SP)+,R0
1915	007150	000205			RTS R5
1916	007152	000000		XPOLY:	0
1917	007154	000000		BCCFBK:	0
1918	007156	000000		CALBCC:	0
1919		120001		CRC16=	120001
1920		102010		CRC.CCITT=	102010
1921					
1922					

INTERRUPT PRIORITY TEST-CPU LEVEL AT 7

1923
1924
1925
1926
1927
1928
1929
1930
1931
1932
1933
1934
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949
1950
1951
1952
1953
1954
1955
1956
1957
1958
1959
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978

007160 012737 000001 001226
007166 012737 007264 001216
007174 012737 000340 177776
007202 052777 000400 172202
007210 004737 005044
007214 004537 006746
007220 006770
007222 006774
007224 340 340
007226 012777 004100 172156
007234 012737 000340 177776
007242 000240
007244 000240
007246 000240
007250 005077 172136
007254 104400
007256 012716 007250
007262 000002

```
***** TEST 1 *****  
*PRIORITY INTERRUPT TEST.  
*SET PROCESSOR STATUS TO PRIORITY 7  
*AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.  
*****  
*****  
*  
TEST 1  
*  
*****  
*****  
TST1: MOV #1, @TSTNO  
MOV @TST2, NEXT  
MOV #340, PS ; LOCK OUT INTERRUPTS  
BIS #MRESET, @TXCSR ; RESET THE DEVICE  
JSR PC, SMALL ; WAIT FOR RESET TO FINISH  
JSR R5, SETVEC ; SET UP VECTORS  
NO. ATRAP ; VECTOR "A"  
NO. BTRAP ; VECTOR "B"  
.BYTE 340, 340 ; LEVEL  
MOV #TXINTE!SYSTST, @TXCSR ; TURN ON DUP TX INT. ENABLE AND ENTER SYSTST MODE  
MOV #340, PS ; SET CPU PRIORITY--CHANGE HERE IF NOT = 5  
NOP ; STALL  
NOP ; DITTO  
NOP ; DITTO  
1$: CLR @TXCSR ; DISABLE THE DUP11  
SCOPE ; SCOPE THIS TEST  
2$: MOV #1$, (SP) ; SETUP FOR RETURN  
RTI ; RETURN
```

***** TEST 2 *****
*PRIORITY INTERRUPT TEST.
*SET PROCESSOR STATUS TO PRIORITY 6
*AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.

1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974
1975
1976
1977
1978

007264 012737 000002 001226
007272 012737 007400 001216
007300 122737 000005 001320
007306 001026
007310 012737 000340 177776
007316 052777 000400 172066
007324 004737 005044
007330 004537 006746
007334 006770
007336 006774
007340 340 340
007342 012777 004100 172042

```
*****  
*  
TEST 2  
*  
*****  
*****  
TST2: MOV #2, @TSTNO  
MOV @TST3, NEXT  
CMPB #5, PRIRTY ; COMPARE REAL WITH NORMAL  
BNE 1$ ; BR IF NOT A MATCH  
MOV #340, PS ; LOCK OUT INTERRUPTS  
BIS #MRESET, @TXCSR ; RESET THE DEVICE  
JSR PC, SMALL ; WAIT FOR RESET TO FINISH  
JSR R5, SETVEC ; SET UP VECTORS  
NO. ATRAP ; VECTOR "A"  
NO. BTRAP ; VECTOR "B"  
.BYTE 340, 340 ; LEVEL  
MOV #TXINTE!SYSTST, @TXCSR ; TURN ON DUP TX INT. ENABLE AND ENTER SYSTST MODE
```


INTERRUPT PRIORITY TEST-CPU LEVEL AT 6

```

1979 007350 012737 000300 177776      MOV      #300,PS      ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
1980 007356 000240                    NOP                    ;STALL
1981 007360 000240                    NOP                    ;DITTO
1982 007362 000240                    NOP                    ;DITTO
1983 007364 005077 172022      1$: CLR      @TXCSR    ;DISABLE THE DUP11
1984 007370 104400                    SCOPE                 ;SCOPE THIS TEST
1985 007372 012716 007364      2$: MOV      #1$, (SP) ;SETUP FOR RETURN
1986 007376 000002                    RTI                   ;RETURN
1987
1988
1989
1990
1991
1992
1993
1994
1995
1996
1997
1998
1999
2000
2001 007400 012737 000003 001226      TST3: MOV      #3,@TSTNO
2002 007406 012737 007514 001216      MOV      #TST4,NEXT
2003 007414 122737 000005 001320      CMPB     #5,PRIRTY   ;COMPARE REAL WITH NORMAL
2004 007422 001026                    BNE      1$          ;BR IF NOT A MATCH
2005 007424 012737 000340 177776      MOV      #340,PS     ;LOCK OUT INTERRUPTS
2006 007432 052777 000400 171752      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2007 007440 004737 005044                    JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
2008 007444 004537 006746                    JSR      R5,SETVEC   ;SET UP VECTORS
2009 007450 006770                    NO. ATRAP            ;VECTOR "A"
2010 007452 006774                    NO. BTRAP            ;VECTOR "B"
2011 007454      340      340      .BYTE 340,340      ;LEVEL
2012 007456 012777 004100 171726      MOV      #TXINTE!SYSTST,@TXCSR ;TURN ON DUP TX INT. ENABLEAND ENTER SYSTST MODE
2013 007464 012737 000240 177776      MOV      #240,PS     ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
2014 007472 000240                    NOP                    ;STALL
2015 007474 000240                    NOP                    ;DITTO
2016 007476 000240                    NOP                    ;DITTO
2017 007500 005077 171706      1$: CLR      @TXCSR    ;DISABLE THE DUP11
2018 007504 104400                    SCOPE                 ;SCOPE THIS TEST
2019 007506 012716 007500      2$: MOV      #1$, (SP) ;SETUP FOR RETURN
2020 007512 000002                    RTI                   ;RETURN
2021
2022
2023
2024
2025
2026
2027
2028
2029
2030
2031
2032
2033
2034

```

```

:***** TEST 3 *****
:*PRIORITY INTERRUPT TEST.
:*SET PROCESSOR STATUS TO PRIORITY 5
:*AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.
:*****

```

```

:*****
: TEST 3
:*****

```

```

:***** TEST 4 *****
:*PRIORITY INTERRUPT TEST.
:*SET PROCESSOR STATUS TO PRIORITY 4
:*AND VERIFY THAT THE DUP11 WILL INTERRUPT.
:*****

```

```

:*****
: TEST 4
:*****

```

INTERRUPT PRIORITY TEST-CPU AT LEVEL 4

```

2035 007514 012737 000004 001226 TST4: MOV #4, @TSTNO
2036 007522 012737 007632 001216 MOV #TST5, NEXT
2037 007530 122737 000005 001320 CMPB #5, PIRTY ;COMPARE REAL WITH NORMAL
2038 007536 001027 BNE 1$ ;BR IF NOT A MATCH
2039 007540 012737 000340 177776 MOV #340, PS ;LOCK OUT INTERRUPTS
2040 007546 052777 000400 171636 BIS #MRESET, @TXCSR ;RESET THE DEVICE
2041 007554 004737 005044 JSR PC, SMALL ;WAIT FOR RESET TO FINISH
2042 007560 004537 006746 JSR R5, SETVEC ;SET UP VECTORS
2043 007564 006770 NO. ATRAP ;VECTOR "A"
2044 007566 007624 2$ ;VECTOR "B"
2045 007570 340 340 .BYTE 340, 340 ;LEVEL
2046 007572 012777 004100 171612 MOV #TXINTE!SYSTST, @TXCSR ;TURN ON DUP TX INT. ENABLE AND ENTER SYSTST MODE
2047 007600 012737 000200 177776 MOV #200, PS ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
2048 007606 000240 NOP ;STALL
2049 007610 000240 NOP ;DITTO
2050 007612 000240 NOP ;DITTO
2051 007614 104013 HLT 13 ;DUP FAILED TO INTERRUPT-POSSIBLY WRONG PRIORITY-CHANGE IF NOT 5
2052 007616 005077 171570 1$: CLR @TXCSR ;DISABLE THE DUP11
2053 007622 104400 SCOPE ;SCOPE THIS TEST
2054 007624 012716 007616 2$: MOV #1$, (SP) ;SETUP FOR RETURN
2055 007630 000002 RTI ;RETURN

```

```

:***** TEST 5 *****
:*TEST TO PROVE THE HALF-DUPLEX FUNCTION
:*PROVE THAT THE RECEIVER WILL NOT RECOGNIZE
:*DATA IF SEND IS ASSERTED.
:*****

```

```

:*****
: TEST 5
:*****

```

```

2068 007632 012737 000005 001226 TST5: MOV #5, @TSTNO
2069 007640 012737 010172 001216 MOV #TST6, NEXT
2070 007646 105737 001322 TSTB TCNFLG
2071 007652 001530 BEQ 1$
2072 007654 012737 000340 177776 MOV #340, PS ;LOCK OUT INTERRUPTS
2073 007662 052777 000400 171522 BIS #MRESET, @TXCSR ;RESET THE DEVICE
2074 007670 004737 005044 JSR PC, SMALL ;WAIT FOR RESET TO FINISH
2075 007674 052777 010010 171510 BIS #MEXT!HDXYEN, @TXCSR ;ENTER MAINT EXT AND HALF-DUPLEX MODES
2076 007702 004537 006746 JSR R5, SETVEC ;SET UP VECTORS
2077 007706 010162 2$ ;RECEIVER
2078 007710 006774 NO. BTRAP ;TRANSMITTER
2079 007712 340 340 .BYTE 340, 340 ;LEVEL
2080 007714 005037 177776 CLR PS ;LOWER PROC. STATUS
2081 007720 052777 000020 171456 BIS #RCVEN, @RXCSR ;TURN ON RECEIVER
2082 007726 052777 000100 171450 BIS #RINTEN, @RXCSR ;TURN ON INT. ENABLE
2083 007734 052777 000020 171450 BIS #SEND, @TXCSR ;TURN ON TRANSMITTER
2084 007742 012737 000005 007772 MOV #5, 68$ ;LOAD THE NUMBER
2085 007750 032777 004000 171436 66$: BIT #TIMER, @TXDBUF ;CHECK THE TIMER BIT
2086 007756 001374 BNE 66$ ;BR IF SET
2087 007760 032777 004000 171426 67$: BIT #TIMER, @TXDBUF ;CHECK THE BIT
2088 007766 001774 BEQ 67$ ;BR IF CLEAR
2089 007770 005327 DEC (PC)+ ;DECREMENT THE NUMBER

```

```

2091 007772 000005      68$: 5      ; OF TIMES TO REPEAT
2092 007774 001365      BNE      66$      ; BR IF MORE TO GO
2093 007776 105777 171410 TSTB    @TXCSR
2094 010002 100401      BMI      3$
2095 010004 104005      HLT      5      ; TXDONE FAILED TO SET
2096 010006 012777 000400 171400 3$: MOV    #T5OM,@TXDBUF ; LOAD TX BUFFER
2097 010014 012737 000005 010044 MOV    #5,73$      ; LOAD THE NUMBER
2098 010022 032777 004000 171364 71$: BIT    #TIMER,@TXDBUF ; CHECK THE TIMER BIT
2099 010030 001374      BNE      71$      ; BR IF SET
2100 010032 032777 004000 171354 72$: BIT    #TIMER,@TXDBUF ; CHECK THE BIT
2101 010040 001774      BEQ      72$      ; BR IF CLEAR
2102 010042 005327      DEC    (PC)+      ; DECREMENT THE NUMBER
2103 010044 000005      73$: 5      ; OF TIMES TO REPEAT
2104 010046 001365      BNE      71$      ; BR IF MORE TO GO
2105 010050 105777 171336 TSTB    @TXCSR      ; CHECK FOR DONE
2106 010054 100401      BMI      4$      ; BR IF SET
2107 010056 104000      HLT
2108      ; EXTERNAL CLOCKING STOPPED
2109      ; OR DATA WAS NOT RECEIVED.CHECK
2110      ; EIA DATA AND CLOCK PATHS
2110 010060 005077 171330 4$: CLR    @TXDBUF      ; LOAD A CHARACTER
2111 010064 105777 171322 TSTB    @TXCSR      ; CHECK FOR DONE
2112 010070 100375      BPL    -4      ; BR IF NOT SET
2113 010072 012777 001000 171314 MOV    #TEOM,@TXDBUF ; END THE MESSAGE
2114 010100 012737 000050 010130 MOV    #40,78$      ; LOAD THE NUMBER
2115 010106 032777 004000 171300 76$: BIT    #TIMER,@TXDBUF ; CHECK THE TIMER BIT
2116 010114 001374      BNE      76$      ; BR IF SET
2117 010116 032777 004000 171270 77$: BIT    #TIMER,@TXDBUF ; CHECK THE BIT
2118 010124 001774      BEQ      77$      ; BR IF CLEAR
2119 010126 005327      DEC    (PC)+      ; DECREMENT THE NUMBER
2120 010130 000050      78$: 40.      ; OF TIMES TO REPEAT
2121 010132 001365      BNE      76$      ; BR IF MORE TO GO
2122 010134 012737 000340 177776 1$: MOV    #340,PS      ; RAISE PROCESSOR STATUS
2123 010142 012706 001150      MOV    #STACK,SP    ; RESET STACK
2124 010146 052777 000400 171236 BIS    #MRESET,@TXCSR ; RESET THE DEVICE
2125 010154 004737 005044 JSR    PC,SMALL      ; WAIT FOR RESET TO FINISH
2126 010160 104400      SCOPE      ; SCOPE THIS TEST
2127
2128 010162 104007      2$: HLT    7      ; RECEIVER INTERRUPTED AND SHOULD
2129 010164 012716 010134 MOV    #1$, (SP)    ; NOT HAVE--THIS IS HALF
2130 010170 000002      RTI      ; DUPLEX.

```

```

***** TEST 6 *****
*TEST OF THE DUP RUNNING A BINARY COUNT
*PATTERN WITHOUT A CRC CALCULATION
*****

```

```

*****
:
: TEST 6
:
: *****

```

```

2142
2143 010172 012737 000006 001226 TST6: MOV    #6,@TSTNO
2144 010200 012737 010632 001216 MOV    #TST7,NEXT
2145 010206 052777 000400 171176 BIS    #MRESET,@TXCSR ; RESET THE DEVICE
2146 010214 004737 005044 JSR    PC,SMALL      ; WAIT FOR RESET TO FINISH

```

2147	010220	012737	000001	001236	MOV	#1,TEMP1	:LOAD DATA
2148	010226	005037	001240		CLR	TEMP2	:CLEAR EXPECTED
2149	010232	012737	000340	177776	MOV	#340,PS	:PS = 7
2150	010240	052777	004000	171144	BIS	#SYSTST, @TXCSR	:ENTER SYSTEM TEST MODE
2151	010246	004537	006746		JSR	R5,SETVEC	:LOAD INTERRUPT VECTORS
2152	010252	010446			11\$:RECEIVER
2153	010254	010526			12\$:TRANSMITER
2154	010256	340	340		.BYTE	340,340	:LEVEL
2155	010260	052777	001000	171122	BIS	#CRCE, @PARCSR	:TURN OFF CRC
2156	010266	052777	000020	171110	BIS	#RCVEN, @RXCSR	:TURN ON THE RECEIVER
2157	010274	052777	000100	171102	BIS	#RINTEN, @RXCSR	:TURN ON REC INTERRUPT ENABLE
2158	010302	105777	171104		15\$: TSTB	@TXCSR	:TEST FOR TX DONE
2159	010306	100375			BPL	15\$:BR IF NOT SET
2160	010310	052777	000020	171074	25\$: BIS	#SEND, @TXCSR	:TURN ON SEND
2161	010316	012777	000400	171070	MOV	#TSOM, @TXDBUF	:TURN ON START OF MESSAGE
2162	010324	012737	000005	010354	MOV	#5,68\$:LOAD THE NUMBER
2163	010332	032777	004000	171054	66\$: BIT	#TIMER, @TXDBUF	:CHECK THE TIMER BIT
2164	010340	001374			BNE	66\$:BR IF SET
2165	010342	032777	004000	171044	67\$: BIT	#TIMER, @TXDBUF	:CHECK THE BIT
2166	010350	001774			BEQ	67\$:BR IF CLEAR
2167	010352	005327			DEC	(PC)+	:DECREMENT THE NUMBER
2168	010354	000005			68\$: 5		:OF TIMES TO REPEAT
2169	010356	001365			BNE	66\$:BR IF MORE TO GO
2170	010360	105777	171026		35\$: TSTB	@TXCSR	:WAIT FOR DONE
2171	010364	100401			BMI	4\$:BR IF SET
2172	010366	104000			HLT		:EXTERNAL CLOCKING STOPPED
2173	010370	005077	171020		45\$: CLR	@TXDBUF	:PUSH OUT DATA
2174	010374	052777	000100	171010	BIS	#TXINTE, @TXCSR	:TURN ON TRANSMITTER INT ENABLE
2175	010402	005037	177776		CLR	PS	:LOWER PROCESOR STATUS
2176	010406				55\$:		
2177	010406	012737	000040	010436	MOV	#32,73\$:LOAD THE NUMBER
2178	010414	032777	004000	170772	71\$: BIT	#TIMER, @TXDBUF	:CHECK THE TIMER BIT
2179	010422	001374			BNE	71\$:BR IF SET
2180	010424	032777	004000	170762	72\$: BIT	#TIMER, @TXDBUF	:CHECK THE BIT
2181	010432	001774			BEQ	72\$:BR IF CLEAR
2182	010434	005327			DEC	(PC)+	:DECREMENT THE NUMBER
2183	010436	000040			73\$: 32.		:OF TIMES TO REPEAT
2184	010440	001365			BNE	71\$:BR IF MORE TO GO
2185	010442	104001			HLT	1	:FAILED TO INTERRUPT IN TIME
2186	010444	104400			65\$: SCOPE		:SCOPE THIS TEST
2187							
2188							
2189							
2190							
2191							
2192							
2193	010446	017737	170734	001324	11\$: MOV	@RXDBUF, DATA	:GET THE REGISTER AND DATA
2194	010454	123737	001240	001324	CMPB	TEMP2, DATA	:CHECK IT
2195	010462	001401			BEQ	+4	:BR IF OK
2196	010464	104002			HLT	2	:COMPARISON ERROR
2197	010466	105237	001240		INCB	TEMP2	:COUNT UP EXPECTED
2198	010472	105737	001240		TSTB	TEMP2	:CHECK TO SEE IF DONE
2199	010476	001012			BNE	7\$:BR IF NO
2200	010500	105777	170700		10\$: TSTB	@RXCSR	:CHECK FOR DONE
2201	010504	100375			BPL	10\$:BR IF NOT YET
2202	010506	032777	001000	170672	BIT	#REOM, @RXDBUF	:CHECK FOR END OF MSG

: INTERRUPT SERVICE ROUTINES

: RECEIVER:

```

2203 010514 001001      BNE      3+4      ;BR IF SET
2204 010516 104003      HALT      3        ;END OF MSG FAILED TO SET
2205 010520 012716 010444  MOV #6$, (SP)    ;CRUNCH STACK
2206
2207 010524 000002      7$: RTI          ;RETURN
2208
2209
2210      ; TRANSMITTER:
2211 010526 113777 001236 170660 12$: MOVB TEMP1, @TXDBUF ;LOAD THE TRANSMITTER BUFFER
2212 010534 105237 001236      INCB TEMP1      ;UP THE COUNT
2213 010540 122737 000377 001236  CMPB #377, TEMP1 ;ARE WE DONE
2214 010546 001026      BNE      13$      ;BR IF NO
2215 010550 012777 010560 170622  MOV #21$, @DUPTVC ;SETUP FOR NEXT PART
2216 010556 000422      BR      13$      ;LEAVE
2217 010560 012777 000377 170626 21$: MOV #377, @TXDBUF ;LOAD BUFFER
2218 010566 012777 010576 170604  MOV #22$, @DUPTVC ;SETUP NEXT PART
2219 010574 000413      BR      13$      ;LEAVE
2220 010576 012777 001000 170610 22$: MOV #TEOM, @TXDBUF ;SET END OF MSG
2221 010604 000240      NOP          ;STALL
2222 010606 000240      NOP          ;DITTO
2223 010610 042777 000120 170574  BIC #SEND!TXINTE, @TXCSR ;TURN OFF TRANSMITTER
2224 010616 012777 006774 170554  MOV #NO.BTRAP, @DUPTVC ;LOAD VECTOR
2225 010624 012716 010406 13$: MOV #5$, (SP)    ;CRUNCH STACK
2226 010630 000002      RTI          ;RETURNS
2227
2228
2229
2230
2231
2232
2233
2234
2235
2236
2237
2238
2239

```

```

:***** TEST 7 *****
:*TEST OF THE DUP RUNNING A BINARY COUNT
:*PATTERN WITH A CRC CALCULATION
:*****

```

```

:*****
:TEST 7
:*****

```

```

2240 010632 012737 000007 001226 TST7: MOV #7, @TSTNO
2241 010640 012737 011432 001216      MOV #TST10, NEXT
2242 010646 052777 000400 170536      BIS #MRESET, @TXCSR ;RESET THE DEVICE
2243 010654 004737 005044      JSR PC, SMALL ;WAIT FOR RESET TO FINISH
2244 010660 005001      CLR R1 ;CLEAR OUT DATA
2245 010662 012737 102010 007152      MOV #CRC.CCITT, XPOLY ;SET UP THE POLYNOMIAL
2246 010670 012737 177777 007156      MOV #-1, CALBCC ;SETUP FOR THE FIRST TIME
2247 010676 013737 007156 010720 16$: MOV CALBCC, 20$ ;ALLOW FOR THE NEXT CHARACTER
2248 010704 010137 010716      MOV R1, 17$ ;LOAD DATA
2249 010710 004537 007000      JSR R5, SIMBCC ;GO CALCULATE SOFTWARE BCC
2250 010714 000010      B. ;BASED ON THESE PARAMETERS
2251 010716 000001      17$: .BLKW 1 ;DATA
2252 010720 000001      20$: .BLKW 1 ;PREVIOUS BCC
2253 010722 105201      INCB R1 ;INCREMENT DATA
2254 010724 001364      BNE 16$ ;BR IF MORE TO GO
2255 010726 012737 000001 001236      MOV #1, TEMP1 ;LOAD DATA
2256 010734 005037 001240      CLR TEMP2 ;CLEAR EXPECTED
2257 010740 012737 000340 177776      MOV #340, PS ;PS = 7
2258 010746 052777 004000 170436      BIS #SYSTST, @TXCSR ;ENTER SYSTEM TEST MODE

```

```

2259 010754 004537 006746 JSR R5,SETVEC ;LOAD INTERRUPT VECTORS
2260 010760 011146 11$ ;RECEIVER
2261 010762 011214 12$ ;TRANSMITER
2262 010764 340 340 .BYTE 340,340 ;LEVEL
2263 010766 052777 000020 170410 BIS #RCVEN,#RXCSR ;TURN ON THE RECEIVER
2264 010774 052777 000100 170402 BIS #RINTEN,#RXCSR ;TURN ON REC INTERRUPT ENABLE
2265 011002 105777 170404 1$: TSTB #TXCSR ;TEST FOR TX DONE
2266 011006 100375 BPL 1$ ;BR IF NOT SET
2267 011010 052777 000020 170374 2$: BIS #SEND,#TXCSR ;TURN ON SEND
2268 011016 012777 000400 170370 MOV #TSOM,#TXDBUF ;TURN ON START OF MESSAGE
2269 011024 012737 000005 011054 MOV #5,68$ ;LOAD THE NUMBER
2270 011032 032777 004000 170354 66$: BIT #TIMER,#TXDBUF ;CHECK THE TIMER BIT
2271 011040 001374 BNE 66$ ;BR IF SET
2272 011042 032777 004000 170344 67$: BIT #TIMER,#TXDBUF ;CHECK THE BIT
2273 011050 001774 BEQ 67$ ;BR IF CLEAR
2274 011052 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2275 011054 000005 68$: 5 ;OF TIMES TO REPEAT
2276 011056 001365 BNE 66$ ;BR IF MORE TO GO
2277 011060 105777 170326 3$: TSTB #TXCSR ;WAIT FOR DONE
2278 011064 100401 BMI 4$ ;BR IF SET
2279 011066 104000 HLT ;EXTERNAL CLOCKING STOPPED
2280 011070 005077 170320 4$: CLR #TXDBUF ;PUSH OUT DATA
2281 011074 052777 000100 170310 BIS #TXINTE,#TXCSR ;TURN ON TRANSMITTER INT ENABLE
2282 011102 005037 177776 CLR PS ;LOWER PROCESOR STATUS
2283 011106 5$:
2284 011106 012737 000040 011136 MOV #32,73$ ;LOAD THE NUMBER
2285 011114 032777 004000 170272 71$: BIT #TIMER,#TXDBUF ;CHECK THE TIMER BIT
2286 011122 001374 BNE 71$ ;BR IF SET
2287 011124 032777 004000 170262 72$: BIT #TIMER,#TXDBUF ;CHECK THE BIT
2288 011132 001774 BEQ 72$ ;BR IF CLEAR
2289 011134 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2290 011136 000040 73$: 32 ;OF TIMES TO REPEAT
2291 011140 001365 BNE 71$ ;BR IF MORE TO GO
2292 011142 104001 HLT 1 ;FAILED TO INTERRUPT IN TIME
2293 011144 104400 6$: SCOPE ;SCOPE THIS TEST
2294
2295
2296 ;INTERRUPT SERVICE ROUTINES
2297 ;-----
2298
2299 ;RECEIVER:
2300 011146 017737 170234 001324 11$: MOV #RXDBUF,DATA ;GET THE REGISTER AND DATA
2301 011154 123737 001240 001324 CMPB TEMP2,DATA ;CHECK IT
2302 011162 001401 BEQ +4 ;BR IF OK
2303 011164 104002 HLT 2 ;COMPARISON ERROR
2304 011166 105237 001240 INCB TEMP2 ;COUNT UP EXPECTED
2305 011172 105737 001240 TSTB TEMP2 ;CHECK TO SEE IF DONE
2306 011176 001005 BNE 7$ ;BR IF NO
2307 011200 004537 006746 JSR R5,SETVEC ;YES--RESET THE VECTORS
2308 011204 011320 14$ ;RECEIVER
2309 011206 011214 12$ ;TRANSMITTER
2310 011210 340 340 .BYTE 340,340 ;LEVEL
2311
2312 011212 000002 7$: RTI ;RETURN
2313
2314
    
```

```

2315      : TRANSMITTER:
2316 011214 113777 001236 170172 12$: MOVB TEMP1,@TXDBUF ;LOAD THE TRANSMITTER BUFFER
2317 011222 105237 001236      INCB TEMP1 ;UP THE COUNT
2318 011226 122737 000377 001236      CMPB #377,TEMP1 ;ARE WE DONE
2319 011234 001026      BNE 13$ ;BR IF NO
2320 011236 012777 011246 170134      MOV #21$,@DUPTVC ;SETUP FOR NEXT PART
2321 011244 000422      BR 13$ ;LEAVE
2322 011246 012777 000377 170140 21$: MOV #377,@TXDBUF ;LOAD BUFFER
2323 011254 012777 011264 170116      MOV #22$,@DUPTVC ;SETUP NEXT PART
2324 011262 000413      BR 13$ ;LEAVE
2325 011264 012777 001000 170122 22$: MOV #TEOM,@TXDBUF ;SET END OF MSG
2326 011272 000240      NOP ;STALL
2327 011274 000240      NOP ;DITTO
2328 011276 042777 000120 170106      BIC #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
2329 011304 012777 006774 170066      MOV #NO.BTRAP,@DUPTVC ;LOAD VECTOR
2330 011312 012716 011106      13$: MOV #5$, (SP) ;CRUNCH STACK
2331 011316 000002      RTI ;RETURNS
2332
2333 011320 117737 170062 001324 14$: MOVB @RXDBUF,DATA ;GET FIRST PART OF CRC
2334 011326 105777 170052      TSTB @RXCSR ;WAIT FOR SECOND PART
2335 011332 100375      BPL -4 ;DITTO
2336 011334 017737 170046 001242      MOV @RXDBUF,TEMP3 ;GET THE REST OF THE CRC
2337 011342 113737 001242 001325      MOVB TEMP3,DATA+1 ;SET UP CRC CHARACTER
2338 011350 012716 011356      MOV #15$, (SP) ;SETUP FOR RETURN
2339 011354 000002      RTI ;RETURN
2340 011356 012737 000340 177776 15$: MOV #340,PS ;RAISE PS
2341 011364 005137 007156      COM CALBCC ;INVERT BCC
2342 011370 023737 007156 001324      CMP CALBCC,DATA ;COMPARE SOFTWARE AND HARDWARE BCC
2343 011376 001401      BEQ +4 ;BR IF OK
2344 011400 104004      HLT 4 ;BCC COMPARISON ERROR
2345 011402 032737 010000 001242      BIT #CRCERR,TEMP3 ;CHECK THE ERROR BIT
2346 011410 001401      BEQ +4 ;BR IF NO ERROR
2347 011412 104004      HLT 4 ;BCC ERROR--RECEIVER DOESN'T
2348      ;AGREE WITH WHAT TX SENT
2349 011414 052777 000400 167770      BIS #MRESET,@TXCSR ;RESET THE DEVICE
2350 011422 004737 005044      JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2351 011426 000137 011144      JMP 6$ ;LEAVE
2352
2353      ;***** TEST 10 *****
2354      ;*TEST OF THE DUP RUNNING A BINARY COUNT
2355      ;*PATTERN WITH A CRC CALCULATION
2356      ;*****
2357
2358      ;*****
2359      ;*
2360      ;* TEST 10
2361      ;*
2362      ;*****
2363      ;*****
2364 011432 012737 000010 001226 10: MOV #10,@TSTNO
2365 011440 012737 012240 001216      MOV #TST11,NEXT
2366 011446 052777 000400 167736      BIS #MRESET,@TXCSR ;RESET THE DEVICE
2367 011454 004737 005044      JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2368 011460 105737 001322      TSTB TCNFLAG
2369 011464 001532      BEQ 6$
2370 011466 005001      CLR R1 ;CLEAR OUT DATA

```

```

2371 011470 012737 102010 007152      MOV      #CRC.CCITT,XPOLY      ;SET UP THE POLYNOMIAL
2372 011476 012737 177777 007156      MOV      #-1,CALBCC           ;SETUP FOR THE FIRST TIME
2373 011504 013737 007156 011526 16$:    MOV      CALBCC,20$          ;ALLOW FOR THE NEXT CHARACTER
2374 011512 010137 011524           MOV      R1,17$              ;LOAD DATA
2375 011516 004537 007000           JSR      RS,SIMBCC           ;GO CALCULATE SOFTWARE BCC
2376 011522 000010           B.                   ;BASED ON THESE PARAMETERS
2377 011524 000001           17$:    .BLKW      1              ;DATA
2378 011526 000001           20$:    .BLKW      1              ;PREVIOUS BCC
2379 011530 105201           INCB     R1                  ;INCREMENT DATA
2380 011532 001364           BNE     16$                  ;BR IF MORE TO GO
2381 011534 012737 000001 001236      MOV      #1,TEMP1           ;LOAD DATA
2382 011542 005037 001240           CLR     TEMP2                ;CLEAR EXPECTED
2383 011546 012737 000340 177776      MOV      #340,PS            ;PS = 7
2384 011554 052777 010000 167630      BIS     #MEXT,@TXCSR        ;
2385 011562 004537 006746           JSR      RS,SETVEC           ;LOAD INTERRUPT VECTORS
2386 011566 011754           11$:    ;RECEIVER
2387 011570 012022           12$:    ;TRANSMITER
2388 011572           340     340                 .BYTE 340,340              ;LEVEL
2389 011574 052777 000020 167602      BIS     #RCVEN,@RXCSR       ;TURN ON THE RECEIVER
2390 011602 052777 000100 167574      BIS     #RINTEN,@RXCSR      ;TURN ON REC INTERRUPT ENABLE
2391 011610 105777 167576           15$:    TSTB     @TXCSR             ;TEST FOR TX DONE
2392 011614 100375           BPL     15$                  ;BR IF NOT SET
2393 011616 052777 000020 167566 25$:    BIS     #SEND,@TXCSR        ;TURN ON SEND
2394 011624 012777 000400 167562      MOV     #TSOM,@TXDBUF        ;TURN ON START OF MESSAGE
2395 011632 012737 000005 011662      MOV     #5,68$              ;LOAD THE NUMBER
2396 011640 032777 004000 167546 66$:    BIT     #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
2397 011646 001374           BNE     66$                  ;BR IF SET
2398 011650 032777 004000 167536 67$:    BIT     #TIMER,@TXDBUF      ;CHECK THE BIT
2399 011656 001774           BEQ     67$                  ;BR IF CLEAR
2400 011660 005327           DEC     (PC)+                ;DECREMENT THE NUMBER
2401 011662 000005           68$:    5                          ;OF TIMES TO REPEAT
2402 011664 001365           BNE     66$                  ;BR IF MORE TO GO
2403 011666 105777 167520           3$:    TSTB     @TXCSR             ;WAIT FOR DONE
2404 011672 100401           BMI     4$                    ;BR IF SET
2405 011674 104000           HLT                    ;EXTERNAL CLOCKING STOPPED
2406 011676 005077 167512 4$:    CLR     @TXDBUF              ;PUSH OUT DATA
2407 011702 052777 000100 167502      BIS     #TXINTE,@TXCSR      ;TURN ON TRANSMITTER INT ENABLE
2408 011710 005037 177776           CLR     PS                    ;LOWER PROCESOR STATUS
2409 011714           5$:
2410 011714 012737 000040 011744      MOV     #32,73$              ;LOAD THE NUMBER
2411 011722 032777 004000 167464 71$:    BIT     #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
2412 011730 001374           BNE     71$                  ;BR IF SET
2413 011732 032777 004000 167454 72$:    BIT     #TIMER,@TXDBUF      ;CHECK THE BIT
2414 011740 001774           BEQ     72$                  ;BR IF CLEAR
2415 011742 005327           DEC     (PC)+                ;DECREMENT THE NUMBER
2416 011744 000040           73$:    32                       ;OF TIMES TO REPEAT
2417 011746 001365           BNE     71$                  ;BR IF MORE TO GO
2418 011750 104001           HLT                    ;FAILED TO INTERRUPT IN TIME
2419 011752 104400           6$:    SCOPE                    ;SCOPE THIS TEST
2420
2421
2422           ;INTERRUPT SERVICE ROUTINES
2423           ;-----
2424
2425           ;RECEIVER:
2426 011754 017737 167426 001324 11$:    MOV     @RXDBUF,DATA          ;GET THE REGISTER AND DATA

```


BINARY PATTERN TEST WITH BCC IN EXTERNAL MODE

```

2427 011762 123737 001240 001324      CMPB   TEMP2,DATA      ;CHECK IT
2428 011770 001401                BEQ    .+4              ;BR IF OK
2429 011772 104002                HLT    2                ;COMPARISON ERROR
2430 011774 105237 001240      INCB   TEMP2           ;COUNT UP EXPECTED
2431 012000 105737 001240      TSTB   TEMP2           ;CHECK TO SEE IF DONE
2432 012004 001005                BNE    7$              ;BR IF NO
2433 012006 004537 006746      JSR    R5,SETVEC      ;YES--RESET THE VECTORS
2434 012012 012126                14$                      ;RECEIVER
2435 012014 012022                12$                      ;TRANSMITTER
2436 012016      340      340      .BYTE  340,340        ;LEVEL
2437
2438 012020 000002                7$:  RTI                ;RETURN
2439
2440
2441
2442 012022 113777 001236 167364 12$:  MOVB   TEMP1,@TXDBUF  ;LOAD THE TRANSMITTER BUFFER
2443 012030 105237 001236                INCB   TEMP1           ;UP THE COUNT
2444 012034 122737 000377 001236      CMPB   #377,TEMP1     ;ARE WE DONE
2445 012042 001026                BNE    13$             ;BR IF NO
2446 012044 012777 012054 167326      MOV    #21$,@DUPTVC   ;SETUP FOR NEXT PART
2447 012052 000422                BR     13$             ;LEAVE
2448 012054 012777 000377 167332 21$:  MOV    #377,@TXDBUF   ;LOAD BUFFER
2449 012062 012777 012072 167310      MOV    #22$,@DUPTVC   ;SETUP NEXT PART
2450 012070 000413                BR     13$             ;LEAVE
2451 012072 012777 001000 167314 22$:  MOV    #TEOM,@TXDBUF  ;SET END OF MSG
2452 012100 000240                NOP                      ;STALL
2453 012102 000240                NOP                      ;DITTO
2454 012104 042777 000120 167300      BIC    #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
2455 012112 012777 006774 167260      MOV    #NO.BTRAP,@DUPTVC ;LOAD VECTOR
2456 012120 012716 011714 13$:  MOV    #5$, (SP)      ;CRUNCH STACK
2457 012124 000002                RTI                ;RETURNS
2458
2459 012126 117737 167254 001324 14$:  MOVB   @RXDBUF,DATA   ;GET FIRST PART OF CRC
2460 012134 105777 167244                TSTB   @RXCSR         ;WAIT FOR SECOND PART
2461 012140 100375                BPL    -4              ;DITTO
2462 012142 017737 167240 001242      MOV    @RXDBUF,TEMP3  ;GET THE REST OF THE CRC
2463 012150 113737 001242 001325      MOVB   TEMP3,DATA+1   ;SET UP CRC CHARACTER
2464 012156 012716 012164                MOV    #15$, (SP)     ;SETUP FOR RETURN
2465 012162 000002                RTI                ;RETURN
2466 012164 012737 000340 177776 15$:  MOV    #340,PS        ;RAISE PS
2467 012172 005137 007156                COM    CALBCC         ;INVERT BCC
2468 012176 023737 007156 001324      CMP    CALBCC,DATA    ;COMPARE SOFTWARE AND HARDWARE BCC
2469 012204 001401                BEQ    .+4              ;BR IF OK
2470 012206 104004                HLT    4                ;BCC COMPARISON ERROR
2471 012210 032737 010000 001242      BIT    #CRCERR,TEMP3  ;CHECK THE ERROR BIT
2472 012216 001401                BEQ    .+4              ;BR IF NO ERROR
2473 012220 104004                HLT    4                ;BCC ERROR--RECEIVER DOESN'T
2474
2475 012222 052777 000400 167162      BIS    #MRESET,@TXCSR ;RESET THE DEVICE
2476 012230 004737 005044                JSR    PC,SMALL       ;WAIT FOR RESET TO FINISH
2477 012234 000137 011752                JMP    6$              ;LEAVE
2478
2479
2480
2481
2482
;***** TEST 11 *****
;*THIS TEST WILL CHECK FOR ABORT SEQUENCE
;*OF THE DUP IN A DATA STREAM
;*****

```

```

2483
2484
2485
2486
2487
2488
2489 012240 012737 000011 001226
2490 012246 012737 012554 001216
2491 012254 052777 000400 167130
2492 012262 004737 005044
2493 012266 004537 006746
2494 012272 012452
2495 012274 006774
2496 012276 340 340
2497 012300 005000
2498 012302 005003
2499 012304 012737 000340 177776
2500 012312 052777 010377 167070
2501 012320 052777 000120 167056
2502 012326 052777 004020 167056
2503 012334 005037 177776
2504 012340 105777 167046
2505 012344 100375
2506 012346 052777 000400 167040
2507 012354 105777 167032
2508 012360 100375
2509 012362 012777 000377 167024
2510 012370 005200
2511 012372 022700 000005
2512 012376 001366
2513 012400 052777 002000 167006
2514 012406 012737 000310 012436
2515 012414 032777 004000 166772
2516 012422 001374
2517 012424 032777 004000 166762
2518 012432 001774
2519 012434 005327
2520 012436 000310
2521 012440 001365
2522 012442 104001
2523 012444 012706 001150
2524 012450 104400
2525
2526 012452 017701 166726
2527 012456 017702 166724
2528 012462 032701 000200
2529 012466 001001
2530 012470 104007
2531 012472 122702 000377
2532 012476 001401
2533 012500 104002
2534 012502 005203
2535 012504 022703 000003
2536 012510 001020
2537 012512 105777 166666
2538 012516 100375

```

```

:*****
:TEST 11
:*****
:*****
:*****
ST11: MOV #11, @TSTNO
MOV #TST12, NEXT
BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
JSR R5, SETVEC ; SET UP INTERRUPT VECTORS
4$ ; BASED ON THESE
NO. BTRAP ; PARAMETERS
.BYTE 340, 340 ; LEVEL
CLR R0 ; CLEAR
CLR R3 ; DITTO
MOV #340, PS ; PS=7
BIS #PRISEC!377, @PARCSR ; LOAD SEC STATION AND ADRS
BIS #RCVEN!RINTEN, @RXCSR ; TURN ON THE RECEIVER
BIS #SEND!SYSTST, @TXCSR ; TURN ON TRANSMITTER
CLR PS
1$: TSTB @TXCSR ; CHECK FOR TXDONE
BPL 1$ ; BR IF NOT SET
BIS #TSOM, @TXDBUF ; TURN ON START OF MSG
2$: TSTB @TXCSR ; WAIT FOR DONE
BPL 2$ ; AND THEN
3$: MOV #377, @TXDBUF ; LOAD A CHARACTER
INC R0 ; UPDATE CHARACTER COUNTER
CMP #5, R0 ; ARE ALL CHARACTERS LOADED?
BNE 2$ ; BR IF NO
BIS #TABORT, @TXDBUF ; TURN ON ABORT
MOV #200, 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$: 200. ; OF TIMES TO REPEAT
BNE 66$ ; BR IF MORE TO GO
HLT 1 ; RECEIVER DID NOT INTERRUPT IN TIME
11$: MOV #STACK, SP ; RESET STACK
SCOPE ; SCOPE THIS TEST
; RECEIVER INTERRUPT SERVICE ROUTINE
4$: MOV @RXCSR, R1 ; GET THE CONTROL REGISTER
MOV @RXDBUF, R2 ; GET THE BUFFER
BIT #RXDONE, R1 ; CHECK FOR DONE
BNE 5$ ; BR IF DONE SET
HLT 7 ; FALSE INTERRUPT
5$: CMPB #377, R2 ; CHECK DATA CHARACTER
BEQ 6$ ; BR IF A MATCH
HLT 2 ; DATA ERROR
6$: INC R3 ; INC THE # OF CHARS TO DO
CMP #3, R3 ; CHECK TO SEE IF DONE
BNE 10$ ; BR IF MORE TO GO
12$: TSTB @RXCSR ; CHECK FOR
BPL 12$ ; DONE

```

2539	012520	017702	166662			MOV	DRXDBUF,R2	;READ THE BUFFER
2540	012524	032702	002000			BIT	#RABORT,R2	;TEST ABORT
2541	012530	001001				BNE	7\$;BR IF SET
2542	012532	104010				HLT	10	;FAILED TO RECEIVE ABORT
2543	012534	012716	012444		7\$:	MOV	#11\$, (SP)	;SET UP FOR RETURN
2544	012540	052777	000400	166644		BIS	#MRESET, @TXCSR	;RESET THE DEVICE
2545	012546	004737	005044			JSR	PC, SMALL	;WAIT FOR RESET TO FINISH
2546	012552	000002			10\$:	RTI		;RETURN

```

***** TEST 12 *****
*THIS TEST PROVES THE RECEIVER WILL STOP
*ACCEPTING DATA IF SHUT OFF IN THE MIDDLE
*OF A MESSAGE, AND THAT IT WILL NOT
*RESTART UNTIL IT RECEIVES A FLAG
*****

```

```

*****
: TEST 12
*****

```

2562	012554	012737	000012	001226		TST12:	MOV	#12, @TSTNO	
2563	012562	012737	013344	001216			MOV	#TST13, NEXT	
2564	012570	052777	000400	166614			BIS	#MRESET, @TXCSR	;RESET THE DEVICE
2565	012576	004737	005044				JSR	PC, SMALL	;WAIT FOR RESET TO FINISH
2566	012602	005000					CLR	RO	;CLEAR FOR SOFTWARE
2567	012604	004537	006746				JSR	RS, SETVEC	;SET UP THE VECTORS
2568	012610	013034					3\$;RECEIVER
2569	012612	013234					14\$;TRANSMITTER
2570	012614	340	340				.BYTE	340, 340	;LEVEL
2571	012616	012737	000340	177776			MOV	#340, PS	;PROC STATUS=7
2572	012624	052777	001000	166556			BIS	#CRCEN, @PARCSR	
2573	012632	052777	000120	166544			BIS	#RCVEN!RINTEN, @RXCSR	;TURN ON RECEIVER
2574	012640	052777	004020	166544			BIS	#SEND!SYSTST, @TXCSR	;START TRANSMITTER
2575	012646	005037	177776				CLR	PS	;LOWER PS
2576	012652	105777	166534		1\$:		TSTB	@TXCSR	;CHECK FOR DONE
2577	012656	100375					BPL	1\$;BR IF NOT YET
2578	012660	052777	000400	166526			BIS	#TSOM, @TXDBUF	;TURN ON START OF MSG
2579	012666	052777	000100	166516			BIS	#TXINTE, @TXCSR	;TURN ON INT. ENABLE
2580	012674	012737	000764	012724			MOV	#500, 68\$;LOAD THE NUMBER
2581	012702	032777	004000	166504	66\$:		BIT	#TIMER, @TXDBUF	;CHECK THE TIMER BIT
2582	012710	001374					BNE	66\$;BR IF SET
2583	012712	032777	004000	166474	67\$:		BIT	#TIMER, @TXDBUF	;CHECK THE BIT
2584	012720	001774					BEQ	67\$;BR IF CLEAR
2585	012722	005327					DEC	(PC)+	;DECREMENT THE NUMBER
2586	012724	000764			68\$:		500.		;OF TIMES TO REPEAT
2587	012726	001365					BNE	66\$;BR IF MORE TO GO
2588	012730	104001					HLT	1	;DEVICE FAILED TO INTERRUPT IN TIME
2589	012732	012706	001150		2\$:		MOV	#STACK, SP	;RESET THE STACK
2590	012736	104400					SCOPE		;SCOPE THIS TEST
2591	012740	004537	006746		20\$:		JSR	RS, SETVEC	;SET UP VECTORS
2592	012744	013164					23\$;RECEIVER
2593	012746	006774					NO.BTRAP		;TRANSMITTER
2594	012750	340	340				.BYTE	340, 340	;LEVEL

```

2595 012752 052777 000020 166424      BIS      #RCVEN, @RXCSR
2596 012760 105777 166426      21$:    TSTB     @TXCSR      ; TEST DONE
2597 012764 100375      BPL      #70, @TXDBUF  ; BR IF NOT SET
2598 012766 012777 000070 166420      MOV      #50, @TXDBUF ; PUSH OUT DATA CHARACTER
2599 012774 012737 000062 013024      MOV      #50, 73$    ; LOAD THE NUMBER
2600 013002 032777 004000 166404      71$:    BIT      #TIMER, @TXDBUF ; CHECK THE TIMER BIT
2601 013010 001374      BNE      71$        ; BR IF SET
2602 013012 032777 004000 166374      72$:    BIT      #TIMER, @TXDBUF ; CHECK THE BIT
2603 013020 001774      BEQ      72$        ; BR IF CLEAR
2604 013022 005327      DEC      (PC)+      ; DECREMENT THE NUMBER
2605 013024 000062      73$:    50.         ; OF TIMES TO REPEAT
2606 013026 001365      BNE      71$        ; BR IF MORE TO GO
2607 013030 104001      HLT      1          ; FAILED TO INTERRUPT IN TIME
2608 013032 000737      BR       2$         ; FINISH
2609                                     ; INTERRUPT SVC ROUTINES
2610
2611                                     ; RECEIVER
2612 013034 017704 166344      3$:    MOV      @RXCSR, R4 ; GET THE CONTROL REGISTER
2613 013040 017705 166342      MOV      @RXDBUF, R5 ; GET THE BUFFER
2614 013044 032705 000400      BIT      #RSOM, R5   ; CHECK FOR START OF MSG
2615 013050 001001      BNE      4$         ; BR IF SET
2616 013052 104011      HLT      1          ; FAILED TO RECEIVE SOM
2617 013054 032704 000200      4$:    BIT      #RXDONE, R4 ; CHECK FOR DONE
2618 013060 001001      BNE      5$         ; BR IF SET
2619 013062 104007      HLT      7          ; FALSE INTERRUPT
2620 013064 122705 000377      5$:    CMPB     #377, R5 ; CHECK DATA
2621 013070 001401      BEQ      6$         ; BR IF A MATCH
2622 013072 104002      HLT      2          ; DATA ERROR
2623 013074 012777 013104 166272      6$:    MOV      #10$, @DUPRVC ; RELOAD THE VECTOR
2624 013102 000002      7$:    RTI          ; RETURN
2625 013104 017705 166276      10$:   MOV      @RXDBUF, R5 ; GET THE BUFFER
2626 013110 122705 000377      CMPB     #377, R5   ; CHECK THE CHARACTER
2627 013114 001401      BEQ      11$        ; BR IF A MATCH
2628 013116 104002      HLT      2          ; DATA ERROR
2629 013120 042777 000020 166256      11$:   BIC      #RCVEN, @RXCSR ; TURN OFF THE RECEIVER
2630 013126 012777 013136 166240      MOV      #12$, @DUPRVC ; RELOAD THE VECTOR
2631 013134 000762      BR       7$         ; RETURN
2632 013136 017704 166242      12$:   MOV      @RXCSR, R4 ; GET THE CONTROL REGISTER
2633 013142 017705 166240      MOV      @RXDBUF, R5 ; GET THE BUFFER
2634 013146 122705 000252      CMPB     #252, R5   ; CHECK THE CHARACTER
2635 013152 001402      BEQ      13$        ; BR IF A MATCH
2636 013154 104007      HLT      7          ; FALSE INTERRUPT
2637 013156 000751      BR       7$         ;
2638 013160 104007      13$:   HLT      7          ; DEVICE INTERRUPTED AFTER RX ENABLE
2639 013162 000747      BR       7$         ; WAS CLEARED
2640 013164 017704 166214      23$:   MOV      @RXCSR, R4 ; GET THE CONTROL REG
2641 013170 017705 166212      MOV      @RXDBUF, R5 ; GET THE BUFFER
2642 013174 032715 000400      BIT      #RSOM, (R5) ; CHECK START OF MSG
2643 013200 001001      BNE      24$        ; BR IF SET
2644 013202 104011      HLT      11         ; SOM FAILED TO SET
2645 013204 122705 000070      24$:   CMPB     #70, R5   ; CHECK DATA
2646 013210 001401      BEQ      25$        ; BR IF A MATCH
2647 013212 104002      HLT      2          ; DATA FAILED TO MATCH AFTER
2648                                     ; RESTARTING RECEIVER
2649 013214      25$:
2650 013214 052777 000400 166170      BIS      #MRESET, @TXCSR ; RESET THE DEVICE

```

```

2651 013222 004737 005044 JSR PC, SMALL ;WAIT FOR RESET TO FINISH
2652 013226 012716 012732 MOV #2$, (SP) ;CRUNCH STACK
2653 013232 000002 RTI ;RETURN
2654 ; TRANSMITTER
2655 013234 105777 166152 14$: TSTB @TXCSR ;CHECK DONE
2656 013240 100401 BMI 30$ ;BR IF SET
2657 013242 104007 HLT 7 ;FALSE INTERRUPT
2658 013244 012777 000377 166142 30$: MOV #377, @TXDBUF ;LOAD A CHARACTER
2659 013252 005200 INC R0 ;INC THE # TO DO
2660 013254 022700 000002 CMP #2, R0 ;CHECK TO SEE IF ALL ARE SENT
2661 013260 001030 BNE 15$ ;BR IF MORE TO GO
2662 013262 012777 013274 166110 MOV #16$, @DUPTVC ;RELOAD THE VECT
2663 013270 005000 CLR R0 ;CLEAR CHAR COUNT
2664 013272 000423 BR 15$
2665 013274 105777 166112 16$: TSTB @TXCSR ;TEST DONE
2666 013300 100401 BMI 17$ ;BR IF SET
2667 013302 104007 HLT 7 ;FALSE INTERRUPT
2668 013304 012777 000252 166102 17$: MOV #252, @TXDBUF ;LOAD A DATA CHARACTER
2669 013312 005200 INC R0 ;INC THE # TO DO
2670 013314 022700 000003 CMP #3, R0 ;CHECK FOR ALL DONE
2671 013320 001010 BNE 15$ ;BR IF MORE TO GO
2672 013322 012777 001400 166064 MOV #TEOM!T$OM, @TXDBUF ;END MSG
2673 013330 042777 000100 166054 BIC #TXINTE, @TXCSR
2674 013336 012716 012740 MOV #20$, (SP) ;CRUNCH STACK
2675 013342 000002 15$: RTI

```

```

***** TEST 13 *****
*THIS TEST WILL TRANSMIT CONTIGUOUS ONES CHARACTERS
*IN SECONDARY MODE WITH A BCC CHECK.
*****

```

```

*****
: TEST 13
*****

```

```

2687 013344 012737 000013 001226 TST13: MOV #13, @TSTNO
2688 013352 012737 014004 001216 MOV #TST14, NEXT
2689 013360 052777 000400 166024 BIS #MRESET, @TXCSR ;RESET THE DEVICE
2690 013366 004737 005044 JSR PC, SMALL ;WAIT FOR RESET TO FINISH
2691 013372 012737 000340 177776 MOV #340, PS ;SET STATUS=7
2692 013400 005000 CLR R0
2693 013402 005002 CLR R2 ;SETUP FOR SOFTWARE
2694 013404 012701 000377 MOV #377, R1 ;CALCULATION OF BCC
2695 013410 012737 102010 007152 MOV #CRC.CCITT, XPOLY ;LOAD THE POLYNOMIAL
2696 013416 012737 177777 007156 MOV #-1, CALBCC ;SETUP FOR FIRST TIME
2697 013424 013737 007156 013446 1$: MOV CALBCC, 3$ ;ALLOW FOR THE NEXT CHARACTER
2698 013432 010137 013444 MOV R1, 2$ ;LOAD DATA
2699 013436 004537 007000 JSR R5, SIMBCC ;GO CALCULATE SOFTWARE BCC
2700 013442 000010 B. ;BASED ON THOSE PARAMETERS
2701 013444 000001 2$: .BLKW 1 ;DATA
2702 013446 000001 3$: .BLKW 1 ;PREVIOUS BCC
2703 013450 005200 INC R0 ;INC THE # OF CHARS TO DO
2704 013452 022700 000005 CMP #5, R0 ;ARE WE DONE?
2705 013456 001362 BNE 1$ ;BR IF NO
2706 013460 005000 CLR R0 ;CLEAR OUT HOLD

```

```

2707 013462 004537 006746 JSR R5,SETVEC ;LOAD INTERRUPT VECTORS
2708 013466 013616 6$ ;RECEIVER
2709 013470 013722 11$ ;TRANSMITTER
2710 013472 340 340 .BYTE 340,340 ;LEVEL
2711 013474 052777 010377 165706 BIS #PR1SEC!377,@PARCSR ;ENTER SECONDARY MODE
2712 013502 052777 000120 165674 BIS #RCVEN!RINTEN,@RXCSR ;TURN ON RECEIVER AND INTERRUPTS
2713 013510 052777 004020 165674 BIS #SEND!SYSTST,@TXCSR ;TURN ON TRANSMITTER
2714 013516 105777 165670 20$: TSTB @TXCSR
2715 013522 100375 BPL 20$
2716 013524 012777 000400 165662 MOV #TSOM,@TXDBUF ;START MESSAGE
2717 013532 052777 000100 165652 BIS @TXINTE,@TXCSR ;TURN ON INTERRUPT ENABLE
2718 013540 005037 177776 CLR PS ;LOWER PS
2719 013544 4$:
2720 013544 012737 000040 013574 MOV #32,68$ ;LOAD THE NUMBER
2721 013552 032777 004000 165634 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2722 013560 001374 BNE 66$ ;BR IF SET
2723 013562 032777 004000 165624 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2724 013570 001774 BEQ 67$ ;BR IF CLEAR
2725 013572 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2726 013574 000040 68$: 32. ;OF TIMES TO REPEAT
2727 013576 001365 BNE 66$ ;BR IF MORE TO GO
2728 013600 104001 HLT 1 ;FAILED TO INTERRUPT IN TIME
2729 013602 5$:
2730 013602 052777 000400 165602 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2731 013610 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2732 013614 104400 SCOPE ;SCOPE THIS TEST
2733
2734 ;INTERRUPT SERVICE ROUTINES.
2735
2736 ;RECEIVER
2737 013616 017737 165564 001324 6$: MOV @RXDBUF,DATA ;GET THE DATA
2738 013624 120137 001324 CMPB R1,DATA ;CHECK IT
2739 013630 001401 BEQ .+4 ;BR IF A MATCH
2740 013632 104002 HLT 2 ;DATA ERROR
2741 013634 005200 INC R0 ;UPDATE THE # OF CHARS TO DO
2742 013636 022700 000004 CMP #4,R0 ;CHECK FOR ALL CHARS DONE
2743 013642 001003 BNE 7$ ;BR IF MORE TO GO
2744 013644 012777 013654 165522 7$: MOV #10$,@DUPRVC ;SETUP TO GET BCC
2745 013652 000002 RTI ;RETURN
2746
2747 013654 117737 165526 001324 10$: MOVB @RXDBUF,DATA ;GET THE FIRST HALF OF BCC
2748 013662 105777 165516 TSTB @RXCSR ;WAIT FOR
2749 013666 100375 BPL .-4 ;THE SECOND HALF
2750 013670 117737 165512 001325 MOVB @RXDBUF,DATA+1 ;GET THE SECOND HALF
2751 013676 005137 007156 COM CALBCC ;INVERT BCC
2752 013702 023737 007156 001324 CMP CALBCC,DATA ;CHECK IT
2753 013710 001401 BEQ .+4 ;BR IF OK
2754 013712 104004 HLT 4 ;BCC COMPARE ERROR
2755 013714 012716 013602 MOV #5$, (SP) ;FINISH TEST
2756 013720 000002 RTI ;RETURN
2757
2758 ;TRANSMITTER
2759 013722 012777 000377 165464 11$: MOV #377,@TXDBUF ;LOAD A DATA CHARACTER
2760 013730 005202 INC R2 ;INC THE # OF CHARS TO DO
2761 013732 022702 000005 CMP #5,R2 ;CHECK TO SEE OF DONE
2762 013736 001017 BNE 13$ ;BR IF MORE TO GO
    
```

CONTIGUOUS ONES OUTPUT TEST

2763	013740	012777	013750	165432
2764	013746	000413		
2765	013750	012777	001000	165436
2766	013756	000240		
2767	013760	000240		
2768	013762	042777	000120	165422
2769	013770	012777	006774	165402
2770	013776	012716	013544	
2771	014002	000002		

```

MOV #12$, @DUPTVC ; SETUP NEXT VECTOR
BR 13$ ; RETURN
12$: MOV #TEOM, @TXDBUF ; END MSG
NOP ; WAIT
NOP ; DITTO
BIC #SEND!TXINTE, @TXCSR ; TURN OFF TRANSMITTER
MOV #NO.BTRAP, @DUPTVC ; RESET THE VECTOR
13$: MOV #4$, (SP) ; GO BACK TO WAIT LOOP
RTI ; RETURN

```

2772				
2773				
2774				
2775				
2776				
2777				
2778				
2779				
2780				
2781				
2782				
2783				

```

;***** TEST 14 *****
;THIS TEST PROVES THE INTERACTION OF DEC MODE,
;TSOM, SYNC, TXACT, TXDONE
;*****

```

```

;*****
; TEST 14
;*****

```

2784	014004	012737	000014	001226
2785	014012	012737	014156	001216
2786	014020	052777	000400	165364
2787	014026	004737	005044	
2788	014032	012777	101026	165350
2789	014040	052777	004000	165344
2790	014046	052777	000020	165336
2791	014054	012777	000426	165332
2792	014062	012737	000005	014112
2793	014070	032777	004000	165316
2794	014076	001374		
2795	014100	032777	004000	165306
2796	014106	001774		
2797	014110	005327		
2798	014112	000005		
2799	014114	001365		
2800	014116	017704	165270	
2801	014122	032704	000200	
2802	014126	001001		
2803	014130	104016		
2804	014132	032704	001000	
2805	014136	001001		
2806	014140	104017		
2807	014142			
2808	014142	052777	000400	165242
2809	014150	004737	005044	
2810	014154	104400		

```

TST14: MOV #14, @TSTNO
MOV #TST15, NEXT
BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
MOV #DECMOD!26!CRCEN, @PARCSR
BIS #SYSTST, @TXCSR ; ENTER SYSTEM TEST MODE
BIS #SEND, @TXCSR ; TURN ON TRANSMITTER
MOV #TSOM!26, @TXDBUF ; OUTPUT A SYNC CHAR
MOV #5, 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$: S ; OF TIMES TO REPEAT
BNE 66$ ; BR IF MORE TO GO
MOV @TXCSR, R4 ; GET THE CSR
BIT #TXDONE, R4 ; CHECK TRANSMITTER DONE
BNE 15$ ; BR IF SET
2804: HLT 16 ; TXDONE FAILED TO SET
15$: BIT #TXACT, R4 ; TEST ACTIVE
BNE 25$ ; BR IF SET
2806: HLT 17 ; ACTIVE FAILED TO SET
25$: BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
SCOPE ; SCOPE THIS TEST

```

2811				
2812				
2813				
2814				
2815				
2816				
2817				
2818				

```

;***** TEST 15 *****
;THIS TEST PROVES THE INTERACTION OF TEOM,
;SEND, TXACT AND TXDONE IN DEC MODE.
;*****

```

```

;*****

```

2819
2820
2821
2822
2823
2824
2825
2826
2827
2828
2829
2830
2831
2832
2833
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
2848
2849
2850
2851
2852
2853
2854
2855
2856
2857
2858
2859
2860
2861
2862
2863
2864
2865
2866
2867
2868
2869
2870
2871
2872
2873
2874

014156 012737 000015 001226
014164 012737 014352 001216
014172 052777 000400 165212
014200 004737 005044
014204 012777 101026 165176
014212 052777 004000 165172
014220 052777 000020 165164
014226 012777 000426 165160
014234 105777 165152
014240 100375
014242 012777 000426 165144
014250 105777 165136
014254 100375
014256 012777 001000 165130
014264 042777 000020 165120
014272 012737 000025 014322
014300 032777 004000 165106
014306 001374
014310 032777 004000 165076
014316 001774
014320 005327
014322 000025
014324 001365
014326 105777 165060
014332 100401
014334 104016
014336 032777 001000 165046
014344 001401
014346 104020
014350 104400

```

: TEST 15 *
: *****
: *****
TST15: MOV #15, @TSTNO
MOV #TST16, NEXT
BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
MOV #DECMOD!26!CRCEN, @PARCSR
BIS #SYSTST, @TXCSR ; ENTER SYSTEM TEST MODE
BIS #SEND, @TXCSR ; TURN ON TRANSMITTER
MOV #TSON!26, @TXDBUF ; OUTPUT A SYNC CHAR
1$: TSTB @TXCSR ; CHECK FOR DONE
BPL 1$ ; BR IF NOT YET
MOV #TSON!26, @TXDBUF ; LOAD A SECOND SYNC
2$: TSTB @TXCSR ; AND NOW WAIT
BPL 2$ ; FOR DONE AGAIN
MOV #TEOM, @TXDBUF ; SET END OF MSG
BIC #SEND, @TXCSR ; TURN OFF TRANSMITTER
MOV #25, 66$ ; 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$: 25 ; OF TIMES TO REPEAT
BNE 66$ ; BR IF MORE TO GO
TSTB @TXCSR ; CHECK DONE
BMI 3$ ; BR IF SET
HLT 16 ; DONE FAILED TO SET AFTER TURNING OFF TX.
3$: BIT #TXACT, @TXCSR ; CHECK ACTIVE
BEQ 4$ ; BR IF OFF
HLT 20 ; ACTIVE IS STILL SET-SHOULD BE RESET
4$: SCOPE ; SCOPE FOR THIS TEST.

```

***** TEST 16 *****
*THIS TEST PROVES THAT THE DUP WILL NOT
*SYNC UP IN LESS THAN TWO SYNCs

```

: *****
: *****
: TEST 16 *
: *****
: *****
TST16: MOV #16, @TSTNO
MOV #TST17, NEXT
BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
MOV #DECMOD!26!CRCEN, @PARCSR
BIS #SYSTST, @TXCSR ; ENTER SYSTEM TEST MODE
BIS #RCVEN, @RXCSR ; LOAD RCVEN
BIS #SEND, @TXCSR ; TURN ON TRANSMITTER

```



```

2875 014430 012777 000426 164756      MOV      #TSOM!26,@TXDBUF      ;OUTPUT A SYNC CHAR
2876 014436 105777 164750      1$:     TSTB     @TXCSR             ;CHECK TRANSMITTER DONE
2877 014442 100375                BPL      1$                  ;WAIT TILL SET
2878 014444 012777 000125 164742      MOV      #125,@TXDBUF        ;LOAD DATA
2879 014452 012737 000005 014502      MOV      #5,68$             ;LOAD THE NUMBER
2880 014460 032777 004000 164726      66$:    BIT      #TIMER,@TXDBUF  ;CHECK THE TIMER BIT
2881 014466 001374                BNE      66$                 ;BR IF SET
2882 014470 032777 004000 164716      67$:    BIT      #TIMER,@TXDBUF  ;CHECK THE BIT
2883 014476 001774                BEQ      67$                 ;BR IF CLEAR
2884 014500 005327                DEC      (PC)+               ;DECREMENT THE NUMBER
2885 014502 000005                5                      ;OF TIMES TO REPEAT
2886 014504 001365                BNE      66$                 ;BR IF MORE TO GO
2887 014506 105777 164672      TSTB     @RXCSR             ;CHECK FOR RECEIVER DONE
2888 014512 100002                BPL      2$                  ;BR IF NOT SET
2889 014514 104021                HLT      21                  ;DEVICE SYNC'S UP IN LESS THAN 2 SYNC'S!!
2890 014516 000472                BR       5$                  ;LEAVE
2891 014520                2$:
2892 014520 052777 000400 164664      BIS      #MRESET,@TXCSR     ;RESET THE DEVICE
2893 014526 004737 005044                JSR      PC,SMALL           ;WAIT FOR RESET TO FINISH
2894 014532 012777 101026 164650      MOV      #RCEN!DECMOD!26,@PARCSR ;LOAD THE MODE AND SYNC CHAR
2895 014540 052777 000020 164636      BIS      #RCVEN,@RXCSR      ;TURN ON RECEIVER
2896 014546 052777 004000 164636      BIS      #SYSTST,@TXCSR     ;ENTER SYSTEM TEST MODE
2897 014554 052777 000020 164630      BIS      #SEND,@TXCSR       ;TURN ON TRANSMITTER
2898 014562 012777 000426 164624      MOV      #TSOM!26,@TXDBUF  ;OUTPUT A SYNC CHAR
2899 014570 105777 164616      69$:    TSTB     @TXCSR             ;CHECK DONE
2900 014574 100375                BPL      69$                 ;BR IF NOT SET
2901 014576 012777 000426 164610      MOV      #TSOM!26,@TXDBUF  ;SEND SYNC
2902 014604 105777 164602      3$:     TSTB     @TXCSR             ;CHECK DONE
2903 014610 100375                BPL      3$                  ;WAIT
2904 014612 012777 000125 164574      MOV      #125,@TXDBUF        ;LOAD DATA
2905 014620 012737 000020 014650      MOV      #20,74$            ;LOAD THE NUMBER
2906 014626 032777 004000 164560      72$:    BIT      #TIMER,@TXDBUF  ;CHECK THE TIMER BIT
2907 014634 001374                BNE      72$                 ;BR IF SET
2908 014636 032777 004000 164550      73$:    BIT      #TIMER,@TXDBUF  ;CHECK THE BIT
2909 014644 001774                BEQ      73$                 ;BR IF CLEAR
2910 014646 005327                DEC      (PC)+               ;DECREMENT THE NUMBER
2911 014650 000020                20                      ;OF TIMES TO REPEAT
2912 014652 001365                BNE      72$                 ;BR IF MORE TO GO
2913 014654 105777 164524      TSTB     @RXCSR             ;CHECK FOR DONE
2914 014660 100401                BMI      4$                  ;BR IF SET
2915 014662 104022                HLT      22                  ;FAILED TO RECEIVE DATA
2916 014664 017737 164516 001236      4$:     MOV      @RXDBUF,TEMP1     ;READ DATA
2917 014672 122737 000125 001236      CMPB     #125,TEMP1         ;CHECK IT
2918 014700 001401                BEQ      5$                  ;BR IF MATCH
2919 014702 104022                HLT      22                  ;DATA COMPARE ERROR
2920 014704                5$:
2921 014704 052777 000400 164500      BIS      #MRESET,@TXCSR     ;RESET THE DEVICE
2922 014712 004737 005044                JSR      PC,SMALL           ;WAIT FOR RESET TO FINISH
2923 014716 104400                SCOPE                       ;SCOPE THIS TEST

```

2924
2925
2926
2927
2928
2929
2930

```

:***** TEST 17 *****
:*THIS TEST PROVES THE RECEIVER WILL STRIP THE FIRST
:*TWO SYNC'S AND WILL PRESENT ALL SUBSEQUENT SYNC'S.
:*****

```

```

2931      ::*****
2932      : *
2933      : TEST 17
2934      : *
2935      :*****
2936      :*****
2937      TST17: MOV      #17, @TSTNO
2938      MOV      @TST20, NEXT
2939      BIS      @MRESET, @TXCSR ; RESET THE DEVICE
2940      JSR      PC, SMALL ; WAIT FOR RESET TO FINISH
2941      MOV      @DECMOD!26!CRCEN, @PARCSR
2942      BIS      @SYSTST, @TXCSR ; ENTER SYSTEM TEST MODE
2943      BIS      @RCVEN, @RXCSR ; LOAD RCVEN
2944      BIS      @SEND, @TXCSR ; TURN ON TRANSMITTER
2945      MOV      @TSOM!26, @TXDBUF ; OUTPUT A SYNC CHAR
2946      64$: BIT      @TIMER, @TXDBUF ; CHECK THE TIMER BIT
2947      BNE      64$ ; BR IF SET
2948      65$: BIT      @TIMER, @TXDBUF ; CHECK THE TIMER BIT
2949      BEQ      65$ ; BR IF CLEAR
2950      69$: TSTB    @TXCSR ; CHECK DONE
2951      BPL      69$ ; BR IF NOT SET
2952      MOV      @TSOM!26, @TXDBUF ; SEND SYNC
2953      70$: TSTB    @TXCSR ; CHECK DONE
2954      BPL      70$ ; BR IF NOT SET
2955      MOV      @TSOM!26, @TXDBUF ; SEND SYNC
2956      MOV      @20, 75$ ; LOAD THE NUMBER
2957      73$: BIT      @TIMER, @TXDBUF ; CHECK THE TIMER BIT
2958      BNE      73$ ; BR IF SET
2959      74$: BIT      @TIMER, @TXDBUF ; CHECK THE BIT
2960      BEQ      74$ ; BR IF CLEAR
2961      DEC      (PC)+ ; DECREMENT THE NUMBER
2962      75$: 20 ; OF TIMES TO REPEAT
2963      BNE      73$ ; BR IF MORE TO GO
2964      TSTB    @RXCSR ; CHECK FOR DONE
2965      BMI      1$ ; BR IF SET
2966      HLT     21 ; DONE NOT SET-DEVICE FAILED TO SYNC UP
2967      1$: MOVB  @RXDBUF, TEMP1 ; READ BUFFER
2968      CMPB   @26, TEMP1 ; CHECK FOR SYNC
2969      BEQ     2$ ; BR IF OK
2970      HLT     22 ; DATA ERROR
2971      2$: SCOPE ; SCOPE THIS TEST
2972
2973
2974      :***** TEST 20 *****
2975      :*THIS TEST PROVES THE DUP11 WILL
2976      :*IDLE SYNC. IDLE 64. SYNC
2977      :*****
2978
2979      :*****
2980      : *
2981      : TEST 20
2982      : *
2983      :*****
2984      :*****
2985      TST20: MOV      #20, @TSTNO
2986      MOV      @TST21, NEXT

```

```

2987 015156 052777 000400 164226 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2988 015164 004737 005044 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2989 015170 012777 101026 164212 MOV #DECMOD!26!CRCEN,@PARCSR
2990 015176 052777 004000 164206 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2991 015204 052777 000020 164172 BIS #RCVEN,@RXCSR ;LOAD RCVEN
2992 015212 052777 000020 164172 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
2993 015220 012777 000426 164166 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2994 015226 105777 164160 64$: TSTB @TXCSR ;CHECK DONE
2995 015232 100375 64$: BPL 64$ ;BR IF NOT SET
2996 015234 012777 000426 164152 MOV #TSOM!26,@TXDBUF ;SEND SYNC
2997 015242 105777 164144 65$: TSTB @TXCSR ;CHECK DONE
2998 015246 100375 65$: BPL 65$ ;BR IF NOT SET
2999 015250 012777 000426 164136 MOV #TSOM!26,@TXDBUF ;SEND SYNC
3000 015256 005037 001236 CLR TEMP1
3001 015262 005037 001240 CLR TEMP2
3002 015266 012737 000100 001236 MOV #64,TEMP1 ;LOAD # OF SYNCs
3003 015274 012737 000010 015324 MOV #10,70$ ;LOAD THE NUMBER
3004 015302 032777 004000 164104 68$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3005 015310 001374 68$: BNE 68$ ;BR IF SET
3006 015312 032777 004000 164074 69$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3007 015320 001774 69$: BEQ 69$ ;BR IF CLEAR
3008 015322 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3009 015324 000010 70$: 10 ;OF TIMES TO REPEAT
3010 015326 001365 68$: BNE 68$ ;BR IF MORE TO GO
3011 015330 105777 164056 1$: TSTB @TXCSR ;CHECK DONE
3012 015334 100401 1$: BMI 2$ ;BR IF SET
3013 015336 104016 16$: HLT 16 ;DONE FAILED TO SET
3014 015340 012777 000426 164046 2$: MOV #TSOM!26,@TXDBUF ;LOAD A SYNC
3015 015346 005337 001236 2$: DEC TEMP1 ;LOWER THE # OF SYNCs TO DO
3016 015352 001001 4$: BNE 4$ ;BR IF MORE TO GO
3017 015354 104400 3$: SCOPE ;SCOPE THIS TEST
3018
3019 015356 105777 164022 4$: TSTB @RXCSR ;CHECK RECEIVER DONE
3020 015362 100375 4$: BPL 4$ ;WAIT TILL SET
3021 015364 017737 164016 001240 MOV @RXDBUF,TEMP2 ;GET THE BUFFER
3022 015372 122737 000026 001240 CMPB #26,TEMP2 ;CHECK IT FOR SYNC
3023 015400 001753 1$: BEQ 1$ ;BR IF OK
3024 015402 104021 21$: HLT 21 ;CHARACTER IS TEMP2 NOT A SYNC!
3025 015404 000763 3$: BR 3$ ;LEAVE TEST
3026
3027
3028 ;***** TEST 21 *****
3029 ;*THIS TEST PROVES THE STRIP SYNC
3030 ;*FUNCTION OF THE RECEIVER. SYNC UP
3031 ;*THE RECEIVER. SEND DATA WITH A SYNC
3032 ;*CHARACTER IMBEDDED AND CHECK FOR
3033 ;*THE SYNC TO BE RECEIVED.
3034 ;*****
3035 ;*****
3036 ;*
3037 ;* TEST 21
3038 ;*
3039 ;*****
3040 ;*****
3041 015406 012737 000021 001226 ST21: MOV #21,@TSTNO
3042 015414 012737 016126 001216 MOV #TST22,NEXT

```

```

3043 015422 012737 000340 177776      MOV      #340,PS          ;RAISE STATUS
3044 015430 004537 006746              JSR      RS,SETVEC      ;SET UP VECTORS
3045 015434 015714              5$          ;BASED ON
3046 015436 006774              NO.BTRAP    ;THESE
3047 015440      340      340      .BYTE     340,340      ;PARAMETERS
3048
3049 015442 052777 000400 163742      BIS      #MRESET,ATXCSR ;RESET THE DEVICE
3050 015450 004737 005044              JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
3051 015454 012777 101026 163726      MOV      #DECMOD!26!CRCEN,APARCSR
3052 015462 052777 004000 163722      BIS      #SYSTST,ATXCSR ;ENTER SYSTEM TEST MODE
3053 015470 052777 000420 163706      BIS      #RCVEN!STPSYN,ARXCSR ;LOAD RCVEN!STPSYN
3054 015476 052777 000020 163706      BIS      #SEND,ATXCSR   ;TURN ON TRANSMITTER
3055 015504 012777 000426 163702      MOV      #TSOM!26,ATXDBUF ;OUTPUT A SYNC CHAR
3056 015512 105777 163674      64$: TSTB   ATXCSR          ;CHECK DONE
3057 015516 100375              BPL      64$          ;BR IF NOT SET
3058 015520 012777 000426 163666      MOV      #TSOM!26,ATXDBUF ;SEND SYNC
3059 015526 105777 163660      65$: TSTB   ATXCSR          ;CHECK DONE
3060 015532 100375              BPL      65$          ;BR IF NOT SET
3061 015534 012777 000426 163652      MOV      #TSOM!26,ATXDBUF ;SEND SYNC
3062 015542 105777 163644      66$: TSTB   ATXCSR          ;CHECK DONE
3063 015546 100375              BPL      66$          ;BR IF NOT SET
3064 015550 012777 000426 163636      MCV      #TSOM!26,ATXDBUF ;SEND SYNC
3065 015556 005037 177776      CLR      PS            ;LOWER PS
3066 015562 052777 000100 163614      BIS      #RINTEN,ARXCSR ;TURN ON INTERRUPTS
3067 015570 105777 163616      1$: TSTB   ATXCSR          ;CHECK TX DONE
3068 015574 100375              BPL      1$          ;WAIT FOR SET
3069 015576 012777 000252 163610      MOV      #252,ATXDBUF    ;LOAD A CHARACTER
3070 015604 105777 163602      2$: TSTB   ATXCSR          ;CHECK TX DONE
3071 015610 100375              BPL      2$          ;WAIT TO BE SET
3072 015612 012777 000026 163574      MOV      #26,ATXDBUF     ;LOAD THE SYNC CHAR
3073 015620 105777 163566      3$: TSTB   ATXCSR          ;CHECK DONE AGAIN
3074 015624 100375              BPL      3$          ;WAIT
3075 015626 012777 000125 163560      MOV      #125,ATXDBUF    ;LOAD ANOTHER CHARACTER
3076 015634 105777 163552      4$: TSTB   ATXCSR          ;CHECK DONE
3077 015640 100375              BPL      4$          ;WAIT
3078 015642 012777 001000 163544      MOV      #TEOM,ATXDBUF   ;SET END OF MESSAGE
3079 015650 042777 000020 163534      BIC      #SEND,ATXCSR    ;TURN OFF TRANSMITTER
3080 015656 012737 000050 015706      MOV      #40,71$        ;LOAD THE NUMBER
3081 015664 032777 004000 163522      69$: BIT    #TIMER,ATXDBUF ;CHECK THE TIMER BIT
3082 015672 001374              BNE      69$          ;BR IF SET
3083 015674 032777 004000 163512      70$: BIT    #TIMER,ATXDBUF ;CHECK THE BIT
3084 015702 001774              BEQ      70$          ;BR IF CLEAR
3085 015704 005327              DEC      (PC)+         ;DECREMENT THE NUMBER
3086 015706 000050              71$: 40.             ;OF TIMES TO REPEAT
3087 015710 001365              BNE      69$          ;BR IF MORE TO GO
3088 015712 104023              HLT      23           ;FAILED TO TAKE A RECEIVER INTERRUPT
3089
3090      ;RECEIVER INTERRUPT SERVICE ROUTINE
3091 015714 017700 163464      5$: MOV     ARXCSR,RO      ;READ CSR
3092 015720 017701 163462      MOV     ARXDBUF,R1      ;READ BUFFER
3093 015724 032700 000200      BIT     #RXDONE,RO      ;CHECK FOR DONE
3094 015730 001001              BNE     6$             ;BR IF SET
3095 015732 104024              HLT     24           ;RX DONE FAILED TO SET-ERRONEOUS INTERRUPT
3096 015734 032700 004000      6$: BIT     #REACT,RO     ;CHECK FOR ACTIVE
3097 015740 001001              BNE     7$             ;BR IF SET
3098 015742 104025              HLT     25           ;RX ACTIVE FAILED TO SET
    
```

```

3099 015744
3100 015744 005701
3101 015746 100001
3102 015750 104026
3103 015752 122701 000252
3104 015756 001401
3105 015760 104022
3106 015762 012777 015772 163404
3107 015770 000455
3108 015772 017700 163406
3109 015776 017701 163404
3110 016002 032700 000200
3111 016006 001001
3112 016010 104024
3113 016012 005701
3114 016014 100001
3115 016016 104026
3116 016020 122701 000026
3117 016024 001422
3118 016026 122701 000125
3119 016032 001402
3120 016034 104022
3121 016036 000415
3122 016040 104021
3123 016042 012777 016052 163324
3124 016050 000425
3125 016052 017700 163326
3126 016056 017701 163324
3127 016062 122701 000125
3128 016066 001401
3129 016070 104022
3130 016072 032777 004000 163304
3131 016100 001001
3132 016102 104025
3133 016104
3134 016104 052777 000400 163300
3135 016112 004737 005044
3136 016116 012706 001150
3137 016122 104400
3138 016124 000002
3139
3140
3141
3142
3143
3144
3145
3146
3147
3148
3149
3150
3151
3152 016126 012737 000022 001226
3153 016134 012737 016610 001216
3154 016142 012737 000340 177776

```

```

7$: TST R1 ;CHECK FOR ERROR
BPL 10$ ;BR IF NO ERROR
HLT 26 ;RECEIVER ERROR
10$: CMPB #252,R1 ;CHECK FOR CORRECT DATA
BEQ 11$ ;BR IF OK
HLT 22 ;DATA FAILED TO MATCH
11$: MOV #12$,@DUPRVC ;LOAD VECTOR
BR 20$ ;CONTINUE
12$: MOV @RXCSR,R0 ;READ CSR
MOV @RXDBUF,R1 ;READ BUFFER
BIT #RXDONE,R0 ;CHECK FOR DONE
BNE 13$ ;BR IF OK
HLT 24 ;RX DONE FAILED TO SET-ERRONEOUS INTERRUPT
13$: TST R1 ;TEST FOR ERROR
BPL 14$ ;BR IF NO ERROR
HLT 26 ;ERROR SET
14$: CMPB #26,R1 ;CHECK CHARACTER
BEQ 15$ ;BR IF OK-IF NOT, THEN
CMPB #125,R1 ;CHECK FOR CLEARING SYNC
BEQ 15$ ;BR IF A NEXT CHARACTER
HLT 22 ;ERRONEOUS CHARACTER
BR 16$ ;BR TO END OF TEST
15$: HLT 21 ;STRIPPED OUT THE SYNC CHAR!!
MOV #21$,@DUPRVC ;SET UP VECTOR
BR 20$ ;LEAVE
21$: MOV @RXCSR,R0 ;GET CSR
MOV @RXDBUF,R1 ;GET BUFFER
CMPB #125,R1 ;CHECK DATA
BEQ 16$ ;BR IF A MATCH
HLT 22 ;DATA COMPARE ERROR
16$: BIT #REACT,@RXCSR ;TEST ACTIVE
BNE 17$ ;BR IF ON
HLT 25 ;ACTIVE SHOULD BE ON
17$: BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
MOV #STACK,SP ;RESET STACK
SCOPE ;SCOPE THIS TEST
20$: RTI ;RETURN

```

```

***** TEST 22 *****
*THIS TEST PROVES THAT A BINARY COUNT
*PATTERN CAN BE RUN IN DEC MODE
*WITHOUT A BCC CALCULATION
*****

```

```

:*****
:
: TEST 22
:
:*****
:*****
TST22: MOV #22,@TSTNO
MOV #TST23,NEXT
MOV #340,PS

```

3155	016150	005000			CLR	R0		; CLR OUT DATA POINTER
3156	016152	005001			CLR	R1		; DITTO
3157	016154	004537	006746		JSR	R5, SETVEC		; SET UP INTERRUPTS
3158	016160	016370			4\$; RECEIVER
3159	016162	016542			17\$; TRANSMITTER
3160	016164	340	340		.BYTE	340, 340		; LEVEL
3161								
3162	016166	052777	000400	163216	BIS	#MRESET, @TXCSR		; RESET THE DEVICE
3163	016174	004737	005044		JSR	PC, SMALL		; WAIT FOR RESET TO FINISH
3164	016200	012777	101026	163202	MOV	#DECMOD!26!CRCEN, @PARCSR		
3165	016206	052777	004000	163176	BIS	#SYSTST, @TXCSR		; ENTER SYSTEM TEST MODE
3166	016214	052777	000020	163162	BIS	#RCVEN, @RXCSR		; LOAD RCVEN
3167	016222	052777	000020	163162	BIS	#SEND, @TXCSR		; TURN ON TRANSMITTER
3168	016230	012777	000426	163156	MOV	#TSOM!26, @TXDBUF		; OUTPUT A SYNC CHAR
3169	016236	032777	004000	163150	64\$: BIT	#TIMER, @TXDBUF		; CHECK THE TIMER BIT
3170	016244	001374			BNE	64\$; BR IF SET
3171	016246	032777	004000	163140	65\$: BIT	#TIMER, @TXDBUF		; CHECK THE TIMER BIT
3172	016254	001774			BEG	65\$; BR IF CLEAR
3173	016256	105777	163130		69\$: TSTB	@TXCSR		; CHECK DONE
3174	016262	100375			BPL	69\$; BR IF NOT SET
3175	016264	012777	000426	163122	MOV	#TSOM!26, @TXDBUF		; SEND SYNC
3176	016272	005037	177776		CLR	PS		
3177	016276	052777	000100	163100	BIS	#RINTEN, @RXCSR		; TURN ON INT ENABLES
3178	016304	052777	000100	163100	BIS	#TXINTE, @TXCSR		; DITTO
3179	016312				30\$: MOV	#200, 74\$; LOAD THE NUMBER
3180	016312	012737	000310	016342	MOV	#200, 74\$		
3181	016320	032777	004000	163066	72\$: BIT	#TIMER, @TXDBUF		; CHECK THE TIMER BIT
3182	016326	001374			BNE	72\$; BR IF SET
3183	016330	032777	004000	163056	73\$: BIT	#TIMER, @TXDBUF		; CHECK THE BIT
3184	016336	001774			BEG	73\$; BR IF CLEAR
3185	016340	005327			DEC	(PC)+		; DECREMENT THE NUMBER
3186	016342	000310			74\$: 200.			; OF TIMES TO REPEAT
3187	016344	001365			BNE	72\$; BR IF MORE TO GO
3188	016346	104023			HLT	23		; FAILED TO FINISH TEST
3189	016350				3\$:			
3190	016350	052777	000400	163034	BIS	#MRESET, @TXCSR		; RESET THE DEVICE
3191	016356	004737	005044		JSR	PC, SMALL		; WAIT FOR RESET TO FINISH
3192	016362	012706	001150		MOV	#STACK, SP		; RESET THE STACK
3193	016366	104400			SCOPE			; SCOPE THIS TEST
3194								
3195								
3196								
3197	016370	017702	163010		4\$: MOV	@RXCSR, R2		; SAVE CSR
3198	016374	017703	163006		MOV	@RXDBUF, R3		; SAVE BUFFER
3199	016400	032702	004000		BIT	#REACT, R2		; TEST RX ACTIVE
3200	016404	001004			BNE	5\$; BR IF OK
3201	016406	104025			HLT	25		; ACTIVE NOT SET
3202	016410	012716	016350		MOV	#3\$, (SP)		; SETUP FOR RETURN
3203	016414	000432			BR	12\$		
3204	016416	032702	000200		5\$: BIT	#RXDONE, R2		; TEST DONE
3205	016422	001004			BNE	6\$; BR IF OK
3206	016424	104024			HLT	24		; FALSE INTERRUPT
3207	016426	012716	016350		MOV	#3\$, (SP)		; SETUP FOR RETURN
3208	016432	000423			BR	12\$		
3209	016434	005703			6\$: TST	R3		; CHECK FOR ERROR
3210	016436	100004			BPL	7\$; BR IF NO ERROR

```

3211 016440 104026          HLT      26          ;DATA ERROR
3212 016442 012716 016350  MOV      #3$, (SP)  ;SET UP RETURN
3213 016446 000415          BR       12$
3214 016450 120103          7$:    CMPB   R1,R3    ;CHECK DATA
3215 016452 001404          BEQ     10$          ;BR IF OK
3216 016454 104022          HLT     22          ;BAD DATA
3217 016456 012716 016350  MOV      #3$, (SP)  ;SETUP RETURN
3218 016462 000407          BR       12$
3219 016464 005201          10$:   INC     R1          ;UPDATE DATA
3220 016466 001002          BNE     11$          ;BR IF MORE TO GO
3221 016470 012716 016350  MOV      #3$, (SP)  ;SETUP RETURN
3222 016474 012777 016504 162672 11$:   MOV      #22$, @DUPRVC ;SETUP NEW RETURN FOR INTERRUPT
3223 016502 000002          12$:   RTI
3224 016504 017702 162674 22$:   MOV      @RXCSR,R2
3225 016510 017703 162672  MOV      @RXDBUF,R3
3226 016514 005703          TST     R3
3227 016516 100001          BPL     23$
3228 016520 104026          HLT     26          ;ERROR
3229 016522 120103          23$:   CMPB   R1,R3
3230 016524 001401          BEQ     24$
3231 016526 104022          HLT     22          ;DATA COMPARE ERROR
3232 016530 105201          24$:   INCB   R1
3233 016532 001363          BNE     12$
3234 016534 012716 016350  MOV      #3$, (SP)
3235 016540 000760          BR       12$
3236
3237          ; TRANSMITTER
3238 016542 010077 162646 17$:   MOV      RO,@TXDBUF ;PUSH OUT DATA
3239 016546 105200          INCB   RO          ;UPDATE IT
3240 016550 001014          BNE     21$
3241 016552 105777 162634 20$:   TSTB   @TXCSR    ;BR IF MORE
3242 016556 100375          BPL     20$        ;CHECK FOR NEXT DONE
3243 016560 052777 001000 162626  BIS      #TEOM,@TXDBUF ;WAIT
3244 016566 042777 000120 162616  BIC      #SEND!TXINTE,@TXCSR ;END MSG
3245 016574 012777 006774 162576  MOV      #NO.BTRAP,@DUPTVC ;SHUT OF TRANSMITTER
3246 016602 012716 016312 21$:   MOV      #30$, (SP) ;RESET VECTOR ADRS
3247 016606 000002          RTI
3248
3249
3250
3251          ;***** TEST 23 *****
3252          ;*THIS TEST PROVES THAT A BINARY COUNT
3253          ;*PATTERN CAN BE RUN IN DEC MODE
3254          ;*WITH A BCC CALCULATION USING
3255          ;*THE CRC16 POLYNOMIAL
3256          ;*****
3257          ;*****
3258          ;*
3259          ;* TEST 23
3260          ;*
3261          ;*****
3262          ;*****
3263 016610 012737 000023 001226 1ST23: MOV      #23,@TSTNO
3264 016616 012737 017432 001216  MOV      #TST24,NEXT
3265 016624 012737 000340 177776  MOV      #340,PS
3266 016632 005000          CLR     RO

```

```

3267 016634 012737 120001 007152      MOV      #CRC16,XPOLY      ;SET THE POLYNOMIAL
3268 016642 005037 007156      CLR      CALBCC           ;CLEAR OUT OLD BCC
3269 016646 013737 007156 016670 1$:  MOV      CALBCC,35$       ;LOAD BCC
3270 016654 010037 016666      MOV      R0,2$           ;LOAD DATA
3271 016660 004537 007000      JSR      R5,SIMBCC        ;CALCULATE A SOFTWARE BCC
3272 016664 000010      B.                               ;BASED
3273 016666 000000      .WORD   0                ;ON THESE
3274 016670 000000 35$:  .WORD   0                ;PARAMETERS
3275 016672 105200      INCB     R0              ;UPDATE DATA
3276 016674 001364      BNE     1$              ;BR IF MORE TO GO
3277 016676 005000      CLR     R0              ;CLR OUT DATA POINTER
3278 016700 005001      CLR     R1              ;DITTO
3279 016702 004537 006746      JSR     R5,SETVEC       ;SET UP INTERRUPTS
3280 016706 017116      4$                               ;RECEIVER
3281 016710 017364      17$                              ;TRANSMITTER
3282 016712      340      340      .BYTE   340,340        ;LEVEL
3283
3284 016714 052777 000400 162470      BIS     #MRESET,@TXCSR   ;RESET THE DEVICE
3285 016722 004737 005044      JSR     PC,SMALL        ;WAIT FOR RESET TO FINISH
3286 016726 012777 100026 162454      MOV     #DECMOD!26,@PARCSR ;LOAD THE MODE AND SYNC CHARACTER
3287 016734 052777 004000 162450      BIS     #SYSTST,@TXCSR   ;ENTER SYSTEM TEST MODE
3288 016742 052777 000020 162434      BIS     #RCVEN,@RXCSR    ;LOAD RCVEN
3289 016750 052777 000020 162434      BIS     #SEND,@TXCSR     ;TURN ON TRANSMITTER
3290 016756 012777 000426 162430      MOV     #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
3291 016764 032777 004000 162422 64$:  BIT     #TIMER,@TXDBUF   ;CHECK THE TIMER BIT
3292 016772 001374      BNE     64$             ;BR IF SET
3293 016774 032777 004000 162412 65$:  BIT     #TIMER,@TXDBUF   ;CHECK THE TIMER BIT
3294 017002 001774      BEQ     65$             ;BR IF CLEAR
3295 017004 105777 162402 69$:  TSTB   @TXCSR           ;CHECK DONE
3296 017010 100375      BPL     69$             ;BR IF NOT SET
3297 017012 012777 000426 162374      MOV     #TSOM!26,@TXDBUF ;SEND SYNC
3298 017020 005037 177776      CLR     PS
3299 017024 052777 000100 162352      BIS     #RINTEN,@RXCSR   ;TURN ON INT ENABLES
3300 017032 052777 000100 162352      BIS     #TXINTE,@TXCSR   ;DITTO
3301 017040      30$:
3302 017040 012737 000310 017070      MOV     #200,74$        ;LOAD THE NUMBER
3303 017046 032777 004000 162340 72$:  BIT     #TIMER,@TXDBUF   ;CHECK THE TIMER BIT
3304 017054 001374      BNE     72$             ;BR IF SET
3305 017056 032777 004000 162330 73$:  BIT     #TIMER,@TXDBUF   ;CHECK THE BIT
3306 017064 001774      BEQ     73$             ;BR IF CLEAR
3307 017066 005327      DEC     (PC)+           ;DECREMENT THE NUMBER
3308 017070 000310      74$:  200.                   ;OF TIMES TO REPEAT
3309 017072 001365      BNE     72$             ;BR IF MORE TO GO
3310 017074 104023      HLT     23              ;FAILED TO FINISH TEST
3311 017076      3$:
3312 017076 052777 000400 162306      BIS     #MRESET,@TXCSR   ;RESET THE DEVICE
3313 017104 004737 005044      JSR     PC,SMALL        ;WAIT FOR RESET TO FINISH
3314 017110 012706 001150      MOV     #STACK,SP       ;RESET THE STACK
3315 017114 104400      SCOPE                   ;SCOPE THIS TEST
3316
3317
3318      ;RECEIVER INT SVC ROUTINE
3319 017116 017702 162262 4$:  MOV     @RXCSR,R2        ;SAVE CSR
3320 017122 017703 162260      MOV     @RXDBUF,R3      ;SAVE BUFFER
3321 017126 032702 004000      BIT     #REACT,R2       ;TEST RX ACTIVE
3322 017132 001004      BNE     5$              ;BR IF OK
    
```


3323	017134	104025				HLT	25		:ACTIVE NOT SET
3324	017136	012716	017076			MOV	#3\$, (SP)		:SETUP FOR RETURN
3325	017142	000433				BR	12\$		
3326	017144	032702	000200		5\$:	BIT	#RXDONE, R2		:TEST DONE
3327	017150	001004				BNE	6\$:BR IF OK
3328	017152	104024				HLT	24		:FALSE INTERRUPT
3329	017154	012716	017076			MOV	#3\$, (SP)		:SETUP FOR RETURN
3330	017160	000424				BR	12\$		
3331	017162	005703			6\$:	TST	R3		:CHECK FOR ERROR
3332	017164	100004				BPL	7\$:BR IF NO ERROR
3333	017166	104026				HLT	26		:DATA ERROR
3334	017170	012716	017076			MOV	#3\$, (SP)		:SET UP RETURN
3335	017174	000416				BR	12\$		
3336	017176	120103			7\$:	CMPB	R1, R3		:CHECK DATA
3337	017200	001404				BEQ	10\$:BR IF OK
3338	017202	104022				HLT	22		:BAD DATA
3339	017204	012716	017076			MOV	#3\$, (SP)		:SETUP RETURN
3340	017210	000410				BR	12\$		
3341	017212	005201			10\$:	INC	R1		:UPDATE DATA
3342	017214	001003				BNE	11\$:BR IF MORE TO GO
3343	017216	012716	017234			MOV	#13\$, (SP)		:SETUP TO FINISH TEST
3344	017222	000403				BR	12\$		
3345	017224	012777	017324	162142	11\$:	MOV	#22\$, @DUPRVC		:SETUP NEW RETURN FOR INTERRUPT
3346	017232	000002			12\$:	RTI			:RETURN
3347	017234	105777	162144		13\$:	TSTB	@RXCSR		:TEST DONE
3348	017240	100375				BPL	13\$:WAIT
3349	017242	017737	162140	001236		MOV	@RXDBUF, TEMP1		:GET DATA
3350	017250	105777	162130		14\$:	TSTB	@RXCSR		:CHECK DONE FOR HALF OF CRC
3351	017254	100375				BPL	14\$:WAIT
3352	017256	017737	162124	001240		MOV	@RXDBUF, TEMP2		:MOVE IT
3353	017264	113737	001240	001237		MOVB	TEMP2, TEMP1+1		:COMBINE BCC CHARACTER
3354	017272	023737	007156	001236		CMP	CALBCC, TEMP1		:BR IF A MATCH
3355	017300	001401				BEQ	15\$:AFTER CHECKING IT
3356	017302	104027				HLT	27		:CRC COMPARE ERROR--THE
3357									:SOFTWARE DOESN'T AGREE
3358									:WITH WHAT THE TRANSMITTER
3359									:SENT. SEE THE FRONT OF
3360									:THE LISTING FOR SPECIAL
3361									:CRC DEBUG AID TEST.
3362	017304	032737	010000	001240	15\$:	BIT	#CRCERR, TEMP2		:CHECK FOR ERROR
3363	017312	001001				BNE	16\$:BR IF OK
3364	017314	104030				HLT	30		:HARDWARE DETECTED CRC ERROR
3365									:RECEIVER DOESN'T AGREE WITH
3366									:WHAT THE TRANSMITTER SENT
3367									:SEE FRONT OF LISTING FOR
3368									:SPECIAL CRC DEBUG AID
3369	017316	012716	017076		16\$:	MOV	#3\$, (SP)		:LOAD END OF TEST
3370	017322	000743				BR	12\$:RETURN
3371	017324	017702	162054		22\$:	MOV	@RXCSR, R2		
3372	017330	017703	162052			MOV	@RXDBUF, R3		
3373	017334	005703				TST	R3		
3374	017336	100001				BPL	23\$		
3375	017340	104026				HLT	26		:ERROR
3376	017342	120103			23\$:	CMPB	R1, R3		
3377	017344	001401				BEQ	24\$		
3378	017346	104022				HLT	22		:DATA COMPARE ERROR

```

3379 017350 105201          24$: INCB R1
3380 017352 001327          BNE 12$
3381 017354 012777 017234 162012  MOV #13$, @DUPRVC
3382 017362 000723          BR 12$
3383
3384          ; TRANSMITTER
3385 017364 010077 162024 17$: MOV R0, @TXDBUF ; PUSH OUT DATA
3386 017370 105200          INCB R0 ; UPDATE IT
3387 017372 001014          BNE 21$ ; BR IF MORE
3388 017374 105777 162012 20$: TSTB @TXCSR ; CHECK FOR NEXT DONE
3389 017400 100375          BPL 20$ ; WAIT
3390 017402 052777 001000 162004  BIS #TEOM, @TXDBUF ; END MSG
3391 017410 042777 000120 161774  BIC #SEND, TXINTE, @TXCSR ; SHUT OF TRANSMITTER
3392 017416 012777 006774 161754  MOV #NO.BTRAP, @DUPTVC ; RESET VECTOR ADRS
3393 017424 012716 017040 21$: MOV #30$, (SP)
3394 017430 000002          RTI ; RETURN
3395
3396
3397
3398
3399          ;***** TEST 24 *****
3400          ;*TEST TO PROVE THE DEVICE IDLES SYNCs AND
3401          ;*WILL SHIFT OUT DATA AT THE APPROPRIATE TIME
3402          ;:*****
3403
3404          ;:*****
3405          ;*
3406          ;* TEST 24
3407          ;*
3408          ;:*****
3409          ;:*****
3410 017432 012737 000024 001226  TST24: MOV #24, @TSTNO
3411 017440 012737 017630 001216  MOV #TST25, NEXT
3412 017446 052777 000400 161736  BIS #MRESET, @TXCSR ; RESET THE DEVICE
3413 017454 004737 005044          JSR PC, SMALL ; WAIT FOR RESET TO FINISH
3414 017460 052777 014000 161724  BIS #MODE, @TXCSR ; ENTER MAINT MODE
3415 017466 012777 000020 161710  MOV #RCVEN, @RXCSR ; TURN ON RECEIVER
3416 017474 012777 100026 161706  MOV #DECMOD!26, @PARCSR ; ENTER DECMODE AND SYNC CHAR
3417 017502 052777 000020 161702  BIS #SEND, @TXCSR ; TURN ON TRANSMITTER
3418 017510 012777 000426 161676  MOV #TSOM!26, @TXDBUF ; PUSH OUT SYNCs
3419 017516 104412 000044          PKCLK 36.
3420 017522 012777 000252 161664  MOV #252, @TXDBUF ; LOAD DATA
3421 017530 104412 000024          PKCLK 20. ; PUSH OUT ANOTHER SYNC
3422 017534 105777 161644          TSTB @RXCSR ; CHECK TO SEE IF SYNC ARRIVED
3423 017540 100401          BMI 1$ ; BR IF YES
3424 017542 104021          HLT 21
3425 017544 017737 161636 001324 1$: MOV @RXDBUF, DATA ; GET THE REC CHAR
3426 017552 122737 000026 001324  CMPB #26, DATA ; CHECK FOR SYNC
3427 017560 001401          BEQ 2$ ; BR IF MATCH
3428 017562 104021          HLT 21 ; FAILED TO RECEIVE THIRD SYNC
3429 017564 042777 000020 161620 2$: BIC #SEND, @TXCSR ; TURN OFF TRANSMITTER
3430 017572 104412 000016          PKCLK 14. ; PUSH OUT DATA
3431 017576 105777 161602          TSTB @RXCSR ; CHECK FOR REC DATA
3432 017602 100401          BMI 3$ ; BR IF YES
3433 017604 104026          HLT 26 ; FAILED TO GET A DATA DONE
3434 017606 017737 161574 001324 3$: MOV @RXDBUF, DATA ; GET THE DATA

```



```

3491 020060 001401
3492 020062 104022
3493 020064
3494 020064 012702 000003
3495 020070 005000
3496 020072 105201
3497 020074 110137 001236
3498 020100 110137 001240
3499 020104 001300
3500 020106 104400

```

```

65: BEQ 6$ ;BR IF OK
      HLT 22 ;SYNC CHAR DOES NOT MATCH SENT
      MOV #3,R2 ;SET UP FOR NEXT SYNC
      CLR R0 ;DITTO
      INCB R1 ;DITTO
      MOVB R1,TEMP1
      MOVB R1,TEMP2
      BNE 1$ ;BR IF MORE TO GO
      SCOPE ;SCOPE THIS TEST

```

```

3501
3502
3503
3504
3505
3506
3507
3508
3509
3510
3511
3512
3513

```

```

***** TEST 26 *****
*THIS TEST PROVES THAT THE CRC ERROR BIT FUNCTIONS
*CORRECTLY. FORCE AN ERROR AND VERIFY THE BIT.
*****
:*****
:
: TEST 26
:
:*****
:*****

```

```

3514 020110 012737 000026 001226
3515 020116 012737 020450 001216
3516 020124 012737 000340 177776
3517 020132 004537 006746
3518 020136 020412
3519 020140 020344
3520 020142 340 340
3521 020144 005001
3522
3523 020146 052777 000400 161236
3524 020154 004737 005044
3525 020160 012777 100026 161222
3526 020166 052777 004000 161216
3527 020174 052777 000020 161202
3528 020202 052777 000020 161202
3529 020210 012777 000426 161176
3530 020216 105777 161170
3531 020222 100375
3532 020224 012777 000426 161162
3533 020232 105777 161154
3534 020236 100375
3535 020240 012777 000426 161146
3536 020246 005037 177776
3537 020252 052777 000100 161124
3538 020260 052777 000100 161124
3539 020266
3540 020266 012737 000040 020316
3541 020274 032777 004000 161112
3542 020302 001374
3543 020304 032777 004000 161102
3544 020312 001774
3545 020314 005327
3546 020316 000040

```

```

TST26: MOV #26,@TSTNO
        MOV #TST27,NEXT
        MOV #340,PS ;RAISE PROCESSOR STATUS
        JSR RS,SETVEC ;SETUP VECTORS
        6$ ;RECEIVER
        3$ ;TRANSMITTER
        .BYTE 340,340 ;LEVEL
        CLR R1 ;CLEAR CHAR COUNT
        BIS #MRESET,@TXCSR ;RESET THE DEVICE
        JSR PC,SMALL ;WAIT FOR RESET TO FINISH
        MOV #DECMOD!26,@PARCSR ;LOAD THE MODE AND SYNC CHARACTER
        BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
        BIS #RCVEN,@RXCSR ;LOAD RCVEN
        BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
        MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
        64$: TSTB @TXCSR ;CHECK DONE
        BPL 64$ ;BR IF NOT SET
        MOV #TSOM!26,@TXDBUF ;SEND SYNC
        65$: TSTB @TXCSR ;CHECK DONE
        BPL 65$ ;BR IF NOT SET
        MOV #TSOM!26,@TXDBUF ;SEND SYNC
        CLR PS ;LOWER PROCESSOR STSTATUS
        BIS #RINTEN,@RXCSR ;TURN ON INTERRUPT ENABLES
        BIS #TXINTE,@TXCSR ;DITTO
        1$: MOV #32,70$ ;LOAD THE NUMBER
        68$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
        BNE 68$ ;BR IF SET
        69$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
        BEQ 69$ ;BR IF CLEAR
        DEC (PC)+ ;DECREMENT THE NUMBER
        70$: 32. ;OF TIMES TO REPEAT

```

```

3547 020320 001365      BNE 68$      ;BR IF MORE TO GO
3548 020322 104023      HLT 23      ;FAILED TO FINISH TEST
3549 020324      2$:
3550 020324 052777 000400 161060  BIS  #MRESET, @TXCSR ;RESET THE DEVICE
3551 020332 004737 005044      JSR  PC, SMALL ;WAIT FOR RESET TO FINISH
3552 020336 012706 001150      MOV  #STACK, SP ;RESET THE STACK
3553 020342 104400
3554
3555
3556      ; INTERRUPT SERVICE ROUTINES
3557      ; TRANSMITTER
3558 020344 005000      3$: CLR  R0      ;CLEAR DATA
3559 020346 010077 161042      MOV  R0, @TXDBUF ;LOAD DATA TO BUFFER
3560 020352 012777 020362 161020  MOV  #4$, @DUPTVC ;SETUP FOR NEXT INTERRUPT
3561 020360 000411      BR   5$      ;LEAVE
3562 020362 012777 001000 161024  4$: MOV  #TEOM, @TXDBUF ;END OF MSG--OUTPUT CRC
3563 020370 042777 000120 161014  BIC  #SEND, TXINTE, @TXCSR ;TRUN OFF THE
3564 020376 012777 006774 160774  MOV  #NO.BTRAP, @DUPTVC ;TRANSMITTER AND TXINTEN
3565 020404 012716 020266      5$: MOV  #1$, (SP) ;SETUP TO RETURN
3566 020410 000002      RTI ;RETURN
3567
3568      ; RECEIVER
3569 020412 017737 160770 001324  6$: MOV  @RXDBUF, DATA ;GET THE DATA
3570 020420 005201      INC  R1      ;CHECK FOR LAST CHAR
3571 020422 022701 000004      CMP  #4, R1 ;AND BRANCH IF
3572 020426 001007      BNE 10$     ;NOT YET
3573 020430 032737 010000 001324  BIT  #CRCERR, DATA ;CHECK FOR CRC ERROR
3574 020436 001401      BEQ 7$     ;BR IF CRC ERROR SEEN
3575 020440 104014      HLT 14     ;FAILED TO CATCH CRC ERROR!!!!
3576 020442 012716 020324      7$: MOV  #2$, (SP) ;FINISH TEST
3577 020446 000002      10$: RTI ;RETURN
3578
3579
3580
3581
3582
3583
3584
3585
3586
3587
3588
3589
3590
3591

```

```

:***** TEST 27 *****
:THIS TEST PROVES THE DEVICE WILL HANDLE THE
:DDCMP PROCALL. SEND AND RECEIVE SYNCs,
:FOLLOWED BY DATA, BCC, DATA AND FINAL BCC.
:*****

```

```

:*****
:TEST 27
:*****

```

```

3592 020450 012737 000027 001226  TST27: MOV  #27, @TSTNO
3593 020456 012737 021454 001216  MOV  #TST30, NEXT
3594 020464 012737 000340 177776  MOV  #340, P$ ;RAISE PROCESSOR STATUS
3595 020472 004537 006746      JSR  R5, SETVEC ;SET UP VECTORS
3596 020476 021104      10$: ;BASED ON
3597 020500 020726      2$: ;THESE
3598 020502 340 340      .BYTE 340, 340 ;PARAMETERS
3599 020504 005037 001236      CLR  TEMP1
3600 020510 005037 001240      CLR  TEMP2
3601 020514 005037 001242      CLR  TEMP3
3602 020520 005037 001244      CLR  TEMP4

```

```

3603 020524 005037 001246 CLR TEMP5
3604
3605 020530 052777 000400 160654 BIS #MRESET, @TXCSR ; RESET THE DEVICE
3606 020536 004737 005044 JSR PC, SMALL ; WAIT FOR RESET TO FINISH
3607 020542 012777 100026 160640 MOV #DECMOD!26, @PARCSR ; LOAD THE MODE AND SYNC CHARACTER
3608 020550 052777 004000 160634 BIS #SYSTST, @TXCSR ; ENTER SYSTEM TEST MODE
3609 020556 052777 000420 160620 BIS #RCVEN!STPSYN, @RXCSR ; LOAD RCVEN!STPSYN
3610 020564 052777 000020 160620 BIS #SEND, @TXCSR ; TURN ON TRANSMITTER
3611 020572 012777 000426 160614 MOV #TSOM!26, @TXDBUF ; OUTPUT A SYNC CHAR
3612 020600 105777 160606 64$: TSTB @TXCSR ; CHECK DONE
3613 020604 100375 BPL 64$ ; BR IF NOT SET
3614 020606 012777 000426 160600 MOV #TSOM!26, @TXDBUF ; SEND SYNC
3615 020614 105777 160572 65$: TSTB @TXCSR ; CHECK DONE
3616 020620 100375 BPL 65$ ; BR IF NOT SET
3617 020622 012777 000426 160564 MOV #TSOM!26, @TXDBUF ; SEND SYNC
3618 020630 052777 000100 160546 BIS #RINTEN, @RXCSR ; TURN ON INTERRUPTS
3619 020636 052777 000100 160546 BIS #TXINTE, @TXCSR ; DITTO
3620 020644 005037 177776 CLR PS ; LOWER PROCESSOR STATUS
3621 020650 100$:
3622 020650 012737 000144 020700 MOV #100, 70$ ; LOAD THE NUMBER
3623 020656 032777 004000 160530 68$: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
3624 020664 001374 BNE 68$ ; BR IF SET
3625 020666 032777 004000 160520 69$: BIT #TIMER, @TXDBUF ; CHECK THE BIT
3626 020674 001774 BEQ 69$ ; BR IF CLEAR
3627 020676 0C5327 DEC (PC)+ ; DECREMENT THE NUMBER
3628 020700 000144 70$: MOV #100, 70$ ; OF TIMES TO REPEAT
3629 020702 001365 BNE 68$ ; BR IF MORE TO GO
3630 020704 104023 HLT 23 ; FAILED TO FINISH TEST
3631 020706 1$:
3632 020706 052777 000400 160476 BIS #MRESET, @TXCSR ; RESET THE DEVICE
3633 020714 004737 005044 JSR PC, SMALL ; WAIT FOR RESET TO FINISH
3634 020720 012706 001150 MOV #STACK, SP ; RESET THE STACK
3635 020724 104400 SCOPE ; SCOPE THIS TEST
3636
3637 ; INTERRUPT SERVICE ROUTINES
3638 ; TRANSMITTER
3639
3640 020726 012777 000252 160460 2$: MOV #252, @TXDBUF ; LOAD FIRST DATA CHAR
3641 020734 012737 000026 001236 MOV #26, TEMP1 ; LOAD DATA
3642 020742 012777 020752 160430 MOV #3$, @DUPTVC ; RELOAD VECTOR
3643 020750 000452 BR 7$ ; LEAVE
3644 020752 013777 001236 160434 3$: MOV TEMP1, @TXDBUF ; MOV DATA TO BUFFER
3645 020760 105237 001236 INCB TEMP1 ; UPDATE DATA
3646 020764 122737 000032 001236 CMPB #32, TEMP1 ; CHECK FOR DONE
3647 020772 001041 BNE 7$ ; BR IF MORE TO SEND
3648 020774 012777 021004 160376 MOV #4$, @DUPTVC ; RELOAD VECTOR
3649 021002 000435 BR 7$ ; RETURN
3650 021004 012777 001000 160402 4$: MOV #TEOM, @TXDBUF ; PUT OUT BCC
3651 021012 012777 021022 160360 MOV #5$, @DUPTVC ; RELOAD VECTOR
3652 021020 000426 BR 7$ ; RETURN
3653 021022 013777 001240 160364 5$: MOV TEMP2, @TXDBUF ; LOAD DATA
3654 021030 105237 001240 INCB TEMP2 ; UPDATE DATA
3655 021034 122737 000100 001240 CMPB #100, TEMP2 ; CHECK FOR FINISH
3656 021042 001015 BNE 7$ ; BR IF MORE TO GO
3657 021044 012777 021054 160326 MOV #6$, @DUPTVC ; RELOAD VECTOR
3658 021052 000411 BR 7$ ; RETURN
    
```

```

3659 021054 012777 001000 160332 6$: MOV #TEOM, @TXDBUF ; PUSH OUT DATA BCC
3660 021062 042777 000120 160322 BIC #SEND, @TXINTE, @TXCSR ; SHUT DOWN TRANSMITTER
3661 021070 012777 006774 160302 MOV #NO.BTRAP, @DUPTVC ; RESET VECTOR
3662 021076 012716 020650 7$: MOV #100$, (SP) ; SETUP RETURN
3663 021102 000002 RTI ; RETURN
3664
3665 ;RECEIVER
3666
3667 021104 017737 160274 001242 10$: MOV @RXCSR, TEMP3 ; SAVE CSR
3668 021112 017737 160270 001244 MOV @RXDBUF, TEMP4 ; SAVE BUFFER
3669 021120 105737 001242 TSTB TEMP3 ; CHECK FOR DONE
3670 021124 100401 BMI 11$ ; BR IF SET
3671 021126 104024 HLT 24 ; FALSE INTERRUPT
3672 021130 005737 001244 11$: TST TEMP4 ; CHECK FOR ERROR
3673 021134 100001 BPL 12$ ; BR IF NO ERROR
3674 021136 104026 HLT 26 ; RECEIVER ERROR
3675 021140 122737 000252 001244 12$: CMPB #252, TEMP4 ; CHECK DATA
3676 021146 001401 BEQ 13$ ; BR IF A MATCH
3677 021150 104022 HLT 22 ; DATA COMPARE ERROR
3678 021152 012737 000026 001246 13$: MOV #26, TEMP5 ; LOAD NEXT EXPECTED
3679 021160 012777 021170 160206 MOV #14$, @DUPRVC ; RELOAD VECTOR
3680 021166 000531 BR 26$ ; LEAVE
3681 021170 017737 160212 001244 14$: MOV @RXDBUF, TEMP4 ; GET DATA
3682 021176 005737 001244 TST TEMP4 ; CHECK FOR ERROR
3683 021202 100001 BPL 15$ ; BR IF NO ERROR
3684 021204 104026 HLT 26 ; DATA ERROR
3685 021206 123737 001246 001244 15$: CMPB TEMP5, TEMP4 ; CHECK DATA
3686 021214 001401 BEQ 16$ ; BR IF A MATCH
3687 021216 104022 HLT 22 ; DATA COMPARE ERROR
3688 021220 105237 001246 16$: INCB TEMP5 ; UPDATE DATA
3689 021224 122737 000032 001246 CMPB #32, TEMP5 ; CHECK FOR FIRST PART FINISH
3690 021232 001107 BNE 26$ ; BR IF MORE TO GO
3691 021234 012777 021244 160132 MOV #17$, @DUPRVC ; SET UP NEXT VECTOR
3692 021242 000503 BR 26$ ; LEAVE
3693 021244 017737 160136 001244 17$: MOV @RXDBUF, TEMP4 ; GET THE BUFFER
3694 021252 005737 001244 TST TEMP4 ; TEST FOR ERROR
3695 021256 100001 BPL .+4 ; BR IF OK
3696 021260 104026 HLT 26 ; RECEIVER ERROR
3697 021262 012777 021272 160104 MOV #18$, @DUPRVC ; RELOAD THE VECTOR
3698 021270 000470 BR 26$ ; LEAVE
3699 021272 017737 160110 001324 18$: MOV @RXDBUF, DATA ; GET DATA
3700 021300 032737 010000 001324 BIT #CRCERR, DATA ; CHECK FOR CRC ERROR
3701 021306 001001 BNE 19$ ; BR IF OK
3702 021310 104014 HLT 14 ; CRC ERROR!!!!!!
3703 021312 012777 021326 160054 19$: MOV #20$, @DUPRVC ; SET UP VECTOR
3704 021320 005037 001330 CLR MIND ; SETUP FOR NEXT DATA
3705 021324 000452 BR 26$ ; LEAVE
3706 021326 017737 160054 001244 20$: MOV @RXDBUF, TEMP4 ; GET DATA
3707 021334 005737 001244 TST TEMP4 ; CHECK FOR ERROR
3708 021340 100001 BPL 21$ ; BR IF NO ERROR
3709 021342 104026 HLT 26 ; RECEIVER ERROR
3710 021344 123737 001330 001244 21$: CMPB MIND, TEMP4 ; CHECK DATA
3711 021352 001401 BEQ 22$ ; BR IF A MATCH
3712 021354 104022 HLT 22 ; DATA ERROR
3713 021356 105237 001330 22$: INCB MIND ; UPDATE SOFTWARE DATA
3714 021362 122737 000100 001330 CMPB #100, MIND ; CHECK FOR FINISH
    
```

3715	021370	001030				BNE	26\$;BR IF MORE TO GO
3716	021372	012777	021402	157774		MOV	#23\$, @DUPRVC		;RELOAD FINAL VECTOR
3717	021400	000424				BR	26\$;LEAVE
3718	021402	017737	160000	001244	23\$:	MOV	@RXDBUF, TEMP4		;GET DATA
3719	021410	005737	001244			TST	TEMP4		;CHECK FOR ERROR
3720	021414	100001				BPL	24\$;BR IF OK
3721	021416	104026				HLT	26		;RECEIVER ERROR ON FIRST OCTET
3722									;OF SECOND BCC
3723	021420	105777	157760		24\$:	TSTB	@RXCSR		;TEST DONE
3724	021424	100375				BPL	24\$;BR IF NOT SET
3725	021426	017737	157754	001324		MOV	@RXDBUF, DATA		;GET SECOND BCC OCTET
3726	021434	032737	010000	001324		BIT	#CRCERR, DATA		;CHECK FOR BCC ERROR
3727	021442	001001				BNE	25\$;BR IF OK
3728	021444	104014				HLT	14		;BCC ERROR ON SECOND PART OF MSG
3729	021446	012716	020706		25\$:	MOV	#1\$, (SP)		;SETUP TO FINISH TEST
3730	021452	000002			26\$:	RTI			;RETURN

```

:***** TEST 30 *****
:THIS TEST IS AN AID FOR DEBUGGING CRC
:ERRORS. A CHARACTER IS LOADED INTO THE
:BUF 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
:*****

```

```

:*****
:TEST 30
:*****
:*****
TST30: MOV #30, @TSTNO
MOV #.EOP, NEXT
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
MOV #CRC16, XPOLY ;LOAD THE POLYNOMIAL
MOV #125, 3$ ;LOAD DATA TO SOFTWARE BCC-CHANGE CHARACTER HERE
MOV 3$, SAVR1
CLR CALBCC ;CLEAR FOR SOFTWARE BCC
MOV CALBCC, 4$
CLR TEMP3
CLR TEMP4 ;CLEAR BIT COUNTER
CLR TEMP5
MOV #DECMOD!26, @PARCSR ;LOAD MODE AND SYNC CHARACTER
BIS #MMODE, @TXCSR ;ENTER MAINT MODE-PROGRAM CLOCKING
BIS #RCVEN!STPSYN, @RXCSR ;TURN ON RECEIVER
BIS #SEND, @TXCSR ;TURN ON TRANSMITTER
MOV #TSOM!26, @TXDBUF ;LOAD A SYNC
PKCLK 36. ;PUSH OUT 2 SYNCs
MOV 3$, @TXDBUF ;LOAD DATA
PKCLK 16. ;PUSH OUT ANOTHER SYNC
1$: PKCLK 2. ;PUSH OUT A BIT

```

3750	021454	012737	000030	001226	
3751	021462	012737	002764	001216	
3752	021470	052777	000400	157714	
3753	021476	004737	005044		
3754	021502	012737	120001	007152	
3755	021510	012737	000125	021656	
3756	021516	013737	021656	001252	
3757	021524	005037	007156		
3758	021530	013737	007156	021660	
3759	021536	005037	001242		
3760	021542	005037	001244		
3761	021546	005037	001246		
3762	021552	012777	100026	157630	
3763	021560	052777	014000	157624	
3764	021566	052777	000420	157610	
3765	021574	052777	000020	157610	
3766	021602	012777	000426	157604	
3767	021610	104412	000044		
3768	021614	013777	021656	157572	
3769	021622	104412	000020		
3770	021626	104412	000002		

3771	021632	013737	001244	001254		MOV	TEMP4,SAVR2	;SET UP TO TYPE
3772	021640	005237	001242			INC	TEMP3	
3773	021644	005237	001244			INC	TEMP4	;UPDATE BIT COUNTER
3774	021650	004537	007000		2\$:	JSR	R5,SIMBCC	;CALCULATE SOFTWARE BCC BASED ON THESE PARAMETERS
3775	021654	000001				1		;SHIFTS
3776	021656	000000			3\$:	.WORD	0	;DATA
3777	021660	000000			4\$:	.WORD	0	;PREVIOUS BCC
3778	021662	004737	021760			JSR	PC,5\$;CHECK TO SEE IF WE SHOULD WAIT FOR SCOPING
3779	021666	000241				CLC		;CLEAR FOR NEXT ROTATE
3780	021670	106037	021656			RORB	3\$;SET UP THE NEXT BIT
3791	021674	013737	007156	021660		MOV	CALBCC,4\$;FOR THE SOFTWARE BCC
3782	021702	022737	000006	001244		CMP	#6,TEMP4	
3783	021710	001002				BNE	.+6	
3784	021712	005077	157476			CLR	@TXDBUF	
3785	021716	022737	000014	001242		CMP	#12.,TEMP3	
3786	021724	001003				BNE	12\$	
3787	021726	012777	001000	157460		MOV	#TEOM,@TXDBUF	
3788	021734	022737	000020	001244	12\$:	CMP	#16.,TEMP4	;ALL DONE WITH THE CHARACTER?
3790								;INCREASE COMPARE # TO FORCE
3791	021742	001331				BNE	1\$;CRC OUT OF THE GENERATOR
3792	021744	052777	000400	157440		BIS	#MRESET,@TXCSR	;BR IF MORE TO GO
3793	021752	004737	005044			JSR	PC,SMALL	;RESET THE DEVICE
3794	021756	104400				SCOPE		;WAIT FOR RESET TO FINISH
3795								;SCOPE THIS TEST
3796	021760	032777	001000	157214	5\$:	BIT	#SW09,@SWR	;SW09=1?
3797	021766	001432				BEQ	6\$;BR IF NO
3798	021770	013704	007156			MOV	CALBCC,R4	;THE DATA CHARACTER IS ALWAYS
3799	021774	012737	000001	001256		MOV	#1,SAVR3	;FOLLOWED BY A ZERO. THE DATA IN
3800	022002	000241				CLC		;CRC SHOWS WHICH BIT OF THE 2 CHARS
3801	022004	006004			11\$:	ROR	R4	;IS BEING GENERATED
3802	022006	006137	001256			ROL	SAVR3	
3803	022012	103374				BCC	11\$	
3804	022014	105737	001246			TSTB	TEMP5	
3805	022020	001006				BNE	10\$	
3806	022022	104402	023152			TYPE	,EM17	;TYPE MSG
3807	022026	104402	023201			TYPE	MH1	;TYPE HEADER
3808	022032	105137	001246			COMB	TEMP5	
3809	022036	104410			10\$:	CONVRT		
3810	022040	023522				DT1		
3811	022042	105777	157136		7\$:	TSTB	@TKCSR	;CHECK TTY DONE--GO SCOPE THE CRC GENERATOR
3812	022046	100375				BPL	7\$;BR IF NOT YET
3813	022050	017701	157132			MOV	@TKDBR,R1	;READ THE BUFFER
3814	022054	000207			6\$:	RTS	PC	;RETURN
3815								
3816								

3817

(1)	022056	042377	050125	043040	EM1:	.ASCIZ	<377>/DUP FAILED TO INTERRUPT IN TIME /		
(1)	022120	042377	052101	020101	EM2:	.ASCIZ	<377>/DATA COMPARE ERROR /		
(1)	022145	377	047105	020104	EM3:	.ASCIZ	<377>/END OF MESSAGE /		
(1)	022166	041377	041503	042440	EM4:	.ASCIZ	<377>/BCC ERROR /		
(1)	022202	052377	040522	051516	EM5:	.ASCIZ	<377>/TRANSMITTER DONE /		
(1)	022225	106	044501	042514	EM6:	.ASCIZ	/FAILED TO SET /		
(1)	022244	051377	041505	044505	EM7:	.ASCIZ	<377>/RECEIVER INTERRUPT IN HALF-DUPLEX /		
(1)	022310	043377	046101	042523	EM10:	.ASCIZ	<377>/FALSE INTERRUPT /		
(1)	022332	040777	047502	052122	EM11:	.ASCIZ	<377>/ABORT SEQUENCE ERROR /		
(1)	022361	377	052123	051101	EM12:	.ASCIZ	<377>/START MESSAGE /		
(1)	022404	052777	042516	050130	EM13:	.ASCIZ	<377>/UNEXPECTED RECEIVER INTERRUPT /		
(1)	022444	052777	042516	050130	EM14:	.ASCIZ	<377>/UNEXPECTED TRANSMITTER INTERRUPT /		
(1)	022507	377	051124	047101	EM20:	.ASCIZ	<377>/TRANSMITTER DONE /		
(1)	022532	051377	041505	044505	EM21:	.ASCIZ	<377>/RECEIVER DONE /		
(1)	022552	052377	040522	051516	EM22:	.ASCIZ	<377>/TRANSMITTER ACTIVE /		
(1)	022577	377	042522	042503	EM23:	.ASCIZ	<377>/RECEIVER ACTIVE /		
(1)	022621	106	044501	042514	EM24:	.ASCIZ	/FAILED TO SET. /		
(1)	022641	106	044501	042514	EM25:	.ASCIZ	/FAILED TO CLEAR. /		
(1)	022663	377	054523	041516	EM26:	.ASCIZ	<377>/SYNC ERROR /		
(1)	022700	042377	052101	020101	EM27:	.ASCIZ	<377>/DATA ERROR /		
(1)	022715	377	042504	044526	EM30:	.ASCIZ	<377>/DEVICE FAILED TO INTERRUPT IN TIME /		
(1)	022762	043377	046101	042523	EM31:	.ASCIZ	<377>/FALSE INTERRUPT /		
(1)	023004	052377	040522	051516	EM32:	.ASCIZ	<377>/TRANSMITTER BCC ERROR IN DEC MODE /		
(1)	023051	377	042522	042503	EM33:	.ASCIZ	<377>/RECEIVER BCC ERROR IN DEC MODE /		
(1)	023113	377	042522	042503	EM15:	.ASCIZ	<377>/RECEIVER ERROR /		
(1)	023134	041377	041503	042440	EM16:	.ASCIZ	<377>/BCC ERROR!! /		
(1)	023152	041777	041522	043440	EM17:	.ASCIZ	<377>/CRC GENERATOR STATUS /		
(1)	023201	377	040504	040524	MH1:	.ASCIZ	<377>/DATA CHAR DATA BIT IN CRC GEN.		CRC FOR THIS BIT /

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

(1)

.ERRTAB:

0

0

0

EM1

;HALT 1

0

0

EM2

;HALT 2

0

0

EM3

;HALT 3

EM6

0

0

EM4

;HALT 4

0

0

EM5

;HALT 5

EM6

0

0

EM7

(1)	023342	000000	0		;HALT 6
(1)	023344	000000	0		
(1)					
(1)	023346	022310	EM10		
(1)	023350	000000	0		;HALT 7
(1)	023352	000000	0		
(1)					
(1)	023354	022332	EM11		
(1)	023356	000000	0		;HALT 10
(1)	023360	000000	0		
(1)					
(1)	023362	022361	EM12		
(1)	023364	022225	EM6		;HALT 11
(1)	023366	000000	0		
(1)					
(1)	023370	022404	EM13		
(1)	023372	000000	0		;HALT 12
(1)	023374	000000	0		
(1)					
(1)	023376	022444	EM14		
(1)	023400	000000	0		;HALT 13
(1)	023402	000000	0		
(1)					
(1)	023404	023134	EM16		
(1)	023406	000000	0		;HALT14
(1)	023410	000000	0		
(1)					
(1)	023412	022532	EM21		
(1)	023414	022621	EM24		;HALT15
(1)	023416	000000	0		
(1)					
(1)					
(1)	023420	022507	EM20		
(1)	023422	022621	EM24		;HALT16
(1)	023424	000000	0		
(1)					
(1)	023426	022552	EM22		
(1)	023430	022621	EM24		;HALT17
(1)	023432	000000	0		
(1)					
(1)	023434	022552	EM22		
(1)	023436	022641	EM25		;HALT20
(1)	023440	000000	0		
(1)					
(1)	023442	022663	EM26		
(1)	023444	000000	0		;HALT21
(1)	023446	000000	0		
(1)					
(1)	023450	022700	EM27		
(1)	023452	000000	0		;HALT22
(1)	023454	000000	0		
(1)					
(1)	023456	022715	EM30		
(1)	023460	000000	0		;HALT23
(1)	023462	000000	0		
(1)					

(1)	023464	022762		EM31	
(1)	023466	000000		0	;HALT24
(1)	023470	000000		0	
(1)					
(1)	023472	022577		EM23	
(1)	023474	022621		EM24	;HALT25
(1)	023476	000000		0	
(1)					
(1)	023500	023113		EM15	
(1)	023502	000000		0	;HALT26
(1)	023504	000000		0	
(1)					
(1)	023506	023004		EM32	
(1)	023510	000000		0	;HALT 27
(1)	023512	000000		0	
(1)					
(1)	023514	023051		EM33	
(1)	023516	000000		0	;HALT 30
(1)	023520	000000		0	
(1)	023522	000003		0	
(1)	023524	006	DT1:	3	
(1)	023526	001252	021	.BYTE	6,17.
(1)	023530	006	017	SAVR1	
(1)	023532	001254		.BYTE	6,15.
(1)	023534	006	002	SAVR2	
(1)	023536	001256		.BYTE	6,2
(1)				SAVR3	
(1)					
(1)	023540		CORMAX:		
3818	000001		.END		

TRP.PC = 006222	1763#													
TSOM = 000400	958#	2096	2161	2268	2394	2506	2578	2672	2716	2791	2831	2834	2875	
	2898	2901	2945	2952	2955	2993	2996	2999	3014	3055	3058	3061	3064	
	3168	3175	3290	3297	3418	3459	3529	3532	3535	3611	3614	3617	3766	
TSTNO 001226	861#	1129#	1710	1842	1848	1250	1935*	1967*	2001*	2035*	2069*	2143*	2240*	
	2364*	2489*	2562*	2687*	2784*	2824*	2867*	2937*	2985*	3041*	3152*	3263*	3410*	
	3451*	3514*	3592*	3750*										
TST1 007160	1845	1862	1935#											
TST10 011432	2241	2364#												
TST11 012240	2365	2489#												
TST12 012554	2490	2562#												
TST13 013344	2563	2687#												
TST14 014004	2688	2784#												
TST15 014156	2785	2824#												
TST16 014352	2825	2867#												
TST17 014720	2868	2937#												
TST2 007264	1936	1967#												
TST20 015142	2938	2985#												
TST21 015406	2986	3041#												
TST22 016126	3042	3152#												
TST23 016610	3153	3263#												
TST24 017432	3264	3410#												
TST25 017630	3411	3451#												
TST26 020110	3452	3514#												
TST27 020450	3515	3592#												
TST3 007400	1968	2001#												
TST30 021454	3593	3750#	3817											
TST31 = ***** U	3751													
TST4 007514	2002	2035#												
TST5 007632	2036	2069#												
TST6 010172	2070	2143#												
TST7 010632	2144	2240#												
TTST 003174	1307*	1308*	1310*	1311*	1367#									
TWOSYN= 000000	799#													
TXACT = 001000	937#	2804	2850											
TXCSR 001412	1029#	1686*	1714*	1717*	1811*	1812*	1813	1938*	1944*	1949*	1972*	1978*	1983*	
	2006#	2012*	2017*	2040*	2046*	2052*	2074*	2076*	2084*	2093	2105	2111	2124*	
	2145*	2150*	2158	2160*	2170	2174*	2223*	2242*	2258*	2265	2267*	2277	2281*	
	2328*	2349*	2366*	2384*	2391	2393*	2403	2407*	2454*	2475*	2491*	2502*	2504	
	2507	2544*	2564*	2574*	2576	2579*	2596	2650*	2655	2665	2673*	2689*	2713*	
	2714	2717*	2730*	2768*	2786*	2789*	2790*	2800	2808*	2826*	2829*	2830*	2832	
	2835	2838*	2847	2850	2869*	2872*	2874*	2876	2892*	2896*	2897*	2899	2902	
	2921*	2939*	2942*	2944*	2950	2953	2987*	2990*	2992*	2994	2997	3011	3049*	
	3052*	3054*	3056	3059	3062	3067	3070	3073	3076	3079*	3134*	3162*	3165*	
	3167*	3173	3178*	3190*	3241	3244*	3284*	3287*	3289*	3295	3300*	3312*	3388	
	3391*	3412*	3414*	3417*	3429*	3462*	3466*	3467	3523*	3526*	3528*	3530	3533	
	3538*	3550*	3563*	3605*	3608*	3610*	3612	3615	3619*	3632*	3660*	3752*	3763*	
	3765*	3792*												
TXDBUF 001414	1030#	1815*	1816*	1817	2086	2088	2096*	2098	2100	2110*	2113*	2115	2117	
	2161*	2163	2165	2173*	2178	2180	2211*	2217*	2220*	2268*	2270	2272	2280*	
	2285	2287	2316*	2322*	2325*	2394*	2396	2398	2406*	2411	2413	2442*	2448*	
	2451*	2506*	2509*	2513*	2515	2517	2578*	2581	2583	2598*	2600	2602	2658*	
	2668*	2672*	2716*	2721	2723	2759*	2765*	2791*	2793	2795	2831*	2834*	2837*	
	2840	2842	2875*	2878*	2880	2882	2898*	2901*	2904*	2906	2908	2945*	2946	
	2948	2952*	2955*	2957	2959	2993*	2996*	2999*	3004	3006	3014*	3055*	3058*	
	3061*	3064*	3069*	3072*	3075*	3078*	3081	3083	3168*	3169	3171	3175*	3181	

CROSS REFERENCE TABLE -- USER SYMBOLS

. INSTR	003412	1000	1420#		
. INST1	003432	1424#	1444		
. MSG	003434	1422*	1425#		
. PARAM	003536	1004	1452#		
. PFAIL	005050	815	1120	1729#	1733
. PKCLK	005006	1014	1711#		
. RESOS	003776	1008	1523#		
. SAVDS	003736	1006	1509#		
. SCOPE	003160	994	1364#		
. SCOP1	003312	996	1394#		
. SETFL	004242	1016	1592#	1604	
. START	001562	833	1118#	1130	
. TRPSR	004316	819	1612#		
. TRPTA	001344	992#	1617		
. TYPE	003336	998	1404#		

\$SMALL	18	1722													
\$SYNC	7178	2899	2950	2953	2994	2997	3056	3059	3062	3173	3295	3530	3533	3612	3615
\$TRPDE	18	993	995	997	999	1001	1003	1005	1007	1009	1011	1013	1015		
\$TSTN	18	1929	1961	1995	2029	2063	2137	2234	2358	2483	2556	2681	2778	2818	2861
	2931	2979	3035	3146	3257	3404	3445	3508	3586	3744					
\$VARIA	18	835													
\$WAIT	18	2085	2097	2114	2162	2176	2269	2283	2395	2409	2514	2580	2599	2719	2792
	2839	2879	2905	2946	2956	3003	3080	3169	3180	3291	3302	3472	3539	3622	
\$XZ	18	1923	1927	1955	1959	1989	1993	2023	2027	2057	2061	2132	2135	2229	2232
	2353	2356	2479	2482	2549	2554	2677	2680	2773	2776	2813	2816	2856	2859	2926
	2929	2974	2977	3028	3034	3141	3145	3251	3256	3399	3402	3441	3444	3504	3507
	3580	3584	3734	3743											

. ABS. 023540 000

ERRORS DETECTED: 0
 DEFAULT GLOBALS GENERATED: 0

DSKZ:DZDPDB, DSKZ:DZDPDB. SEQ/SOL/CRF/DOC/NL: TOC=DZDPDB.MAC, DZDPDB.P11
 RUN-TIME: 9 14 1 SECONDS
 RUN-TIME RATIO: 33/25=1.3
 CORE USED: 26K (51 PAGES)

DOCUMENT PAGES: 93