

# DUP11

TRANSMITTER TEST  
MD-11-DZDPB-A

EP-DZDPB-A-DL-A  
COPYRIGHT © 1976  
FICHE 1 OF 1

NOV 1976  
**digital**  
MADE IN USA

The microfiche contains 110 frames of data, organized into 10 rows and 11 columns. Each frame displays a small-scale version of a transmitter test page, likely containing technical specifications, test results, or diagrams. The text within the frames is too small to be legible, but the overall layout suggests a comprehensive set of test data for the MD-11-DZDPB-A transmitter.

NOV 1976  
MADE IN USA



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

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.

DZDPB DOES READ/WRITE CHECKING, BUS ADDRESS CHECKING, DEVICE AND BUS RESET TESTS ON THE CONTROL AND STATUS REGISTERS. TESTS ARE MADE TO PROVE THERE IS NO INTERACTION WITHIN AND BETWEEN REGISTERS. THE REGISTERS ARE CHECKED BOTH A BIT AT A TIME AND ALL AT ONCE.

IN ADDITION, THE TRANSMITTER SDLC FUNCTIONS ARE CHECKED IN MAINTENANCE INTERNAL MODE. I.E. CLOCKING OF THE DEVICE IS DONE BY THE PROGRAM. THE DEVICE IS SET UP, A SPECIFIC NUMBER OF HALF-CLOCKS ARE DONE, AND A TEST IS MADE FOR A SIGNIFICANT EVENT.

IN CHECKING DATA, THE SOFTWARE EMULATES THE HARDWARE AND USES THE PROCESSOR CARRY BIT AFTER A ROTATE TO DETERMINE WHAT THE OUTPUT SHOULD BE AT THE TRANSMITTER BIT WINDOW A BIT AT A TIME. THE PROGRAM THEN COMPARES THE EMULATED SOFTWARE BIT TO THE HARDWARE BIT OUTPUT. THE PROCESS IS REPEATED UNTIL ALL DATA IS CHECKED.

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

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

98  
99  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117

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 SDLC AND DECMODE DATA AND FUNCTION TESTS

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

118  
119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166

2.0 REQUIREMENTS

2.1 EQUIPMENT

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

2.2 STORAGE

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

3.0 LOADING PROCEDURE

3.1 METHOD

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

ABSOLUTE LOADER STARTING ADDRESS = \*+500

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

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

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

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

167  
 168  
 169  
 170  
 171  
 172  
 173  
 174  
 175  
 176  
 177  
 178  
 179  
 180  
 181  
 182  
 183  
 184  
 185  
 186  
 187  
 188  
 189  
 190  
 191  
 192  
 193  
 194  
 195  
 196  
 197  
 198  
 199  
 200  
 201  
 202  
 203  
 204  
 205  
 206  
 207  
 208  
 209  
 210  
 211  
 212  
 213  
 214  
 215  
 216  
 217  
 218  
 219  
 220  
 221  
 222

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

DZOPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 7  
DZOPBA.CMB

223  
224

SW 06 SET: RESERVED  
SW 05 SET: RESERVED

225  
226  
227  
228  
229  
230  
231  
232  
233  
234  
235  
236  
237  
238  
239  
240  
241  
242  
243  
244  
245  
246  
247  
248  
249  
250  
251  
252  
253  
254  
255  
256  
257  
258  
259  
260  
261  
262  
263  
264  
265  
266  
267  
268  
269  
270  
271  
272  
273  
274  
275  
276  
277  
278  
279  
280

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

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

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 03 RESELECT DUP11'S DESIRED ACTIVE. PLEASE NOTE THAT A MESSAGE IS TYPED OUT FOR SETTING THE SWITCH REGISTER EQUAL TO DUP11'S ACTIVE. THIS MEANS IF THE SYSTEM HAS THREE 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

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 9  
DZDPBA.CMB

281  
282

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

283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303  
304  
305  
306  
307  
308  
309  
310  
311  
312  
313  
314  
315  
316  
317  
318  
319  
320  
321  
322  
323  
324  
325  
326  
327  
328  
329

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

4.1.3 SWITCH REGISTER PRIORITIES

A) ERROR SWITCHES

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

B) SCOPE SWITCHES

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

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

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

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

4.2 STARTING ADDRESS

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

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

330  
331  
332  
333  
334  
335  
336  
337  
338  
339  
340  
341  
342  
343  
344  
345  
346  
347  
348  
349  
350  
351  
352  
353  
354  
355  
356  
357  
358  
359  
360  
361  
362  
363  
364  
365  
366  
367  
368  
369  
370  
371  
372  
373  
374  
375  
376  
377  
378  
379  
380  
381  
382

## 5.0 OPERATING PROCEDURE

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

## 5.1 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

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

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

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

## 6.0 ERRORS

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

## 6.1 ERROR RECOVERY

IF FOR SOME REASON THE 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.

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

7.0 RESTRICTIONS

7.1 STARTING RESTRICTIONS

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

7.2 OPERATING RESTRICTIONS

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.

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

## 8.0 MISCELLANEOUS

## 8.1 EXECUTION TIME

ALL 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 DZDPBA 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.

DUPCR00-DUPCR07  
(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

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 14  
DZDPBA.CMB

479  
480

THAT DUP11 NO. 00,01,02,03,04 WILL BE TESTED.  
EXAMPLE: (DUPACTV) /00000000000010001 MEANS



C02

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 16  
DZDPBA.CMB

537  
538

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



539  
540  
541  
542  
543  
544  
545  
546  
547  
548  
549  
550  
551  
552  
553  
554  
555  
556  
557  
558  
559  
560  
561  
562  
563  
564  
565  
566  
567

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.

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 18  
 DZDPBA.CMB INTRODUCTION TO DUP11 DIAGNOSTIC

568  
569  
570  
571  
572  
573  
574  
575  
576  
577  
578  
579  
580  
581  
582  
583  
584  
585  
586  
587  
588  
589  
590  
591  
592  
593  
594  
595  
596  
597  
598  
599  
600  
601

```

: *MAINDEC-11-DZDPB-A /<377>/BASIC DUP11 AND OFFLINE SDLC TRANSMITTER TESTS
: *COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
: *-----

```

```

: STARTING PROCEDURE
: LOAD PROGRAM
: LOAD ADDRESS 000200
: PRESS START
: PROGRAM WILL TYPE "MAINDEC-11-DZDPB-A /<377>/BASIC DUP11 AND OFFLINE SDLC TRANS
: 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

```

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 19  
 DZDPBA.CMB GENERAL DEFINITIONS AND EQUIVALENCIES

```

602
603
604           ;REGISTER DEFINITIONS
605           ;-----
606
607           000000      R0=%0           ;GENERAL REGISTER
608           000001      R1=%1           ;GENERAL REGISTER
609           000002      R2=%2           ;GENERAL REGISTER
610           000003      R3=%3           ;GENERAL REGISTER
611           000004      R4=%4           ;GENERAL REGISTER
612           000005      R5=%5           ;GENERAL REGISTER
613           000006      SP=%6          ;PROCESSOR STACK POINTER
614           000007      PC=%7          ;PROGRAM COUNTER
615
616           ;LOCATION EQUIVALENCIES
617           ;-----
618
619           177776      PS=177776       ;PROCESSOR STATUS WORD
620           001150      STACK=1150      ;START OF PROCESSOR STACK
621
622           ;INSTRUCTION DEFINITIONS
623           ;-----
624
625           005746      PUSH1SP=5746    ;DECREMENT PROCESSOR STACK 1 WORD
626           005726      POP1SP=5726    ;INCREMENT PROCESSOR STACK 1 WORD
627           010046      PUSHRO=10046    ;SAVE R0 ON STACK
628           012600      POPRO=12600     ;RESTORE R0 FROM STACK
629           024646      PUSH2SP=24646   ;DECREMENT STACK TWICE
630           022626      POP2SP=22626   ;INCREMENT STACK TWICE
631           .EQUIV EMT,HLT ;BASIC DEFINITION OF ERROR CALL
632
633
634           100000      BIT15=100000
635           040000      BIT14=40000
636           020000      BIT13=20000
637           010000      BIT12=10000
638           004000      BIT11=4000
639           002000      BIT10=2000
640           001000      BIT9=1000
641           000400      BIT8=400
642           000200      BIT7=200
643           000100      BIT6=100
644           000040      BIT5=40
645           000020      BIT4=20
646           000010      BIT3=10
647           000004      BIT2=4
648           000002      BIT1=2
649           000001      BIT0=1
650
651

```

```

652      ;*****
653      ;-----
654      ; TRAPCATCHER FOR ILLEGAL INTERRUPTS
655      ; THE STANDARD "TRAP CATCHER" IS PLACED
656      ; BETWEEN ADDRESS 0 TO ADDRESS 776.
657      ; IT LOOKS LIKE "PC+2 HALT".
658      ;-----
659      ;*****
660
661      000000      .=0
662      ;-----
663      ; STANDARD INTERRUPT VECTORS
664      ;-----
665      000024      .=24
666      000024      004776      .PFAIL      ; POWER FAIL HANDLER
667      000026      000340      340          ; SERVICE AT LEVEL 7
668      000030      004350      .HLT          ; ERROR HANDLER
669      000032      000340      340          ; SERVICE AT LEVEL 7
670      000034      004316      .TRPSRV     ; GENERAL HANDLER DISPATCH SERVICE
671      000036      000340      340          ; SERVICE AT LEVEL 7
672
673      000040      000000      .=40
674      000042      000000      0          ; SAVE FOR ACT-11 OR DDP2
675      000044      000000      0          ; RETURN ADDRESS IF UNDER ACT-11 OR DDP2
676      000046      003104      0          ; SAVE FOR ACT-11 OR DDP2
677      000052      000052      SENDAD     ; FOR USE WITH ACT-11 OR DDP2
678      000052      000000      0          ; ACT-11 PROGRAM CHARACTERISTICS
679
680      000174      000174      .=174
681      000174      000000      DISPREG:0  ; SOFTWARE DISPLAY REGISTER
682      000176      000000      SWREG: 0   ; SOFTWARE SWITCH REGISTER
683      000200      000200      .=200
684      000200      000137      001562      JMP      .START ; GO TO START OF PROGRAM
685
686
687      001000      001000      040515      047111      .=1000
688      001000      005377      MTITLE: .ASCIZ <377><12>/MAINDEC-11-DZDPB-A /<377>/BASIC DUP11 AND OFFLINE SOLC TRANSMI
689      001200      001200      .=1200
690      ;-----
691      ; SWR AND LIGHTS
692      ;-----
693      001200      177570      DISPLAY: 177570      ; 11/45 CONSOLE LIGHTS
694      001202      177570      SWR: 177570        ; INDIRECT POINTER TO SWITCH REGISTER
695
696      ;-----
697      ; INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
698      ;-----
699      001204      177560      TKCSR: 177560     ; TELETYPE KEYBOARD CONTROL REGISTER
700      001206      177562      TKDBR: 177562     ; TELETYPE KEYBOARD DATA BUFFER
701      001210      177564      TPCSR: 177564     ; TELEPRINTER CONTROL REGISTER
702      001212      177566      TPDBR: 177566     ; TELEPRINTER DATA BUFFER
703
704      ;-----
705      ; PROGRAM CONTROL PARAMETERS
706      ;-----
  
```

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 21  
 DZDPBA.CMB PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

707	001214	000000	RETURN:	0	; SCOPE ADDRESS FOR LOOP ON TEST
708	001216	000000	NEXT:	0	; ADDRESS OF NEXT TEST TO BE EXECUTED
709	001220	000000	LOCK:	0	; ADDRESS FOR LOCK ON CURRENT DATA
710	001222	000001	ICOUNT:	1	; NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
711	001224	000000	LPCNT:	0	; NUMBER OF ITERATIONS COMPLETED
712	001226	000000	TSTNO:	0	; NUMBER OF TEST IN PROGRESS
713	001230	000000	PASCNT:	0	; NUMBER OF PASSES COMPLETED
714	001232	000000	ERRCNT:	0	; TOTAL NUMBER OF ERRORS
715	001234	000000	LSTERR:	0	; PC OF LAST ERROR CALL
716					
717			; PROGRAM VARIABLES		
718			;-----		
719					
720	001236	000000	TEMP1:	0	; TEMPORARY STORAGE
721	001240	000000	TEMP2:	0	; TEMPORARY STORAGE
722	001242	000000	TEMP3:	0	; TEMPORARY STORAGE
723	001244	000000	TEMP4:	0	; TEMPORARY STORAGE
724	001246	000000	TEMP5:	0	; TEMPORARY STORAGE
725	001250	000000	SAVR0:	0	; R0 STORAGE
726	001252	000000	SAVR1:	0	; R1 STORAGE
727	001254	000000	SAVR2:	0	; R2 STORAGE
728	001256	000000	SAVR3:	0	; R3 STORAGE
729	001260	000000	SAVR4:	0	; R4 STORAGE
730	001262	000000	SAVR5:	0	; R5 STORAGE
731	001264	000000	SAVSP:	0	; STACK POINTER STORAGE
732	001266	000000	SAVPC:	0	; PROGRAM COUNTER STORAGE
733					
734	001270	000000	SAVR0A:	0	; R0 STORAGE
735	001272	000000	SAVR1A:	0	; R1 STORAGE
736	001274	000000	SAVR2A:	0	; R2 STORAGE
737	001276	000000	SAVR3A:	0	; R3 STORAGE
738	001300	000000	SAVR4A:	0	; R4 STORAGE
739	001302	000000	SAVR5A:	0	; R5 STORAGE
740	001304	000000	SAVSPA:	0	; STACK POINTER STORAGE
741	001306	000000	SAVPCA:	0	; PROGRAM COUNTER STORAGE
742					
743	001310	000001	DUPACTV:	.BLKB 1	; DUP11'S SELECTED ACTIVE.
744	001311	000001	DUPNUM:	.BLKB 1	; OCTAL NUMBER OF DUP11'S.
745	001312	000001	SAVACT:	.BLKB 1	; ORIGINAL ACTV. DEVICES.
746	001313	000001	SAVNUM:	.BLKB 1	; WORKABLE NUMBER.
747	001314	000001	RUN:	.BLKB 1	; POINTER ONE PAST RUNNING DEVICE.
748		001316	.EVEN		
749	001316	001500	CREAM:	DUP.MAP	; TABLE POINTER.

```

750
751                                     ;CONTROL REGISTER DEFINITIONS
752                                     ;-----
753                                     ;RXCSR BIT DEFINITIONS
754      100000      DSCA=BIT15      ;DATA SET CHANGE A
755      040000      RING=BIT14      ;RING
756      020000      CTS=BIT13      ;CLR TO SEND
757      010000      CARDET=BIT12    ;CARRIER DETECT
758      004000      RECACT=BIT11    ;REC ACTIVE
759      002000      SRD=BIT10      ;SEC REC DATA
760      001000      DSR=BIT9       ;DATA SET RDY
761      000400      STPSYN=BIT8     ;STRIP SYNC
762      000200      RXDONE=BIT7     ;REC DONE
763      000100      RINTEN=BIT6    ;REC INTR ENABLE
764      000040      DSINTE=BIT5    ;DSC INTR ENABLE
765      000020      RCVEN=BIT4     ;REC ENABLE
766      000010      STD=BIT3       ;SEC XMIT DATA
767      000001      RTS=BIT2       ;REQ TO SEND
768      000002      DTR=BIT1       ;DATA TERM RDY
769      000001      DSCB=BIT0      ;DATA SET CHANGE B
770                                     ;RXDBUF BIT DEFINITIONS
771      100000      RXDERR=BIT15    ;REC DATA ERROR
772      040000      OVRUN=BIT14    ;OVERRUN ERROR
773      010000      CRCERR=BIT12   ;CRC ERROR
774      002000      RABORT=BIT10   ;REC ABORT
775      001000      REOM=BIT9      ;REC END OF MESSAGE
776      000400      RSOM=BIT8      ;REC START OF MESSAGE
777                                     ;PARCSR BIT DEFINITIONS
778      100000      DECMOD=BIT15    ;DEC MODE (DDCMP)
779      001000      CRCEN=BIT9      ;CRC ENABLE
780      010000      PRISEC=BIT12   ;PRI/SEC SELECT
781                                     ;TXCSR BIT DEFINITIONS
782      100000      TXDLAT=BIT15    ;TX DATA LATE
783      040000      MTDATA=BIT14    ;MAINT DATA OUT
784      020000      CLK=BIT13      ;CLK
785      010000      MMODEB=BIT12   ;MAINT MODE B
786      004000      MMODEA=BIT11   ;MAINT MODE A
787      002000      BITW=BIT10     ;BIT WINDOW INPUT
788      001000      TXACT=BIT9     ;TX ACTIVE
789      000400      MRESET=BIT8    ;MASTER RESET
790      000200      TXDONE=BIT7    ;XMIT DONE
791      000100      TXINTE=BIT6    ;XMIT DONE INTR ENABLE
792      000020      SEND=BIT4     ;SEND
793      000010      HDXEN=BIT3     ;HDX/FDX
794                                     ;TXCSR WRD DEFINITIONS
795      000000      USER=0          ;USER MODE
796      014000      MMODE=14000    ;MAINT INT MODE
797      010000      MEXT=10000     ;MAINT EXT MODE
798      004000      SYSTST=4000    ;SYSTEM TEST MODE
799
800                                     ;TXDBUF BIT DEFINITIONS
801                                     ;-----
802      100000      RCRC7T=BIT15
803      040000      RCRCIN=BIT14
804      020000      TCRC7T=BIT13
805      010000      TCRCIN=BIT12
  
```

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 23  
DZDPBA.CMB PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

806	004000	TIMER=BIT11	; MAINTENANCE TIMER
807	002000	TABORT=BIT10	; TRANSMIT ABORT
808	001000	TEOM=BIT9	; TRANSMIT END OF MESSAGE
809	000400	TSOM=BIT8	; TRANSMIT START OF MESSAGE
810			
811		; MISC. PROGRAM DEFINITIONS	
812		-----	
813	001320	PRIRTY: .WORD	0
814	001322	TCNFLG: .BLKB	1
815	001323	OPCLRJ: .BLKB	1
816	001324	DATA: .WORD	0
817	001326	SHIFTS: .WORD	0
818	001330	MIND: .WORD	0
819	001332	FLAG: .WORD	0
820	001334	STJMFL: .BLKW	1
821	001336	SRJMFL: .BLKW	1
822			
823			

Handwritten marks and scribbles at the bottom right of the page.

824  
825  
826  
827  
828  
829  
830  
831  
832  
833  
834  
835  
836  
837  
838  
839  
840  
841  
842  
843  
844  
845  
846  
847  
848  
849  
850  
851  
852  
853  
854  
855  
856  
857  
858  
859  
860  
861  
862  
863  
864  
865  
866  
867  
868  
869  
870

001340 000  
001341 000  
001342 000  
001343 000  
  
000000  
  
  
  
001344 104400  
001344 003160  
104401  
001346 003312  
104402  
001350 003336  
104403  
001352 003412  
104404  
001354 003516  
104405  
001356 003536  
104406  
001360 003736  
104407  
001362 003776  
104410  
001364 004030  
104411  
001366 004034  
104412  
001370 004734  
104413  
001372 004242

PROGRAM CONTROL FLAGS  
-----

INIFLG: .BYTE 0 ;PROGRAM INITIALIZATION FLAG  
ERRFLG: .BYTE 0 ;ERROR OCCURED FLAG  
LOKFLG: .BYTE 0 ;LOCK ON CURRENT TEST FLAG  
QV.FLG: .BYTE 0 ;QUICK VERIFY FLAG.  
;ON FIRST PASS OF EACH DUP11 ITERATIONS  
;WILL BE SUPPRESSED

.EVEN  
\$Y=0

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

\*\*\*\*\*

-----  
TRPTAB:  
SCOPE=TRAP+0 ;CALL TO SCOPE LOOP AND ITERATION HANDLER  
;SCOPE  
SCOPI=TRAP+1 ;CALL TO LOOP ON CURRENT DATA HANDLER  
;SCOPI  
TYPE=TRAP+2 ;CALL TO TELETYPE OUTPUT ROUTINE  
;TYPE  
INSTR=TRAP+3 ;CALL TO ASCII STRING INPUT ROUTINE  
;INSTR  
INSTER=TRAP+4 ;CALL TO INPUT ERROR HANDLER  
;INSTER  
PARAM=TRAP+5 ;CALL TO NUMERICAL DATA 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

-----  
\*\*\*\*\*



DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 25  
 DZDPBA.CMB PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

871                                     ;DUP11 VECTOR AND REGISTER INDIRECT POINTERS
872
873 001374 000000      DUPRVC: 0          ; POINTER TO DUP11 RECEIVER INTERRUPT VECTOR
874 001376 000000      DUPRPS: 0          ; POINTER TO DUP11 RECEIVER INTERRUPT SERVICE PS
875 001400 000000      DUPTVC: 0          ; POINTER TO DUP11 TRANSMITTER INTERRUPT VECTOR
876 001402 000000      DUPTPS: 0          ; POINTER TO DUP11 TRANSMITTER INTERRUPT SERVICE PS
877 001404 000000      RXCSR: 0           ; POINTER TO DUP11 RECEIVER STATUS REGISTER
878 001406 000000      RXDBUF: 0          ; POINTER TO DUP11 RECEIVER DATA BUFFER
879 001410 000000      PARCSR: 0          ; POINTER TO DUP11 PARAMETER STATUS REGISTER
880 001412 000000      TXCSR: 0           ; POINTER TO DUP11 TRANSMITTER STATUS REGISTER
881 001414 000000      TXDBUF: 0          ; POINTER TO DUP11 TRANSMITTER DATA BUFFER
882 001416 000000      DUPSEC: 0          ; POINTER TO DUP11 SECONDARY REGISTER SELECT REGISTER
883 001420 000000      HUPPSR: 0          ; POINTER TO PARAMETER STATUS HIGH BYTE
884 001422 000000      HUPRBF: 0          ; POINTER TO RECEIVER BUFFER HIGH BYTE
885 001424 000000      HUPCR: 0           ; POINTER TO RECEIVER CONTROL REG HIGH BYTE
886 001426 000000      HUPTBF: 0          ; POINTER TO TRANSMITTER BUFFER HIGH BYTE
887 001430 000000      HUPTCR: 0          ; POINTER TO TRANSMITTER CONTROL REG HIGH BYTE
888
889
890                                     ;DUP11 CONTROL INDICATORS FOR CURRENT DUP11 UNDER TEST
891 -----
892
893 001432 000          MASK.A: .BYTE 000    ; LAST CHAR TO TEST AND PARITY MASK
894
895 001433 010          CLK.A: .BYTE 8.      ; NUMBER OF CLOCKS NEEDED FOR ONE CHAR
896
897 001434 000000      LOO.OO: 000000      ; PARAMETERS
898

```

:DUP11 STATUS TABLE AND ADDRESS ASSIGNMENTS

Address	Hex Value	Hex Value	Label	Description
899				
900				
901				
902	001500	001500	.=1500	
903	001500	000001	DUP.MAP:	
904	001500	000001	DUPCR0: .BLKW 1	:CONTROL STATUS REGISTER FOR DUP11 NUMBER 0
905	001502	000001	DUPTR0: .BLKW 1	:VECTOR "A" FOR DUP11 NUMBER 0
906	001504	000001	DUP0.A: .BLKW 1	:PARAMETER FOR DUP11 NUMBER 0
907				
908	001506	000001	DUPCR1: .BLKW 1	:CONTROL STATUS REGISTER FOR DUP11 NUMBER 1
909	001510	000001	DUPTR1: .BLKW 1	:VECTOR "A" FOR DUP11 NUMBER 1
910	001512	000001	DUP1.A: .BLKW 1	:PARAMETER FOR DUP11 NUMBER 1
911				
912	001514	000001	DUPCR2: .BLKW 1	:CONTROL STATUS REGISTER FOR DUP11 NUMBER 2
913	001516	000001	DUPTR2: .BLKW 1	:VECTOR "A" FOR DUP11 NUMBER 2
914	001520	000001	DUP2.A: .BLKW 1	:PARAMETER FOR DUP11 NUMBER 2
915				
916	001522	000001	DUPCR3: .BLKW 1	:CONTROL STATUS REGISTER FOR DUP11 NUMBER 3
917	001524	000001	DUPTR3: .BLKW 1	:VECTOR "A" FOR DUP11 NUMBER 3
918	001526	000001	DUP3.A: .BLKW 1	:PARAMETER FOR DUP11 NUMBER 3
919				
920	001530	000001	DUPCR4: .BLKW 1	:CONTROL STATUS REGISTER FOR DUP11 NUMBER 4
921	001532	000001	DUPTR4: .BLKW 1	:VECTOR "A" FOR DUP11 NUMBER 4
922	001534	000001	DUP4.A: .BLKW 1	:PARAMETER FOR DUP11 NUMBER 4
923				
924	001536	000001	DUPCR5: .BLKW 1	:CONTROL STATUS REGISTER FOR DUP11 NUMBER 5
925	001540	000001	DUPTR5: .BLKW 1	:VECTOR "A" FOR DUP11 NUMBER 5
926	001542	000001	DUP5.A: .BLKW 1	:PARAMETER FOR DUP11 NUMBER 5
927				
928	001544	000001	DUPCR6: .BLKW 1	:CONTROL STATUS REGISTER FOR DUP11 NUMBER 6
929	001546	000001	DUPTR6: .BLKW 1	:VECTOR "A" FOR DUP11 NUMBER 6
930	001550	000001	DUP6.A: .BLKW 1	:PARAMETER FOR DUP11 NUMBER 6
931				
932	001552	000001	DUPCR7: .BLKW 1	:CONTROL STATUS REGISTER FOR DUP11 NUMBER 7
933	001554	000001	DUPTR7: .BLKW 1	:VECTOR "A" FOR DUP11 NUMBER 7
934	001556	000001	DUP7.A: .BLKW 1	:PARAMETER FOR DUP11 NUMBER 7
935				
936	001560	000000	DUP.END:	000000
937				
938				
939				
940				
941				

```

15 14 13 12 11 10 09 08 07 06 05 04 03 02 01 00
I I I I I I I I I I I I I I I I I I
I C O N T R O L I R E G I S T E R I
I I I I I I I I I I I I I I I I I
-----
I I I I I I I I I I * I V I E I C I T I O I R I * I
I I I I I I I I I I I I I I I I I I I I I
-----
I I I I I I I I I I * I * I S Y N C 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-

```

961
962
963
964
965
965
967
968
969 001562 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
970 001570 012706 001150 MOV #STACK,SP ;SET UP STACK
971 001574 012737 004776 000024 MOV #.PFAIL,2#24 ;SET UP POWER FAIL VECTOR
972 001602 113737 001311 001313 MOVB DUPNUM,SAVNUM ;SAVE NUMBER OF DEVICES IN SYSTEM
973 001610 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
974 001514 105037 001341 CLRB ERRFLG ;CLEAR ERROR FLAG
975 001620 105037 001343 CLRB QV.FLG ;ZERO QUICK VERIFY FLAG
976 001624 012737 001500 001316 MOV #DUP.MAP,CREAM ;GET MAP POINTER.
977 001632 112737 000001 001314 MOVB #1,RUN ;POINT POINTER TO FIRST DEVICE.
978 001640 005037 001232 CLR ERACNT ;CLEAR ERROR COUNT
979 001644 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
980 001650 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
981 001656 012737 001562 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
982 ;TESTING STARTS
983 001664 013746 000006 MOV 2#6,-(SP) ;SAVE CURRENT VECTORS
984 001670 013746 000004 MOV 2#4,-(SP)
985 001674 012737 001710 000004 MOV #12$,2#4 ;SETUP FOR TIMEOUT
986 001702 005777 177274 TST 2$SWR ;REFERENCE HARDWARE SWITCH REG
987 001706 000407 BR 13$ ;BR IF IT EXISTS
988 001710 012737 000176 001202 12$: MOV #SWREG,SWR ;POINT TO SOFT SWR
989 001716 012737 000174 001200 MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
990 001724 022626 CMP (SP)+,(SP)+ ;ADJUST STACK
991 001726 012637 000004 13$: MOV (SP)+,2#4 ;RESTORE VECTORS
992 001732 012637 000006 MOV (SP)+,2#6
993 001736 105737 001340 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
994 001742 001401 BEQ 11$
995 001744 000410 BR 6$
996 001746 022737 003104 000042 11$: CMP #SENDAD,2#42 ;IF ACT-11 AUTO MODE,
997 001754 001404 BEQ 6$ ;DON'T TYPE ID
998 001756 104402 001000 TYPE #MTITLE ;TYPE TITLE MESSAGE
999 001762 105137 001340 COMB INIFLG ;IF NOT SET FLAG AND DO
1000 001766 105777 177210 6$: TSTB 2$SWR ;BIT7=1??
1001 001772 100002 BPL 10$
1002 001774 000137 002520 JMP 1$
1003 002000 10$:
1004 002000 032777 000001 177174 BIT #SW00,2$SWR ;ENTER PARAMETERS
1005 002006 001002 BNE +6 ;YES
1006 002010 000137 002360 JMP 21$ ;NO
1007 002014 105137 001332 COMB FLAG
1008 002020 112737 000001 001340 MOVB #1,INIFLG ;SET TO MANUAL ENTRY
1009 002026 012700 001500 MOV #DUP.MAP,RO ;CLR MAP
1010 002032 005020 68$: CLR (RO)+
1011 002034 020027 001560 CMP RO,#DUP.END ;DONE WITH MAP?
1012 002040 001374 BNE 68$ ;BR IF NO
1013 002042 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
1014 002044 005422 MCSR ;MESSAGE
1015 002046 104405 PARAM ;CONVERT STRING
1016 002050 160000 160000 ;LOW LIMIT

```

1017	002052	175500				175500	:HIGH LIMIT
1018	002054	001500				DUPCR0	:STORE AT THIS LOCATION
1019	002056	001			.BYTE	1	:MASK
1020	002057	001			.BYTE	1	:HOW MANY TIMES + 2
1021	002060	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1022	002062	005441				MVEC	:MESSAGE
1023	002064	104405				PARAM	:CONVERT STRING
1024	002066	000300				300	:LOW LIMIT
1025	002070	000770				770	:HIGH LIMIT
1026	002072	001502				DUPTRO	:STORE AT THIS LOCATION
1027	002074	001			.BYTE	1	:MASK
1028	002075	001			.BYTE	1	:HOW MANY TIMES + 2
1029	002076	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1030	002100	005631				MVAR	:MESSAGE
1031	002102	104405				PARAM	:CONVERT STRING
1032	002104	000004				4	:LOW LIMIT
1033	002106	000007				7	:HIGH LIMIT
1034	002110	001240				TEMP2	:STORE AT THIS LOCATION
1035	002112	000			.BYTE	0	:MASK
1036	002113	001			.BYTE	1	:HOW MANY TIMES + 2
1037	002114	013737	001240	001320		MOV	TEMP2,PRIRTY :SAVE PRIORITY
1038	002122	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1039	002124	005576				MTOTAL	:MESSAGE
1040	002126	104405				PARAM	:CONVERT STRING
1041	002130	000001				1	:LOW LIMIT
1042	002132	000010				8.	:HIGH LIMIT
1043	002134	001236				TEMP1	:STORE AT THIS LOCATION
1044	002136	000			.BYTE	0	:MASK
1045	002137	001			.BYTE	1	:HOW MANY TIMES + 2
1046	002140	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1047	002142	005454				MJMFR	:MESSAGE
1048	002144	104413				SETFLG	:SET FLAG BASED UPON INPUT STRING
1049	002146	001323				OPCLRJ	:THIS FLAG
1050	002150	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1051	002152	005527				MTCN	:MESSAGE
1052	002154	104413				SETFLG	:SET FLAG BASED UPON INPUT STRING
1053	002156	001322				TCNFLG	:THIS FLAG
1054	002160	105737	001322			TSTB	TCNFLG
1055	002164	001410				BEQ	71S
1056	002166	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1057	002170	005655				MSTJM	:MESSAGE
1058	002172	104413				SETFLG	:SET FLAG BASED UPON INPUT STRING
1059	002174	001334				STJMFL	:THIS FLAG
1060	002176	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1061	002200	005710				MSRJM	:MESSAGE
1062	002202	104413				SETFLG	:SET FLAG BASED UPON INPUT STRING
1063	002204	001336				SRJMFL	:THIS FLAG
1064	002206	105737	001323		71S:	TSTB	OPCLRJ
1065	002212	001403				BEQ	69S
1066	002214	052737	100000	001504		BIS	#BIT15,DUPO.A
1067	002222	105737	001322		69S:	TSTB	TCNFLG
1068	002226	001403				BEQ	70S
1069	002230	052737	040000	001504		BIS	#BIT14,DUPO.A
1070	002236	112737	000001	001312	70S:	MOVB	#1,SAVACT
1071	002244	113737	001236	001311		MOVB	TEMP1,DUPNUM
1072	002252	113737	001236	001313		MOVB	TEMP1,SAVNUM

1073	002260	005337	001236	65\$:	DEC	TEMP1		
1074	002264	001404			BEQ	64\$		
1075	002268	000261			SEC			
1076	002270	106137	001312		ROLB	SAVACT		
1077	002274	000771			BR	65\$		
1078	002276	113737	001312	001240	64\$:	MOVB	SAVACT,TEMP2	;# OF TIMES
1079	002304	113737	001312	001310		MOVB	SAVACT,DUPACTV	
1080	002312	000241				CLC		
1081	002314	106037	001240			RORB	TEMP2	
1082	002320	012700	001500			MOV	#DUPCR0,R0	
1083	002324	012701	001506			MOV	#DUPCR1,R1	
1084	002330	000241			67\$:	CLC		
1085	002332	106037	001240			RORB	TEMP2	
1086	002336	103051				BCC	66\$	
1087	002340	012011				MOV	(R0)+(R1)	
1088	002342	062721	000010			ADD	#10,(R1)+	;CSR
1089	002346	012011				MOV	(R0)+(R1)	
1090	002350	062721	000010			ADD	#10,(R1)+	;VECTOR
1091	002354	012021				MOV	(R0)+(R1)+	;PARAMETERS
1092	002356	000764				BR	67\$	
1093	002360	012700	001500	21\$:		MOV	#DUP.MAP,R0	;SETUP TO CLEAR MAP
1094	002364	005020		20\$:		CLR	(R0)+	;CLEAR
1095	002366	020027	001560			CMP	R0,#DUP.END	;CHECK FOR FINISH
1096	002372	001374				BNE	20\$	;BR IF MORE TO GO
1097	002374	012700	001500			MOV	#DUP.MAP,R0	;SETUP TO DEFAULT
1098	002400	012710	160050			MOV	#160050,(R0)	;LOAD CSR
1099	002404	012760	000770	000002		MOV	#770,2(R0)	;LOAD VECTOR
1100	002412	012760	140026	000004		MOV	#140026,4(R0)	;LOAD PARAMETERS AND SYNC
1101	002420	112737	00000F	001320		MOVB	#5,PRTY	;LOAD PRIORITY
1102	002426	012700	000001			MOV	#1,R0	;SAVE CORE THIS WAY
1103	002432	110037	001310			MOVB	R0,DUPACTV	;PRESET PROGRAM CONTROLS
1104	002436	110037	001311			MOVB	R0,DUPNUM	;DITTO
1105	002442	110037	001312			MOVB	R0,SAVACT	;DITTO
1106	002446	110037	001313			MOVB	R0,SAVNUM	;DITTO
1107	002452	110037	001322			MOVB	R0,TCNFLAG	;DITTO
1108	002456	110037	001323			MOVB	R0,OPCLRJ	;DITTO
1109	002462			66\$:				
1110	002462	104402	005743	16\$:		TYPE	XHEAD	;TYPE HEADER
1111	002466	012737	001500	001236		MOV	#DUP.MAP,TEMP1	;SET POINTER
1112	002474	017737	176536	001240	5\$:	MOV	#TEMP1,TEMP2	;SET DATA
1113	002502	001406				BEQ	1\$	;ALL DONE WITH DATA
1114	002504	104410				CONVRT		
1115	002506	005772				XSTATQ		
1116	002510	062737	000002	001236		ADD	#2,TEMP1	;UPDATE POINTER
1117	002516	000766				BR	5\$	
1118	002520	032777	000001	176454	1\$:	BIT	#SW00,#SWR	
1119	002526	001405				BEQ	7\$	
1120	002530	005737	001332			TST	FLAG	
1121	002534	001002				BNE	7\$	
1122	002536	000137	002000			JMP	10\$	
1123	002542	005037	001332		7\$:	CLR	FLAG	
1124	002546	005737	000042			TST	#42	;IS PROGRAM RUNNING UNDER MONITOR
1125	002552	001030				BNE	3\$	;BR IF YES
1126	002554	032777	000010	176420		BIT	#SW03,#SWR	;SELECT SPECIFIC DEVICES??
1127	002562	001424				BEQ	3\$	;BR IF NO.
1128	002564	104402	005342			TYPE	,MNEW	;TYPE THE MESSAGE.

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 31  
 DZDPBA.CMB PROGRAM INITIALIZATION AND START UP.

```

1129 002570 005000          CLR      RO          ;ZERO DATA LIGHTS
1130 002572 000000          HALT
1131 002574 127737 176402 001312  CMPB    @SWR, SAVACT ;WAIT FOR USER TO TELL WHAT DEVICES TO RUN
1132 002602 101404          BLOS    2$          ;IS THE NUMBER VALID?
1133 002604 104402 005203    TYPE    ,MERR3      ;BR IF NUMBER IS OK.
1134 002610 000000          HALT          ;TELL USER OF INVALID NUMBER.
1135 002612 000776          BR      -2          ;STOP EVERY THING.
1136 002614 117737 176362 001310 2$:  MOVB   @SWR, DUPACTV ;RESTART THE PROGRAM AGAIN.
1137 002622 113700 001310    MOVB   DUPACTV, RO  ;GET NEW DEVICE PATTERN
1138 002626 042700 177400    BIC    @1C(377), RO ;SHOW THE USER WHAT HE SELECTED.
1139 002632 000000          HALT          ;USE ONLY LOW BYTE.
1140 002634 012700 000300 3$:  MOV    @300, RO     ;CONTINUE DYNAMIC SWITCHES.
1141 002640 012701 000302    MOV    @302, R1    ;PREPARE TO CLEAR THE FLOATING
1142 002644 010120          MOV    R1, (R0)+   ;VECTOR AREA. 300-776
1143 002646 005021          CLR    (R1)+       ;START PUTTING "PC+2 - HALT"
1144 002650 022021          CMP    (R0)+, (R1)+ ;IN VECTOR AREA.
1145 002652 022700 001000    CMP    @1000, RO   ;POP POINTERS
1146 002656 001372          BNE    4$          ;ALL DONE??
1147
1148
1149
1150
1151 002660 012737 000340 177776 .BEGIN: MOV    @340, PS      ;LOCK OUT INTERRUPTS
1152 002666 012706 001150    MOV    @STACK, SP ;SET UP STACK
1153 002672 005737 000042    TST    @#42       ;IS PROGRAM UNDER MONITOR CONTROL
1154 002676 001023          BNE    2$          ;BR IF YES
1155 002700 032777 000004 176274  BIT    @BIT2, @SWR ;CHECK FOR LOCK ON TEST
1156 002706 001411          BEQ    1$          ;BR IF NO LOCK DESIRED.
1157 002710 104402 005241    TYPE    ,MLOCK    ;TYPE LOCK SELECTED.
1158 002714 012737 000240 003174  MOV    @NOP, TTST  ;ADJUST SCOPE ROUTINE.
1159 002722 012737 000240 003176  MOV    @NOP, TTST+2 ;SET UP TO LOCK
1160 002730 000406          BR     2$          ;CONTINUE ALONG.
1161 002732 013737 003306 003174 1$:  MOV    BRW, TTST   ;PREPARE NORMAL SCOPE ROUTINE
1162 002740 013737 003310 003176  MOV    BRX, TTST+2 ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1163 002746 012737 006154 001214 2$:  MOV    @CYCLE, RETURN ;START AT "CYCLE" FIND WHICH DEVICE TO TEST
1164 002754 104402 005131    TYPE    ,MR
1165 002760 000177 176230    JMP    @RETURN    ;START TESTING

```

;TEST START AND RESTART

-----

```

1166                                     ;END OF PASS
1167                                     ;TYPE NAME OF TEST
1168                                     ;UPDATE PASS COUNT
1169                                     ;CHECK FOR EXIT TO ACT-11
1170                                     ;RESTART TEST
1171
1172 002764 005037 001234      .EOP: CLR      LSTERR      ;CLEAR LAST ERROR PC
1173 002770 105037 001341      CLR RB  ERRFLG      ;CLEAR ERROR FLAG
1174 002774 005237 001230      INC      PASCNT      ;UPDATE PASS COUNT
1175 003000 013777 001230      MOV      PASCNT, @DISPLAY ;DISPLAY PASS COUNT
1176 003006 104402 005105      TYPE    ,MEPASS     ;TYPE END PASS
1177 003012 104402 005270      TYPE    ,MCSRX      ;TYPE CSR
1178 003016 104411 003130      CNVRT   ,XCSR       ;SHOW IT
1179 003022 104402 005276      TYPE    ,MVECX      ;TYPE VECTOR
1180 003026 104411 003136      CNVRT   ,XVEC       ;SHOW IT
1181 003032 104402 005304      TYPE    ,MPASSX     ;TYPE PASSES
1182 003036 104411 003144      CNVRT   ,XPASS      ;SHOW IT
1183 003042 104402 005315      TYPE    ,MERRX      ;TYPE ERRORS
1184 003046 104411 003152      CNVRT   ,XERR       ;SHOW IT
1185 003052 105337 001313      DECB   SAVNUM      ;ARE ALL DEVICES TESTED?
1186 003056 001017 000000      BNE     RESTRT      ;BR IF NO.
1187 003060 112737 000377      MOV B  @377, @V.FLG ;SET THE QUICK VERIFY FLAG.
1188 003066 113737 001311      MOV B  DUPNUM, SAVNUM ;RESTORE THE COUNT
1189 003074 013701 000042      MOV    @42, R1     ;CHECK FOR ACT-11 OR DDP
1190 003100 001406 000000      BEQ    RESTRT      ;IF NOT, CONTINUE TESTING
1191 003102 000005 000000      RESET
1192 003104                                     SENDAD:
1193 003104 004711 000000      JSR    PC, (R1)
1194 003106 000240 000000      NOP
1195 003110 000240 000000      NOP
1196 003112 000240 000000      NOP
1197 003114 000240 000000      NOP
1198 003116 012737 006154      RESTRT: MOV @CYCLE, RETURN
1199 003124 000137 006154      JMP    CYCLE
1200 003130 000001 000000      XCSR:  1
1201 003132 006 002      .BYTE  6,2
1202 003134 001404 000000      RXCSR
1203 003136 000001 000000      XVEC:  1
1204 003140 003 002      .BYTE  3,2
1205 003142 001374 000000      DUPRVC
1206 003144 000001 000000      XPASS: 1
1207 003146 006 002      .BYTE  6,2
1208 003150 001230 000000      PASCNT
1209 003152 000001 000000      XERR:  1
1210 003154 006 002      .BYTE  6,2
1211 003156 001232 000000      ERRCNT
1212
1213                                     ;SCOPE LOOP AND INTERATION HANDLER
1214
1215 003160 005037 001234      .SCOPE: CLR      LSTERR      ;CLEAR LAST ERROR PC
1216 003164 010016 000000      MOV    RD, (SP)    ;SAVE RD ON STACK
1217 003166 032777 040000      BIT    @BIT14, @SWR ;! 00P ON TEST?
1218 003174 001407 000000      TTST:  BEQ    1$     ;BR IF NO (IF LOCK SW01 = 1; THIS LOCATION = 240)
1219 003176 000437 000000      BR     3$          ;GO TO 3$ (DITTO)
1220 003200 105777 176000      TSTB  @TKCSR      ;KYBD DONE?
1221 003204 100034 000000      BPL   3$          ;BR IF NO (LOCK: HIT A KEY ON TTY TO GO TO NEXT TEST)

```



```

1222 003206 017700 175774      MOV      @TKDBR,RO      ;CLR DONE BIT
1223 003212 000415              BR      2$             ;CONTINUE
1224 003214 032777 004000 175760 1$:  BIT      #SW11,@SWR    ;DELETE ITERATION (QUICK PASS)?
1225 003222 001011              BNE     2$             ;BR IF YES
1226 003224 105737 001343      TSTB   QV.FLG         ;HAS FIRST PASS BEEN COMPLETED?
1227 003230 001406              BEQ     2$             ;BR IF QUICK VERIFY
1228 003232 005237 001224      INC     LPCNT          ;UPDATE ITERATION COUNTER
1229 003236 023737 001224 001222  CMP     LPCNT,ICOUNT   ;ALL ITERATIONS DONE?
1230 003244 001014              BNE     3$             ;BR IF NOT YET
1231 003246 105037 001341      CLR    ERRFLG         ;PREPARE FOR NEW TEST
1232 003252 005037 001224      CLR    LPCNT          ;START ICOUNT AT ZERO
1233 003256 005037 001220      CLR    LOCK           ;
1234 003262 012737 000050 001222  MOV     #50,ICOUNT     ;RESET ITERATIONS
1235 003270 013737 001216 001214  MOV     NEXT,RETURN    ;GET NEXT TEST
1236 003276 011600      3$:  MOV     (SP),RO       ;POP RO OFF STACK
1237 003300 022626      POP2SP                ;FAKE AN RTI
1238 003302 000177 175706      JMP     @RETURN        ;GO DO THE TEST
1239 003306 001407      BRW:  1407
1240 003310 000437      BRX:  437
1241
1242
1243      ;CHECK FOR FREEZE ON CURRENT DATA
1244      -----
1245 003312 032777 001000 175662 .SCOP1: BIT      #SW09,@SWR    ;IS SW09=1(SET)?
1246 003320 001405              BEQ     1$             ;BR IF NOT SET.
1247 003322 005737 001220      TST    LOCK           ;
1248 003326 001402              BEQ     1$             ;
1249 003330 013716 001220      MOV     LOCK,(SP)     ;GOTO THE ADDRESS IN LOCK.
1250 003334 000002      1$:  RTI                ;GO BACK.
1251
1252      ;TELETYPE OUTPUT ROUTINE
1253      -----
1254
1255 003336 010546      .TYPE: MOV     R5,-(SP)   ;SAVE R5 ON THE STACK.
1256 003340 017605 000002      MOV     @2(SP),R5     ;GET ADDRESS OF MESSAGE.
1257 003344 062766 000002 000002  ADD     #2,2(SP)      ;POP OVER ADDRESS.
1258 003352 032777 010000 175622 1$:  BIT      #SW12,@SWR   ;INHIBIT ALL PRINT OUT??
1259 003360 001012              BNE     3$             ;BR IF NO PRINT OUT WANTED (SW12=1)
1260 003362 105715              TSTB   (R5)           ;IS NUMBER MINUS? (MSB=1(BIT7))
1261 003364 100002              BPL     2$             ;BR IF NUMBER IS PLUS
1262 003366 104402 005064      TYPE   MCRLF          ;TYPE A CR/LF!
1263 003372 105777 175612      2$:  TSTB   @TPCSR     ;TTY READY?
1264 003376 100375              BPL     2$             ;BR IF NO.
1265 003400 112577 175606      MOV    (R5)+,@TPDBR   ;PRINT CURRENT CHAR.
1266 003404 001362              BNE     1$             ;IF NOT ZERO KEEP PRINTING!
1267 003406 012605      3$:  MOV     (SP)+,R5     ;END OF OUTPUT. RESTORE R5
1268 003410 000002      RTI                ;GO HOME
1269
1270
1271 003412 010346      .INSTR: MOV     R3,-(SP) ;SAVE R3 ON STACK
1272 003414 010446      MOV     R4,-(SP)     ;SAVE R4 ON STACK
1273 003416 017637 000004 003434      MOV     @4(SP),MSG   ;
1274 003424 062766 000002 000004  ADD     #2,4(SP)     ;
1275 003432 104402      .INST1: TYPE
1276 003434 000000      .MSG:  0
1277 003436 012704 006110      MOV     #INBUF,R4

```

DZDPRA MACY11 27(732) 21-OCT-76 15:33 PAGE 34  
 DZDPRA.CMB END OF PASS ROUTINE

1278	003442	012703	000007		MOV	#7,R3		
1279	003446	105777	175532	1\$:	TSTB	@TKCSR		
1280	003452	100375			BPL	1\$		
1281	003454	117714	175526		MOVB	@TKDBR,(R4)		
1282	003460	142714	000200		BICB	#200,(R4)		
1283	003464	122427	000015		CMPB	(R4)+,#15		
1284	003470	001417			BEQ	INSTR2		
1285	003472	105777	175512	2\$:	TSTB	@TPCSR		
1286	003476	100375			BPL	2\$		
1287	003500	017777	175502	175504	MOV	@TKDBR,@TPDBR		
1288	003506	005303			DEC	R3		
1289	003510	001356			BNE	1\$		
1290	003512	012604			MOV	(SP)+,R4		
1291	003514	012603			MOV	(SP)+,R3		
1292	003516	010346		.INSTE:	MOV	R3,-(SP)		
1293	003520	010446			MOV	R4,-(SP)		
1294	003522	104402	005060		TYPE	,MOM		
1295	003526	000741			BR	.INST1		
1296	003530	012604		INSTR2:	MOV	(SP)+,R4	;RESTORE R4	
1297	003532	012603			MOV	(SP)+,R3	;RESTORE R3	
1298	003534	000002			RTI			
1299								
1300								
1301								
1302								
1303	003536	010546		.PARAM:	MOV	R5,-(SP)		
1304	003540	010446			MOV	R4,-(SP)		
1305	003542	016605	000004		MOV	4(SP),R5		
1306	003546	012537	003726		MOV	(R5)+,LOLIM		
1307	003552	012537	003730		MOV	(R5)+,HILIM		
1308	003556	012537	003732		MOV	(R5)+,DEVADR		
1309	003562	112537	003734		MOVB	(R5)+,LOBITS		
1310	003566	112537	003735		MOVB	(R5)+,ADRCNT		
1311	003572	010566	000004		MOV	R5,4(SP)		
1312	003576	005005		PARAM1:	CLR	R5		
1313	003600	012704	006110		MOV	#INBUF,R4		
1314	003604	122714	000015		CMPB	#15,(R4)		
1315	003610	001420			BEQ	PARERR		
1316	003612	121427	000060	1\$:	CMPB	(R4),#60		
1317	003616	002415			BLT	PARERR		
1318	003620	121427	000067		CMPB	(R4),#67		
1319	003624	003012			BGT	PARERR		
1320	003626	142714	000060		BICB	#60,(R4)		
1321	003632	152405			BISB	(R4)+,R5		
1322	003634	122714	000015		CMPB	#15,(R4)		
1323	003640	001406			BEQ	LIMITS		
1324	003642	006305			ASL	R5		
1325	003644	006305			ASL	R5		
1326	003646	006305			ASL	R5		
1327	003650	000760			BR	1\$		
1328	003652	104404		PARERR:	INSTR			
1329	003654	000750			BR	PARAM1		
1330								
1331								
1332								
1333								

;CONVERT ASCII STRING TO OCTAL  
 -----  
 ;TEST TO SEE IF NUMBER IS WITHIN LIMITS  
 -----

```

1334 003656 020537 003730
1335 003662 101373
1336 003664 020537 003726
1337 003670 103770
1338 003672 133705 003734
1339 003676 001365
1340
1341
1342
1343 003700 013704 003732
1344 003704 010524
1345 003706 062705 000002
1346 003712 105337 003735
1347 003716 001372
1348 003720 012604
1349 003722 012605
1350 003724 000002
1351 003726 000000
1352 003730 000000
1353 003732 000000
1354 003734 000000
1355 003735
1356
1357
1358
1359
1360 003736 016637 000004 001266
1361
1362
1363
1364 003744 010537 001262
1365 003750 010437 001260
1366 003754 010337 001256
1367 003760 010237 001254
1368 003764 010137 001252
1369 003770 010037 001250
1370 003774 000002
1371
1372
1373
1374 003776 013700 001250
1375 004002 013701 001252
1376 004006 013702 001254
1377 004012 013703 001256
1378 004016 013704 001260
1379 004022 013705 001262
1380 004026 000002
1381
1382
1383
1384
1385
1386 004030 104402 005064
1387 004034 010046
1388 004036 010146
1389 004040 010346
  
```

```

LIMITS: CMP      R5, HILIM
        BHI      PARERR
        CMP      R5, LOLIM
        BLO      PARERR
        BITB     LOBITS, R5
        BNE      PARERR

;STORE NUMBER AT SPECIFIED ADDRESS

IS:    MOV      DEVADR, R4
        MOV      R5, (R4)+
        ADD      #2, R5
        DECB     ADRCNT
        BNE      IS
        MOV      (SP)+, R4
        MOV      (SP)+, R5
        RTI

LOLIM: 0
HILIM: 0
DEVADR: 0
LOBITS: 0
ADRCNT=LOBITS+1

;SAVE PC OF TEST THAT FAILED AND R0-R5
-----

.SAV05: MOV      4(SP), SAVPC      ;SAVE R7 (PC)
        ;SAVE R0-R5

SV05:   MOV      R5, SAVR5      ;SAVE R5
        MOV      R4, SAVR4      ;SAVE R4
        MOV      R3, SAVR3      ;SAVE R3
        MOV      R2, SAVR2      ;SAVE R2
        MOV      R1, SAVR1      ;SAVE R1
        MOV      R0, SAVR0      ;SAVE R0
        RTI                    ;LEAVE.

;RESTORE R0-R5

.RES05: MOV      SAVR0, R0      ;RESTORE R0
        MOV      SAVR1, R1      ;RESTORE R1
        MOV      SAVR2, R2      ;RESTORE R2
        MOV      SAVR3, R3      ;RESTORE R3
        MOV      SAVR4, R4      ;RESTORE R4
        MOV      SAVR5, R5      ;RESTORE R5
        RTI                    ;LEAVE

;CONVERT OCTAL NUMBER TO ASCII AND OUTPUT TO TELEPRINTER
-----

.CONVR: TYPE     MCRLF
.CNVRT: MOV      R0, -(SP)
        MOV      R1, -(SP)
        MOV      R3, -(SP)
  
```

1390	004042	010446			MOV	R4,-(SP)
1391	004044	010546			MOV	R5,-(SP)
1392	004046	017601	000012		MOV	2(2(SP),R1
1393	004052	062766	000002	000012	ADD	#2,12(SP)
1394	004060	012137	004234		MOV	(R1)+,WRDCNT
1395	004064	112137	004236	1\$:	MOVB	(R1)+,CHRCNT
1396	004070	112137	004237		MOVB	(R1)+,SPACNT
1397	004074	013137	004240		MOV	2(R1)+,BINWRD
1398	004100	013704	004240	2\$:	MOV	BINWRD,R4
1399	004104	113705	004236		MOVB	CHRCNT,R5
1400	004110	012700	006004		MOV	#TEMP,R0
1401	004114	010403		3\$:	MOV	R4,R3
1402	004116	042703	177770		BIC	#177770,R3
1403	004122	062703	000060		ADD	#060,R3
1404	004126	110320			MOVB	R3,(R0)+
1405	004130	000241			CLC	
1406	004132	006004			ROR	R4
1407	004134	000241			CLC	
1408	004136	006004			ROR	R4
1409	004140	000241			CLC	
1410	004142	006004			ROR	R4
1411	004144	005305			DEC	R5
1412	004146	001362			BNE	3\$
1413	004150	012703	006046		MOV	#MDATA,R3
1414	004154	114023		4\$:	MOVB	-(R0),(R3)+
1415	004156	105337	004236		DECB	CHRCNT
1416	004162	001374			BNE	4\$
1417	004164	105737	004237		TSTB	SPACNT
1418	004170	001405			BEG	6\$
1419	004172	112723	000040	5\$:	MOVB	#040,(R3)+
1420	004176	105337	004237		DECB	SPACNT
1421	004202	001373			BNE	5\$
1422	004204	105013		6\$:	CLRB	(R3)
1423	004206	104402	006046		TYPE	,MDATA
1424	004212	005337	004234		DEC	WRDCNT
1425	004216	001322			BNE	1\$
1426	004220	012605			MOV	(SP)+,R5
1427	004222	012604			MOV	(SP)+,R4
1428	004224	012603			MOV	(SP)+,R3
1429	004226	012601			MOV	(SP)+,R1
1430	004230	012600			MOV	(SP)+,R0
1431	004232	000002			RTI	
1432	004234	000000				
1433	004236	000000				
1434		004237				
1435	004240	000000				
1436						
1437						
1438						
1439						
1440						
1441						
1442						
1443	004242	017605	000000		.SETFLG:MOV	2(SP),R5
1444	004246	042737	000040	006110	BIC	#40,INBUF
1445	004254	122737	000116	006110	CMPB	#'N,INBUF ; IS IT "N" ?

WRDCNT: 0  
 CHRCNT: 0  
 SPACNT=CHRCNT+1  
 BINWRD: 0

;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

```

1446 004262 001002          BNE      1$
1447 004264 105015          CLRB    (R5)      ;000
1448 004266 000406          BR      2$
1449 004270 122737 000131 006110 1$:  CMPB   #'Y,INBUF ;IS IT 'Y' ?
1450 004276 001005          BNE     3$
1451 004300 112715 177777          MOVB   #-1,(R5)  ;377
1452 004304 062716 000002          2$:  ADD    #2,(SP)
1453 004310 000002          RTI
1454 004312 104404          3$:  INSTER ;RETRY
1455 004314 000752          BR     .SETFLG
1456
1457
1458          ;TRAP DISPATCH SERVICE
1459          ;ARGUMENT OF TRAP IS EXTRACTED
1460          ;AND USED AS OFFSET TO OBTAIN POINTER
1461          ;TO SELECTED SUBROUTINE
1462
1463 004316 011646          .TRPSR: MOV   (SP),-(SP) ;GET PC OF RETURN
1464 004320 162716 000002          SUB    #2,(SP) ;=PC OF TRAP
1465 004324 017616 000000          MOV    @2(SP),(SP) ;GET TRP
1466 004330 006316          TRPOK: ASL   (SP) ;MULTIPLY TRAP ARG BY 2
1467 004332 042716 177001          BIC   #177001,(SP) ;CLEAR UNWANTED BITS
1468 004336 062716 001344          ADD   #.TRPTAB,(SP) ;POINTER TO SUBROUTINE ADDRESS
1469 004342 017616 000000          MOV   @2(SP),(SP) ;SUBROUTINE ADDRESS
1470 004346 000136          JMP   @2(SP)+ ;GO TO SUBROUTINE
1471
1472          ;ERROR HANDLER
1473          ;-----
1474
1475 004350 032777 010000 174624 .HLT: BIT    #SW12,@SWR ;BELL ON ERROR?
1476 004356 001406          BEQ   XBX ;BR IF NO BELL
1477 004360 105777 174624          TSTB @TPCSR ;TTY READY.
1478 004364 100003          BPL   XBX ;DON'T WAIT IF TTY NOT READY.
1479 004366 112777 000207 174616          MOVB  #207,@TPDBR ;PUSH A BELL AT THE TTY.
1480 004374 032777 020000 174600 XBX: BIT    #SW13,@SWR ;DELETE ERROR PRINT OUT?
1481 004402 001105          BNE   HALTS ;BR IF NO PRINT OUT WANTED.
1482 004404 021637 001234          CMP   (SP),LSTERR ;WAS THIS ERROR FOUND LAST TIME?
1483 004410 001404          BEQ   1$ ;BR IF YES
1484 004412 011637 001234          MOV   (SP),LSTERR ;RECORD BEING HERE
1485 004416 105037 001341          CLRB ERRFLG ;PREPARE HEADER
1486 004422 104406          1$:  SAVOS ;SAVE ALL PROC REGISTERS
1487 004424 011605          MOV   (SP),R5 ;GET THE PC OF ERROR
1488 004426 162705 000002          SUB   #2,R5 ;GET ADDRESS OF TRAP CALL
1489 004432 011504          MOV   (R5),R4 ;GET HLT INSTRUCTION
1490 004434 006304          ASL   R4 ;MULT BY TWO
1491 004436 061504          ADD   (R5),R4 ;DOUBLE IT
1492 004440 006304          ASL   R4 ;MULT AGAIN
1493 004442 042704 177001          BIC   #177001,R4 ;CLEAR JUNK
1494 004446 062704 027624          ADD   #.ERRTAB,R4 ;GET POINTER
1495 004452 012437 004566          MOV   (R4)+,ERRMSG ;GET ERROR MESSAGE
1496 004456 012437 004600          MOV   (R4)+,DATAHD ;GET DATA HEADRER
1497 004462 011437 004612          MOV   (R4),DATABP ;GET DATA TABLE
1498 004466 105737 001341          TSTB ERRFLG ;TYPE HEADREER
1499 004472 001403          BEQ   TYPMSG ;BR IF YES
1500 004474 005737 004612          TST  DATABP ;DOES DATA TABLE EXIST?
1501 004500 001040          BNE   TYPDAT ;BR IF YES.

```

1502	004502	104402	005064		TYPMSG:	TYPE	,MCRLF	
1503	004506	104402	005064			TYPE	,MCRLF	
1504	004512	005737	001220			TST	LOCK	
1505	004516	001402				BEQ	1\$	
1506	004520	104402	005340			TYPE	,MASTEK	
1507	004524	104402	005326		1\$:	TYPE	,MTSTN	
1508	004530	104411	004726			CNVRT	,XTSTN	;SHOW IT
1509	004534	104402	005415			TYPE	,MERRPC	;TYPE PC.
1510	004540	104411	004720			CNVRT	,ERTABO	;SHOW IT
1511	004544	104402	005064			TYPE	,MCRLF	;GIVE A CR/LF
1512	004550	112737	177777	001341		MOVB	1-1,ERRFLG	;NO MORE HEADER UNLESS NO DATA TABLE.
1513	004556	005737	004566			TST	ERRMSG	;IS THERE AN ERROR MESSAGE?
1514	004562	001402				BEQ	WRKO.FM	;BR IF NO.
1515	004564	104402				TYPE		;TYPE
1516	004566	000000			ERRMSG:	0		ERROR MESSAGE
1517	004570				WRKO.FM:			
1518	004570	005737	004600			TST	DATAHD	;DATA HEADER?
1519	004574	001402				BEQ	TYPDAT	;BR IF NO
1520	004576	104402				TYPE		;TYPE
1521	004600	000000			DATAHD:	0		DATA HEADER
1522	004602	005737	004612		TYPDAT:	TST	DATABP	;DATA TABLE?
1523	004606	001402				BEQ	RESREG	;BR IF NO.
1524	004610	104410				CONVRT		;SHOW
1525	004612	000000			DATABP:	0		DATA TABLE
1526	004614	104407			RESREG:	RESOS		RESTORE PROC REGISTERS
1527	004616	022737	003104	000042	HALTS:	CMP	#SENDAD,2#42	;IF ACT-11 AUTO MODE--HALT!!
1528	004624	001403				BEQ	1\$	
1529	004626	005777	174350			TST	2\$SWR	;HALT ON ERROR?
1530	004632	100010				BPL	EXITER	;BR IF NO HALT ON ERROR
1531	004634	010046			1\$:	PUSHRO		;SAVE RO
1532	004636	016600	000002			MOV	2(SP),RO	;SHOW ERROR PC IN DATA LIGHTS
1533	004642	042777	014000	174542		BIC	#SYSTST!MEXT,2TXCSR	
1534	004650	000000				HALT		;HALT
1535	004652	012600				POPPO		;GET RO
1536	004654	005237	001232		EXITER:	INC	ERRCNT	;UPDATE ERROR COUNT
1537	004660	032777	000400	174314		BIT	#SW08,2\$SWR	;GOTO TOP OF TEST?
1538	004666	001007				BNE	1\$	;BR IF YES
1539	004670	032777	002000	174304		BIT	#SW10,2\$SWR	;GOTO NEXT TEST?
1540	004676	001407				BEQ	2\$	;BR IF NO
1541	004700	013737	001216	001214		MOV	NEXT RETURN	;SET FOR NEXT TEST
1542	004706	012706	001150		1\$:	MOV	#STACK,SP	;RESET SP
1543	004712	000177	174276			JMP	2\$RETURN	;GOTO SPECIFIED TEST
1544	004716	000002			2\$	RTI		;RETURN
1545	004720	000001			ERRCNT:	1		
1546	004722	006	002			.BYTE	6,2	
1547	004724	001266				SAVPC		
1548	004726	000001			XTSTN:	1		
1549	004730	003	002			.BYTE	3,2	
1550	004732	001226				TSTNO		
1551	004734	017600	000000		.PKCLK:	MOV	2(SP),RO	;GET THE # OF TICKS TO POKE
1552	004740	062716	000002			ADD	#2,(SP)	;POP OVER THE #
1553	004744				1\$:			
1554	004744	052777	020000	174440		BIS	#CLK,2TXCSR	;POKE CLOCK UP
1555	004752	005300				DEC	RO	;ARE WE DONE?
1556	004754	001405				BEQ	2\$	;YES-GO TO 2\$
1557	004756	042777	020000	174426		BIC	#CLK,2TXCSR	;POKE CLOCK DOWN

```

1558 004764 005300          DEC      RO          ;ARE WE DONE?
1559 004766 001366          BNE     1$          ;NO-REPEAT
1560 004770 000002          2$:    RTI          ;RETURN
1561
1562
1563          ;WAIT ROUTINE
1564 004772 000240          SMALL: NOP          ;STALL
1565 004774 000207          RTS     PC          ;RETURN
1566
1567          ;POWER FAIL ROUTINE
1568
1569 004776 012737 005006 000024 .PFAIL: MOV     #PWRUP,24 ;LOAD PFAIL VECTOR FOR POWER UP
1570 005004 000000          HALT
1571 005006 000005          PWRUP: RESET        ;WAIT TTY TO COME UP
1572 005010 012706 001150          MOV     #STACK,SP   ;REINIT STACK POINTER
1573 005014 012737 004776 000024          MOV     #.PFAIL,24 ;LOAD PFAIL VECTOR FOR POWER DOWN
1574 005022 104402          TYPE
1575 005024 005067          MPOWER
1576 005026 000177 174162          JMP     @RETURN
1577          ;CLRVEC,ROUTINE TO FILL COMMUNICATION VECTOR AREA WITH .+2,HALT
1578
1579 005032 012702 000300          CLRVEC: MOV     #300,R2 ;R2 COMM VECTOR AREA ADRS
1580 005036 012701 000302          MOV     #302,R1      ;INIT R1 WITH ADRS OF HALT
1581 005042 010122          1$:    MOV     R1,(R2)+ ;MOV .+2 TO PC
1582 005044 005022          CLR     (R2)+        ;MOV HALT TO PC
1583 005046 022121          CMP     (R1)+,(R1)+ ;INC TO NEXT VECTOR AREA
1584 005050 022701 000776          CMP     #776,R1     ;END OF VECTOR AREA
1585 005054 001372          BNE     1$          ;NO
1586 005056 000207          RTS     PC          ;RETURN
1587
1588
1589
1590 005060 020040 000077          MQM:   .ASCIZ  / ?/
      (2) 005064 005015 000          MCRLF: .ASCIZ  <15><12>
      (2) 005067 377 053520 020122          MPOWER: .ASCIZ <377>/PWR FAILED. /
      (2) 005105 015 042777 042116          MEPASS: .ASCIZ <15><377>/END PASS DZDPBA /
      (2) 005131 377 000122          MR:    .ASCIZ  <377>/R/
      (2) 005134 050377 047522 051107          MERR2: .ASCIZ  <377>/PROGRAM INDICATES NO DEVICES PRESENT./
      (2) 005203 377 047111 052523          MERR3: .ASCIZ  <377>/INSUFFICIENT DATA!/
      (2) 005227 377 042524 052123          MTSTPC: .ASCIZ <377>/TEST PC-/
      (2) 005241 377 047514 045503          MLOCK: .ASCIZ  <377>/LOCK ON SELECTED TEST/
      (2) 005270 051503 035122 000040          MCSRX: .ASCIZ  /CSR: /
      (2) 005276 042526 035103 000040          MVECX: .ASCIZ  /VEC: /
      (2) 005304 040520 051523 051505          MPASSX: .ASCIZ  /PASSES: /
      (2) 005315 105 051122 051117          MERRX: .ASCIZ  /ERRORS: /
      (2) 005326 042524 052123 047040          MTSTN: .ASCIZ  /TEST NO: /
      (2) 005340 000052          MASTEK: .ASCIZ  /*/
      (2) 005342 051777 052105 051440          MNEW:  .ASCIZ  <377>/SET SWITCH REG TO DUP11'S DESIRED ACTIVE./
      (2) 005415 120 035103 000040          MERRPC: .ASCIZ  /PC: /
      (2) 005422 051377 041505 041440          MCSR:  .ASCIZ  <377>/REC CSR ADRS /
      (2) 005441 377 042526 020103          MVEC:  .ASCIZ  <377>/VEC ADRS /
      (2) 005454 044777 020123 044124          MJMPR: .ASCIZ  <377>/IS THE OPTIONAL CLR JMPR IN? (Y OR N) /
      (2) 005527 377 051511 052040          MTCN:  .ASCIZ  <377>/IS THE H325 CONNECTOR ON? (Y OR N) /
      (2) 005576 021777 047440 020106          MTOTAL: .ASCIZ  <377>/# OF DUP'S (IN OCTAL) /
      (2) 005631 377 051120 047511          MPAR:  .ASCIZ  <377>/PRIORITY (4 TO 7) /
      (2) 005655 377 042523 020103          MSTJM: .ASCIZ  <377>/SEC TX JMPR IN? (Y OR N) /
  
```

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 40  
 DZDPBA.CMB END OF PASS ROUTINE

(2)	005710	051777	041505	051040	MSRJM:	.ASCIZ	<377>/SEC RX	JMPR IN? (Y OR N) /
(2)	005743	377	040515	020120	XHEAD:	.ASCIZ	<377>/MAP OF	DUP11 STATUS/<377>
(2)		005772				.EVEN		
(2)	005772	000002			XSTATQ:	2		
1591	005774	006	003			.BYTE	6,3	
1592	005776	001236				TEMP1		
1593	006000	006	002			.BYTE	6,2	
1594	006002	001240				TEMP2		
1595						.EVEN		
1596								
1597	006004	000000			TEMP:	0		
1598		006046			.=. +40			
1599	006046	000000			MDATA:	0		
1600		006110			.=. +40			
1601	006110	000000			INBUF:	0		
1602		006152			.=. +40			
1603	006152	000001			TRP.PC:	.BLKW 1		
1604								



1605  
1606  
1607  
1608  
1609  
1610  
1611  
1612  
1613  
1614  
1615  
1616  
1617  
1618  
1619  
1620  
1621  
1622  
1623  
1624  
1625  
1626  
1627  
1628  
1629  
1630  
1631  
1632  
1633  
1634  
1635  
1636  
1637  
1638  
1639  
1640  
1641  
1642  
1643  
1644  
1645  
1646  
1647  
1648  
1649  
1650  
1651  
1652  
1653  
1654  
1655  
1656  
1657  
1658  
1659  
1660

006154 105737 001310  
006160 001004  
006162 104402 005134  
006166 000000  
006170 000776  
006172 133737 001314 001310  
006200 001020  
006202 000241  
006204 106137 001314  
006210 105537 001314  
006214 062737 000006 001316  
006222 022737 001560 001316  
006230 001360  
006232 012737 001500 001316  
006240 000754  
006242 000241  
006244 106137 001314  
006250 105537 001314  
006254 013700 001316  
006260 062737 000006 001316  
006266 022737 001560 001316  
006274 001003  
006276 012737 001500 001316  
006304 012037 001404  
006310 012037 001374  
006314 012037 001434  
006320 012700 000002  
006324 013737 001404 001424  
006332 005237 001424  
006336 013737 001424 001406  
006344 005237 001406  
006350 013737 001406 001416  
006356 013737 001406 001410  
006364 013737 001406 001422  
006372 005237 001422  
006376 013737 001422 001420  
006404 013737 001420 001412  
006412 005237 001412  
006416 013737 001412 001430  
006424 005237 001430  
006430 013737 001430 001414  
006436 005237 001414  
006442 013737 001414 001426  
006450 005237 001426  
006454 013737 001374 001376

CYCLE: TSTB DUPACTV  
BNE 1\$  
TYPE ,MERR2  
HALT  
BR -2  
1\$: BITB RUN,DUPACTV  
BNE 2\$  
CLC  
ROLB RUN  
ADCB RUN  
ADD #6,CREAM  
CMP #DUP.END,CREAM  
BNE 1\$  
MOV #DUP.MAP,CREAM  
BR 1\$  
2\$: CLC  
ROLB RUN  
ADCB RUN  
MOV CREAM,PO  
ADD #6,CREAM  
CMP #DUP.END,CREAM  
BNE 3\$  
MOV #DUP.MAP,CREAM  
3\$: MOV (RO)+,RXCSR  
MOV (RO)+,DUPRVC  
MOV (RO)+,LOO.00  
MOV #2,RO  
MOV RXCSR,HUPRCR  
INC HUPRCR  
MOV HUPRCR,RXDBUF  
INC RXDBUF  
MOV RXDBUF,DUPSEC  
MOV RXDBUF,PARCSR  
MOV RXDBUF,HUPRBF  
INC HUPRBF  
MOV HUPRBF,HUPPSR  
MOV HUPPSR,TXCSR  
INC TXCSR  
MOV TXCSR,HUPTCR  
INC HUPTCR  
MOV HUPTCR,TXDBUF  
INC TXDBUF  
MOV TXDBUF,HUPTBF  
INC HUPTBF  
MOV DUPRVC,DUPRPS

: ROUTINE USED TO "CYCLE" THROUGH UP TO EIGHT DUP11'S  
: THIS ROUTINE SETS UP THE CONTROL ADDRESS FOR THE DIAGNOSTIC  
: AND RUNS THE SPECIFIED DUP11'S. THIS ROUTINE \*MUST\*  
: BE RUN FIRST BEFORE ENTERING THE DIAGNOSTIC FOR THE  
: SETUP NECESSARY.  
: ARE ANY DUP11'S TO BE TESTED?  
: BR IF OK.  
: NO DUP11'S SELECTED!!  
: STOP THE SHOW.  
: DISQUALIFY CONT. SW.  
: IS THIS ONE "ACTIVE"  
: BR IF GOOD ONE FOUND.  
: CLEAR PROC. CARRY BIT.  
: UPDATE POINTER  
: CATCH CARRY FROM RUN  
: UPDATE ADDRESS POINTER.  
: KEEP GOING; NOT ALL TESTED FOR.  
: RESET ADDRESS POINTER.  
: KEEP LOOKING FOR ACTIVE DUP11  
: CLEAR PROC. CARRY.  
: UPDATE POINTER.  
: CATCH CARRY.  
: GET ADDRESS POINTER.  
: UPDATE.  
: ALL DONE?  
: BR IF NO.  
: RESTORE POINTER.  
: LOAD SYSTEM CTRL. REG  
: LOAD VECTOR  
: GET PARAMETERS  
: SAVE CORE THIS WAY!  
: GET CONTROL REG HIGH BYTE  
: GOT IT  
: GET RX CONTROL REG BUFFER  
: GOT IT  
: GOT SECONDARY REG SELECT REG  
: GOT PARAMETER STATUS REGISTER  
: GET RX BUFFER HIGH BYTE  
: GOT IT  
: GOT PAR STATUS REG HIGH BYTE  
: GET TX CONTROL REGISTER  
: GOT IT  
: GET TX CONTROL REG HIGH BYTE  
: GOT IT  
: BET TX BUFFER  
: GOT IT  
: GET TX BUFFER HIGH BYTE  
: GOT IT  
: RX VECTOR

1661	006462	060037	001376			ADD	RO, DUPRPS	;RX PRIORITY LEVEL
1662	006466	013737	001376	001400		MOV	DUPRPS, DUPTVC	
1663	006474	060037	001400			ADD	RO, DUPTVC	;TX VECTOR
1664	006500	013737	001400	001402		MOV	DUPTVC, DUPTPS	
1665	006506	060037	001402			ADD	RO, DUPTPS	;TX PRIORITY LEVEL
1666								
1667								
1668	006512	012700	001434			MOV	#L00.00, RO	;LOAD STAUS 00-00
1669	006516	012701	001432			MOV	#MASK.A, R1	;PREPARE MASK.
1670	006522	012702	001433			MOV	#CLK.A, R2	;PREPARE CLOCKS
1671	006526	004737	006672			JSR	PC, FIX.00	;GO AND CALCULATE CONFIGURATION.
1672	006532	005737	000042			TST	#42	
1673	006536	001050				BNE	4\$	
1674	006540	032777	000002	172434		BIT	#SW01, #SWR	;IF SW01=1, GET STARTING TEST #
1675	006546	001444				BEQ	4\$	
1676	006550	104402	005064		7\$:	TYPE	, MCRLF	
1677	006554	104403				INSTR	; OUTPUT MESSAGE & GET INPUT STRING	
1678	006556	005326				MTSTN	; MESSAGE	
1679	006560	104405				PARAM	; CONVERT STRING	
1680	006562	000001				I	; LOW LIMIT	
1681	006564	001000				I000	; HIGH LIMIT	
1682	006566	001226				TSTNO	; STORE AT THIS LOCATION	
1683	006570	000			.BYTE	D	; MASK	
1684	006571	001			.BYTE	I	; HOW MANY TIMES + 2	
1685	006572	012700	007106			MOV	#TST1, RO	
1686	006576	022710	012737		5\$:	CMP	#12737, (RO)	
1687	006602	001017				BNE	6\$	
1688	006604	023760	001226	000002		CMP	TSTNO, 2(RO)	
1689	006612	001013				BNE	6\$	
1690	006614	022760	001226	000004		CMP	#TSTNO, 4(RO)	
1691	006622	001007				BNE	6\$	
1692	006624	010037	001214			MOV	RO, RETURN	;SAVE PC
1693	006630	104402	005064			TYPE	, MCRLF	
1694	006634	104402	005131			TYPE	, MR	
1695	006640	000412				BR	8\$	
1696	006642	005720			6\$:	TST	(RO)+	
1697	006644	020027	026214			CMP	RO, #TLAST+10	
1698	006650	001352				BNE	5\$	
1699	006652	104402	005060			TYPE	, MQM	
1700	006656	000734				BR	7\$	
1701								
1702	006660	012737	007106	001214	4\$:	MOV	#TST1, RETURN	;PREPARE RETURN ADDRESS
1703	006666	000177	172322		8\$:	JMP	#RETURN	;GO START TESTING.
1704								
1705	006672	011003			FIX.00:	MOV	(RO), R3	;GET PARAMETERS.
1706	006674	000207			5\$:	RTS	PC	;

```

1707
1708
1709
1710 006676 012104          ACC:  MOV      (R1)+,R4      ;GET THE FLAG FOR # OF CLOCK TICKS
1711 006700 104412 000002 1$:  PKCLK    ,2
1712 006704 000241          CLC      ;PUT CARRY IN A KNOWN STATE
1713 006706 032777 040000 172476 BIT      @MTDATA,@TXCSR ;FIND OUT IF BIT IS A 1 OR 0
1714 006714 001401          BEQ      .+4           ;BR IF 0
1715 006716 000261          SEC      ;SET THE BIT
1716 006720 006004          ROR      R4           ;PICK UP CARRY AND PUSH INTO R4
1717 006722 103366          BCC      1$         ;BRANCH IF MORE TO GO
1718 006724 000201          RTS      R1
1719 006726 005037 001246  ABRT:  CLR      TEMPS
1720 006732 012137 001244 1$:  MOV      (R1)+,TEMP4    ;GET THE # OF ABORTS TO DO
1721 006736 104412 000002 1$:  PKCLK    ,2           ;POKE OUT A BIT
1722 006742 032777 040000 172442 BIT      @MTDATA,@TXCSR ;CHECK MAINT DATA OUT
1723 006750 001001          BNE      2$         ;BRANCH IF IT IS A ONE
1724 006752 104013          HLT      13         ;OUTPUT WAS A ZERO--NG
1725 006754 005237 001246 2$:  INC      TEMPS      ;INC THE # OF BITS OUTPUT
1726 006760 022737 000010 001246 CMP      @B.,TEMP5    ;IS THE CHARACTER DONE?
1727 006766 001363          BNE      1$         ;BRANCH IF NOT DONE
1728 006770 005337 001244          DEC      TEMP4     ;LOWER THE #TO DO
1729 006774 001360          BNE      1$         ;BRANCH IF NOT DONE
1730 006776 000201          RTS      R1
1731
1732 007000 012137 001244  FLG:  MOV      (R1)+,TEMP4    ;GET THE # OF FLAGS
1733 007004 104412 000002 64$:  PKCLK    ,2           ;POKE OUT THE FIRST BIT OF THE FLAG
1734 007010 032777 040000 172374 BIT      @MTDATA,@TXCSR ;CHECK MAINT DATA OUT
1735 007016 001401          BEQ      65$        ;BRANCH IF 0
1736 007020 104012          HLT      12         ;DUP FAILED TO CLOCK OUT FIRST BIT
1737 007022 005037 001246 65$:  CLR      TEMPS      ;SETUP FOR 1'S OUTPUT
1738 007026 104412 000002 1$:  PKCLK    ,2           ;CONTINUE TO POKE OUT BITS
1739 007032 032777 040000 172352 BIT      @MTDATA,@TXCSR ;TEST MAINT DATA OUT
1740 007040 001001          BNE      2$         ;BRANCH IF A 1
1741 007042 104013          HLT      13         ;DUP FAILED TO CLOCK A ONE
1742 007044 005237 001246 2$:  INC      TEMPS      ;KEEP UP WITH THE # OF 1'S OUTPUT
1743 007050 022737 000006 001246 CMP      @6,TEMP5    ;ARE WE DONE WITH SIX ONES?
1744 007056 001363          BNE      1$         ;NO+BRANCH BACK
1745 007060 104412 000002          PKCLK    ,2           ;YES,OUTPUT THE LAST 0
1746 007064 032777 040000 172320 BIT      @MTDATA,@TXCSR ;CHECK IT
1747 007072 001401          BEQ      3$         ;BRANCH IF A 0
1748 007074 104012          HLT      12         ;LAST BIT OF FLAG WAS NOT CORECT
1749 007076 005337 001244 3$:  DEC      TEMP4     ;ARE WE DONE WITH FLAGS?
1750 007102 001340          BNE      64$        ;BR IF NO
1751 007104 000201          RTS      R1
1752

```

1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762  
1763  
1764  
1765  
1766  
1767  
1768  
1769  
1770  
1771  
1772  
1773  
1774  
1775  
1776  
1777  
1778  
1779  
1780  
1781  
1782  
1783  
1784  
1785  
1786  
1787  
1788  
1789  
1790  
1791  
1792  
1793  
1794  
1795  
1796  
1797  
1798  
1799  
1800  
1801  
1802  
1803  
1804  
1805  
1806  
1807  
1808

007106 012737 000001 001226  
007114 012737 007212 001216  
007122 012737 007154 001220  
007130 012700 000004  
007134 013701 001404  
007140 012737 007204 000004  
007146 012737 000340 000006  
007154 005711  
007156 104401  
007160 062701 000002  
007164 005300  
007166 001372  
007170 012737 000006 000004  
007176 005037 000006  
007202 104400  
007204 011602  
007206 104001  
007210 000002  
  
007212 012737 000002 001226  
007220 012737 007370 001216  
007226 012737 007252 001220  
007234 012700 007360  
007240 013703 001404  
007244 005013  
007246 012702 000004  
007252 011005  
007254 010513  
007256 011304  
007260 042704 000200  
007264 042704 000200  
007270 020504

```
***** TEST 1 *****  
: *VERIFY THAT ADDRESSING DEVICE DOES *NOT* CAUSE  
: *A TIME-OUT TRAP.  
: *****  
: *****  
: TEST 1 *  
: *****  
: *****  
TST1: MOV #1, J#TSTNO  
      MOV #TST2, NEXT  
      MOV #1$ , LOCK  
      MOV #4, R0 ;SET FOR MAX. 4 PRI. REGISTERS  
      MOV RXCSR, R1 ;GET FIRST PRI. ADDRESS  
      MOV #2$ , 4 ;SET FOR TIME-OUT TRAP.  
      MOV #340, 6 ;SAFE GUARD.  
1$: TST (R1) ;REFERENCE THE ADDRESS.  
   SCOPE1 ;IF SW09=1; GOTO 1$  
   ADD #2, R1 ;UPDATE TO NEXT ADDRESS.  
   DEC R0 ;ARE ALL ADDRESS CHECKED?  
   BNE 1$ ;BR IF NO.  
   MOV #6, 4 ;RESET TRAP ZONE.  
   CLR J#6  
2$: SCOPE ;SCOPE THIS TEST  
   MOV (SP), R2 ;SAVE THE TRAP PC  
   HLT 1 ;REPORT TIME-OUT TRAP  
   RTI ;RETURN TO MAIN PROGRAM
```

```
***** TEST 2 *****  
: *PRIMARY REGISTER ADDRESSING TEST  
: *LOAD EACH PRIMARY REGISTER WITH A  
: *DIFFERENT NUMBER AND VERIFY EACH  
: *WAS INDIVIDUALLY ADDRESSED  
: *****  
: *****  
: TEST 2 *  
: *****  
: *****  
TST2: MOV #2, J#TSTNO  
      MOV #TST3, NEXT  
      MOV #1$ , LOCK  
      MOV #3$ , R0 ;SET THE TABLE POINTER  
      MOV RXCSR, R3 ;SET THE DUP HARDWARE POINTER  
      CLR (R3) ;CLR THE REGISTER BEFORE STARTING  
      MOV #4, R2 ;SET FOR 4 PRIMARY REGISTERS  
1$: MOV (R0), R5 ;SET "EXPECTED"  
   MOV R5, (R3) ;WRITE "EXPECTED" TO THE REGISTER  
   MOV (R3), R4 ;READ THE REGISTER BACK  
   BIC #BIT7, R4 ;CLR UNWANTED BIT  
   BIC #BIT7, R4 ;CLR UNWANTED BITS  
   CMP R5, R4 ;DOES EXPECTED=RECEIVED?
```

```

1809 007272 001401          BEQ      2$          ;BR IF YES
1810 007274 104003          HLT      3          ;THIS IS A DATA ERROR. IT IS **NOT**
1811                                     ;A DUAL ADDRESSING ERROR!!!!!!
1812 007276 104401          2$:      SCOPE1      ;SW09=1?
1813 007300 022023          CMP      (R0)+,(R3)+ ;POP DATA AND HARDWARE POINTERS
1814 007302 005302          DEC      R2          ;UPDATE THE REGISTER COUNTER
1815 007304 001362          BNE     1$          ;BRANCH IF MORE TO GO
1816                                     ;;NOW CHECK FOR DUAL ADDRESSING
1817 007306 012700 007360    MOV      #3$,R0      ;SET THE TABLE POINTER
1818 007312 013703 001404    MOV      RXCSR,R3    ;SET THE DUP HARDWARE POINTER
1819 007316 012737 007330 001220 MOV      #4$,LOCK    ;SET FOR SW09=1
1820 007324 012702 000004    MOV      #4,R2       ;SET FOR 4 PRIMARY REGISTERS
1821 007330 011005          4$:      MOV      (R0),R5 ;SET "EXPECTED"
1822 007332 011304          MOV      (R3),R4     ;READ THE REGISTER BACK
1823 007334 042704 000200    BIC     #BIT7,R4     ;CLR UNWANTED BITS
1824 007340 020504          CMP      R5,R4
1825 007342 001401          BEQ     5$          ;BRANCH IF OK
1826 007344 104003          HLT     3          ;THIS IS A DUAL ADDRESSING ERROR
1827 007346 104401          5$:      SCOPE1      ;SW09=1?
1828 007350 022023          CMP      (R0)+,(R3)+ ;POP POINTERS
1829 007352 005302          DEC      R2          ;UPDATE THE REGISTER COUNTER
1830 007354 001365          BNE     4$          ;BRANCH IF MORE TO GO
1831 007356 104400          SCOPE   ;SCOPE THIS TEST
1832 007360 000020          3$:      .WORD    00020 ;RXCSR
1833 007362 000000          .WORD    00000 ;RXDBUF AND PARCSR
1834 007364 000010          .WORD    00010 ;TXCSR
1835 007366 000100          .WORD    00100 ;TXDBUF

```

```

***** TEST 3 *****
;RECEIVER CONTROL REGISTER RESET TEST. TEST THAT AFTER
;RECEIVER CONTROL REGISTER IS WRITTEN AND A BUS RESET IS
;DONE THAT RECEIVER CONTROL REGISTER IS CLEARED.
*****

```

```

*****
;
; TEST 3
;
*****

```

```

1849
1850 007370 012737 000003 001226 TST3:  MOV      #3, @TSTNO
1851 007376 012737 007432 001216      MOV      @TST4, NEXT
1852 007404 005005          CLR      R5          ;SET "EXPECTED"
1853 007406 013703 001404    MOV      RXCSR,R3    ;GET THE RECEIVER CONTROL REGISTER
1854 007412 012713 177777    MOV      #-1,(R3)   ;LOAD RECEIVER CONTROL REGISTER WITH ALL ONES
1855 007416 000005          RESET
1856 007420 011304          MOV      (R3),R4     ;READ THE RECEIVER CONTROL REGISTER
1857 007422 020504          CMP      R5,R4       ;R5=GOOD, R4=?
1858 007424 001401          BEQ     1$          ;BR IF OK
1859 007426 104002          HLT     2          ;COMPARISON ERROR
1860 007430 104400          1$:      SCOPE   ;SCOPE THIS TEST

```

```

***** TEST 4 *****
;RECEIVER BUFFER REGISTER RESET TEST. TEST THAT AFTER A BUS

```

1865  
 1866  
 1867  
 1868  
 1869  
 1870  
 1871  
 1872  
 1873  
 1874  
 1875  
 1876  
 1877  
 1878  
 1879  
 1880  
 1881  
 1882  
 1883  
 1884  
 1885  
 1886  
 1887  
 1888  
 1889  
 1890  
 1891  
 1892  
 1893  
 1894  
 1895  
 1896  
 1897  
 1898  
 1899  
 1900  
 1901  
 1902  
 1903  
 1904  
 1905  
 1906  
 1907  
 1908  
 1909  
 1910  
 1911  
 1912  
 1913  
 1914  
 1915  
 1916  
 1917  
 1918  
 1919  
 1920

007432	012737	000004	001226
007440	012737	007470	001216
007446	005005		
007450	013703	001406	
007454	000005		
007456	011304		
007460	020504		
007462	001401		
007464	104002		
007466	104400		
007470	012737	000005	001226
007476	012737	007532	001216
007504	005005		
007506	013703	001410	
007512	012713	177777	
007516	000005		
007520	011304		
007522	020504		
007524	001401		
007526	104002		
007530	104400		

;\*RESET IS DONE THAT RECEIVER BUFFER REGISTER IS CLEARED.  
 ;\*\*\*\*\*

\*\*\*\*\*  
 TEST 4  
 \*\*\*\*\*

```

TST4:  MOV   #4, @TSTNO
        MOV   @TST5, NEXT
        CLR   R5           ;SET "EXPECTED"
        MOV   RXDBUF, R3    ;GET THE RECEIVER BUFFER REGISTER
        RESET
        MOV   (R3), R4      ;READ THE RECEIVER BUFFER REGISTER
        CMP   R5, R4        ;R5=GOOD, R4= ?
        BEQ   IS           ;BR IF OK
        HLT   2             ;COMPARISON ERROR
    IS:  SCOPE               ;SCOPE THIS TEST
    
```

\*\*\*\*\* TEST 5 \*\*\*\*\*  
 ;PARAMETER STATUS REGISTER RESET TEST. TEST THAT AFTER  
 ;PARAMETER STATUS REGISTER IS WRITTEN AND A BUS RESET IS  
 ;DONE THAT PARAMETER STATUS REGISTER IS CLEARED.  
 ;\*\*\*\*\*

\*\*\*\*\*  
 TEST 5  
 \*\*\*\*\*

```

TST5:  MOV   #5, @TSTNO
        MOV   @TST6, NEXT
        CLR   R5           ;SET "EXPECTED"
        MOV   PARCSR, R3    ;GET THE PARAMETER STATUS REGISTER
        MOV   #-1, (R3)     ;LOAD PARAMETER STATUS REGISTER WITH ALL ONES
        RESET
        MOV   (R3), R4      ;READ THE PARAMETER STATUS REGISTER
        CMP   R5, R4        ;R5=GOOD, R4= ?
        BEQ   IS           ;BR IF OK
        HLT   2             ;COMPARISON ERROR
    IS:  SCOPE               ;SCOPE THIS TEST
    
```

\*\*\*\*\* TEST 6 \*\*\*\*\*  
 ;TRANSMITTER CONTROL REGISTER RESET TEST. TEST THAT AFTER  
 ;TRANSMITTER CONTROL REGISTER IS WRITTEN AND A BUS RESET IS  
 ;DONE THAT TRANSMITTER CONTROL REGISTER IS CLEARED.  
 ;\*\*\*\*\*

\*\*\*\*\*  
 TEST 6  
 \*\*\*\*\*

```

1921
1922
1923 007532 012737 000006 001226
1924 007540 012737 007600 001216
1925 007546 005005
1926 007550 013703 001412
1927 007554 012713 177777
1928 007560 000005
1929 007562 011304
1930 007564 042704 000200
1931 007570 020504
1932 007572 001401
1933 007574 104002
1934 007576 104400
1935
1936
1937
1938
1939
1940
1941
1942
1943
1944
1945
1946
1947
1948
1949 007600 012737 000007 001226
1950 007606 012737 007642 001216
1951 007614 005005
1952 007616 013703 001414
1953 007622 012713 177777
1954 007626 000005
1955 007630 011304
1956 007632 020504
1957 007634 001401
1958 007636 104002
1959 007640 104400
1960
1961
1962
1963
1964
1965
1966
1967
1968
1969
1970
1971
1972
1973
1974 007642 012737 000010 001226
1975 007650 012737 007726 001216
1976 007656 013703 001412

```

```

:*****
:*****
TST6: MOV #6, @TSTNO
      MOV #TST7, NEXT
      CLR R5 ;SET "EXPECTED"
      MOV TXCSR, R3 ;GET THE TRANSMITTER CONTROL REGISTER
      MOV #-1, (R3) ;LOAD TRANSMITTER CONTROL REGISTER WITH ALL ONES
      RESET
      MOV (R3), R4 ;READ THE TRANSMITTER CONTROL REGISTER
      BIC #BIT7, R4 ;CLR UNWANTED BITS
      CMP R5, R4 ;R5=GOOD, R4= ?
      BEQ 1$ ;BR IF OK
      HLT 2 ;COMPARISON ERROR
1$: SCOPE ;SCOPE THIS TEST

```

```

:***** TEST 7 *****
:*TRANSMITTER BUFFER REGISTER RESET TEST. TEST THAT AFTER
:*TRANSMITTER BUFFER REGISTER IS WRITTEN AND A BUS RESET IS
:*DONE THAT TRANSMITTER BUFFER REGISTER IS CLEARED.
:*****

```

```

:*****
:*****
TEST 7
:*****
:*****
TST7: MOV #7, @TSTNO
      MOV #TST10, NEXT
      CLR R5 ;SET "EXPECTED"
      MOV TXDBUF, R3 ;GET THE TRANSMITTER BUFFER REGISTER
      MOV #-1, (R3) ;LOAD TRANSMITTER BUFFER REGISTER WITH ALL ONES
      RESET
      MOV (R3), R4 ;READ THE TRANSMITTER BUFFER REGISTER
      CMP R5, R4 ;R5=GOOD, R4= ?
      BEQ 1$ ;BR IF OK
      HLT 2 ;COMPARISON ERROR
1$: SCOPE ;SCOPE THIS TEST

```

```

:***** TEST 10 *****
:*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
:*SET BIT3, VERIFY BIT3 WAS SET.
:*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
:*****

```

```

:*****
:*****
TEST 10
:*****
:*****
TST10: MOV #10, @TSTNO
       MOV #TST11, NEXT
       MOV TXCSR, R3 ;SET REGISTER TO BE TESTED.

```

250

```

1977 007662 012705 000010      MOV      #BIT3,R5      ;SET "EXPECTED "
1978 007666 010513      MOV      R5,(R3)      ;WRITE THE REGISTER.
1979 007670 011304      MOV      (R3),R4      ;READ THE REGISTER.
1980 007672 042704 000200      BIC      #BIT7,R4     ;CLEAR UNWANTED BITS
1981 007676 020504      CMP      R5,R4        ;R5=GOOD; R4=UNKNOWN.
1982 007700 001401      BEQ      1$           ;ARE THEY THE SAME?
1983 007702 104003      HLT      3            ;COMPARISON ERROR.
1984 007704 040513      1$:      BIC      R5,(R3)     ;CLEAR BIT3
1985 007706 011304      MOV      (R3),R4     ;READ THE REGISTER.
1986 007710 042704 000200      BIC      #BIT7,R4     ;CLEAR UNWANTED BITS
1987 007714 005005      CLR      R5          ;SET "EXPECTED"
1988 007716 020504      CMP      R5,R4        ;R5=GOOD; R4=?
1989 007720 001401      BEQ      2$           ;BR IF OK
1990 007722 104003      HLT      3            ;COMPARISON ERROR
1991 007724 104400      2$:      SCOPE        ;SCOPE THIS TEST

```

1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032

```

:***** TEST 11 *****
:*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
:*SET BIT4, VERIFY BIT4 WAS SET.
:*CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
:*****

```

```

:*****
:TEST 11
:*****
:*****

```

```

↑TST11:  MOV      #11,#TSTNO      ;SET REGISTER TO BE TESTED.
          MOV      #TST12,NEXT  ;SET "EXPECTED "
          MOV      TXCSR,R3     ;WRITE THE REGISTER.
          MOV      #BIT4,R5     ;READ THE REGISTER.
          MOV      R5,(R3)      ;CLEAR UNWANTED BITS
          MOV      (R3),R4      ;R5=GOOD; R4=UNKNOWN.
          BIC      #BIT7,R4     ;ARE THEY THE SAME?
          CMP      R5,R4        ;COMPARISON ERROR.
          BEQ      1$           ;CLEAR BIT4
          HLT      3            ;READ THE REGISTER.
          1$:      BIC      R5,(R3) ;CLEAR UNWANTED BITS
          MOV      (R3),R4     ;SET "EXPECTED"
          BIC      #BIT7,R4     ;R5=GOOD; R4=?
          CLR      R5          ;BR IF OK
          CMP      R5,R4        ;COMPARISON ERROR
          BEQ      2$           ;SCOPE THIS TEST
          HLT      3
          2$:      SCOPE

```

```

:***** TEST 12 *****
:*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
:*SET BIT10, VERIFY BIT10 WAS SET.
:*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.
:*****

```

```

:*****

```



```

2033
2034
2035
2036
2037
2038 010012 012737 000012 001226
2039 010020 012737 010076 001216
2040 010026 013703 001412
2041 010032 012705 002000
2042 010036 010513
2043 010040 011304
2044 010042 042704 000200
2045 010046 020504
2046 010050 001401
2047 010052 104003
2048 010054 040513
2049 010056 011304
2050 010060 042704 000200
2051 010064 005005
2052 010066 020504
2053 010070 001401
2054 010072 104003
2055 010074 104400
2056
2057
2058
2059
2060
2061
2062
2063
2064
2065
2066
2067
2068
2069
2070 010076 012737 000013 001226
2071 010104 012737 010162 001216
2072 010112 013703 001412
2073 010116 012705 004000
2074 010122 010513
2075 010124 011304
2076 010126 042704 000200
2077 010132 020504
2078 010134 001401
2079 010136 104003
2080 010140 040513
2081 010142 011304
2082 010144 042704 000200
2083 010150 005005
2084 010152 020504
2085 010154 001401
2086 010156 104003
2087 010160 104400
2088

```

```

: TEST 12 *
:*****
:*****
TST12: MOV #12,0#TSTNO
MOV #TST13,NEXT
MOV TXCSR,R3 ;SET REGISTER TO BE TESTED.
MOV #BIT10,R5 ;SET "EXPECTED"
MOV R5,(R3) ;WRITE THE REGISTER.
MOV (R3),R4 ;READ THE REGISTER.
BIC #BIT7,R4 ;CLEAR UNWANTED BITS
CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
BEQ 1$ ;ARE THEY THE SAME?
HLT 3 ;COMPARISON ERROR.
1$: BIC R5,(R3) ;CLEAR BIT10
MOV (R3),R4 ;READ THE REGISTER.
BIC #BIT7,R4 ;CLEAR UNWANTED BITS
CLR R5 ;SET "EXPECTED"
CMP R5,R4 ;R5=GOOD; R4=?
BEQ 2$ ;BR IF OK
HLT 3 ;COMPARISON ERROR
2$: SCOPE ;SCOPE THIS TEST

```

```

:***** TEST 13 *****
:TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.
:SET BIT11, VERIFY BIT11 WAS SET.
:CLEAR BIT11, VERIFY BIT11 WAS CLEARED.
:*****

```

```

:*****
: TEST 13 *
:*****
:*****
TST13: MOV #13,0#TSTNO
MOV #TST14,NEXT
MOV TXCSR,R3 ;SET REGISTER TO BE TESTED.
MOV #BIT11,R5 ;SET "EXPECTED"
MOV R5,(R3) ;WRITE THE REGISTER.
MOV (R3),R4 ;READ THE REGISTER.
BIC #BIT7,R4 ;CLEAR UNWANTED BITS
CMP R5,R4 ;R5=GOOD; R4=UNKNOWN.
BEQ 1$ ;ARE THEY THE SAME?
HLT 3 ;COMPARISON ERROR.
1$: BIC R5,(R3) ;CLEAR BIT11
MOV (R3),R4 ;READ THE REGISTER.
BIC #BIT7,R4 ;CLEAR UNWANTED BITS
CLR R5 ;SET "EXPECTED"
CMP R5,R4 ;R5=GOOD; R4=?
BEQ 2$ ;BR IF OK
HLT 3 ;COMPARISON ERROR
2$: SCOPE ;SCOPE THIS TEST

```

2089  
2090  
2091  
2092  
2093  
2094  
2095  
2096  
2097  
2098  
2099  
2100  
2101  
2102  
2103  
2104  
2105  
2106  
2107  
2108  
2109  
2110  
2111  
2112  
2113  
2114  
2115  
2116  
2117  
2118  
2119  
2120  
2121  
2122  
2123  
2124  
2125  
2126  
2127  
2128  
2129  
2130  
2131  
2132  
2133  
2134  
2135  
2136  
2137  
2138  
2139  
2140  
2141  
2142  
2143  
2144

010162 012737 000014 001226  
010170 012737 010246 001216  
010176 013703 001412  
010202 012705 010000  
010206 010513  
010210 011304  
010212 042704 000200  
010216 020504  
010220 001401  
010222 104003  
010224 040513  
010226 011304  
010230 042704 000200  
010234 005005  
010236 020504  
010240 001401  
010242 104003  
010244 104400  
  
010246 012737 000015 001226  
010254 012737 010332 001216  
010262 013703 001412  
010266 012705 020000  
010272 010513  
010274 011304  
010276 042704 000200  
010302 020504  
010304 001401  
010306 104003  
010310 040513

```
***** TEST 14 *****  
*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.  
*SET BIT12, VERIFY BIT12 WAS SET.  
*CLEAR BIT12, VERIFY BIT12 WAS CLEARED.  
*****  
*****  
*  
TEST 14  
*  
*****  
*****  
TST14: MOV #14, J#TSTNO  
MOV #TST15, NEXT  
MOV TXCSR, R3 ; SET REGISTER TO BE TESTED.  
MOV #BIT12, R5 ; SET "EXPECTED"  
MOV R5, (R3) ; WRITE THE REGISTER.  
MOV (R3), R4 ; READ THE REGISTER.  
BIC #BIT7, R4 ; CLEAR UNWANTED BITS  
CMP R5, R4 ; R5=GOOD; R4=UNKNOWN.  
BEQ 1$ ; ARE THEY THE SAME?  
HLT 3 ; COMPARISON ERROR.  
1$: BIC R5, (R3) ; CLEAR BIT12  
MOV (R3), R4 ; READ THE REGISTER.  
BIC #BIT7, R4 ; CLEAR UNWANTED BITS  
CLR R5 ; SET "EXPECTED"  
CMP R5, R4 ; R5=GOOD; R4=?  
BEQ 2$ ; BR IF OK  
HLT 3 ; COMPARISON ERROR  
2$: SCOPE ; SCOPE THIS TEST
```

```
***** TEST 15 *****  
*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.  
*SET BIT13, VERIFY BIT13 WAS SET.  
*CLEAR BIT13, VERIFY BIT13 WAS CLEARED.  
*****  
*****  
*  
TEST 15  
*  
*****  
*****  
TST15: MOV #15, J#TSTNO  
MOV #TST16, NEXT  
MOV TXCSR, R3 ; SET REGISTER TO BE TESTED.  
MOV #BIT13, R5 ; SET "EXPECTED"  
MOV R5, (R3) ; WRITE THE REGISTER.  
MOV (R3), R4 ; READ THE REGISTER.  
BIC #BIT7, R4 ; CLEAR UNWANTED BITS  
CMP R5, R4 ; R5=GOOD; R4=UNKNOWN.  
BEQ 1$ ; ARE THEY THE SAME?  
HLT 3 ; COMPARISON ERROR.  
1$: BIC R5, (R3) ; CLEAR BIT13
```

```

2145 010312 011304          MOV      (R3),R4          ;READ THE REGISTER.
2146 010314 042704 000200  BIC      #BIT7,R4        ;CLEAR UNWANTED BITS
2147 010320 005005          CLR      R5              ;SET "EXPECTED"
2148 010322 020504          CMP      R5,R4          ;R5=GOOD; R4=?
2149 010324 001401          BEQ     2$              ;BR IF OK
2150 010326 104003          HLT     3               ;COMPARISON ERROR
2151 010330 104400          2$:    SCOPE           ;SCOPE THIS TEST

```

```

***** TEST 16 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BIT0, VERIFY BIT0 WAS SET.
*CLEAR BIT0, VERIFY BIT0 WAS CLEARED.
*****

```

```

*****
*
* TEST 16
*
*****

```

```

2166 010332 012737 000016 001226  TST16: MOV      #16,#TSTNO
2167 010340 012737 010406 001216  MOV      #TST17,NEXT
2168 010346 013703          MOV      TXDBUF,R3      ;SET REGISTER TO BE TESTED.
2169 010352 012705 000001          MOV      #BIT0,R5      ;SET "EXPECTED"
2170 010356 010513          MOV      R5,(R3)       ;WRITE THE REGISTER.
2171 010360 011304          MOV      (R3),R4       ;READ THE REGISTER.
2172 010362 020504          CMP      R5,R4         ;R5=GOOD; R4=UNKNOWN.
2173 010364 001401          BEQ     1$             ;ARE THEY THE SAME?
2174 010366 104003          HLT     3               ;COMPARISON ERROR.
2175 010370 040513          1$:    BIC      R5,(R3)  ;CLEAR BIT0
2176 010372 011304          MOV      (R3),R4       ;READ THE REGISTER.
2177 010374 005005          CLR      R5            ;SET "EXPECTED"
2178 010376 020504          CMP      R5,R4         ;R5=GOOD; R4=?
2179 010400 001401          BEQ     2$             ;BR IF OK
2180 010402 104003          HLT     3               ;COMPARISON ERROR
2181 010404 104400          2$:    SCOPE           ;SCOPE THIS TEST

```

```

***** TEST 17 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BIT1, VERIFY BIT1 WAS SET.
*CLEAR BIT1, VERIFY BIT1 WAS CLEARED.
*****

```

```

*****
*
* TEST 17
*
*****

```

```

2196 010406 012737 000017 001226  TST17: MOV      #17,#TSTNO
2197 010414 012737 010462 001216  MOV      #TST20,NEXT
2198 010422 013703          MOV      TXDBUF,R3      ;SET REGISTER TO BE TESTED.
2199 010426 012705 000002          MOV      #BIT1,R5      ;SET "EXPECTED"
2200 010432 010513          MOV      R5,(R3)       ;WRITE THE REGISTER.

```

```

2201 010434 011304          MOV      (R3),R4          ;READ THE REGISTER.
2202 010436 020504          CMP      R5,R4           ;R5=GOOD; R4=UNKNOWN.
2203 010440 001401          BEQ     1$              ;ARE THEY THE SAME?
2204 010442 104003          HLT     3               ;COMPARISON ERROR.
2205 010444 040513          1$:    BIC     R5,(R3)     ;CLEAR BIT1
2206 010446 011304          MOV      (R3),R4          ;READ THE REGISTER.
2207 010450 005005          CLR     R5              ;SET "EXPECTED"
2208 010452 020504          CMP     R5,R4           ;R5=GOOD; R4=?
2209 010454 001401          BEQ     2$              ;BR IF OK
2210 010456 104003          HLT     3               ;COMPARISON ERROR
2211 010460 104400          2$:    SCOPE           ;SCOPE THIS TEST

```

```

***** TEST 20 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BIT2, VERIFY BIT2 WAS SET.
*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.
*****

```

```

*****
*
* TEST 20
*
*****

```

```

2226 010462 012737 000020 001226 TST20: MOV      #20,2#TSTNO
2227 010470 012737 010536 001216      MOV      #TST21,NEXT
2228 010476 013703          MOV      TXDBUF,R3       ;SET REGISTER TO BE TESTED.
2229 010502 012705 001414          MOV      #BIT2,R5        ;SET "EXPECTED"
2230 010506 010513          MOV      R5,(R3)        ;WRITE THE REGISTER.
2231 010510 011304          MOV      (R3),R4        ;READ THE REGISTER.
2232 010512 020504          CMP     R5,R4           ;R5=GOOD; R4=UNKNOWN.
2233 010514 001401          BEQ     1$              ;ARE THEY THE SAME?
2234 010516 104003          HLT     3               ;COMPARISON ERROR.
2235 010520 040513          1$:    BIC     R5,(R3)     ;CLEAR BIT2
2236 010522 011304          MOV      (R3),R4        ;READ THE REGISTER.
2237 010524 005005          CLR     R5              ;SET "EXPECTED"
2238 010526 020504          CMP     R5,R4           ;R5=GOOD; R4=?
2239 010530 001401          BEQ     2$              ;BR IF OK
2240 010532 104003          HLT     3               ;COMPARISON ERROR
2241 010534 104400          2$:    SCOPE           ;SCOPE THIS TEST

```

```

***** TEST 21 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BIT3, VERIFY BIT3 WAS SET.
*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
*****

```

```

*****
*
* TEST 21
*
*****

```

```

2256 010536 012737 000021 001226 TST21: MOV      #21,2#TSTNO

```

```

2257 010544 012737 010612 001216      MOV      #TST22,NEXT
2258 010552 013703 001414          MOV      TXDBUF,R3          ;SET REGISTER TO BE TESTED.
2259 010556 012705 000010          MOV      #BIT3,R5          ;SET "EXPECTED"
2260 010562 010513          MOV      R5,(R3)          ;WRITE THE REGISTER.
2261 010564 011304          MOV      (R3),R4          ;READ THE REGISTER.
2262 010566 020504          CMP      R5,R4            ;R5=GOOD; R4=UNKNOWN.
2263 010570 001401          BEQ      1$              ;ARE THEY THE SAME?
2264 010572 104003          HLT      3                ;COMPARISON ERROR.
2265 010574 040513          1$:    BIC      R5,(R3)      ;CLEAR BIT3
2266 010576 011304          MOV      (R3),R4          ;READ THE REGISTER.
2267 010600 005005          CLR      R5              ;SET "EXPECTED"
2268 010602 020504          CMP      R5,R4            ;R5=GOOD; R4=?
2269 010604 001401          BEQ      2$              ;BR IF OK
2270 010606 104003          HLT      3                ;COMPARISON ERROR
2271 010610 104400          2$:    SCOPE              ;SCOPE THIS TEST

```

```

***** TEST 22 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BIT4, VERIFY BIT4 WAS SET.
*CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
*****

```

```

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

```

```

2285 010612 012737 000022 001226      TST22: MOV      #22,#TSTNO
2287 010620 012737 010666 001216      MOV      #TST23,NEXT
2288 010626 013703 001414          MOV      TXDBUF,R3          ;SET REGISTER TO BE TESTED.
2289 010632 012705 000020          MOV      #BIT4,R5          ;SET "EXPECTED"
2290 010636 010513          MOV      R5,(R3)          ;WRITE THE REGISTER.
2291 010640 011304          MOV      (R3),R4          ;READ THE REGISTER.
2292 010642 020504          CMP      R5,R4            ;R5=GOOD; R4=UNKNOWN.
2293 010644 001401          BEQ      1$              ;ARE THEY THE SAME?
2294 010646 104003          HLT      3                ;COMPARISON ERROR.
2295 010650 040513          1$:    BIC      R5,(R3)      ;CLEAR BIT4
2296 010652 011304          MOV      (R3),R4          ;READ THE REGISTER.
2297 010654 005005          CLR      R5              ;SET "EXPECTED"
2298 010656 020504          CMP      R5,R4            ;R5=GOOD; R4=?
2299 010660 001401          BEQ      2$              ;BR IF OK
2300 010662 104003          HLT      3                ;COMPARISON ERROR
2301 010664 104400          2$:    SCOPE              ;SCOPE THIS TEST

```

```

***** TEST 23 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BITS, VERIFY BITS WAS SET.
*CLEAR BITS, VERIFY BITS WAS CLEARED.
*****

```

```

*****
: TEST 23

```

2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312

```

2313
2314
2315
2316 010666 012737 000023 001226 TST23: MOV #23, R3
2317 010674 012737 010742 001216 MOV #TST24, NEXT
2318 010702 013703 001414 MOV TXDBUF, R3 ;SET REGISTER TO BE TESTED.
2319 010706 012705 000040 MOV #BIT5, R5 ;SET "EXPECTED"
2320 010712 010513 MOV R5, (R3) ;WRITE THE REGISTER.
2321 010714 011304 MOV (R3), R4 ;READ THE REGISTER.
2322 010716 020504 CMP R5, R4 ;R5=GOOD; R4=UNKNOWN.
2323 010720 001401 BEQ 1$ ;ARE THEY THE SAME?
2324 010722 104003 HLT 3 ;COMPARISON ERROR.
2325 010724 040513 1$: BIC R5, (R3) ;CLEAR BIT5
2326 010726 011304 MOV (R3), R4 ;READ THE REGISTER.
2327 010730 005005 CLR R5 ;SET "EXPECTED"
2328 010732 020504 CMP R5, R4 ;R5=GOOD; R4=?
2329 010734 001401 BEQ 2$ ;BR IF OK
2330 010736 104003 HLT 3 ;COMPARISON ERROR
2331 010740 104400 2$: SCOPE ;SCOPE THIS TEST
2332
2333

```

```

***** TEST 24 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BIT6, VERIFY BIT6 WAS SET.
*CLEAR BIT6, VERIFY BIT6 WAS CLEARED.
*****

```

```

2340
2341
2342
2343
2344
2345
2346 010742 012737 000024 001226 TST24: MOV #24, R3
2347 010750 012737 011016 001216 MOV #TST25, NEXT
2348 010756 013703 001414 MOV TXDBUF, R3 ;SET REGISTER TO BE TESTED.
2349 010762 012705 000100 MOV #BIT6, R5 ;SET "EXPECTED"
2350 010766 010513 MOV R5, (R3) ;WRITE THE REGISTER.
2351 010770 011304 MOV (R3), R4 ;READ THE REGISTER.
2352 010772 020504 CMP R5, R4 ;R5=GOOD; R4=UNKNOWN.
2353 010774 001401 BEQ 1$ ;ARE THEY THE SAME?
2354 010776 104003 HLT 3 ;COMPARISON ERROR.
2355 011000 040513 1$: BIC R5, (R3) ;CLEAR BIT6
2356 011002 011304 MOV (R3), R4 ;READ THE REGISTER.
2357 011004 005005 CLR R5 ;SET "EXPECTED"
2358 011006 020504 CMP R5, R4 ;R5=GOOD; R4=?
2359 011010 001401 BEQ 2$ ;BR IF OK
2360 011012 104003 HLT 3 ;COMPARISON ERROR
2361 011014 104400 2$: SCOPE ;SCOPE THIS TEST
2362
2363

```

```

***** TEST 25 *****
*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.
*SET BIT7, VERIFY BIT7 WAS SET.
*CLEAR BIT7, VERIFY BIT7 WAS CLEARED.
*****

```

```

2364
2365
2366
2367
2368

```

2369  
2370  
2371  
2372  
2373  
2374  
2375  
2376  
2377  
2378  
2379  
2380  
2381  
2382  
2383  
2384  
2385  
2386  
2387  
2388  
2389  
2390  
2391  
2392  
2393  
2394  
2395  
2396  
2397  
2398  
2399  
2400  
2401  
2402  
2403  
2404  
2405  
2406  
2407  
2408  
2409  
2410  
2411  
2412  
2413  
2414  
2415  
2416  
2417  
2418  
2419  
2420  
2421  
2422  
2423  
2424

011016 012737 000025 001226  
011024 012737 011072 001216  
011032 013703 001414  
011036 012705 000200  
011042 010513  
011044 011304  
011046 020504  
011050 001401  
011052 104003  
011054 040513  
011056 011304  
011060 005005  
011062 020504  
011064 001401  
011066 104003  
011070 104400

```
*****  
*  
: TEST 25  
*  
*****  
*****  
TST25: MOV #25, R2  
MOV #TST26, NEXT  
MOV TXDBUF, R3  
MOV #BIT7, R5  
MOV R5, (R3)  
MOV (R3), R4  
CMP R5, R4  
BEQ 1$  
HLT 3  
1$: BIC R5, (R3)  
MOV (R3), R4  
CLR R5  
CMP R5, R4  
BEQ 2$  
HLT 3  
2$: SCOPE  
*****  
;SET REGISTER TO BE TESTED.  
;SET "EXPECTED"  
;WRITE THE REGISTER.  
;READ THE REGISTER.  
;R5=GOOD; R4=UNKNOWN.  
;ARE THEY THE SAME?  
;COMPARISON ERROR.  
;CLEAR BIT7  
;READ THE REGISTER.  
;SET "EXPECTED"  
;R5=GOOD; R4=?  
;BR IF OK  
;COMPARISON ERROR  
;SCOPE THIS TEST
```

\*\*\*\*\* TEST 26 \*\*\*\*\*  
\*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.  
\*SET BIT8, VERIFY BIT8 WAS SET.  
\*CLEAR BIT8, VERIFY BIT8 WAS CLEARED.  
\*\*\*\*\*

```
*****  
*  
: TEST 26  
*  
*****  
*****  
TST26: MOV #26, R2  
MOV #TST27, NEXT  
MOV TXDBUF, R3  
MOV #BIT8, R5  
MOV R5, (R3)  
MOV (R3), R4  
CMP R5, R4  
BEQ 1$  
HLT 3  
1$: BIC R5, (R3)  
MOV (R3), R4  
CLR R5  
CMP R5, R4  
BEQ 2$  
HLT 3  
2$: SCOPE  
*****  
;SET REGISTER TO BE TESTED.  
;SET "EXPECTED"  
;WRITE THE REGISTER.  
;READ THE REGISTER.  
;R5=GOOD; R4=UNKNOWN.  
;ARE THEY THE SAME?  
;COMPARISON ERROR.  
;CLEAR BIT8  
;READ THE REGISTER.  
;SET "EXPECTED"  
;R5=GOOD; R4=?  
;BR IF OK  
;COMPARISON ERROR  
;SCOPE THIS TEST
```

\*\*\*\*\* TEST 27 \*\*\*\*\*

2455  
2456  
2457  
2458  
2459  
2460  
2461  
2462  
2463  
2464  
2465  
2466  
2467  
2468  
2469  
2470  
2471  
2472  
2473  
2474  
2475  
2476  
2477  
2478  
2479  
2480

:\*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.  
:\*SET BIT9, VERIFY BIT9 WAS SET.  
:\*CLEAR BIT9, VERIFY BIT9 WAS CLEARED.  
:\*\*\*\*\*

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

011146 012737 000027 001226  
011154 012737 011222 001216  
011162 013703 001414  
011166 012705 001000  
011172 010513  
011174 011304  
011176 020504  
011200 001401  
011202 104003  
011204 040513  
011206 011304  
011210 005005  
011212 020504  
011214 001401  
011216 104003  
011220 104400

TST27: MOV #27, #TSTNO  
MOV #TST30, NEXT  
MOV TXDBUF, R3 ; SET REGISTER TO BE TESTED.  
MOV #BIT9, R5 ; SET "EXPECTED"  
MOV R5, (R3) ; WRITE THE REGISTER.  
MOV (R3), R4 ; READ THE REGISTER.  
CMP R5, R4 ; R5=GOOD; R4=UNKNOWN.  
BEQ 1\$ ; ARE THEY THE SAME?  
HLT 3 ; COMPARISON ERROR.  
1\$: BIC R5, (R3) ; CLEAR BIT9  
MOV (R3), R4 ; READ THE REGISTER.  
CLR R5 ; SET "EXPECTED"  
CMP R5, R4 ; R5=GOOD; R4=?  
BEQ 2\$ ; BR IF OK  
HLT 3 ; COMPARISON ERROR  
2\$: SCOPE ; SCOPE THIS TEST

:\*\*\*\*\* TEST 30 \*\*\*\*\*  
:\*TRANSMITTER DATA BUFFER REGISTER READ/WRITE BIT TEST.  
:\*SET BIT10, VERIFY BIT10 WAS SET.  
:\*CLEAR BIT10, VERIFY BIT10 WAS CLEARED.  
:\*\*\*\*\*

:\*\*\*\*\*  
: TEST 30 \*  
:\*\*\*\*\*  
:\*\*\*\*\*

011222 012737 000030 001226  
011230 012737 011276 001216  
011236 013703 001414  
011242 012705 002000  
011246 010513  
011250 011304  
011252 020504  
011254 001401  
011256 104003  
011260 040513  
011262 011304  
011264 005005  
011266 020504  
011270 001401  
011272 104003

TST30: MOV #30, #TSTNO  
MOV #TST31, NEXT  
MOV TXDBUF, R3 ; SET REGISTER TO BE TESTED.  
MOV #BIT10, R5 ; SET "EXPECTED"  
MOV R5, (R3) ; WRITE THE REGISTER.  
MOV (R3), R4 ; READ THE REGISTER.  
CMP R5, R4 ; R5=GOOD; R4=UNKNOWN.  
BEQ 1\$ ; ARE THEY THE SAME?  
HLT 3 ; COMPARISON ERROR.  
1\$: BIC R5, (R3) ; CLEAR BIT10  
MOV (R3), R4 ; READ THE REGISTER.  
CLR R5 ; SET "EXPECTED"  
CMP R5, R4 ; R5=GOOD; R4=?  
BEQ 2\$ ; BR IF OK  
HLT 3 ; COMPARISON ERROR



2481  
2482  
2483  
2484  
2485  
2486  
2487  
2488  
2489  
2490  
2491  
2492  
2493  
2494  
2495  
2496  
2497  
2498  
2499  
2500  
2501  
2502  
2503  
2504  
2505  
2506  
2507  
2508  
2509  
2510  
2511  
2512  
2513  
2514  
2515  
2516  
2517  
2518  
2519  
2520  
2521  
2522  
2523  
2524  
2525  
2526  
2527  
2528  
2529  
2530  
2531  
2532  
2533  
2534  
2535  
2536

011274 104400

25: SCOPE ;SCOPE THIS TEST

\*\*\*\*\* TEST 31 \*\*\*\*\*  
\*RECEIVER CONTROL REGISTER READ/WRITE BIT 1 RESET AND CLEAR TEST  
\*WRITE BIT 1 AND TEST THAT IT WILL BE CLEARED AFTER A  
\*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION  
\*\*\*\*\*

\*\*\*\*\*  
TEST 31  
\*\*\*\*\*

011276 012737 000031 001226  
011304 012737 011372 001216  
011312 013703 001404  
011316 052713 000002  
011322 005005  
011324 052777 000400 170060  
011332 004737 004772  
011336 032713 000002  
011342 001402  
011344 011304  
011346 104003  
011350 052713 000002 15:  
011354 005013  
011356 032713 000002  
011362 001402  
011364 011304  
011366 104003  
011370 104400 25:  
SCOPE

\*\*\*\*\*  
TST31: MOV #31, @TSTNO  
MOV #TST32, NEXT  
MOV RXCSR, R3 ;GET THE RECEIVER CONTROL REGISTER  
BIS #BIT1, (R3) ;SET BIT 1 AT RECEIVER CONTROL REGISTER  
CLR R5 ;SET "EXPECTED"  
BIS #MRESET, @TXCSR ;RESET THE DEVICE  
JSR PC, SMALL ;WAIT FOR RESET TO FINISH  
BIT #BIT1, (R3) ;TEST BIT 1  
BEQ 15 ;BIT 1 IS CLEARED  
MOV (R3), R4 ;LOAD "FOUND"  
HLT 3 ;BIT 1 IS SET AND SHOULDN'T BE  
15: BIS #BIT1, (R3) ;SET BIT 1 AGAIN  
CLR (R3) ;CLEAR THE RECEIVER CONTROL REGISTER  
BIT #BIT1, (R3) ;TEST TO SEE IF BIT 1 CLEARED  
BEQ 25 ;BIT 1 IS OK  
MOV (R3), R4 ;LOAD "FOUND"  
HLT 3 ;BIT 1 FAILED TO CLEAR  
25: SCOPE ;SCOPE THIS TEST  
\*\*\*\*\*

\*\*\*\*\* TEST 32 \*\*\*\*\*  
\*RECEIVER CONTROL REGISTER READ/WRITE BIT 2 RESET AND CLEAR TEST  
\*WRITE BIT 2 AND TEST THAT IT WILL BE CLEARED AFTER A  
\*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION  
\*\*\*\*\*

\*\*\*\*\*  
TEST 32  
\*\*\*\*\*

011372 012737 000032 001226  
011400 012737 011466 001216  
011406 013703 001404  
011412 052713 000004  
011416 005005  
011420 052777 000400 167764  
011426 004737 004772  
011432 032713 000004  
011436 001402

\*\*\*\*\*  
TST32: MOV #32, @TSTNO  
MOV #TST33, NEXT  
MOV RXCSR, R3 ;GET THE RECEIVER CONTROL REGISTER  
BIS #BIT2, (R3) ;SET BIT 2 AT RECEIVER CONTROL REGISTER  
CLR R5 ;SET "EXPECTED"  
BIS #MRESET, @TXCSR ;RESET THE DEVICE  
JSR PC, SMALL ;WAIT FOR RESET TO FINISH  
BIT #BIT2, (R3) ;TEST BIT 2  
BEQ 15 ;BIT 2 IS CLEARED  
15: SCOPE ;SCOPE THIS TEST  
\*\*\*\*\*

```

2537 011440 011304      MOV      (R3),R4      ;LOAD "FOUND"
2538 011442 104003      HLT      3            ;BIT 2 IS SET AND SHOULDN'T BE
2539 011444 052713 000004 15:  BIS      #BIT2,(R3)   ;SET BIT 2 AGAIN
2540 011450 005013      CLR      (R3)        ;CLEAR THE RECEIVER CONTROL REGISTER
2541 011452 032713 000004  BIT      #BIT2,(R3)   ;TEST TO SEE IF BIT 2 CLEARED
2542 011456 001402      BEQ      25          ;BIT 2 IS OK
2543 011460 011304      MOV      (R3),R4      ;LOAD "FOUND"
2544 011462 104003      HLT      3            ;BIT 2 FAILED TO CLEAR
2545 011464 104400 25:  SCOPE          ;SCOPE THIS TEST

```

```

:***** TEST 33 *****
:RECEIVER CONTROL REGISTER READ/WRITE BIT 3 RESET AND CLEAR TEST
:WRITE BIT 3 AND TEST THAT IT WILL BE CLEARED AFTER A
:DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
:*****

```

```

:*****
:TEST 33
:*****

```

```

2559 *****
2560 011466 012737 000033 001226 TST33: MOV      #33,#TSTNO
2561 011474 012737 011562 001216      MOV      #TST34,NEXT
2562 011502 C13703 001404      MOV      RXCSR,R3    ;GET THE RECEIVER CONTROL REGISTER
2563 011506 052713 000010  BIS      #BIT3,(R3)   ;SET BIT 3 AT RECEIVER CONTROL REGISTER
2564 011512 005005      CLR      R5          ;SET "EXPECTED"
2565 011514 052777 000400 167670  BIS      #MRESET,#TXCSR ;RESET THE DEVICE
2566 011522 004737 004772      JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
2567 011526 032713 000010  BIT      #BIT3,(R3)   ;TEST BIT 3
2568 011532 001402      BEQ      15          ;BIT 3 IS CLEARED
2569 011534 011304      MOV      (R3),R4      ;LOAD "FOUND"
2570 011536 104003      HLT      3            ;BIT 3 IS SET AND SHOULDN'T BE
2571 011540 052713 000010 15:  BIS      #BIT3,(R3)   ;SET BIT 3 AGAIN
2572 011544 005013      CLR      (R3)        ;CLEAR THE RECEIVER CONTROL REGISTER
2573 011546 032713 000010  BIT      #BIT3,(R3)   ;TEST TO SEE IF BIT 3 CLEARED
2574 011552 001402      BEQ      25          ;BIT 3 IS OK
2575 011554 011304      MOV      (R3),R4      ;LOAD "FOUND"
2576 011556 104003      HLT      3            ;BIT 3 FAILED TO CLEAR
2577 011560 104400 25:  SCOPE          ;SCOPE THIS TEST

```

```

:***** TEST 34 *****
:RECEIVER CONTROL REGISTER READ/WRITE BIT 4 RESET AND CLEAR TEST
:WRITE BIT 4 AND TEST THAT IT WILL BE CLEARED AFTER A
:DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
:*****

```

```

:*****
:TEST 34
:*****

```

```

2592 011562 012737 000034 001226 TST34: MOV      #34,#TSTNO

```

```

2593 011570 012737 011656 001216      MOV      #TST35,NEXT
2594 011576 013703 001404      MOV      RXCSR,R3          ;GET THE RECEIVER CONTROL REGISTER
2595 011602 052713 000020      BIS      #BIT4,(R3)        ;SET BIT 4 AT RECEIVER CONTROL REGISTER
2596 011606 005005                CLR      R5                ;SET "EXPECTED"
2597 011610 052777 000400 167574      BIS      #MRESET,@TXCSR   ;RESET THE DEVICE
2598 011616 004737 004772                JSR      PC_SMALL          ;WAIT FOR RESET TO FINISH
2599 011622 032713 000020      BIT      #BIT4,(R3)        ;TEST BIT 4
2600 011626 001402                BEQ      1$                ;BIT 4 IS CLEARED
2601 011630 011304                MOV      (R3),R4           ;LOAD "FOUND"
2602 011632 104003                HLT      3                  ;BIT 4 IS SET AND SHOULDN'T BE
2603 011634 052713 000020 1$:      BIS      #BIT4,(R3)        ;SET BIT 4 AGAIN
2604 011640 005013                CLR      (R3)              ;CLEAR THE RECEIVER CONTROL REGISTER
2605 011642 032713 000020      BIT      #BIT4,(R3)        ;TEST TO SEE IF BIT 4 CLEARED
2606 011646 001402                BEQ      2$                ;BIT 4 IS OK
2607 011650 011304                MOV      (R3),R4           ;LOAD "FOUND"
2608 011652 104003                HLT      3                  ;BIT 4 FAILED TO CLEAR
2609 011654 104400 2$:      SCOPE                    ;SCOPE THIS TEST
2610
2611
2612
2613
2614
2615
2616
2617
2618
2619
2620
2621
2622
2623

```

```

:***** TEST 35 *****
:*RECEIVER CONTROL REGISTER READ/WRITE BIT 8 RESET AND CLEAR TEST
:*WRITE BIT 8 AND TEST THAT IT WILL BE CLEARED AFTER A
:*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
:*****

```

```

:*****
:
: TEST 35
:
:*****

```

```

2624 011656 012737 000035 001226  TST35: MOV      #35,@TSTNO
2625 011664 012737 011752 001216      MOV      #TST36,NEXT
2626 011672 013703 001404      MOV      RXCSR,R3          ;GET THE RECEIVER CONTROL REGISTER
2627 011676 052713 000400      BIS      #BIT8,(R3)        ;SET BIT 8 AT RECEIVER CONTROL REGISTER
2628 011702 005005                CLR      R5                ;SET "EXPECTED"
2629 011704 052777 000400 167500      BIS      #MRESET,@TXCSR   ;RESET THE DEVICE
2630 011712 004737 004772                JSR      PC_SMALL          ;WAIT FOR RESET TO FINISH
2631 011716 032713 000400      BIT      #BIT8,(R3)        ;TEST BIT 8
2632 011722 001402                BEQ      1$                ;BIT 8 IS CLEARED
2633 011724 011304                MOV      (R3),R4           ;LOAD "FOUND"
2634 011726 104003                HLT      3                  ;BIT 8 IS SET AND SHOULDN'T BE
2635 011730 052713 000400 1$:      BIS      #BIT8,(R3)        ;SET BIT 8 AGAIN
2636 011734 005013                CLR      (R3)              ;CLEAR THE RECEIVER CONTROL REGISTER
2637 011736 032713 000400      BIT      #BIT8,(R3)        ;TEST TO SEE IF BIT 8 CLEARED
2638 011742 001402                BEQ      2$                ;BIT 8 IS OK
2639 011744 011304                MOV      (R3),R4           ;LOAD "FOUND"
2640 011746 104003                HLT      3                  ;BIT 8 FAILED TO CLEAR
2641 011750 104400 2$:      SCOPE                    ;SCOPE THIS TEST
2642
2643
2644
2645
2646
2647
2648

```

```

:***** TEST 36 *****
:*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 3 RESET AND CLEAR TEST
:*WRITE BIT 3 AND TEST THAT IT WILL BE CLEARED AFTER A
:*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
:*****

```

2649  
2650  
2651  
2652  
2653  
2654  
2655  
2656  
2657  
2658  
2659  
2660  
2661  
2662  
2663  
2664  
2665  
2666  
2667  
2668  
2669  
2670  
2671  
2672  
2673  
2674  
2675  
2676  
2677  
2678  
2679  
2680  
2681  
2682  
2683  
2684  
2685  
2686  
2687  
2688  
2689  
2690  
2691  
2692  
2693  
2694  
2695  
2696  
2697  
2698  
2699  
2700  
2701  
2702  
2703  
2704

011752 012737 000036 001226  
011760 012737 012046 001216  
011766 013703 001412  
011772 052713 000010  
011776 005005  
012000 052777 000400 167404  
012006 004737 004772  
012012 032713 000010  
012016 001402  
012020 011304  
012022 104003  
012024 052713 000010  
012030 005013  
012032 032713 000010  
012036 001402  
012040 011304  
012042 104003  
012044 104400

```

:*****
: TEST 36
:*****
:*****
TST36: MOV #36, @TSTNO
MOV #TST37, NEXT
MOV TXCSR, R3 ;GET THE TRANSMITTER CONTROL REGISTER
BIS #BIT3, (R3) ;SET BIT 3 AT TRANSMITTER CONTROL REGISTER
CLR R5 ;SET "EXPECTED"
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
BIT #BIT3, (R3) ;TEST BIT 3
BEQ 1$ ;BIT 3 IS CLEARED
MOV (R3), R4 ;LOAD "FOUND"
HLT 3 ;BIT 3 IS SET AND SHOULDN'T BE
1$: BIS #BIT3, (R3) ;SET BIT 3 AGAIN
CLR (R3) ;CLEAR THE TRANSMITTER CONTROL REGISTER
BIT #BIT3, (R3) ;TEST TO SEE IF BIT 3 CLEARED
BEQ 2$ ;BIT 3 IS OK
MOV (R3), R4 ;LOAD "FOUND"
HLT 3 ;BIT 3 FAILED TO CLEAR
2$: SCOPE ;SCOPE THIS TEST

```

\*\*\*\*\* TEST 37 \*\*\*\*\*  
\*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 4 RESET AND CLEAR TEST  
\*WRITE BIT 4, AND TEST THAT IT WILL BE CLEARED AFTER A  
\*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION  
\*\*\*\*\*

```

:*****
: TEST 37
:*****
:*****
TST37: MOV #37, @TSTNO
MOV #TST40, NEXT
MOV TXCSR, R3 ;GET THE TRANSMITTER CONTROL REGISTER
BIS #BIT4, (R3) ;SET BIT 4 AT TRANSMITTER CONTROL REGISTER
CLR R5 ;SET "EXPECTED"
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
BIT #BIT4, (R3) ;TEST BIT 4
BEQ 1$ ;BIT 4 IS CLEARED
MOV (R3), R4 ;LOAD "FOUND"
HLT 3 ;BIT 4 IS SET AND SHOULDN'T BE
1$: BIS #BIT4, (R3) ;SET BIT 4 AGAIN
CLR (R3) ;CLEAR THE TRANSMITTER CONTROL REGISTER
BIT #BIT4, (R3) ;TEST TO SEE IF BIT 4 CLEARED
BEQ 2$ ;BIT 4 IS OK
MOV (R3), R4 ;LOAD "FOUND"
HLT 3 ;BIT 4 FAILED TO CLEAR

```

2705 012140 104400 2S: SCOPE ;SCOPE THIS TEST

2706  
2707  
2708  
2709  
2710  
2711  
2712  
2713  
2714  
2715  
2716  
2717  
2718  
2719  
2720  
2721  
2722  
2723  
2724  
2725  
2726  
2727  
2728  
2729  
2730  
2731  
2732  
2733  
2734  
2735  
2736  
2737  
2738  
2739  
2740  
2741  
2742  
2743  
2744  
2745  
2746  
2747  
2748  
2749  
2750  
2751  
2752  
2753  
2754  
2755  
2756  
2757  
2758  
2759  
2760

012142 012737 000040 001226  
012150 012737 012236 001216  
012156 013703 001412  
012162 052713 002000  
012166 005005  
012170 052777 000400 167214  
012176 004737 004772  
012202 032713 002000  
012206 001402  
012210 011304  
012212 104003  
012214 052713 002000 1S:  
012220 005013  
012222 032713 002000  
012226 001402  
012230 011304  
012232 104003  
012234 104400 2S:

\*\*\*\*\* TEST 40 \*\*\*\*\*  
\*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 10 RESET AND CLEAR TEST  
\*WRITE BIT 10, AND TEST THAT IT WILL BE CLEARED AFTER A  
\*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 40  
\*  
\*\*\*\*\*  
\*\*\*\*\*  
TST40: MOV #40, @TSTNO  
MOV #TST41, NEXT  
MOV TXCSR, R3 ;GET THE TRANSMITTER CONTROL REGISTER  
BIS #BIT10, (R3) ;SET BIT 10 AT TRANSMITTER CONTROL REGISTER  
CLR R5 ;SET "EXPECTED"  
BIS #MRESET, @TXCSR ;RESET THE DEVICE  
JSR PC, SMALL ;WAIT FOR RESET TO FINISH  
BIT #BIT10, (R3) ;TEST BIT 10  
BEQ 1S ;BIT 10 IS CLEARED  
MOV (R3), R4 ;LOAD "FOUND"  
HLT 3 ;BIT 10 IS SET AND SHOULDN'T BE  
1S: BIS #BIT10, (R3) ;SET BIT 10 AGAIN  
CLR (R3) ;CLEAR THE TRANSMITTER CONTROL REGISTER  
BIT #BIT10, (R3) ;TEST TO SEE IF BIT 10 CLEARED  
BEQ 2S ;BIT 10 IS OK  
MOV (R3), R4 ;LOAD "FOUND"  
HLT 3 ;BIT 10 FAILED TO CLEAR  
2S: SCOPE ;SCOPE THIS TEST

\*\*\*\*\* TEST 41 \*\*\*\*\*  
\*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 11 RESET AND CLEAR TEST  
\*WRITE BIT 11, AND TEST THAT IT WILL BE CLEARED AFTER A  
\*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 41  
\*  
\*\*\*\*\*  
\*\*\*\*\*  
TST41: MOV #41, @TSTNO  
MOV #TST42, NEXT  
MOV TXCSR, R3 ;GET THE TRANSMITTER CONTROL REGISTER  
BIS #BIT11, (R3) ;SET BIT 11 AT TRANSMITTER CONTROL REGISTER  
CLR R5 ;SET "EXPECTED"  
BIS #MRESET, @TXCSR ;RESET THE DEVICE  
JSR PC, SMALL ;WAIT FOR RESET TO FINISH  
BIT #BIT11, (R3) ;TEST BIT 11  
BEQ 1S ;BIT 11 IS CLEARED

```

2761 012304 011304          MOV      (R3),R4          ;LOAD "FOUND"
2762 012306 104003          HLT      3                ;BIT 11 IS SET AND SHOULDN'T BE
2763 012310 052713 004000    15:     BIS      #BIT11,(R3) ;SET BIT 11 AGAIN
2764 012314 005013          CLR      (R3)            ;CLEAR THE TRANSMITTER CONTROL REGISTER
2765 012316 032713 004000    BIT      #BIT11,(R3)     ;TEST TO SEE IF BIT 11 CLEARED
2766 012322 001402          BEQ     25                ;BIT 11 IS OK
2767 012324 011304          MOV      (R3),R4          ;LOAD "FOUND"
2768 012326 104003          HLT      3                ;BIT 11 FAILED TO CLEAR
2769 012330 104400    25:     SCOPE           ;SCOPE THIS TEST

```

```

2770
2771
2772
2773
2774
2775
2776
2777
2778
2779
2780
2781
2782
2783
2784
2785
2786
2787
2788
2789
2790
2791
2792
2793
2794
2795
2796
2797
2798
2799
2800
2801
2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816

```

```

:***** TEST 42 *****
:*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 12 RESET AND CLEAR TEST
:*WRITE BIT 12,AND TEST THAT IT WILL BE CLEARED AFTER A
:*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
:*****

```

```

:*****
:TEST 42
:*****
:*****

```

```

2784 012332 012737 000042 001226 TST42: MOV      #42,#TSTNO
2785 012340 012737 012426 001216    MOV      #TST43,NEXT
2786 012346 013703 001412          MOV      TXCSR,R3        ;GET THE TRANSMITTER CONTROL REGISTER
2787 012352 052713 010000    BIS      #BIT12,(R3)     ;SET BIT 12 AT TRANSMITTER CONTROL REGISTER
2788 012356 005005          CLR      R5              ;SET "EXPECTED"
2789 012360 052777 000400 167024    BIS      #MRESET,#TXCSR ;RESET THE DEVICE
2790 012366 004737 004772          JSR     PC,SMALL        ;WAIT FOR RESET TO FINISH
2791 012372 032713 010000    BIT      #BIT12,(R3)     ;TEST BIT 12
2792 012376 001402          BEQ     15                ;BIT 12 IS CLEARED
2793 012400 011304          MOV      (R3),R4          ;LOAD "FOUND"
2794 012402 104003          HLT      3                ;BIT 12 IS SET AND SHOULDN'T BE
2795 012404 052713 010000    15:     BIS      #BIT12,(R3) ;SET BIT 12 AGAIN
2796 012410 005013          CLR      (R3)            ;CLEAR THE TRANSMITTER CONTROL REGISTER
2797 012412 032713 010000    BIT      #BIT12,(R3)     ;TEST TO SEE IF BIT 12 CLEARED
2798 012416 001402          BEQ     25                ;BIT 12 IS OK
2799 012420 011304          MOV      (R3),R4          ;LOAD "FOUND"
2800 012422 104003          HLT      3                ;BIT 12 FAILED TO CLEAR
2801 012424 104400    25:     SCOPE           ;SCOPE THIS TEST

```

```

2802
2803
2804
2805
2806
2807
2808
2809
2810
2811
2812
2813
2814
2815
2816

```

```

:***** TEST 43 *****
:*TRANSMITTER CONTROL REGISTER READ/WRITE BIT 13 RESET AND CLEAR TEST
:*WRITE BIT 13,AND TEST THAT IT WILL BE CLEARED AFTER A
:*DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
:*****

```

```

:*****
:TEST 43
:*****
:*****

```

```

2816 012426 012737 000043 001226 TST43: MOV      #43,#TSTNO

```

# K05

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 63  
DZDPBA.CMB TRANSMITTER CONTROL REGISTER RESET AND CLEAR TEST BIT 13

```
2817 012434 012737 012522 001216      MOV      #TST44,NEXT
2818 012442 013703 001412      MOV      TXCSR,R3          ;GET THE TRANSMITTER CONTROL REGISTER
2819 012446 052713 020000      BIS      #BIT13,(R3)      ;SET BIT 13 AT TRANSMITTER CONTROL REGISTER
2820 012452 005005          CLR      R5              ;SET "EXPECTED"
2821 012454 052777 000400 166730      BIS      #MRESET,@TXCSR  ;RESET THE DEVICE
2822 012462 004737 004772      JSR      PC,SMALL        ;WAIT FOR RESET TO FINISH
2823 012466 032713 020000      BIT      #BIT13,(R3)     ;TEST BIT 13
2824 012472 001402          BEQ      1$              ;BIT 13 IS CLEARED
2825 012474 011304          MOV      (R3),R4        ;LOAD "FOUND"
2826 012476 104003          HLT      3              ;BIT 13 IS SET AND SHOULDN'T BE
2827 012500 052713 020000 1$:      BIS      #BIT13,(R3)     ;SET BIT 13 AGAIN
2828 012504 005013          CLR      (R3)          ;CLEAR THE TRANSMITTER CONTROL REGISTER
2829 012506 032713 020000      BIT      #BIT13,(R3)     ;TEST TO SEE IF BIT 13 CLEARED
2830 012512 001402          BEQ      2$              ;BIT 13 IS OK
2831 012514 011304          MOV      (R3),R4        ;LOAD "FOUND"
2832 012516 104003          HLT      3              ;BIT 13 FAILED TO CLEAR
2833 012520 104400 2$:      SCOPE                    ;SCOPE THIS TEST
2834
2835
2836
2837
2838
2839
2840
2841
2842
2843
2844
2845
2846
2847
```

```
***** TEST 44 *****
*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 0 RESET AND CLEAR TEST
*WRITE BIT 0 AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
*****
```

```
2848 012522 012737 000044 001226 1$:      MOV      #44,@TSTNO
2849 012530 012737 012616 001216      MOV      #TST45,NEXT
2850 012536 013703 001414      MOV      TXDBUF,R3       ;GET THE TRANSMITTER BUFFER REGISTER
2851 012542 052713 000001      BIS      #BIT0,(R3)     ;SET BIT 0 AT TRANSMITTER BUFFER REGISTER
2852 012546 005005          CLR      R5              ;SET "EXPECTED"
2853 012550 052777 000400 166634      BIS      #MRESET,@TXCSR  ;RESET THE DEVICE
2854 012556 004737 004772      JSR      PC,SMALL        ;WAIT FOR RESET TO FINISH
2855 012562 032713 000001      BIT      #BIT0,(R3)     ;TEST BIT 0
2856 012566 001402          BEQ      1$              ;BIT 0 IS CLEARED
2857 012570 011304          MOV      (R3),R4        ;LOAD "FOUND"
2858 012572 104003          HLT      3              ;BIT 0 IS SET AND SHOULDN'T BE
2859 012574 052713 000001 1$:      BIS      #BIT0,(R3)     ;SET BIT 0 AGAIN
2860 012600 005013          CLR      (R3)          ;CLEAR THE TRANSMITTER BUFFER REGISTER
2861 012602 032713 000001      BIT      #BIT0,(R3)     ;TEST TO SEE IF BIT 0 CLEARED
2862 012606 001402          BEQ      2$              ;BIT 0 IS OK
2863 012610 011304          MOV      (R3),R4        ;LOAD "FOUND"
2864 012612 104003          HLT      3              ;BIT 0 FAILED TO CLEAR
2865 012614 104400 2$:      SCOPE                    ;SCOPE THIS TEST
2866
2867
2868
2869
2870
2871
2872
```

```
***** TEST 45 *****
*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 1 RESET AND CLEAR TEST
*WRITE BIT 1 AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
*****
```

```

2873
2874
2875
2876
2877
2878
2879
2880 012616 012737 000045 001226 TST45: MOV #45,@TSTNO
2881 012624 012737 012712 001216 MOV #TST46,NEXT
2882 012632 013703 001414 MOV TXDBUF,R3 ;GET THE TRANSMITTER BUFFER REGISTER
2883 012636 052713 000002 BIS #BIT1,(R3) ;SET BIT 1 AT TRANSMITTER BUFFER REGISTER
2884 012642 005005 CLR R5 ;SET "EXPECTED"
2885 012644 052777 000400 166540 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2886 012652 004737 004772 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2887 012656 032713 000002 BIT #BIT1,(R3) ;TEST BIT 1
2888 012662 001402 BEQ 1$ ;BIT 1 IS CLEARED
2889 012664 011304 MOV (R3),R4 ;LOAD "FOUND"
2890 012666 104003 HLT 3 ;BIT 1 IS SET AND SHOULDN'T BE
2891 012670 052713 000002 1$: BIS #BIT1,(R3) ;SET BIT 1 AGAIN
2892 012674 005013 CLR (R3) ;CLEAR THE TRANSMITTER BUFFER REGISTER
2893 012676 032713 000002 BIT #BIT1,(R3) ;TEST TO SEE IF BIT 1 CLEARED
2894 012702 001402 BEQ 2$ ;BIT 1 IS OK
2895 012704 011304 MOV (R3),R4 ;LOAD "FOUND"
2896 012706 104003 HLT 3 ;BIT 1 FAILED TO CLEAR
2897 012710 104400 2$: SCOPE ;SCOPE THIS TEST
2898
2899
2900
2901
2902
2903
2904
2905
2906
2907
2908
2909
2910
2911
2912
2913
2914
2915
2916
2917
2918
2919
2920
2921
2922
2923
2924
2925
2926
2927
2928

```

```

***** TEST 46 *****
*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 2 RESET AND CLEAR TEST
*WRITE BIT 2 AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
*****

```

```

2906
2907
2908
2909
2910
2911
2912 012712 012737 000046 001226 TST46: MOV #46,@TSTNO
2913 012720 012737 013006 001216 MOV #TST47,NEXT
2914 012726 013703 001414 MOV TXDBUF,R3 ;GET THE TRANSMITTER BUFFER REGISTER
2915 012732 052713 000004 BIS #BIT2,(R3) ;SET BIT 2 AT TRANSMITTER BUFFER REGISTER
2916 012736 005005 CLR R5 ;SET "EXPECTED"
2917 012740 052777 000400 166444 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2918 012746 004737 004772 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2919 012752 032713 000004 BIT #BIT2,(R3) ;TEST BIT 2
2920 012756 001402 BEQ 1$ ;BIT 2 IS CLEARED
2921 012760 011304 MOV (R3),R4 ;LOAD "FOUND"
2922 012762 104003 HLT 3 ;BIT 2 IS SET AND SHOULDN'T BE
2923 012764 052713 000004 1$: BIS #BIT2,(R3) ;SET BIT 2 AGAIN
2924 012770 005013 CLR (R3) ;CLEAR THE TRANSMITTER BUFFER REGISTER
2925 012772 032713 000004 BIT #BIT2,(R3) ;TEST TO SEE IF BIT 2 CLEARED
2926 012776 001402 BEQ 2$ ;BIT 2 IS OK
2927 013000 011304 MOV (R3),R4 ;LOAD "FOUND"
2928 013002 104003 HLT 3 ;BIT 2 FAILED TO CLEAR

```



2929 013004 104400 2\$: SCOPE ;SCOPE THIS TEST

2930  
2931  
2932  
2933  
2934  
2935  
2936  
2937  
2938  
2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953  
2954  
2955  
2956  
2957  
2958  
2959  
2960  
2961  
2962  
2963  
2964  
2965  
2966  
2967  
2968  
2969  
2970  
2971  
2972  
2973  
2974  
2975  
2976  
2977  
2978  
2979  
2980  
2981  
2982  
2983  
2984

013006 012737 000047 001226  
013014 012737 013102 001216  
013022 013703 001414  
013026 052713 000010  
013032 005005  
013034 052777 000400 166350  
013042 004737 004772  
013046 032713 000010  
013052 001402  
013054 011304  
013056 104003  
013060 052713 000010 1\$:  
013064 005013  
013066 032713 000010  
013072 001402  
013074 011304  
013076 104003  
013100 104400 2\$:

\*\*\*\*\* TEST 47 \*\*\*\*\*  
\*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 3 RESET AND CLEAR TEST  
\*WRITE BIT 3 AND TEST THAT IT WILL BE CLEARED AFTER A  
\*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 47  
\*  
\*\*\*\*\*  
\*\*\*\*\*  
TST47: MOV #47, @TSTNO ;GET THE TRANSMITTER BUFFER REGISTER  
MOV #TST50, NEXT ;SET BIT 3 AT TRANSMITTER BUFFER REGISTER  
MOV TXDBUF, R3 ;SET "EXPECTED"  
BIS #BIT3, (R3) ;RESET THE DEVICE  
CLR R5 ;WAIT FOR RESET TO FINISH  
BIS #MRESET, @TXCSR ;TEST BIT 3  
JSR PC, SMALL ;BIT 3 IS CLEARED  
BIT #BIT3, (R3) ;LOAD "FOUND"  
BEQ 1\$ ;BIT 3 IS SET AND SHOULDN'T BE  
MOV (R3), R4 ;SET BIT 3 AGAIN  
HLT 3 ;CLEAR THE TRANSMITTER BUFFER REGISTER  
1\$: BIS #BIT3, (R3) ;TEST TO SEE IF BIT 3 CLEARED  
CLR (R3) ;BIT 3 IS OK  
BIT #BIT3, (R3) ;LOAD "FOUND"  
BEQ 2\$ ;BIT 3 FAILED TO CLEAR  
MOV (R3), R4 ;SCOPE THIS TEST  
HLT 3  
2\$: SCOPE

\*\*\*\*\* TEST 50 \*\*\*\*\*  
\*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 4 RESET AND CLEAR TEST  
\*WRITE BIT 4 AND TEST THAT IT WILL BE CLEARED AFTER A  
\*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 50  
\*  
\*\*\*\*\*  
\*\*\*\*\*  
TST50: MOV #50, @TSTNO ;GET THE TRANSMITTER BUFFER REGISTER  
MOV #TST51, NEXT ;SET BIT 4 AT TRANSMITTER BUFFER REGISTER  
MOV TXDBUF, R3 ;SET "EXPECTED"  
BIS #BIT4, (R3) ;RESET THE DEVICE  
CLR R5 ;WAIT FOR RESET TO FINISH  
BIS #MRESET, @TXCSR ;TEST BIT 4  
JSR PC, SMALL ;BIT 4 IS CLEARED  
BIT #BIT4, (R3)  
BEQ 1\$

```

2985 013150 011304      MOV      (R3),R4      ;LOAD "FOUND"
2986 013152 104003      HLT      3            ;BIT 4 IS SET AND SHOULDN'T BE
2987 013154 052713 000020 1$:      BIS      #BIT4,(R3)  ;SET BIT 4 AGAIN
2988 013160 005013      CLR      (R3)        ;CLEAR THE TRANSMITTER BUFFER REGISTER
2989 013162 032713 000020      BIT      #BIT4,(R3)  ;TEST TO SEE IF BIT 4 CLEARED
2990 013166 001402      BEQ      2$          ;BIT 4 IS OK
2991 013170 011304      MOV      (R3),R4      ;LOAD "FOUND"
2992 013172 104003      HLT      3            ;BIT 4 FAILED TO CLEAR
2993 013174 104400 2$:      SCOPE              ;SCOPE THIS TEST

```

```

2994
2995
2996      ;***** TEST 51 *****
2997      ;*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 5 RESET AND CLEAR TEST
2998      ;*WRITE BIT 5, AND TEST THAT IT WILL BE CLEARED AFTER A
2999      ;*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
3000      ;*****

```

```

3001
3002      ;*****
3003      ;*
3004      ;* TEST 51
3005      ;*
3006      ;*****

```

```

3007
3008 013176 012737 000051 001226 TST51: MOV      #51,@TSTNO
3009 013204 012737 013272 001216      MOV      #TST52,NEXT
3010 013212 013703 001414      MOV      TXDBUF,R3    ;GET THE TRANSMITTER BUFFER REGISTER
3011 013216 052713 000040      BIS      #BITS,(R3)  ;SET BIT 5 AT TRANSMITTER BUFFER REGISTER
3012 013222 005005      CLR      R5          ;SET "EXPECTED"
3013 013224 052777 000400 166160      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3014 013232 004737 004772      JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
3015 013236 032713 000040      BIT      #BITS,(R3)  ;TEST BIT 5
3016 013242 001402      BEQ      1$          ;BIT 5 IS CLEARED
3017 013244 011304      MOV      (R3),R4      ;LOAD "FOUND"
3018 013246 104003      HLT      3            ;BIT 5 IS SET AND SHOULDN'T BE
3019 013250 052713 000040 1$:      BIS      #BITS,(R3)  ;SET BIT 5 AGAIN
3020 013254 005013      CLR      (R3)        ;CLEAR THE TRANSMITTER BUFFER REGISTER
3021 013256 032713 000040      BIT      #BITS,(R3)  ;TEST TO SEE IF BIT 5 CLEARED
3022 013262 001402      BEQ      2$          ;BIT 5 IS OK
3023 013264 011304      MOV      (R3),R4      ;LOAD "FOUND"
3024 013266 104003      HLT      3            ;BIT 5 FAILED TO CLEAR
3025 013270 104400 2$:      SCOPE              ;SCOPE THIS TEST

```

```

3026
3027
3028      ;***** TEST 52 *****
3029      ;*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 6 RESET AND CLEAR TEST
3030      ;*WRITE BIT 6, AND TEST THAT IT WILL BE CLEARED AFTER A
3031      ;*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
3032      ;*****

```

```

3033      ;*****
3034      ;*
3035      ;* TEST 52
3036      ;*
3037      ;*****

```

```

3038
3039
3040 013272 012737 000052 001226 TST52: MOV      #52,@TSTNO

```

```

3041 013300 012737 013366 001216      MOV      #TST53,NEXT
3042 013306 013703 001414      MOV      TXDBUF,R3          ;GET THE TRANSMITTER BUFFER REGISTER
3043 013312 052713 000100      BIS      #BIT6,(R3)        ;SET BIT 6 AT TRANSMITTER BUFFER REGISTER
3044 013316 005005          CLR      R5                ;SET "EXPECTED"
3045 013320 052777 000400 166064      BIS      #MRESET,@TXCSR   ;RESET THE DEVICE
3046 013326 004737 004772      JSR      PC,SMALL         ;WAIT FOR RESET TO FINISH
3047 013332 032713 000100      BIT      #BIT6,(R3)       ;TEST BIT 6
3048 013336 001402          BEQ      1$               ;BIT 6 IS CLEARED
3049 013340 011304          MOV      (R3),R4         ;LOAD "FOUND"
3050 013342 104003          HLT      3                ;BIT 6 IS SET AND SHOULDN'T BE
3051 013344 052713 000100 1$:      BIS      #BIT6,(R3)       ;SET BIT 6 AGAIN
3052 013350 005013          CLR      (R3)           ;CLEAR THE TRANSMITTER BUFFER REGISTER
3053 013352 032713 000100      BIT      #BIT6,(R3)       ;TEST TO SEE IF BIT 6 CLEARED
3054 013356 001402          BEQ      2$               ;BIT 6 IS OK
3055 013360 011304          MOV      (R3),R4         ;LOAD "FOUND"
3056 013362 104003          HLT      3                ;BIT 6 FAILED TO CLEAR
3057 013364 104400 2$:      SCOPE                    ;SCOPE THIS TEST

```

```

:***** TEST 53 *****
:TRANSMITTER BUFFER REGISTER READ/WRITE BIT 7 RESET AND CLEAR TEST
:WRITE BIT 7, AND TEST THAT IT WILL BE CLEARED AFTER A
:DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
:*****

```

```

:*****
:TEST 53
:*****

```

```

3071 *****
3072 013366 012737 000053 001226  TST53: MOV      #53,@TSTNO
3073 013374 012737 013462 001216      MOV      #TST54,NEXT
3074 013402 013703 001414      MOV      TXDBUF,R3          ;GET THE TRANSMITTER BUFFER REGISTER
3075 013406 052713 000200      BIS      #BIT7,(R3)        ;SET BIT 7 AT TRANSMITTER BUFFER REGISTER
3076 013412 005005          CLR      R5                ;SET "EXPECTED"
3077 013414 052777 000400 165770      BIS      #MRESET,@TXCSR   ;RESET THE DEVICE
3078 013422 004737 004772      JSR      PC,SMALL         ;WAIT FOR RESET TO FINISH
3079 013426 032713 000200      BIT      #BIT7,(R3)       ;TEST BIT 7
3080 013432 001402          BEQ      1$               ;BIT 7 IS CLEARED
3081 013434 011304          MOV      (R3),R4         ;LOAD "FOUND"
3082 013436 104003          HLT      3                ;BIT 7 IS SET AND SHOULDN'T BE
3083 013440 052713 000200 1$:      BIS      #BIT7,(R3)       ;SET BIT 7 AGAIN
3084 013444 005013          CLR      (R3)           ;CLEAR THE TRANSMITTER BUFFER REGISTER
3085 013446 032713 000200      BIT      #BIT7,(R3)       ;TEST TO SEE IF BIT 7 CLEARED
3086 013452 001402          BEQ      2$               ;BIT 7 IS OK
3087 013454 011304          MOV      (R3),R4         ;LOAD "FOUND"
3088 013456 104003          HLT      3                ;BIT 7 FAILED TO CLEAR
3089 013460 104400 2$:      SCOPE                    ;SCOPE THIS TEST

```

```

:***** TEST 54 *****
:TRANSMITTER BUFFER REGISTER READ/WRITE BIT 8 RESET AND CLEAR TEST
:WRITE BIT 8, AND TEST THAT IT WILL BE CLEARED AFTER A
:DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION
:*****

```

```

3090
3091
3092
3093
3094
3095
3096

```

3097  
3098  
3099  
3100  
3101  
3102  
3103  
3104  
3105  
3106  
3107  
3108  
3109  
3110  
3111  
3112  
3113  
3114  
3115  
3116  
3117  
3118  
3119  
3120  
3121  
3122  
3123  
3124  
3125  
3126  
3127  
3128  
3129  
3130  
3131  
3132  
3133  
3134  
3135  
3136  
3137  
3138  
3139  
3140  
3141  
3142  
3143  
3144  
3145  
3146  
3147  
3148  
3149  
3150  
3151  
3152

013462 012737 000054 001226  
013470 012737 013556 001216  
013476 013703 001414  
013502 052713 000400  
013506 005005  
013510 052777 000400 165674  
013516 004737 004772  
013522 032713 000400  
013526 001402  
013530 011304  
013532 104003  
013534 052713 000400  
013540 005013  
013542 032713 000400  
013546 001402  
013550 011304  
013552 104003  
013554 104400

```

:*****
: TEST 54
:*****
:*****
TST54: MOV #54, @TSTNO
MOV @TST55, NEXT
MOV TXDBUF, R3
BIS #BIT8, (R3)
CLR R5
BIS #MRESET, @TXCSR
JSR PC, SMALL
BIT #BIT8, (R3)
BEQ 1$
MOV (R3), R4
HLT 3
1$: BIS #BIT8, (R3)
CLR (R3)
BIT #BIT8, (R3)
BEQ 2$
MOV (R3), R4
HLT 3
2$: SCOPE

```

: GET THE TRANSMITTER BUFFER REGISTER  
: SET BIT 8 AT TRANSMITTER BUFFER REGISTER  
: SET "EXPECTED"  
: RESET THE DEVICE  
: WAIT FOR RESET TO FINISH  
: TEST BIT 8  
: BIT 8 IS CLEARED  
: LOAD "FOUND"  
: BIT 8 IS SET AND SHOULDN'T BE  
: SET BIT 8 AGAIN  
: CLEAR THE TRANSMITTER BUFFER REGISTER  
: TEST TO SEE IF BIT 8 CLEARED  
: BIT 8 IS OK  
: LOAD "FOUND"  
: BIT 8 FAILED TO CLEAR  
: SCOPE THIS TEST

:\*\*\*\*\* TEST 55 \*\*\*\*\*  
: \*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 9 RESET AND CLEAR TEST  
: \*WRITE BIT 9 AND TEST THAT IT WILL BE CLEARED AFTER A  
: \*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION  
:\*\*\*\*\*

```

:*****
: TEST 55
:*****
:*****
TST55: MOV #55, @TSTNO
MOV @TST56, NEXT
MOV TXDBUF, R3
BIS #BIT9, (R3)
CLR R5
BIS #MRESET, @TXCSR
JSR PC, SMALL
BIT #BIT9, (R3)
BEQ 1$
MOV (R3), R4
HLT 3
1$: BIS #BIT9, (R3)
CLR (R3)
BIT #BIT9, (R3)
BEQ 2$
MOV (R3), R4
HLT 3
2$: SCOPE

```

: GET THE TRANSMITTER BUFFER REGISTER  
: SET BIT 9 AT TRANSMITTER BUFFER REGISTER  
: SET "EXPECTED"  
: RESET THE DEVICE  
: WAIT FOR RESET TO FINISH  
: TEST BIT 9  
: BIT 9 IS CLEARED  
: LOAD "FOUND"  
: BIT 9 IS SET AND SHOULDN'T BE  
: SET BIT 9 AGAIN  
: CLEAR THE TRANSMITTER BUFFER REGISTER  
: TEST TO SEE IF BIT 9 CLEARED  
: BIT 9 IS OK  
: LOAD "FOUND"  
: BIT 9 FAILED TO CLEAR

3153 013650 104400 25: SCOPE ;SCOPE THIS TEST

3154  
3155  
3156  
3157  
3158  
3159  
3160  
3161  
3162  
3163  
3164  
3165  
3166  
3167  
3168  
3169  
3170  
3171  
3172  
3173  
3174  
3175  
3176  
3177  
3178  
3179  
3180  
3181  
3182  
3183  
3184  
3185  
3186  
3187  
3188  
3189  
3190  
3191  
3192  
3193  
3194  
3195  
3196  
3197  
3198  
3199  
3200  
3201  
3202  
3203  
3204  
3205  
3206  
3207  
3208

013652 012737 000056 001226  
013660 012737 013746 001216  
013666 013703 001414  
013672 052713 002000  
013676 005005  
013700 052777 000400 165504  
013706 004737 004772  
013712 032713 002000  
013716 001402  
013720 011304  
013722 104003  
013724 052713 002000 15:  
013730 005013  
013732 032713 002000  
013736 001402  
013740 011304  
013742 104003  
013744 104400 25:

\*\*\*\*\* TEST 56 \*\*\*\*\*  
\*TRANSMITTER BUFFER REGISTER READ/WRITE BIT 10 RESET AND CLEAR TEST  
\*WRITE BIT 10, AND TEST THAT IT WILL BE CLEARED AFTER A  
\*DEVICE RESET AND TRANSMITTER BUFFER REGISTER CLR INSTRUCTION  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 56  
\*  
\*\*\*\*\*  
\*\*\*\*\*  
TST56: MOV #56, @TSTNO  
MOV #TST57, NEXT  
MOV TXDBUF, R3 ;GET THE TRANSMITTER BUFFER REGISTER  
BIS #BIT10, (R3) ;SET BIT 10 AT TRANSMITTER BUFFER REGISTER  
CLR R5 ;SET "EXPECTED"  
BIS #MRESET, @TXCSR ;RESET THE DEVICE  
JSR PC, SMALL ;WAIT FOR RESET TO FINISH  
BIT #BIT10, (R3) ;TEST BIT 10  
BEQ IS ;BIT 10 IS CLEARED  
MOV (R3), R4 ;LOAD "FOUND"  
HLT 3 ;BIT 10 IS SET AND SHOULDN'T BE  
15: BIS #BIT10, (R3) ;SET BIT 10 AGAIN  
CLR (R3) ;CLEAR THE TRANSMITTER BUFFER REGISTER  
BIT #BIT10, (R3) ;TEST TO SEE IF BIT 10 CLEARED  
BEQ 25 ;BIT 10 IS OK  
MOV (R3), R4 ;LOAD "FOUND"  
HLT 3 ;BIT 10 FAILED TO CLEAR  
25: SCOPE ;SCOPE THIS TEST

\*\*\*\*\* TEST 57 \*\*\*\*\*  
\*RECEIVER BUFFER REGISTER TEST  
\*TEST THAT RECEIVER BUFFER REGISTER CANNOT BE WRITTEN.  
\*WRITE RECEIVER BUFFER REGISTER WITH ALL 1'S  
\*AND VERIFY THAT ALL 0'S ARE READ BACK.  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 57  
\*  
\*\*\*\*\*  
\*\*\*\*\*  
TST57: MOV #57, @TSTNO  
MOV #TST60, NEXT  
MOV RXDBUF, R3  
CLR R5 ;GET THE REGISTER.  
MOV #-1, (R3) ;SET EXPECTED (ZERO)  
MOV (R3), R4 ;WRITE REGISTER WITH ALL 1'S  
CMP R5, R4 ;READ THE REGISTER.  
;IS THE REGISTER EQUAL TO ZERO.

3209 014000 001401  
3210 014002 104003  
3211 014004 104400

IS: BEQ 1\$ ;BR IF OK.  
HLT 3 ;REGISTER NOT ZERO.  
SCOPE ;SCOPE THIS TEST.

\*\*\*\*\* TEST 60 \*\*\*\*\*

\*PARAMETER STATUS REGISTER TEST  
\*TEST THAT PARAMETER STATUS REGISTER CANNOT BE WRITTEN  
\*READ THE PARAMETER STATUS REGISTER AND STORE THE DATA;  
\*COMPLEMENT THE DATA AND WRITE THE PARAMETER STATUS REGISTER  
\*VERIFYING THAT THE PARAMETER STATUS REGISTER DID NOT CHANGE.  
\*\*\*\*\*

3212  
3213  
3214  
3215  
3216  
3217  
3218  
3219  
3220  
3221  
3222  
3223  
3224  
3225  
3226  
3227

\*\*\*\*\*  
\*  
TEST 60  
\*  
\*\*\*\*\*

3228  
3229 014006 012737 000060 001226  
3230 014014 012737 014050 001216  
3231 014022 013703 001410

\*\*\*\*\*  
TST60: MOV #60, J#TSTNO  
MOV #TST61, NEXT  
MOV PARCSR, R3

3232  
3233 014026 011305  
3234 014030 010504  
3235 014032 005104  
3236 014034 010413  
3237 014036 011304  
3238 014040 020504  
3239 014042 001401  
3240 014044 104003  
3241 014046 104400

MOV (R3), R5 ;GET THE REGISTER.  
MOV R5, R4 ;READ THE REGISTER INTO R5  
COM R4 ;SAVE REG INTO R4  
MOV R4, (R3) ;MAKE R4 OPPOSITE TO REGISTER  
MOV (R3), R4 ;WRITE THE REGISTER WITH THE COMPLIMENT  
CMP R5, R4 ;READ THE REGISTER.  
BEQ 1\$ ;IS THE REGISTER EQUAL TO ZERO.  
HLT 3 ;BR IF OK.  
IS: SCOPE ;REGISTER NOT ZERO.  
SCOPE THIS TEST.

3242  
3243  
3244  
3245  
3246  
3247  
3248  
3249

\*\*\*\*\* TEST 61 \*\*\*\*\*  
\*RECEIVER CONTROL REGISTER BIT 0 READ ONLY DEVICE RESET AND CLEAR TEST  
\*WRITE BIT 0 A ONE AND VERIFY A ZERO IS READ BACK  
\*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
\*\*\*\*\*

3250  
3251  
3252  
3253  
3254

\*\*\*\*\*  
\*  
TEST 61  
\*  
\*\*\*\*\*

3255  
3256 014050 012737 000061 001226  
3257 014056 012737 014162 001216  
3258 014064 013703 001404  
3259 014070 012705 000001  
3260 014074 010513  
3261 014076 011304  
3262 014100 005005  
3263 014102 020504  
3264 014104 001401

\*\*\*\*\*  
TST61: MOV #61, J#TSTNO ;GET THE RECEIVER CONTROL REGISTER  
MOV #TST62, NEXT ;GET BIT 0  
MOV RXCSR, R3 ;WRITE BIT 0 TO RECEIVER CONTROL REGISTER  
MOV #BIT0, R5 ;READ RECEIVER CONTROL REGISTER BACK  
MOV R5, (R3) ;SET "EXPECTED"  
MOV (R3), R4 ;R5=GOOD, R4= ?  
CLR R5 ;BIT 0 IS OK  
CMP R5, R4  
BEQ 5\$

```

3265 014106 104003          HLT      3          ;BIT FAILED TO CLR
3266 014110 012705 000001 5$:      MOV     #BIT0,R5 ;RELOAD THE BIT
3267 014114 010513          MOV     R5,(R3)   ;WRITE BIT 0 TO THE REG
3268 014116 052777 000400 165266  BIS     #MRESET,@TXCSR ;RESET THE DEVICE
3269 014124 004737 004772          JSR     PC,SMALL  ;WAIT FOR RESET TO FINISH
3270 014130 011304          MOV     (R3),R4   ;GET BIT 0
3271 014132 032704 000001          BIT     #BIT0,R4  ;TEST BIT 0 FOR RESET CLR
3272 014136 001401          BEQ     7$       ;BIT 0 IS OK
3273 014140 104003          HLT      3          ;BIT FAILED TO CLEAR
3274 014142 010513 7$:      MOV     R5,(R3)   ;SET BIT 0
3275 014144 005013          CLR     (R3)     ;CLR RECEIVER CONTROL REGISTER
3276 014146 011304          MOV     (R3),R4  ;READ THE RECEIVER CONTROL REGISTER
3277 014150 005005          CLR     R5       ;SET "EXPECTED"
3278 014152 020504          CMP     R5,R4    ;R5=GOOD,R4=?
3279 014154 001401          BEQ     10$      ;BIT 0 IS OK
3280 014156 104003          HLT      3          ;BIT FAILED TO CLEAR
3281 014160 104400 10$:     SCOPE ;SCOPE THIS TEST
  
```

```

:***** TEST 62 *****
:RECEIVER CONTROL REGISTER BIT 7 READ ONLY DEVICE RESET AND CLEAR TEST
:WRITE BIT 7 A ONE AND VERIFY A ZERO IS READ BACK
:REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
:*****
  
```

```

:*****
:TEST 62
:*****
  
```

```

3296 014162 012737 000062 001226 TST62: MOV     #62,@TSTNO
3297 014170 012737 014274 001216  MOV     @TST63,NEXT
3298 014176 013703          MOV     RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
3299 014202 012705 000200          MOV     #BIT7,R5 ;GET BIT 7
3300 014206 010513          MOV     R5,(R3)  ;WRITE BIT 7 TO RECEIVER CONTROL REGISTER
3301 014210 011304          MOV     (R3),R4  ;READ RECEIVER CONTROL REGISTER BACK
3302 014212 005005          CLR     R5       ;SET "EXPECTED"
3303 014214 020504          CMP     R5,R4    ;R5=GOOD,R4=?
3304 014216 001401          BEQ     5$       ;BIT 7 IS OK
3305 014220 104003          HLT      3          ;BIT FAILED TO CLR
3306 014222 012705 000200 5$:     MOV     #BIT7,R5 ;RELOAD THE BIT
3307 014226 010513          MOV     R5,(R3)  ;WRITE BIT 7 TO THE REG
3308 014230 052777 000400 165154  BIS     #MRESET,@TXCSR ;RESET THE DEVICE
3309 014236 004737 004772          JSR     PC,SMALL  ;WAIT FOR RESET TO FINISH
3310 014242 011304          MOV     (R3),R4  ;GET BIT 7
3311 014244 032704 000200          BIT     #BIT7,R4 ;TEST BIT 7 FOR RESET CLR
3312 014250 001401          BEQ     7$       ;BIT 7 IS OK
3313 014252 104003          HLT      3          ;BIT FAILED TO CLEAR
3314 014254 010513 7$:     MOV     R5,(R3)  ;SET BIT 7
3315 014256 005013          CLR     (R3)     ;CLR RECEIVER CONTROL REGISTER
3316 014260 011304          MOV     (R3),R4  ;READ THE RECEIVER CONTROL REGISTER
3317 014262 005005          CLR     R5       ;SET "EXPECTED"
3318 014264 020504          CMP     R5,R4    ;R5=GOOD,R4=?
3319 014266 001401          BEQ     10$      ;BIT 7 IS OK
3320 014270 104003          HLT      3          ;BIT FAILED TO CLEAR
  
```

3321 014272 104400 10\$: SCOPE ;SCOPE THIS TEST

3322  
3323  
3324  
3325  
3326  
3327  
3328  
3329  
3330  
3331  
3332  
3333  
3334  
3335  
3336 014274 012737 000063 001226  
3337 014302 012737 014406 001216  
3338 014310 013703 001404  
3339 014314 012705 001000  
3340 014320 010513  
3341 014322 011304  
3342 014324 005005  
3343 014326 020504  
3344 014330 001401  
3345 014332 104003  
3346 014334 012705 001000 5\$:  
3347 014340 010513  
3348 014342 052777 000400 165042  
3349 014350 004737 004772  
3350 014354 011304  
3351 014356 032704 001000  
3352 014362 001401  
3353 014364 104003  
3354 014366 010513 7\$:  
3355 014370 005013  
3356 014372 011304  
3357 014374 005005  
3358 014376 020504  
3359 014400 001401  
3360 014402 104003  
3361 014404 104400 10\$:  
3362  
3363  
3364  
3365  
3366  
3367  
3368  
3369  
3370  
3371  
3372  
3373  
3374  
3375  
3376 014406 012737 000064 001226

\*\*\*\*\* TEST 63 \*\*\*\*\*  
\*RECEIVER CONTROL REGISTER BIT 9 READ ONLY DEVICE RESET AND CLEAR TEST  
\*WRITE BIT 9 A ONE AND VERIFY A ZERO IS READ BACK  
\*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 63  
\*  
\*\*\*\*\*

\*\*\*\*\*  
TST63: MOV #63, @TSTNO  
MOV @TST64, NEXT  
MOV RXCSR, R3 ;GET THE RECEIVER CONTROL REGISTER  
MOV #BIT9, R5 ;GET BIT 9  
MOV R5, (R3) ;WRITE BIT 9 TO RECEIVER CONTROL REGISTER  
MOV (R3), R4 ;READ RECEIVER CONTROL REGISTER BACK  
CLR R5 ;SET "EXPECTED"  
CMP R5, R4 ;R5=GOOD, R4= ?  
BEQ 5\$ ;BIT 9 IS OK  
HLT 3 ;BIT FAILED TO CLR  
5\$: MOV #BIT9, R5 ;RELOAD THE BIT  
MOV R5, (R3) ;WRITE BIT 9 TO THE REG  
BIS #MRESET, @TXCSR ;RESET THE DEVICE  
JSR PC, SMALL ;WAIT FOR RESET TO FINISH  
MOV (R3), R4 ;GET BIT 9  
BIT #BIT9, R4 ;TEST BIT 9 FOR RESET CLR  
BEQ 7\$ ;BIT 9 IS OK  
HLT 3 ;BIT FAILED TO CLEAR  
7\$: MOV R5, (R3) ;SET BIT 9  
CLR (R3) ;CLR RECEIVER CONTROL REGISTER  
MOV (R3), R4 ;READ THE RECEIVER CONTROL REGISTER  
CLR R5 ;SET "EXPECTED"  
CMP R5, R4 ;R5=GOOD, R4= ?  
BEQ 10\$ ;BIT 9 IS OK  
HLT 3 ;BIT FAILED TO CLEAR  
10\$: SCOPE ;SCOPE THIS TEST  
\*\*\*\*\*

\*\*\*\*\* TEST 64 \*\*\*\*\*  
\*RECEIVER CONTROL REGISTER BIT 10 READ ONLY DEVICE RESET AND CLEAR TEST  
\*WRITE BIT 10 A ONE AND VERIFY A ZERO IS READ BACK  
\*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 64  
\*  
\*\*\*\*\*

\*\*\*\*\*  
TST64: MOV #64, @TSTNO  
\*\*\*\*\*



```

3377 014414 012737 014520 001216      MOV      #TST65,NEXT
3378 014422 013703 001404      MOV      RXCSR,R3          ;GET THE RECEIVER CONTROL REGISTER
3379 014426 012705 002000      MOV      #BIT10,R5        ;GET BIT 10
3380 014432 010513      MOV      R5,(R3)          ;WRITE BIT 10 TO RECEIVER CONTROL REGISTER
3381 014434 011304      MOV      (R3),R4          ;READ RECEIVER CONTROL REGISTER BACK
3382 014436 005005      CLR      R5               ;SET "EXPECTED"
3383 014440 020504      CMP      R5,R4            ;R5=GOOD,R4=?
3384 014442 001401      BEQ      5$               ;BIT 10 IS OK
3385 014444 104003      HLT      3                ;BIT FAILED TO CLR
3386 014446 012705 002000 5$:      MOV      #BIT10,R5        ;RELOAD THE BIT
3387 014452 010513      MOV      R5,(R3)          ;WRITE BIT 10 TO THE REG
3388 014454 052777 000400 164730  BIS      #MRESET,@TXCSR   ;RESET THE DEVICE
3389 014462 004737 004772      JSR      PC,SMALL        ;WAIT FOR RESET TO FINISH
3390 014466 011304      MOV      (R3),R4          ;GET BIT 10
3391 014470 032704 002000      BIT      #BIT10,R4        ;TEST BIT 10 FOR RESET CLR
3392 014474 001401      BEQ      7$               ;BIT 10 IS OK
3393 014476 104003      HLT      3                ;BIT FAILED TO CLEAR
3394 014500 010513 7$:      MOV      R5,(R3)          ;SET BIT 10
3395 014502 005013      CLR      (R3)             ;CLR RECEIVER CONTROL REGISTER
3396 014504 011304      MOV      (R3),R4          ;READ THE RECEIVER CONTROL REGISTER
3397 014506 005005      CLR      R5               ;SET "EXPECTED"
3398 014510 020504      CMP      R5,R4            ;R5=GOOD,R4=?
3399 014512 001401      BEQ      10$              ;BIT 10 IS OK
3400 014514 104003      HLT      3                ;BIT FAILED TO CLEAR
3401 014516 104400 10$:      SCOPE

```

```

:***** TEST 65 *****
:*RECEIVER CONTROL REGISTER BIT 11 READ ONLY DEVICE RESET AND CLEAR TEST
:*WRITE BIT 11 A ONE AND VERIFY A ZERO IS READ BACK
:*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
:*****

```

```

:*****
:TEST 65
:*****

```

```

3416 014520 012737 000065 001226  TST65:  MOV      #65,@TSTNO
3417 014526 012737 014632 001216      MOV      #TST66,NEXT
3418 014534 013703 001404      MOV      RXCSR,R3          ;GET THE RECEIVER CONTROL REGISTER
3419 014540 012705 004000      MOV      #BIT11,R5        ;GET BIT 11
3420 014544 010513      MOV      R5,(R3)          ;WRITE BIT 11 TO RECEIVER CONTROL REGISTER
3421 014546 011304      MOV      (R3),R4          ;READ RECEIVER CONTROL REGISTER BACK
3422 014550 005005      CLR      R5               ;SET "EXPECTED"
3423 014552 020504      CMP      R5,R4            ;R5=GOOD,R4=?
3424 014554 001401      BEQ      5$               ;BIT 11 IS OK
3425 014556 104003      HLT      3                ;BIT FAILED TO CLR
3426 014560 012705 004000 5$:      MOV      #BIT11,R5        ;RELOAD THE BIT
3427 014564 010513      MOV      R5,(R3)          ;WRITE BIT 11 TO THE REG
3428 014566 052777 000400 164616  BIS      #MRESET,@TXCSR   ;RESET THE DEVICE
3429 014574 004737 004772      JSR      PC,SMALL        ;WAIT FOR RESET TO FINISH
3430 014600 011304      MOV      (R3),R4          ;GET BIT 11
3431 014602 032704 004000      BIT      #BIT11,R4        ;TEST BIT 11 FOR RESET CLR
3432 014606 001401      BEQ      7$               ;BIT 11 IS OK

```

```

3433 014610 104003          HLT      3          ;BIT FAILED TO CLEAR
3434 014612 010513      7$:  MOV     R5,(R3)    ;SET BIT 11
3435 014614 005013      CLR     (R3)       ;CLR RECEIVER CONTROL REGISTER
3436 014616 011304      MOV     (R3),R4    ;READ THE RECEIVER CONTROL REGISTER
3437 014620 005005      CLR     R5         ;SET "EXPECTED"
3438 014622 020504      CMP     R5,R4     ;R5=GOOD,R4=?
3439 014624 001401      BEQ    10$        ;BIT 11 IS OK
3440 014626 104003      HLT     3          ;BIT FAILED TO CLEAR
3441 014630 104400      10$:  SCOPE       ;SCOPE THIS TEST

```

```

***** TEST 66 *****
*RECEIVER CONTROL REGISTER BIT 12 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 12 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
TEST 66
*****

```

```

3455 014632 012737 000066 001226 TST66: MOV     #66,@TSTNO
3456 014640 012737 014744 001216 MOV     @TST67,NEXT
3457 014646 013703 001404 MOV     RXCSR,R3    ;GET THE RECEIVER CONTROL REGISTER
3458 014652 012705 010000 MOV     #BIT12,R5   ;GET BIT 12
3459 014656 010513 MOV     R5,(R3)     ;WRITE BIT 12 TO RECEIVER CONTROL REGISTER
3460 014660 011304 MOV     (R3),R4    ;READ RECEIVER CONTROL REGISTER BACK
3461 014662 005005 CLR     R5         ;SET "EXPECTED"
3462 014664 020504 CMP     R5,R4     ;R5=GOOD,R4=?
3463 014666 001401 BEQ    5$         ;BIT 12 IS OK
3464 014670 104003 HLT     3          ;BIT FAILED TO CLR
3465 014672 012705 010000 5$:  MOV     #BIT12,R5  ;RELOAD THE BIT
3466 014676 010513 MOV     R5,(R3)    ;WRITE BIT 12 TO THE REG
3467 014700 052777 000400 164504 BIS     @MRESET,@TXCSR ;RESET THE DEVICE
3468 014706 004737 004772 JSR    PC,SMALL    ;WAIT FOR RESET TO FINISH
3469 014712 011304 MOV     (R3),R4    ;GET BIT 12
3470 014714 032704 010000 BIT     #BIT12,R4  ;TEST BIT 12 FOR RESET CLR
3471 014720 001401 BEQ    7$         ;BIT 12 IS OK
3472 014722 104003 HLT     3          ;BIT FAILED TO CLEAR
3473 014724 010513 7$:  MOV     R5,(R3)    ;SET BIT 12
3474 014726 005013 CLR     (R3)       ;CLR RECEIVER CONTROL REGISTER
3475 014730 011304 MOV     (R3),R4    ;READ THE RECEIVER CONTROL REGISTER
3476 014732 005005 CLR     R5         ;SET "EXPECTED"
3477 014734 020504 CMP     R5,R4     ;R5=GOOD,R4=?
3478 014736 001401 BEQ    10$        ;BIT 12 IS OK
3479 014740 104003 HLT     3          ;BIT FAILED TO CLEAR
3480 014742 104400 10$:  SCOPE       ;SCOPE THIS TEST

```

```

***** TEST 67 *****
*RECEIVER CONTROL REGISTER BIT 13 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 13 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

3481
3482
3483
3484
3485
3486
3487
3488

```

3489  
3490  
3491  
3492  
3493  
3494  
3495  
3496  
3497  
3498  
3499  
3500  
3501  
3502  
3503  
3504  
3505  
3506  
3507  
3508  
3509  
3510  
3511  
3512  
3513  
3514  
3515  
3516  
3517  
3518  
3519  
3520  
3521  
3522  
3523  
3524  
3525  
3526  
3527  
3528  
3529  
3530  
3531  
3532  
3533  
3534  
3535  
3536  
3537  
3538  
3539  
3540  
3541  
3542  
3543  
3544

014744 012737 000067 001226  
014752 012737 015056 001216  
014760 013703 001404  
014764 012705 020000  
014770 010513  
014772 011304  
014774 005005  
014776 020504  
015000 001401  
015002 104003  
015004 012705 020000  
015010 010513  
015012 052777 000400 164372  
015020 004737 004772  
015024 011304  
015026 032704 020000  
015032 001401  
015034 104003  
015036 010513  
015040 005013  
015042 011304  
015044 005005  
015046 020504  
015050 001401  
015052 104003  
015054 104400

```
*****  
*  
: TEST 67  
*  
*****  
*****  
TST67: MOV #67, @TSTNO  
MOV #TST70, NEXT  
MOV RXCSR, R3 ; GET THE RECEIVER CONTROL REGISTER  
MOV #BIT13, R5 ; GET BIT 13  
MOV R5, (R3) ; WRITE BIT 13 TO RECEIVER CONTROL REGISTER  
MOV (R3), R4 ; READ RECEIVER CONTROL REGISTER BACK  
CLR R5 ; SET "EXPECTED"  
CMP R5, R4 ; R5=GOOD, R4= ?  
BEQ 5$ ; BIT 13 IS OK  
HLT 3 ; BIT FAILED TO CLR  
5$: MOV #BIT13, R5 ; RELOAD THE BIT  
MOV R5, (R3) ; WRITE BIT 13 TO THE REG  
BIS #MRESET, @TXCSR ; RESET THE DEVICE  
JSR PC, SMALL ; WAIT FOR RESET TO FINISH  
MOV (R3), R4 ; GET BIT 13  
BIT #BIT13, R4 ; TEST BIT 13 FOR RESET CLR  
BEQ 7$ ; BIT 13 IS OK  
HLT 3 ; BIT FAILED TO CLEAR  
7$: MOV R5, (R3) ; SET BIT 13  
CLR (R3) ; CLR RECEIVER CONTROL REGISTER  
MOV (R3), R4 ; READ THE RECEIVER CONTROL REGISTER  
CLR R5 ; SET "EXPECTED"  
CMP R5, R4 ; R5=GOOD, R4= ?  
BEQ 10$ ; BIT 13 IS OK  
HLT 3 ; BIT FAILED TO CLEAR  
10$: SCOPE ; SCOPE THIS TEST
```

\*\*\*\*\* TEST 70 \*\*\*\*\*  
\*RECEIVER CONTROL REGISTER BIT 14 READ ONLY DEVICE RESET AND CLEAR TEST  
\*WRITE BIT 14 A ONE AND VERIFY A ZERO IS READ BACK  
\*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
\*\*\*\*\*

```
*****  
*  
: TEST 70  
*  
*****  
*****  
TST70: MOV #70, @TSTNO  
MOV #TST71, NEXT  
MOV RXCSR, R3 ; GET THE RECEIVER CONTROL REGISTER  
MOV #BIT14, R5 ; GET BIT 14  
MOV R5, (R3) ; WRITE BIT 14 TO RECEIVER CONTROL REGISTER  
MOV (R3), R4 ; READ RECEIVER CONTROL REGISTER BACK  
CLR R5 ; SET "EXPECTED"  
CMP R5, R4 ; R5=GOOD, R4= ?  
BEQ 5$ ; BIT 14 IS OK
```

```

3545 015114 104003          HLT      3          ;BIT FAILED TO CLR
3546 015116 012705 040000 5$:  MOV     #BIT14,R5 ;RELOAD THE BIT
3547 015122 010513          MOV     R5,(R3)   ;WRITE BIT 14 TO THE REG
3548 015124 052777 000400 164260 BIS     #MRESET,@TXCSR ;RESET THE DEVICE
3549 015132 004737 004772 JSR     PC,SMALL  ;WAIT FOR RESET TO FINISH
3550 015136 011304          MOV     (R3),R4   ;GET BIT 14
3551 015140 032704 040000 BIT     #BIT14,R4 ;TEST BIT 14 FOR RESET CLR
3552 015144 001401          BEQ     7$        ;BIT 14 IS OK
3553 015146 104003          HLT      3          ;BIT FAILED TO CLEAR
3554 015150 010513 7$:  MOV     R5,(R3)   ;SET BIT 14
3555 015152 005013          CLR     (R3)     ;CLR RECEIVER CONTROL REGISTER
3556 015154 011304          MOV     (R3),R4  ;READ THE RECEIVER CONTROL REGISTER
3557 015156 005005          CLR     R5       ;SET "EXPECTED"
3558 015160 020504          CMP     R5,R4    ;R5=GOOD,R4=?
3559 015162 001401          BEQ     10$      ;BIT 14 IS OK
3560 015164 104003          HLT      3          ;BIT FAILED TO CLEAR
3561 015166 104400 10$: SCOPE ;SCOPE THIS TEST

```

```

***** TEST 71 *****
*RECEIVER CONTROL REGISTER BIT 15 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 15 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
*
TEST 71
*
*****

```

```

3576 015170 012737 000071 001226 TST71: MOV     #71,@TSTNO
3577 015176 012737 015302 001216 MOV     #TST72,NEXT
3578 015204 013703 001404          MOV     RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
3579 015210 012705 100000          MOV     #BIT15,R5 ;GET BIT 15
3580 015214 010513          MOV     R5,(R3)   ;WRITE BIT 15 TO RECEIVER CONTROL REGISTER
3581 015216 011304          MOV     (R3),R4  ;READ RECEIVER CONTROL REGISTER BACK
3582 015220 005005          CLR     R5       ;SET "EXPECTED"
3583 015222 020504          CMP     R5,R4    ;R5=GOOD,R4=?
3584 015224 001401          BEQ     5$        ;BIT 15 IS OK
3585 015226 104003          HLT      3          ;BIT FAILED TO CLR
3586 015230 012705 100000 5$:  MOV     #BIT15,R5 ;RELOAD THE BIT
3587 015234 010513          MOV     R5,(R3)   ;WRITE BIT 15 TO THE REG
3588 015236 052777 000400 164146 BIS     #MRESET,@TXCSR ;RESET THE DEVICE
3589 015244 004737 004772 JSR     PC,SMALL  ;WAIT FOR RESET TO FINISH
3590 015250 011304          MOV     (R3),R4   ;GET BIT 15
3591 015252 032704 100000 BIT     #BIT15,R4 ;TEST BIT 15 FOR RESET CLR
3592 015256 001401          BEQ     7$        ;BIT 15 IS OK
3593 015260 104003          HLT      3          ;BIT FAILED TO CLEAR
3594 015262 010513 7$:  MOV     R5,(R3)   ;SET BIT 15
3595 015264 005013          CLR     (R3)     ;CLR RECEIVER CONTROL REGISTER
3596 015266 011304          MOV     (R3),R4  ;READ THE RECEIVER CONTROL REGISTER
3597 015270 005005          CLR     R5       ;SET "EXPECTED"
3598 015272 020504          CMP     R5,R4    ;R5=GOOD,R4=?
3599 015274 001401          BEQ     10$      ;BIT 15 IS OK
3600 015276 104003          HLT      3          ;BIT FAILED TO CLEAR

```

3601 015300 104400 10\$: SCOPE ;SCOPE THIS TEST

3602  
3603  
3604  
3605  
3606  
3607  
3608  
3609  
3610  
3611  
3612  
3613  
3614  
3615  
3616  
3617  
3618  
3619  
3620  
3621  
3622  
3623  
3624  
3625  
3626  
3627  
3628  
3629  
3630  
3631  
3632  
3633  
3634  
3635  
3636  
3637  
3638  
3639  
3640  
3641  
3642  
3643  
3644  
3645  
3646  
3647  
3648  
3649  
3650  
3651  
3652  
3653  
3654  
3655  
3656

015300 104400  
015302 012737 000072 001226  
015310 012737 015414 001216  
015316 013703 001406  
015322 012705 000001  
015326 010513  
015330 011304  
015332 005005  
015334 020504  
015336 001401  
015340 104003  
015342 012705 000001 5\$:  
015346 010513  
015350 052777 000400 164034  
015356 004737 004772  
015362 011304  
015364 032704 000001  
015370 001401  
015372 104003  
015374 010513 7\$:  
015376 005013  
015400 011304  
015402 005005  
015404 020504  
015406 001401  
015410 104003  
015412 104400 10\$:

\*\*\*\*\* TEST 72 \*\*\*\*\*  
\*RECEIVER BUFFER REGISTER BIT 0 READ ONLY DEVICE RESET AND CLEAR TEST  
\*WRITE BIT 0 A ONE AND VERIFY A ZERO IS READ BACK  
\*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 72  
\*  
\*\*\*\*\*

\*\*\*\*\*  
TST72: MOV #72, @TSTNO  
MOV #TST73, NEXT  
MOV RXDBUF, R3 ;GET THE RECEIVER BUFFER REGISTER  
MOV #BIT0, R5 ;GET BIT 0  
MOV R5, (R3) ;WRITE BIT 0 TO RECEIVER BUFFER REGISTER  
MOV (R3), R4 ;READ RECEIVER BUFFER REGISTER BACK  
CLR R5 ;SET "EXPECTED"  
CMP R5, R4 ;R5=GOOD, R4= ?  
BEQ 5\$ ;BIT 0 IS OK  
HLT 3 ;BIT FAILED TO CLR  
5\$: MOV #BIT0, R5 ;RELOAD THE BIT  
MOV R5, (R3) ;WRITE BIT 0 TO THE REG  
BIS #MRESET, @TXCSR ;RESET THE DEVICE  
JSR PC, SMALL ;WAIT FOR RESET TO FINISH  
MOV (R3), R4 ;GET BIT 0  
BIT #BIT0, R4 ;TEST BIT 0 FOR RESET CLR  
BEQ 7\$ ;BIT 0 IS OK  
HLT 3 ;BIT FAILED TO CLEAR  
7\$: MOV R5, (R3) ;SET BIT 0  
CLR (R3) ;CLR RECEIVER BUFFER REGISTER  
MOV (R3), R4 ;READ THE RECEIVER BUFFER REGISTER  
CLR R5 ;SET "EXPECTED"  
CMP R5, R4 ;R5=GOOD, R4= ?  
BEQ 10\$ ;BIT 0 IS OK  
HLT 3 ;BIT FAILED TO CLEAR  
10\$: SCOPE ;SCOPE THIS TEST

\*\*\*\*\* TEST 73 \*\*\*\*\*  
\*RECEIVER BUFFER REGISTER BIT 1 READ ONLY DEVICE RESET AND CLEAR TEST  
\*WRITE BIT 1 A ONE AND VERIFY A ZERO IS READ BACK  
\*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 73  
\*  
\*\*\*\*\*

015414 012737 000073 001226 TST73: MOV #73, @TSTNO

```

3657 015422 012737 015526 001216      MOV      #TST74,NEXT
3658 015430 013703 001406      MOV      RXDBUF,R3          ;GET THE RECEIVER BUFFER REGISTER
3659 015434 012705 000002      MOV      #BIT1,R5          ;GET BIT 1
3660 015440 010513      MOV      R5,(R3)           ;WRITE BIT 1 TO RECEIVER BUFFER REGISTER
3661 015442 011304      MOV      (R3),R4           ;READ RECEIVER BUFFER REGISTER BACK
3662 015444 005005      CLR      R5                ;SET "EXPECTED"
3663 015446 020504      CMP      R5,R4             ;R5=GOOD,R4=?
3664 015450 001401      BEQ      5$                ;BIT 1 IS OK
3665 015452 104003      HLT      3                 ;BIT FAILED TO CLR
3666 015454 012705 000002      5$:    MOV      #BIT1,R5          ;RELOAD THE BIT
3667 015460 010513      MOV      R5,(R3)           ;WRITE BIT 1 TO THE REG
3668 015462 052777 000400 163722      BIS      #MRESET,@TXCSR    ;RESET THE DEVICE
3669 015470 004737 004772      JSR      PC,SMALL          ;WAIT FOR RESET TO FINISH
3670 015474 011304      MOV      (R3),R4           ;GET BIT 1
3671 015476 032704 000002      BIT      #BIT1,R4          ;TEST BIT 1 FOR RESET CLR
3672 015502 001401      BEQ      7$                ;BIT 1 IS OK
3673 015504 104003      HLT      3                 ;BIT FAILED TO CLEAR
3674 015506 010513      7$:    MOV      R5,(R3)           ;SET BIT 1
3675 015510 005013      CLR      (R3)              ;CLR RECEIVER BUFFER REGISTER
3676 015512 011304      MOV      (R3),R4           ;READ THE RECEIVER BUFFER REGISTER
3677 015514 005005      CLR      R5                ;SET "EXPECTED"
3678 015516 020504      CMP      R5,R4             ;R5=GOOD,R4=?
3679 015520 001401      BEQ      10$               ;BIT 1 IS OK
3680 015522 104003      HLT      3                 ;BIT FAILED TO CLEAR
3681 015524 104400      10$:   SCOPE                ;SCOPE THIS TEST

```

```

;***** TEST 74 *****
;RECEIVER BUFFER REGISTER BIT 2 READ ONLY DEVICE RESET AND CLEAR TEST
;WRITE BIT 2 A ONE AND VERIFY A ZERO IS READ BACK
;REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
;*****

```

```

;*****
;TEST 74
;*****
;*****

```

```

3696 015526 012737 000074 001226  TST74:  MOV      #74,@TSTNO
3697 015534 012737 015640 001216      MOV      #TST75,NEXT
3698 015542 013703 001406      MOV      RXDBUF,R3          ;GET THE RECEIVER BUFFER REGISTER
3699 015546 012705 000004      MOV      #BIT2,R5          ;GET BIT 2
3700 015552 010513      MOV      R5,(R3)           ;WRITE BIT 2 TO RECEIVER BUFFER REGISTER
3701 015554 011304      MOV      (R3),R4           ;READ RECEIVER BUFFER REGISTER BACK
3702 015556 005005      CLR      R5                ;SET "EXPECTED"
3703 015560 020504      CMP      R5,R4             ;R5=GOOD,R4=?
3704 015562 001401      BEQ      5$                ;BIT 2 IS OK
3705 015564 104003      HLT      3                 ;BIT FAILED TO CLR
3706 015566 012705 000004      5$:    MOV      #BIT2,R5          ;RELOAD THE BIT
3707 015572 010513      MOV      R5,(R3)           ;WRITE BIT 2 TO THE REG
3708 015574 052777 000400 163610      BIS      #MRESET,@TXCSR    ;RESET THE DEVICE
3709 015602 004737 004772      JSR      PC,SMALL          ;WAIT FOR RESET TO FINISH
3710 015606 011304      MOV      (R3),R4           ;GET BIT 2
3711 015610 032704 000004      BIT      #BIT2,R4          ;TEST BIT 2 FOR RESET CLR
3712 015614 001401      BEQ      7$                ;BIT 2 IS OK

```

```

3713 015616 104003          HLT      3          ;BIT FAILED TO CLEAR
3714 015620 010513      7$:  MOV      R5,(R3)      ;SET BIT 2
3715 015622 005013          CLR      (R3)        ;CLR RECEIVER BUFFER REGISTER
3716 015624 011304          MOV      (R3),R4     ;READ THE RECEIVER BUFFER REGISTER
3717 015626 005005          CLR      R5          ;SET "EXPECTED"
3718 015630 020504          CMP      R5,R4       ;R5=GOOD,R4=?
3719 015632 001401          BEQ      10$         ;BIT 2 IS OK
3720 015634 104003          HLT      3          ;BIT FAILED TO CLEAR
3721 015636 104400      10$:  SCOPE          ;SCOPE THIS TEST
  
```

```

3722
3723
3724
3725
3726
3727
3728
3729
3730
3731
3732
3733
3734
3735
3736
3737
3738
3739
3740
3741
3742
3743
3744
3745
3746
3747
3748
3749
3750
3751
3752
3753
3754
3755
3756
3757
3758
3759
3760
3761
3762
3763
3764
3765
3766
3767
3768
  
```

```

:***** TEST 75 *****
:*RECEIVER BUFFER REGISTER BIT 3 READ ONLY DEVICE RESET AND CLEAR TEST
:*WRITE BIT 3 A ONE AND VERIFY A ZERO IS READ BACK
:*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
:*****
  
```

```

:*****
:TEST 75
:*****
  
```

```

3736 015640 012737 000075 001226 7$:  MOV      #75,@TSTNO
3737 015646 012737 015752 001216 7$:  MOV      #TST76,NEXT
3738 015654 013703 001406          MOV      RXDBUF,R3   ;GET THE RECEIVER BUFFER REGISTER
3739 015660 012705 000010          MOV      #BIT3,R5    ;GET BIT 3
3740 015664 010513          MOV      R5,(R3)     ;WRITE BIT 3 TO RECEIVER BUFFER REGISTER
3741 015666 011304          MOV      (R3),R4     ;READ RECEIVER BUFFER REGISTER BACK
3742 015670 005005          CLR      R5          ;SET "EXPECTED"
3743 015672 020504          CMP      R5,R4       ;R5=GOOD,R4=?
3744 015674 001401          BEQ      5$          ;BIT 3 IS OK
3745 015676 104003          HLT      3          ;BIT FAILED TO CLR
3746 015700 012705 000010      5$:  MOV      #BIT3,R5    ;RELOAD THE BIT
3747 015704 010513          MOV      R5,(R3)     ;WRITE BIT 3 TO THE REG
3748 015706 052777 000400 163476 5$:  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3749 015714 004737 004772          JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
3750 015720 011304          MOV      (R3),R4     ;GET BIT 3
3751 015722 032704 000010          BIT      #BIT3,R4    ;TEST BIT 3 FOR RESET CLR
3752 015726 001401          BEQ      7$          ;BIT 3 IS OK
3753 015730 104003          HLT      3          ;BIT FAILED TO CLEAR
3754 015732 010513      7$:  MOV      R5,(R3)     ;SET BIT 3
3755 015734 005013          CLR      (R3)        ;CLR RECEIVER BUFFER REGISTER
3756 015736 011304          MOV      (R3),R4     ;READ THE RECEIVER BUFFER REGISTER
3757 015740 005005          CLR      R5          ;SET "EXPECTED"
3758 015742 020504          CMP      R5,R4       ;R5=GOOD,R4=?
3759 015744 001401          BEQ      10$         ;BIT 3 IS OK
3760 015746 104003          HLT      3          ;BIT FAILED TO CLEAR
3761 015750 104400      10$:  SCOPE          ;SCOPE THIS TEST
  
```

```

:***** TEST 76 *****
:*RECEIVER BUFFER REGISTER BIT 4 READ ONLY DEVICE RESET AND CLEAR TEST
:*WRITE BIT 4 A ONE AND VERIFY A ZERO IS READ BACK
:*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
:*****
  
```

3769  
3770  
3771  
3772  
3773  
3774  
3775  
3776  
3777  
3778  
3779  
3780  
3781  
3782  
3783  
3784  
3785  
3786  
3787  
3788  
3789  
3790  
3791  
3792  
3793  
3794  
3795  
3796  
3797  
3798  
3799  
3800  
3801  
3802  
3803  
3804  
3805  
3806  
3807  
3808  
3809  
3810  
3811  
3812  
3813  
3814  
3815  
3816  
3817  
3818  
3819  
3820  
3821  
3822  
3823  
3824

015752 012737 000076 001226  
015760 012737 016064 001216  
015766 013703 001406  
015772 012705 000020  
015776 010513  
016000 011304  
016002 005005  
016004 020504  
016006 001401  
016010 104003  
016012 012705 000020  
016016 010513  
016020 052777 000400 163364  
016026 004737 004772  
016032 011304  
016034 032704 000020  
016040 001401  
016042 104003  
016044 010513  
016046 005013  
016050 011304  
016052 005005  
016054 020504  
016056 001401  
016060 104003  
016062 104400

```

:*****
: TEST 76
:*****
:*****
TST76: MOV #76, @TSTNO
MOV #TST77, NEXT
MOV RXDBUF, R3 ;GET THE RECEIVER BUFFER REGISTER
MOV #BIT4, R5 ;GET BIT 4
MOV R5, (R3) ;WRITE BIT 4 TO RECEIVER BUFFER REGISTER
MOV (R3), R4 ;READ RECEIVER BUFFER REGISTER BACK
CLR R5 ;SET "EXPECTED"
CMP R5, R4 ;R5=GOOD, R4= ?
BEQ SS ;BIT 4 IS OK
HLT 3 ;BIT FAILED TO CLR
SS: MOV #BIT4, R5 ;RELOAD THE BIT
MOV R5, (R3) ;WRITE BIT 4 TO THE REG
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
MOV (R3), R4 ;GET BIT 4
BIT #BIT4, R4 ;TEST BIT 4 FOR RESET CLR
BEQ 7S ;BIT 4 IS OK
HLT 3 ;BIT FAILED TO CLEAR
7S: MOV R5, (R3) ;SET BIT 4
CLR (R3) ;CLR RECEIVER BUFFER REGISTER
MOV (R3), R4 ;READ THE RECEIVER BUFFER REGISTER
CLR R5 ;SET "EXPECTED"
CMP R5, R4 ;R5=GOOD, R4= ?
BEQ 10S ;BIT 4 IS OK
HLT 3 ;BIT FAILED TO CLEAR
10S: SCOPE ;SCOPE THIS TEST

```

\*\*\*\*\* TEST 77 \*\*\*\*\*  
\*RECEIVER BUFFER REGISTER BIT 5 READ ONLY DEVICE RESET AND CLEAR TEST  
\*WRITE BIT 5 A ONE AND VERIFY A ZERO IS READ BACK  
\*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
\*\*\*\*\*

```

:*****
: TEST 77
:*****
:*****
TST77: MOV #77, @TSTNO
MOV #TST100, NEXT
MOV RXDBUF, R3 ;GET THE RECEIVER BUFFER REGISTER
MOV #BIT5, R5 ;GET BIT 5
MOV R5, (R3) ;WRITE BIT 5 TO RECEIVER BUFFER REGISTER
MOV (R3), R4 ;READ RECEIVER BUFFER REGISTER BACK
CLR R5 ;SET "EXPECTED"
CMP R5, R4 ;R5=GOOD, R4= ?
BEQ SS ;BIT 5 IS OK

```



```

3825 016122 104003          HLT      3          ;BIT FAILED TO CLR
3826 016124 012705 000040 5$:      MOV      #BIT5,R5 ;RELOAD THE BIT
3827 016130 010513          MOV      R5,(R3)  ;WRITE BIT 5 TO THE REG
3828 016132 052777 000400 163252  BIS      #MRESET,ATXCSR ;RESET THE DEVICE
3829 016140 004737 004772  JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
3830 016144 011304          MOV      (R3),R4  ;GET BIT 5
3831 016146 032704 000040  BIT      #BIT5,R4  ;TEST BIT 5 FOR RESET CLR
3832 016152 001401          BEQ      7$      ;BIT 5 IS OK
3833 016154 104003          HLT      3          ;BIT FAILED TO CLEAR
3834 016156 010513 7$:      MOV      R5,(R3)  ;SET BIT 5
3835 016160 005013          CLR      (R3)    ;CLR RECEIVER BUFFER REGISTER
3836 016162 011304          MOV      (R3),R4 ;READ THE RECEIVER BUFFER REGISTER
3837 016164 005005          CLR      R5      ;SET "EXPECTED"
3838 016166 020504          CMP      R5,R4   ;R5=GOOD,R4=?
3839 016170 001401          BEQ      10$     ;BIT 5 IS OK
3840 016172 104003          HLT      3          ;BIT FAILED TO CLEAR
3841 016174 104400 10$:     SCOPE    ;SCOPE THIS TEST

```

```

***** TEST 100 *****
;RECEIVER BUFFER REGISTER BIT 6 READ ONLY DEVICE RESET AND CLEAR TEST
;WRITE BIT 6 A ONE AND VERIFY A ZERO IS READ BACK
;REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
: TEST 100
*****

```

```

3850
3851
3852
3853
3854
3855
3856 016176 012737 000100 001226 TST100: MOV      #100,ATSTNO
3857 016204 012737 016310 001216  MOV      #TST101,NEXT
3858 016212 013703 001406  MOV      RXDBUF,R3 ;GET THE RECEIVER BUFFER REGISTER
3859 016216 012705 000100  MOV      #BIT6,R5 ;GET BIT 6
3860 016222 010513  MOV      R5,(R3)  ;WRITE BIT 6 TO RECEIVER BUFFER REGISTER
3861 016224 011304  MOV      (R3),R4  ;READ RECEIVER BUFFER REGISTER BACK
3862 016226 005005  CLR      R5      ;SET "EXPECTED"
3863 016230 020504  CMP      R5,R4   ;R5=GOOD,R4=?
3864 016232 001401  BEQ      5$      ;BIT 6 IS OK
3865 016234 104003  HLT      3          ;BIT FAILED TO CLR
3866 016236 012705 000100 5$:      MOV      #BIT6,R5 ;RELOAD THE BIT
3867 016242 010513  MOV      R5,(R3)  ;WRITE BIT 6 TO THE REG
3868 016244 052777 000400 163140  BIS      #MRESET,ATXCSR ;RESET THE DEVICE
3869 016252 004737 004772  JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
3870 016256 011304  MOV      (R3),R4  ;GET BIT 6
3871 016260 032704 000100  BIT      #BIT6,R4  ;TEST BIT 6 FOR RESET CLR
3872 016264 001401  BEQ      7$      ;BIT 6 IS OK
3873 016266 104003  HLT      3          ;BIT FAILED TO CLEAR
3874 016270 010513 7$:      MOV      R5,(R3)  ;SET BIT 6
3875 016272 005013  CLR      (R3)    ;CLR RECEIVER BUFFER REGISTER
3876 016274 011304  MOV      (R3),R4 ;READ THE RECEIVER BUFFER REGISTER
3877 016276 005005  CLR      R5      ;SET "EXPECTED"
3878 016300 020504  CMP      R5,R4   ;R5=GOOD,R4=?
3879 016302 001401  BEQ      10$     ;BIT 6 IS OK
3880 016304 104003  HLT      3          ;BIT FAILED TO CLEAR

```

3881 016306 104400 10\$: SCOPE ;SCOPE THIS TEST

3882  
3883  
3884  
3885  
3886  
3887  
3888  
3889  
3890  
3891  
3892  
3893  
3894  
3895  
3896  
3897  
3898  
3899  
3900  
3901  
3902  
3903  
3904  
3905  
3906  
3907  
3908  
3909  
3910  
3911  
3912  
3913  
3914  
3915  
3916  
3917  
3918  
3919  
3920  
3921  
3922  
3923  
3924  
3925  
3926  
3927  
3928  
3929  
3930  
3931  
3932  
3933  
3934  
3935  
3936

016310 012737 000101 001226  
016316 012737 016422 001216  
016324 013703 001406  
016330 012705 000200  
016334 010513  
016336 011304  
016340 005005  
016342 020504  
016344 001401  
016346 104003  
016350 012705 000200 5\$:  
016354 010513  
016356 052777 000400 163026  
016364 004737 004772  
016370 011304  
016372 032704 000200  
016376 001401  
016400 104003  
016402 010513 7\$:  
016404 005013  
016406 011304  
016410 005005  
016412 020504  
016414 001401  
016416 104003  
016420 104400 10\$:

\*\*\*\*\* TEST 101 \*\*\*\*\*  
\*RECEIVER BUFFER REGISTER BIT 7 READ ONLY DEVICE RESET AND CLEAR TEST  
\*WRITE BIT 7 A ONE AND VERIFY A ZERO IS READ BACK  
\*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 101  
\*  
\*\*\*\*\*

\*\*\*\*\*  
TST101: MOV #101, @TSTNO  
MOV @TST102, NEXT  
MOV RXDBUF, R3 ;GET THE RECEIVER BUFFER REGISTER  
MOV #BIT7, R5 ;GET BIT 7  
MOV R5, (R3) ;WRITE BIT 7 TO RECEIVER BUFFER REGISTER  
MOV (R3), R4 ;READ RECEIVER BUFFER REGISTER BACK  
CLR R5 ;SET "EXPECTED"  
CMP R5, R4 ;R5=GOOD, R4= ?  
BEQ 5\$ ;BIT 7 IS OK  
HLT 3 ;BIT FAILED TO CLR  
5\$: MOV #BIT7, R5 ;RELOAD THE BIT  
MOV R5, (R3) ;WRITE BIT 7 TO THE REG  
BIS #MRESET, @TXCSR ;RESET THE DEVICE  
JSR PC, SMALL ;WAIT FOR RESET TO FINISH  
MOV (R3), R4 ;GET BIT 7  
BIT #BIT7, R4 ;TEST BIT 7 FOR RESET CLR  
BEQ 7\$ ;BIT 7 IS OK  
HLT 3 ;BIT FAILED TO CLEAR  
7\$: MOV R5, (R3) ;SET BIT 7  
CLR (R3) ;CLR RECEIVER BUFFER REGISTER  
MOV (R3), R4 ;READ THE RECEIVER BUFFER REGISTER  
CLR R5 ;SET "EXPECTED"  
CMP R5, R4 ;R5=GOOD, R4= ?  
BEQ 10\$ ;BIT 7 IS OK  
HLT 3 ;BIT FAILED TO CLEAR  
10\$: SCOPE ;SCOPE THIS TEST

\*\*\*\*\* TEST 102 \*\*\*\*\*  
\*RECEIVER BUFFER REGISTER BIT 8 READ ONLY DEVICE RESET AND CLEAR TEST  
\*WRITE BIT 8 A ONE AND VERIFY A ZERO IS READ BACK  
\*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 102  
\*  
\*\*\*\*\*

016422 012737 000102 001226 TST102: MOV #102, @TSTNO

```

3937 016430 012737 016534 001216 MOV #TST103,NEXT
3938 016436 013703 001406 MOV RXDBUF,R3 ;GET THE RECEIVER BUFFER REGISTER
3939 016442 012705 000400 MOV #BIT8,R5 ;GET BIT 8
3940 016446 010513 MOV R5,(R3) ;WRITE BIT 8 TO RECEIVER BUFFER REGISTER
3941 016450 011304 MOV (R3),R4 ;READ RECEIVER BUFFER REGISTER BACK
3942 016452 005005 CLR R5 ;SET "EXPECTED"
3943 016454 020504 CMP R5,R4 ;R5=GOOD,R4=?
3944 016456 001401 BEQ 5$ ;BIT 8 IS OK
3945 016460 104003 HLT 3 ;BIT FAILED TO CLR
3946 016462 012705 000400 5$: MOV #BIT8,R5 ;RELOAD THE BIT
3947 016466 010513 MOV R5,(R3) ;WRITE BIT 8 TO THE REG
3948 016470 052777 000400 162714 BIS #MRESET,DTXCSR ;RESET THE DEVICE
3949 016476 004737 004772 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3950 016502 011304 MOV (R3),R4 ;GET BIT 8
3951 016504 032704 000400 BIT #BIT8,R4 ;TEST BIT 8 FOR RESET CLR
3952 016510 001401 BEQ 7$ ;BIT 8 IS OK
3953 016512 104003 HLT 3 ;BIT FAILED TO CLEAR
3954 016514 010513 7$: MOV R5,(R3) ;SET BIT 8
3955 016516 005013 CLR (R3) ;CLR RECEIVER BUFFER REGISTER
3956 016520 011304 MOV (R3),R4 ;READ THE RECEIVER BUFFER REGISTER
3957 016522 005005 CLR R5 ;SET "EXPECTED"
3958 016524 020504 CMP R5,R4 ;R5=GOOD,R4=?
3959 016526 001401 BEQ 10$ ;BIT 8 IS OK
3960 016530 104003 HLT 3 ;BIT FAILED TO CLEAR
3961 016532 104400 10$: SCOPE ;SCOPE THIS TEST

```

```

***** TEST 103 *****
*RECEIVER BUFFER REGISTER BIT 9 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 9 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
TEST 103
*****

```

```

3976 016534 012737 000103 001226 TST103: MOV #103,#TSTNO
3977 016542 012737 016646 001216 MOV #TST104,NEXT
3978 016550 013703 001406 MOV RXDBUF,R3 ;GET THE RECEIVER BUFFER REGISTER
3979 016554 012705 001000 MOV #BIT9,R5 ;GET BIT 9
3980 016560 010513 MOV R5,(R3) ;WRITE BIT 9 TO RECEIVER BUFFER REGISTER
3981 016562 011304 MOV (R3),R4 ;READ RECEIVER BUFFER REGISTER BACK
3982 016564 005005 CLR R5 ;SET "EXPECTED"
3983 016566 020504 CMP R5,R4 ;R5=GOOD,R4=?
3984 016570 001401 BEQ 5$ ;BIT 9 IS OK
3985 016572 104003 HLT 3 ;BIT FAILED TO CLR
3986 016574 012705 001000 5$: MOV #BIT9,R5 ;RELOAD THE BIT
3987 016600 010513 MOV R5,(R3) ;WRITE BIT 9 TO THE REG
3988 016602 052777 000400 162602 BIS #MRESET,DTXCSR ;RESET THE DEVICE
3989 016610 004737 004772 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3990 016614 011304 MOV (R3),R4 ;GET BIT 9
3991 016616 032704 001000 BIT #BIT9,R4 ;TEST BIT 9 FOR RESET CLR
3992 016622 001401 BEQ 7$ ;BIT 9 IS OK

```

```

3993 016624 104003          HLT      3          ;BIT FAILED TO CLEAR
3994 016626 010513          7$:    MOV     R5,(R3) ;SET BIT 9
3995 016630 005013          CLR     (R3)       ;CLR RECEIVER BUFFER REGISTER
3996 016632 011304          MOV     (R3),R4    ;READ THE RECEIVER BUFFER REGISTER
3997 016634 005005          CLR     R5         ;SET "EXPECTED"
3998 016636 020504          CMP     R5,R4     ;R5=GOOD,R4=?
3999 016640 001401          BEQ    10$        ;BIT 9 IS OK
4000 016642 104003          HLT     3          ;BIT FAILED TO CLEAR
4001 016644 104400          10$:   SCOPE      ;SCOPE THIS TEST

```

```

:***** TEST 104 *****
:*RECEIVER BUFFER REGISTER BIT 10 READ ONLY DEVICE RESET AND CLEAR TEST
:*WRITE BIT 10 A ONE AND VERIFY A ZERO IS READ BACK
:*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
:*****

```

```

:*****
:TEST 104
:*****

```

```

4016 016646 012737 000104 001226 TST104: MOV     #104,#TSTNO
4017 016654 012737 016760 001216      MOV     #TST105,NEXT
4018 016662 013703          MOV     RXDBUF,R3 ;GET THE RECEIVER BUFFER REGISTER
4019 016666 012705          MOV     #BIT10,R5 ;GET BIT 10
4020 016672 010513          MOV     R5,(R3)   ;WRITE BIT 10 TO RECEIVER BUFFER REGISTER
4021 016674 011304          MOV     (R3),R4   ;READ RECEIVER BUFFER REGISTER BACK
4022 016676 005005          CLR     R5         ;SET "EXPECTED"
4023 016700 020504          CMP     R5,R4     ;R5=GOOD,R4=?
4024 016702 001401          BEQ    5$         ;BIT 10 IS OK
4025 016704 104003          HLT     3          ;BIT FAILED TO CLR
4026 016706 012705 002000          5$:    MOV     #BIT10,R5 ;RELOAD THE BIT
4027 016712 010513          MOV     R5,(R3)   ;WRITE BIT 10 TO THE REG
4028 016714 052777 000400 162470      BIS     #MRESET,#TXCSR ;RESET THE DEVICE
4029 016722 004737 004772          JSR     PC,SMALL  ;WAIT FOR RESET TO FINISH
4030 016726 011304          MOV     (R3),R4   ;GET BIT 10
4031 016730 032704 002000          BIT     #BIT10,R4 ;TEST BIT 10 FOR RESET CLR
4032 016734 001401          BEQ    7$         ;BIT 10 IS OK
4033 016736 104003          HLT     3          ;BIT FAILED TO CLEAR
4034 016740 010513          7$:    MOV     R5,(R3)   ;SET BIT 10
4035 016742 005013          CLR     (R3)       ;CLR RECEIVER BUFFER REGISTER
4036 016744 011304          MOV     (R3),R4   ;READ THE RECEIVER BUFFER REGISTER
4037 016746 005005          CLR     R5         ;SET "EXPECTED"
4038 016750 020504          CMP     R5,R4     ;R5=GOOD,R4=?
4039 016752 001401          BEQ    10$        ;BIT 10 IS OK
4040 016754 104003          HLT     3          ;BIT FAILED TO CLEAR
4041 016756 104400          10$:   SCOPE      ;SCOPE THIS TEST

```

```

:***** TEST 105 *****
:*RECEIVER BUFFER REGISTER BIT 12 READ ONLY DEVICE RESET AND CLEAR TEST
:*WRITE BIT 12 A ONE AND VERIFY A ZERO IS READ BACK
:*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
:*****

```

```

4042
4043
4044
4045
4046
4047
4048

```

4049  
4050  
4051  
4052  
4053  
4054  
4055  
4056  
4057  
4058  
4059  
4060  
4061  
4062  
4063  
4064  
4065  
4066  
4067  
4068  
4069  
4070  
4071  
4072  
4073  
4074  
4075  
4076  
4077  
4078  
4079  
4080  
4081  
4082  
4083  
4084  
4085  
4086  
4087  
4088  
4089  
4090  
4091  
4092  
4093  
4094  
4095  
4096  
4097  
4098  
4099  
4100  
4101  
4102  
4103  
4104

016760 012737 000105 001226  
016766 012737 017072 001216  
016774 013703 001406  
017000 012705 010000  
017004 010513  
017006 011304  
017010 005005  
017012 020504  
017014 001401  
017016 104003  
017020 012705 010000  
017024 010513  
017026 052777 000400 162356  
017034 004737 004772  
017040 011304  
017042 032704 010000  
017046 001401  
017050 104003  
017052 010513  
017054 005013  
017056 011304  
017060 005005  
017062 020504  
017064 001401  
017066 104003  
017070 104400

```
*****  
*  
: TEST 105  
*  
*****  
*****  
TST105: MOV #105, @TSTNO  
MOV #TST106, NEXT  
MOV RXDBUF, R3 ;GET THE RECEIVER BUFFER REGISTER  
MOV #BIT12, R5 ;GET BIT 12  
MOV R5, (R3) ;WRITE BIT 12 TO RECEIVER BUFFER REGISTER  
MOV (R3), R4 ;READ RECEIVER BUFFER REGISTER BACK  
CLR R5 ;SET "EXPECTED"  
CMP R5, R4 ;R5=GOOD, R4= ?  
BEQ 5$ ;BIT 12 IS OK  
HLT 3 ;BIT FAILED TO CLR  
5$: MOV #BIT12, R5 ;RELOAD THE BIT  
MOV R5, (R3) ;WRITE BIT 12 TO THE REG  
BIS #MRESET, @TXCSR ;RESET THE DEVICE  
JSR PC, SMALL ;WAIT FOR RESET TO FINISH  
MOV (R3), R4 ;GET BIT 12  
BIT #BIT12, R4 ;TEST BIT 12 FOR RESET CLR  
BEQ 7$ ;BIT 12 IS OK  
HLT 3 ;BIT FAILED TO CLEAR  
7$: MOV R5, (R3) ;SET BIT 12  
CLR (R3) ;CLR RECEIVER BUFFER REGISTER  
MOV (R3), R4 ;READ THE RECEIVER BUFFER REGISTER  
CLR R5 ;SET "EXPECTED"  
CMP R5, R4 ;R5=GOOD, R4= ?  
BEQ 10$ ;BIT 12 IS OK  
HLT 3 ;BIT FAILED TO CLEAR  
10$: SCOPE ;SCOPE THIS TEST
```

\*\*\*\*\* TEST 106 \*\*\*\*\*  
\*RECEIVER BUFFER REGISTER BIT 14 READ ONLY DEVICE RESET AND CLEAR TEST  
\*WRITE BIT 14 A ONE AND VERIFY A ZERO IS READ BACK  
\*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
\*\*\*\*\*

4096  
4097  
4098  
4099  
4100  
4101  
4102  
4103  
4104

017072 012737 000106 001226  
017100 012737 017204 001216  
017106 013703 001406  
017112 012705 040000  
017116 010513  
017120 011304  
017122 005005  
017124 020504  
017126 001401

```
*****  
*  
: TEST 106  
*  
*****  
*****  
TST106: MOV #106, @TSTNO  
MOV #TST107, NEXT  
MOV RXDBUF, R3 ;GET THE RECEIVER BUFFER REGISTER  
MOV #BIT14, R5 ;GET BIT 14  
MOV R5, (R3) ;WRITE BIT 14 TO RECEIVER BUFFER REGISTER  
MOV (R3), R4 ;READ RECEIVER BUFFER REGISTER BACK  
CLR R5 ;SET "EXPECTED"  
CMP R5, R4 ;R5=GOOD, R4= ?  
BEQ 5$ ;BIT 14 IS OK
```

```

4105 017130 104003          HLT      3          ;BIT FAILED TO CLR
4106 017132 012705 040000 5$:  MOV     #BIT14,R5 ;RELOAD THE BIT
4107 017136 010513          MOV     R5,(R3)   ;WRITE BIT 14 TO THE REG
4108 017140 052777 000400 162244 BIS     #MRESET,@TXCSR ;RESET THE DEVICE
4109 017146 004737 004772          JSR     PC,SMALL  ;WAIT FOR RESET TO FINISH
4110 017152 011304          MOV     (R3),R4   ;GET BIT 14
4111 017154 032704 040000          BIT     #BIT14,R4 ;TEST BIT 14 FOR RESET CLR
4112 017160 001401          BEQ     7$        ;BIT 14 IS OK
4113 017162 104003          HLT      3          ;BIT FAILED TO CLEAR
4114 017164 010513          7$:  MOV     R5,(R3)   ;SET BIT 14
4115 017166 005013          CLR     (R3)      ;CLR RECEIVER BUFFER REGISTER
4116 017170 011304          MOV     (R3),R4   ;READ THE RECEIVER BUFFER REGISTER
4117 017172 005005          CLR     R5        ;SET "EXPECTED"
4118 017174 020504          CMP     R5,R4     ;R5=GOOD,R4=?
4119 017176 001401          BEQ     10$       ;BIT 14 IS OK
4120 017200 104003          HLT      3          ;BIT FAILED TO CLEAR
4121 017202 104400          10$:  SCOPE        ;SCOPE THIS TEST

```

```

***** TEST 107 *****
*RECEIVER BUFFER REGISTER BIT 15 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 15 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
*
* TEST 107
*
*****

```

```

4130
4131
4132
4133
4134
4135
4136 017204 012737 000107 001226 TST107: MOV     #107,@TSTNO
4137 017212 012737 017316 001216          MOV     #TST10,NEXT
4138 017220 013703          MOV     RXDBUF,R3 ;GET THE RECEIVER BUFFER REGISTER
4139 017224 012705 100000          MOV     #BIT15,R5 ;GET BIT 15
4140 017230 010513          MOV     R5,(R3)   ;WRITE BIT 15 TO RECEIVER BUFFER REGISTER
4141 017232 011304          MOV     (R3),R4   ;READ RECEIVER BUFFER REGISTER BACK
4142 017234 005005          CLR     R5        ;SET "EXPECTED"
4143 017236 020504          CMP     R5,R4     ;R5=GOOD,R4=?
4144 017240 001401          BEQ     5$        ;BIT 15 IS OK
4145 017242 104003          HLT      3          ;BIT FAILED TO CLR
4146 017244 012705 100000 5$:  MOV     #BIT15,R5 ;RELOAD THE BIT
4147 017250 010513          MOV     R5,(R3)   ;WRITE BIT 15 TO THE REG
4149 017252 052777 000400 162132 BIS     #MRESET,@TXCSR ;RESET THE DEVICE
4149 017260 004737 004772          JSR     PC,SMALL  ;WAIT FOR RESET TO FINISH
4150 017264 011304          MOV     (R3),R4   ;GET BIT 15
4151 017266 032704 100000          BIT     #BIT15,R4 ;TEST BIT 15 FOR RESET CLR
4152 017272 001401          BEQ     7$        ;BIT 15 IS OK
4153 017274 104003          HLT      3          ;BIT FAILED TO CLEAR
4154 017276 010513          7$:  MOV     R5,(R3)   ;SET BIT 15
4155 017300 005013          CLR     (R3)      ;CLR RECEIVER BUFFER REGISTER
4156 017302 011304          MOV     (R3),R4   ;READ THE RECEIVER BUFFER REGISTER
4157 017304 005005          CLR     R5        ;SET "EXPECTED"
4158 017306 020504          CMP     R5,R4     ;R5=GOOD,R4=?
4159 017310 001401          BEQ     10$       ;BIT 15 IS OK
4160 017312 104003          HLT      3          ;BIT FAILED TO CLEAR

```

```

4161 017314 104400          10$: SCOPE                ;SCOPE THIS TEST
4162
4163          ;***** TEST 110 *****
4164          ;*THIS TEST VERIFIES BIT7 OF THE TRANSMITTER CONTROL REGISTER
4165          ;*TEST THAT BIT 7 SETS AFTER A RESET AND MRESET
4166          ;*VERIFY THAT WRITING THE LOW BYTE OF
4167          ;*THE TRANSMITTER BUFFER REGISTER CLEARS BIT 7
4168          ;*****
4169
4170          ;*****
4171          ;*
4172          ;* TEST 110
4173          ;*
4174          ;*****
4175          ;*****
4176 017316 012737 000110 001226 TST110: MOV      #110, @TSTNO
4177 017324 012737 017424 001216      MOV      @TST111, NEXT
4178 017332 013702 001414          MOV      TXDBUF, R2          ;LOAD SECOND REG
4179 017336 013703 001412          MOV      TXCSR, R3         ;GET THE TRANSMITTER CONTROL REGISTER
4180 017342 012705 000200          MOV      #BIT7, R5        ;SET "EXPECTED"
4181 017346 000005          RESET
4182 017350 011304          MOV      (R3), R4         ;GET THE BIT
4183 017352 020504          CMP      R5, R4          ;R5=GOOD, R4=?
4184 017354 001401          BEQ     1$              ;ARE THEY THE SAME?
4185 017356 104003          HLT     3               ;NO--REPORT THE ERROR
4186 017360 005005          1$: CLR      R5          ;SET "EXPECTED"
4187 017362 005012          CLR      (R2)          ;WRITE THE LOW BYTE OF THE TXDBUF
4188 017364 011304          MOV      (R3), R4         ;READ BACK BIT 7
4189 017366 020504          CMP      R5, R4         ;R5=GOOD, R4=?
4190 017370 001401          BEQ     2$              ;ARE THEY THE SAME?
4191 017372 104003          HLT     3               ;NO-BIT 7 FAILED TO CLEAR
4192 017374 012705 000200          2$: MOV      #BIT7, R5        ;SET "EXPECTED"
4193 017400 052777 000400 162004      BIS      #MRESET, @TXCSR ;RESET THE DEVICE
4194 017406 004737 004772          JSR     PC, SMALL        ;WAIT FOR RESET TO FINISH
4195 017412 011304          MOV      (R3), R4         ;READ BACK BIT 7
4196 017414 020504          CMP      R5, R4         ;R5=GOOD, R4=?
4197 017416 001401          BEQ     3$              ;BRANCH IF BIT 7 OK
4198 017420 104003          HLT     3               ;BIT 7 FAILED TO SET AFTER A MRESET
4199 017422 104400          3$: SCOPE                ;SCOPE THIS TEST
4200
4201
4202
4203          ;***** TEST 111 *****
4204          ;*TRANSMITTER CONTROL REGISTER BIT 9 READ ONLY DEVICE RESET AND CLEAR TEST
4205          ;*WRITE BIT 9 A ONE AND VERIFY A ZERO IS READ BACK
4206          ;*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
4207          ;*****
4208
4209          ;*****
4210          ;*
4211          ;* TEST 111
4212          ;*
4213          ;*****
4214          ;*****
4215 017424 012737 000111 001226 TST111: MOV      #111, @TSTNO
4216 017432 012737 017546 001216      MOV      @TST112, NEXT
    
```

```

4217 017440 013703 001412      MOV      TXCSR,R3      ;GET THE TRANSMITTER CONTROL REGISTER
4218 017444 012705 001000      MOV      #BIT9,R5     ;GET BIT 9
4219 017450 010513              MOV      R5,(R3)      ;WRITE BIT 9 TO TRANSMITTER CONTROL REGISTER
4220 017452 011304              MOV      (R3),R4      ;READ TRANSMITTER CONTROL REGISTER BACK
4221 017454 005005              CLR      R5           ;SET "EXPECTED"
4222 017456 042704 000200      BIC      #BIT7,R4     ;CLEAR UNWANTED BITS
4223 017462 020504              CMP      R5,R4        ;R5=GOOD,R4=?
4224 017464 001401              BEQ      5$           ;BIT 9 IS OK
4225 017466 104003              HLT      3            ;BIT FAILED TO CLR
4226 017470 012705 001000      5$:      MOV      #BIT9,R5     ;RELOAD THE BIT
4227 017474 010513              MOV      R5,(R3)      ;WRITE BIT 9 TO THE REG
4228 017476 052777 000400 161706  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
4229 017504 004737 004772      JSR      PC,SMALL     ;WAIT FOR RESET TO FINISH
4230 017510 011304              MOV      (R3),R4      ;GET BIT 9
4231 017512 032704 001000      BIT      #BIT9,R4     ;TEST BIT 9 FOR RESET CLR
4232 017516 001401              BEQ      7$           ;BIT 9 IS OK
4233 017520 104003              HLT      3            ;BIT FAILED TO CLEAR
4234 017522 010513              7$:      MOV      R5,(R3)     ;SET BIT 9
4235 017524 005013              CLR      (R3)         ;CLR TRANSMITTER CONTROL REGISTER
4236 017526 011304              MOV      (R3),R4      ;READ THE TRANSMITTER CONTROL REGISTER
4237 017530 005005              CLR      R5           ;SET "EXPECTED"
4238 017532 042704 000200      BIC      #BIT7,R4     ;CLEAR UNWANTED BITS
4239 017536 020504              CMP      R5,R4        ;R5=GOOD,R4=?
4240 017540 001401              BEQ      10$          ;BIT 9 IS OK
4241 017542 104003              HLT      3            ;BIT FAILED TO CLEAR
4242 017544 104400      10$:     SCOPE         ;SCOPE THIS TEST

```

```

***** TEST 112 *****
*TRANSMITTER CONTROL REGISTER BIT 14 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 14 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
:TEST 112
*****
*****

```

```

4257 017546 012737 000112 001226  TST112: MOV      #112,@TSTNO
4258 017554 012737 017670 001216  MOV      #TST113,NEXT
4259 017562 013703 001412      MOV      TXCSR,R3     ;GET THE TRANSMITTER CONTROL REGISTER
4260 017566 012705 040000      MOV      #BIT14,R5    ;GET BIT 14
4261 017572 010513              MOV      R5,(R3)      ;WRITE BIT 14 TO TRANSMITTER CONTROL REGISTER
4262 017574 011304              MOV      (R3),R4      ;READ TRANSMITTER CONTROL REGISTER BACK
4263 017576 005005              CLR      R5           ;SET "EXPECTED"
4264 017600 042704 000200      BIC      #BIT7,R4     ;CLEAR UNWANTED BITS
4265 017604 020504              CMP      R5,R4        ;R5=GOOD,R4=?
4266 017606 001401              BEQ      5$           ;BIT 14 IS OK
4267 017610 104003              HLT      3            ;BIT FAILED TO CLR
4268 017612 012705 040000      5$:      MOV      #BIT14,R5    ;RELOAD THE BIT
4269 017616 010513              MOV      R5,(R3)      ;WRITE BIT 14 TO THE REG
4270 017620 052777 000400 161564  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
4271 017626 004737 004772      JSR      PC,SMALL     ;WAIT FOR RESET TO FINISH
4272 017632 011304              MOV      (R3),R4      ;GET BIT 14

```



```

4273 017634 032704 040000 BIT #BIT14,R4 ;TEST BIT 14 FOR RESET CLR
4274 017640 001401 BEQ 7$ ;BIT 14 IS OK
4275 017642 104003 HLT 3 ;BIT FAILED TO CLEAR
4276 017644 010513 7$: MOV R5,(R3) ;SET BIT 14
4277 017646 005013 CLR (R3) ;CLR TRANSMITTER CONTROL REGISTER
4278 017650 011304 MOV (R3),R4 ;READ THE TRANSMITTER CONTROL REGISTER
4279 017652 005005 CLR R5 ;SET "EXPECTED"
4280 017654 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
4281 017660 020504 CMP R5,R4 ;R5=GOOD,R4=?
4282 017662 001401 BEQ 10$ ;BIT 14 IS OK
4283 017664 104003 HLT 3 ;BIT FAILED TO CLEAR
4284 017666 104400 10$: SCOPE ;SCOPE THIS TEST

```

```

***** TEST 113 *****
*TRANSMITTER CONTROL REGISTER BIT 15 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 15 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
TEST 113
*****

```

```

4299 017670 012737 000113 001226 TST113: MOV #113,@TSTNO
4300 017676 012737 020012 001216 MOV #TST114,NEXT
4301 017704 013703 MOV TXCSR,R3 ;GET THE TRANSMITTER CONTROL REGISTER
4302 017710 012705 100000 MOV #BIT15,R5 ;GET BIT 15
4303 017714 010513 MOV R5,(R3) ;WRITE BIT 15 TO TRANSMITTER CONTROL REGISTER
4304 017716 011304 MOV (R3),R4 ;READ TRANSMITTER CONTROL REGISTER BACK
4305 017720 005005 CLR R5 ;SET "EXPECTED"
4306 017722 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
4307 017726 020504 CMP R5,R4 ;R5=GOOD,R4=?
4308 017730 001401 BEQ 5$ ;BIT 15 IS OK
4309 017732 104003 HLT 3 ;BIT FAILED TO CLR
4310 017734 012705 100000 5$: MOV #BIT15,R5 ;RELOAD THE BIT
4311 017740 010513 MOV R5,(R3) ;WRITE BIT 15 TO THE REG
4312 017742 052777 000400 161442 BIS #MRESET,@TXCSR ;RESET THE DEVICE
4313 017750 004737 004772 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
4314 017754 011304 MOV (R3),R4 ;GET BIT 15
4315 017756 032704 100000 BIT #BIT15,R4 ;TEST BIT 15 FOR RESET CLR
4316 017762 001401 BEQ 7$ ;BIT 15 IS OK
4317 017764 104003 HLT 3 ;BIT FAILED TO CLEAR
4318 017766 010513 7$: MOV R5,(R3) ;SET BIT 15
4319 017770 005013 CLR (R3) ;CLR TRANSMITTER CONTROL REGISTER
4320 017772 011304 MOV (R3),R4 ;READ THE TRANSMITTER CONTROL REGISTER
4321 017774 005005 CLR R5 ;SET "EXPECTED"
4322 017776 042704 000200 BIC #BIT7,R4 ;CLEAR UNWANTED BITS
4323 020002 020504 CMP R5,R4 ;R5=GOOD,R4=?
4324 020004 001401 BEQ 10$ ;BIT 15 IS OK
4325 020006 104003 HLT 3 ;BIT FAILED TO CLEAR
4326 020010 104400 10$: SCOPE ;SCOPE THIS TEST
4327
4328

```

```

4329
4330
4331
4332
4333
4334
4335
4336
4337
4338
4339
4340
4341 020012 012737 000114 001226
4342 020020 012737 020124 001216
4343 020026 013703 001414
4344 020032 012705 010000
4345 020036 010513
4346 020040 011304
4347 020042 005005
4348 020044 020504
4349 020046 001401
4350 020050 104003
4351 020052 012705 010000
4352 020056 010513
4353 020060 052777 000400 161324
4354 020066 004737 004772
4355 020072 011304
4356 020074 032704 010000
4357 020100 001401
4358 020102 104003
4359 020104 010513
4360 020106 005013
4361 020110 011304
4362 020112 005005
4363 020114 020504
4364 020116 001401
4365 020120 104003
4366 020122 104400
4367
4368
4369
4370
4371
4372
4373
4374
4375
4376
4377
4378
4379
4380
4381 020124 012737 000115 001226
4382 020132 012737 020236 001216
4383 020140 013703 001414
4384 020144 012705 020000

```

```

***** TEST 114 *****
*TRANSMITTER BUFFER REGISTER BIT 12 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 12 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
*
TEST 114
*
*****

```

```

*****
TST114: MOV #114, @TSTNO
MOV #TST115, NEXT
MOV TXDBUF, R3 ;GET THE TRANSMITTER BUFFER REGISTER
MOV #BIT12, R5 ;GET BIT 12
MOV R5, (R3) ;WRITE BIT 12 TO TRANSMITTER BUFFER REGISTER
MOV (R3), R4 ;READ TRANSMITTER BUFFER REGISTER BACK
CLR R5 ;SET "EXPECTED"
CMP R5, R4 ;R5=GOOD, R4= ?
BEQ 5$ ;BIT 12 IS OK
HLT 3 ;BIT FAILED TO CLR
5$: MOV #BIT12, R5 ;RELOAD THE BIT
MOV R5, (R3) ;WRITE BIT 12 TO THE REG
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
MOV (R3), R4 ;GET BIT 12
BIT #BIT12, R4 ;TEST BIT 12 FOR RESET CLR
BEQ 7$ ;BIT 12 IS OK
HLT 3 ;BIT FAILED TO CLEAR
7$: MOV R5, (R3) ;SET BIT 12
CLR (R3) ;CLR TRANSMITTER BUFFER REGISTER
MOV (R3), R4 ;READ THE TRANSMITTER BUFFER REGISTER
CLR R5 ;SET "EXPECTED"
CMP R5, R4 ;R5=GOOD, R4= ?
BEQ 10$ ;BIT 12 IS OK
HLT 3 ;BIT FAILED TO CLEAR
10$: SCOPE ;SCOPE THIS TEST

```

```

***** TEST 115 *****
*TRANSMITTER BUFFER REGISTER BIT 13 READ ONLY DEVICE RESET AND CLEAR TEST
*WRITE BIT 13 A ONE AND VERIFY A ZERO IS READ BACK
*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
*****

```

```

*****
*
TEST 115
*
*****

```

```

*****
TST115: MOV #115, @TSTNO
MOV #TST116, NEXT
MOV TXDBUF, R3 ;GET THE TRANSMITTER BUFFER REGISTER
MOV #BIT13, R5 ;GET BIT 13

```

4385	020150	010513			MOV	R5, (R3)	:WRITE BIT 13 TO TRANSMITTER BUFFER REGISTER
4386	020152	011304			MOV	(R3), R4	:READ TRANSMITTER BUFFER REGISTER BACK
4387	020154	005005			CLR	R5	:SET "EXPECTED"
4388	020156	020504			CMP	R5, R4	:R5=GOOD, R4= ?
4389	020160	001401			BEQ	5\$	:BIT 13 IS OK
4390	020162	104003			HLT	3	:BIT FAILED TO CLR
4391	020164	012705	020000	5\$:	MOV	#BIT13, R5	:RELOAD THE BIT
4392	020170	010513			MOV	R5, (R3)	:WRITE BIT 13 TO THE REG
4393	020172	052777	000400	161212	BIS	#MRESET, @TXCSR	:RESET THE DEVICE
4394	020200	004737	004772		JSR	PC, SMALL	:WAIT FOR RESET TO FINISH
4395	020204	011304			MOV	(R3), R4	:GET BIT 13
4396	020206	032704	020000		BIT	#BIT13, R4	:TEST BIT 13 FOR RESET CLR
4397	020212	001401			BEQ	7\$	:BIT 13 IS OK
4398	020214	104003			HLT	3	:BIT FAILED TO CLEAR
4399	020216	010513		7\$:	MOV	R5, (R3)	:SET BIT 13
4400	020220	005013			CLR	(R3)	:CLR TRANSMITTER BUFFER REGISTER
4401	020222	011304			MOV	(R3), R4	:READ THE TRANSMITTER BUFFER REGISTER
4402	020224	005005			CLR	R5	:SET "EXPECTED"
4403	020226	020504			CMP	R5, R4	:R5=GOOD, R4= ?
4404	020230	001401			BEQ	10\$	:BIT 13 IS OK
4405	020232	104003			HLT	3	:BIT FAILED TO CLEAR
4406	020234	104400		10\$:	SCOPE		:SCOPE THIS TEST

\*\*\*\*\* TEST 116 \*\*\*\*\*  
 \*TRANSMITTER BUFFER REGISTER BIT 14 READ ONLY DEVICE RESET AND CLEAR TEST  
 \*WRITE BIT 14 A ONE AND VERIFY A ZERO IS READ BACK  
 \*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.  
 \*\*\*\*\*

\*\*\*\*\*  
 \*  
 TEST 116  
 \*  
 \*\*\*\*\*

4420					*****			
4421	020236	012737	000116	001226	TST116:	MOV	#116, @TSTNO	
4422	020244	012737	020350	001216		MOV	#TST117, NEXT	
4423	020252	013703	001414			MOV	TXDBUF, R3	:GET THE TRANSMITTER BUFFER REGISTER
4424	020256	012705	04000C			MOV	#BIT14, R5	:GET BIT 14
4425	020262	010513				MOV	R5, (R3)	:WRITE BIT 14 TO TRANSMITTER BUFFER REGISTER
4426	020264	011304				MOV	(R3), R4	:READ TRANSMITTER BUFFER REGISTER BACK
4427	020266	005005				CLR	R5	:SET "EXPECTED"
4428	020270	020504				CMP	R5, R4	:R5=GOOD, R4= ?
4429	020272	001401				BEQ	5\$	:BIT 14 IS OK
4430	020274	104003				HLT	3	:BIT FAILED TO CLR
4431	020276	012705	040000	5\$:	MOV	#BIT14, R5	:RELOAD THE BIT	
4432	020302	010513			MOV	R5, (R3)	:WRITE BIT 14 TO THE REG	
4433	020304	052777	000400	161100	BIS	#MRESET, @TXCSR	:RESET THE DEVICE	
4434	020312	004737	004772		JSR	PC, SMALL	:WAIT FOR RESET TO FINISH	
4435	020316	011304			MOV	(R3), R4	:GET BIT 14	
4436	020320	032704	040000		BIT	#BIT14, R4	:TEST BIT 14 FOR RESET CLR	
4437	020324	001401			BEQ	7\$	:BIT 14 IS OK	
4438	020326	104003			HLT	3	:BIT FAILED TO CLEAR	
4439	020330	010513		7\$:	MOV	R5, (R3)	:SET BIT 14	
4440	020332	005013			CLR	(R3)	:CLR TRANSMITTER BUFFER REGISTER	

```

4441 020334 011304      MOV      (R3),R4      ;READ THE TRANSMITTER BUFFER REGISTER
4442 020336 005005      CLR      R5          ;SET "EXPECTED"
4443 020340 020504      CMP      R5,R4       ;R5=GOOD,R4=?
4444 020342 001401      BEQ      10$         ;BIT 14 IS OK
4445 020344 104003      HLT      3           ;BIT FAILED TO CLEAR
4446 020346 104400      10$:     SCOPE       ;SCOPE THIS TEST
  
```

```

4447
4448
4449
4450
4451
4452
4453
4454
4455
4456
4457
4458
4459
4460
4461
4462
4463
4464
4465
4466
4467
4468
4469
4470
4471
4472
4473
4474
4475
4476
4477
4478
4479
4480
4481
4482
4483
4484
4485
4486
4487
4488
4489
4490
4491
4492
4493
4494
4495
4496
  
```

```

:***** TEST 117 *****
:*TRANSMITTER BUFFER REGISTER BIT 15 READ ONLY DEVICE RESET AND CLEAR TEST
:*WRITE BIT 15 A ONE AND VERIFY A ZERO IS READ BACK
:*REPEAT FOR DEVICE RESET AND CLR INSTRUCTIONS.
:*****
  
```

```

:*****
:
: TEST 117
:
:*****
  
```

```

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

```

4461 020350 012737 000117 001226 TST117: MOV      #117,2#TSTNO
4462 020356 012737 020462 001216      MOV      #TST120,NEXT
4463 020364 013703 001414      MOV      TXDBUF,R3      ;GET THE TRANSMITTER BUFFER REGISTER
4464 020370 012705 100000      MOV      #BIT15,R5      ;GET BIT 15
4465 020374 010513      MOV      R5,(R3)        ;WRITE BIT 15 TO TRANSMITTER BUFFER REGISTER
4466 020376 011304      MOV      (R3),R4        ;READ TRANSMITTER BUFFER REGISTER BACK
4467 020400 005005      CLR      R5             ;SET "EXPECTED"
4468 020402 020504      CMP      R5,R4         ;R5=GOOD,R4=?
4469 020404 001401      BEQ      5$            ;BIT 15 IS OK
4470 020406 104003      HLT      3             ;BIT FAILED TO CLR
4471 020410 012705 100000      5$:     MOV      #BIT15,R5 ;RELOAD THE BIT
4472 020414 010513      MOV      R5,(R3)        ;WRITE BIT 15 TO THE REG
4473 020416 052777 000400 160766      BIS      #MRESET,2TXCSR ;RESET THE DEVICE
4474 020424 004737 004772      JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
4475 020430 011304      MOV      (R3),R4        ;GET BIT 15
4476 020432 032704 100000      BIT      #BIT15,R4      ;TEST BIT 15 FOR RESET CLR
4477 020436 001401      BEQ      7$            ;BIT 15 IS OK
4478 020440 104003      HLT      3             ;BIT FAILED TO CLEAR
4479 020442 010513      7$:     MOV      R5,(R3)    ;SET BIT 15
4480 020444 005013      CLR      (R3)          ;CLR TRANSMITTER BUFFER REGISTER
4481 020446 011304      MOV      (R3),R4        ;READ THE TRANSMITTER BUFFER REGISTER
4482 020450 005005      CLR      R5             ;SET "EXPECTED"
4483 020452 020504      CMP      R5,R4         ;R5=GOOD,R4=?
4484 020454 001401      BEQ      10$           ;BIT 15 IS OK
4485 020456 104003      HLT      3             ;BIT FAILED TO CLEAR
4486 020460 104400      10$:     SCOPE       ;SCOPE THIS TEST
  
```

```

:***** TEST 120 *****
:*PARAMETER STATUS REGISTER BIT 9 WRITE ONLY DEVICE RESET AND CLEAR TEST
:*TEST THAT BIT 9 CANNOT BE WRITTEN AND READ
:*BACK THE SAME. READ BIT 9, COMPLEMENT IT AND WRITE IT
:*VERIFYING THAT IT DID NOT CHANGE. WRITE BIT 9.
:*DO A DEVICE RESET AND VERIFY THE BIT WAS CLEARED.
:*REPEAT FOR A CLR INSTRUCTION.
:*****
  
```

4544  
4545  
4546  
4547  
4548  
4549  
4550  
4551  
4552  
4553  
4554  
4555  
4556  
4557  
4558  
4559  
4560  
4561  
4562  
4563  
4564  
4565  
4566  
4567  
4568  
4569  
4570  
4571  
4572  
4573  
4574  
4575  
4576  
4577  
4578  
4579  
4580  
4581  
4582  
4583  
4584  
4585  
4586  
4587  
4588  
4589  
4590  
4591  
4592  
4593  
4594  
4595  
4596  
4597  
4598  
4599  
4600

020462 012737 000120 001226  
020470 012737 020610 001216  
020476 013703 001410  
020502 012705 001000  
020506 011305  
020510 010504  
020512 032704 001000  
020516 001401  
020520 104003  
020522 005104  
020524 010413  
020526 011304  
020530 020504  
020532 001401  
020534 104003  
020536 012705 001000  
020542 010513  
020544 052777 000400 160640  
020552 004737 004772  
020556 011304  
020560 032704 001000  
020564 001401  
020566 104003  
020570 010513  
020572 005013  
020574 011304  
020576 005005  
020600 020504  
020602 001401  
020604 104003  
020606 104400

```

:*****
:TEST 120
:*****
:*****
TST120: MOV #120, @TSTNO
MOV @TST121, NEXT
MOV PARCSR, R3 ;GET THE PARAMETER STATUS REGISTER
MOV #BIT9, R5 ;GET BIT 9
MOV (R3), R5 ;READ THE REGISTER
MOV R5, R4 ;SAVE THE BIT
BIT #BIT9, R4 ;CHECK BIT 9
BEQ 1$ ;BIT 9 IS OK
HLT 3 ;BIT FAILED TO CLEAR
1$: COM R4 ;REVERSE THE BIT
MOV R4, (R3) ;WRITE THE BIT TO THE REG
MOV (R3), R4 ;READ IT BACK
CMP R5, R4 ;R5=GOOD, R4= ?
BEQ 3$ ;BR IF OK
HLT 3 ;COMPARE ERROR
3$: MOV #BIT9, R5 ;LOAD THE BIT
MOV R5, (R3) ;WRITE THE BIT TO THE REG
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
MOV (R3), R4 ;GET BIT 9
BIT #BIT9, R4 ;TEST BIT 9 FOR 0 AFTER RESET
BEQ 4$ ;BIT 9 IS OK
HLT 3 ;BIT IS NOT A 0
4$: MOV R5, (R3) ;WRITE THE BIT TO THE REG
CLR (R3) ;CLR THE PARAMETER STATUS REGISTER
MOV (R3), R4 ;GET THE PARAMETER STATUS REGISTER
CLR R5 ;SET "EXPECTED"
CMP R5, R4 ;R5=GOOD, R4= ?
BEQ 10$ ;BIT 9 IS OK
HLT 3 ;BIT FAILED TO CLEAR
10$: SCOPE ;SCOPE THIS TEST

```

```

:***** TEST 121 *****
:*PARAMETER STATUS REGISTER BIT 12 WRITE ONLY DEVICE RESET AND CLEAR TEST
:*TEST THAT BIT 12 CANNOT BE WRITTEN AND READ
:*BACK THE SAME. READ BIT 12, COMPLEMENT IT AND WRITE IT
:*VERIFYING THAT IT DID NOT CHANGE. WRITE BIT 12.
:*DO A DEVICE RESET AND VERIFY THE BIT WAS CLEARED.
:*REPEAT FOR A CLR INSTRUCTION.
:*****

```

```

:*****
:TEST 121
:*****
:*****
TST121: MOV #121, @TSTNO

```

```

4553 020616 012737 020736 001216 MUV #TST122,NEXT
4554 020624 013703 001410 MOV PARCSR,R3 ;GET THE PARAMETER STATUS REGISTER
4555 020630 012705 010000 MOV #BIT12,R5 ;GET BIT 12
4556 020634 011305 MOV (R3),R5 ;READ THE REGISTER
4557 020636 010504 MOV R5,R4 ;SAVE THE BIT
4558 020640 032704 010000 BIT #BIT12,R4 ;CHECK BIT 12
4559 020644 001401 BEQ IS ;BIT 12 IS OK
4560 020646 104003 HLT 3 ;BIT FAILED TO CLEAR
4561 020650 005104 15: COM R4 ;REVERSE THE BIT
4562 020652 010413 MOV R4,(R3) ;WRITE THE BIT TO THE REG
4563 020654 011304 MOV (R3),R4 ;READ IT BACK
4564 020656 020504 CMP R5,R4 ;R5=GOOD, R4= ?
4565 020660 001401 BEQ 35 ;BR IF OK
4566 020662 104003 HLT 3 ;COMPARE ERROR
4567 020664 012705 010000 35: MOV #BIT12,R5 ;LOAD THE BIT
4568 020670 010513 MOV R5,(R3) ;WRITE THE BIT TO THE REG
4569 020672 052777 000400 160512 BIS #MRESET,DTXCSR ;RESET THE DEVICE
4570 020700 004737 004772 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
4571 020704 011304 MOV (R3),R4 ;GET BIT 12
4572 020706 032704 010000 BIT #BIT12,R4 ;TEST BIT 12 FOR 0 AFTER RESET
4573 020712 001401 BEQ 45 ;BIT 12 IS OK
4574 020714 104003 HLT 3 ;BIT IS NOT A 0
4575 020716 010513 45: MOV R5,(R3) ;WRITE THE BIT TO THE REG
4576 020720 005013 CLR (R3) ;CLR THE PARAMETER STATUS REGISTER
4577 020722 011304 MOV (R3),R4 ;GET THE PARAMETER STATUS REGISTER
4578 020724 005005 CLR R5 ;SET "EXPECTED"
4579 020726 020504 CMP R5,R4 ;R5=GOOD, R4= ?
4580 020730 001401 BEQ 105 ;BIT 12 IS OK
4581 020732 104003 HLT 3 ;BIT FAILED TO CLEAR
4582 020734 104400 105: SCOPE ;SCOPE THIS TEST

```

```

***** TEST 122 *****
*PARAMETER STATUS REGISTER BIT 15 WRITE ONLY DEVICE RESET AND CLEAR TEST
*TEST THAT BIT 15 CANNOT BE WRITTEN AND READ
*BACK THE SAME. READ BIT 15, COMPLEMENT IT AND WRITE IT
*VERIFYING THAT IT DID NOT CHANGE. WRITE BIT 15.
*DO A DEVICE RESET AND VERIFY THE BIT WAS CLEARED.
*REPEAT FOR A CLR INSTRUCTION.
*****

```

```

*****
: TEST 122
*****

```

```

4599 *****
4600 020736 012737 000122 001226 TST122: MOV #122,DTSTNO
4601 020744 012737 021064 001216 MOV #TST123,NEXT
4602 020752 013703 001410 MOV PARCSR,R3 ;GET THE PARAMETER STATUS REGISTER
4603 020756 012705 100000 MOV #BIT15,R5 ;GET BIT 15
4604 020762 011305 MOV (R3),R5 ;READ THE REGISTER
4605 020764 010504 MOV R5,R4 ;SAVE THE BIT
4606 020766 032704 100000 BIT #BIT15,R4 ;CHECK BIT 15
4607 020772 001401 BEQ IS ;BIT 15 IS OK
4608 020774 104003 HLT 3 ;BIT FAILED TO CLEAR

```

```

4609 020776 005104      1S:  CUM      R4      ; REVERSE THE BIT
4610 021000 010413      MOV     R4,(R3) ; WRITE THE BIT TO THE REG
4611 021002 011304      MOV     (R3),R4 ; READ IT BACK
4612 021004 020504      CMP     R5,R4   ; R5=GOOD, R4= ?
4613 021006 001401      BEQ    3$       ; BR IF OK
4614 021010 104003      HLT    3        ; COMPARE ERROR
4615 021012 012705 100000 3S:  MOV     #BIT15,R5 ; LOAD THE BIT
4616 021016 010513      MOV     R5,(R3) ; WRITE THE BIT TO THE REG
4617 021020 052777 000400 160364 BIS     #MRESET, @TXCSR ; RESET THE DEVICE
4618 021026 004737 004772 JSR     PC, SMALL ; WAIT FOR RESET TO FINISH
4619 021032 011304      MOV     (R3),R4 ; GET BIT 15
4620 021034 032704 100000 BIT     #BIT15,R4 ; TEST BIT 15 FOR 0 AFTER RESET
4621 021040 001401      BEQ    4$       ; BIT 15 IS OK
4622 021042 104003      HLT    3        ; BIT IS NOT A 0
4623 021044 010513      4S:  MOV     R5,(R3) ; WRITE THE BIT TO THE REG
4624 021046 005013      CLR     (R3)    ; CLR THE PARAMETER STATUS REGISTER
4625 021050 011304      MOV     (R3),R4 ; GET THE PARAMETER STATUS REGISTER
4626 021052 005005      CLR     R5     ; SET "EXPECTED"
4627 021054 020504      CMP     R5,R4   ; R5=GOOD, R4= ?
4628 021056 001401      BEQ    10$      ; BIT 15 IS OK
4629 021060 104003      HLT    3        ; BIT FAILED TO CLEAR
4630 021062 104400      10S: SCOPE     ; SCOPE THIS TEST

```

```

***** TEST 123 *****
*TRANSMITTER CONTROL REGISTER BIT 8 WRITE ONLY DEVICE RESET AND CLEAR TEST
*TEST THAT BIT 8 CANNOT BE WRITTEN AND READ
*BACK THE SAME. READ BIT 8, COMPLEMENT IT AND WRITE IT
*VERIFYING THAT IT DID NOT CHANGE. WRITE BIT 8.
*DO A DEVICE RESET AND VERIFY THE BIT WAS CLEARED.
*REPEAT FOR A CLR INSTRUCTION.
*****

```

```

*****
*
TEST 123
*
*****

```

```

4648 021064 012737 000123 001226 TST123: MOV     #123, @TSTNO
4649 021072 012737 021216 001216 MOV     @TST124, NEXT
4650 021100 013703 001412 MOV     TXCSR, R3 ; GET THE TRANSMITTER CONTROL REGISTER
4651 021104 012705 000400 MOV     #BIT8, R5 ; GET BIT 8
4652 021110 011305 MOV     (R3), R5 ; READ THE REGISTER
4653 021112 010504 MOV     R5, R4   ; SAVE THE BIT
4654 021114 032704 000400 BIT     #BIT8, R4 ; CHECK BIT 8
4655 021120 001401 BEQ    1$       ; BIT 8 IS OK
4656 021122 104003 HLT    3        ; BIT FAILED TO CLEAR
4657 021124 005104      1S:  COM     R4     ; REVERSE THE BIT
4658 021126 010413      MOV     R4,(R3) ; WRITE THE BIT TO THE REG
4659 021130 011304      MOV     (R3),R4 ; READ IT BACK
4660 021132 020504      CMP     R5,R4   ; R5=GOOD, R4= ?
4661 021134 001401      BEQ    3$       ; BR IF OK
4662 021136 104003      HLT    3        ; COMPARE ERROR
4663 021140 012705 000400 3S:  MOV     #BIT8, R5 ; LOAD THE BIT
4664 021144 010513      MOV     R5,(R3) ; WRITE THE BIT TO THE REG

```

E08

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 96  
DZDPBA.CMB TRANSMITTER CONTROL REGISTER WRITE ONLY TEST BIT 8

```

4665 021146 052777 000400 160236      BIS      #MRESET, @TXCSR  ;RESET THE DEVICE
4666 021154 004737 004772              JSR      PC, SMALL   ;WAIT FOR RESET TO FINISH
4667 021160 011304              MOV      (R3), R4    ;GET BIT 8
4668 021162 032704 000400      BIT      #BIT8, R4   ;TEST BIT 8 FOR 0 AFTER RESET
4669 021166 001401              BEQ      4$          ;BIT 8 IS OK
4670 021170 104003              HLT      3           ;BIT IS NOT A 0
4671 021172 010513          4$:      MOV      R5, (R3)  ;WRITE THE BIT TO THE REG
4672 021174 005013              CLR      (R3)        ;CLR THE TRANSMITTER CONTROL REGISTER
4673 021176 011304              MOV      (R3), R4    ;GET THE TRANSMITTER CONTROL REGISTER
4674 021200 005005              CLR      R5          ;SET "EXPECTED"
4675 021202 042704 000200      BIC      #BIT7, R4   ;CLR UNWANTED BITS
4676 021206 020504              CMP      R5, R4      ;R5=GOOD, R4=?
4677 021210 001401              BEQ      10$         ;BIT 8 IS OK
4678 021212 104003              HLT      3           ;BIT FAILED TO CLEAR
4679 021214 104400          10$:     SCOPE      ;SCOPE THIS TEST

```

```

;***** TEST 124 *****
;RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
;SET BIT8, VERIFY BIT8 WAS SET.
;CLEAR BIT8, VERIFY BIT8 WAS CLEARED.
;*****

```

```

;*****
; TEST 124
;*****

```

```

4694 021216 012737 000124 001226 001226  TST124: MOV      #124, @TSTNO
4695 021224 012737 021272 001216          MOV      #TST125, NEXT
4696 021232 013703 001404              MOV      RXCSR, R3   ;SET REGISTER TO BE TESTED.
4697 021236 012705 000400              MOV      #BIT8, R5   ;SET "EXPECTED"
4698 021242 010513              MOV      R5, (R3)    ;WRITE THE REGISTER.
4699 021244 011304              MOV      (R3), R4    ;READ THE REGISTER.
4700 021246 020504              CMP      R5, R4      ;R5=GOOD; R4=UNKNOWN.
4701 021250 001401              BEQ      1$          ;ARE THEY THE SAME?
4702 021252 104003              HLT      3           ;COMPARISON ERROR.
4703 021254 040513          1$:      BIC      R5, (R3)    ;CLEAR BIT8
4704 021256 011304              MOV      (R3), R4    ;READ THE REGISTER.
4705 021260 005005              CLR      R5          ;SET "EXPECTED"
4706 021262 020504              CMP      R5, R4      ;R5=GOOD; R4=?
4707 021264 001401              BEQ      2$          ;BR IF OK
4708 021266 104003              HLT      3           ;COMPARISON ERROR
4709 021270 104400          2$:      SCOPE      ;SCOPE THIS TEST

```

```

;***** TEST 125 *****
;RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
;SET BIT6, VERIFY BIT6 WAS SET.
;CLEAR BIT6, VERIFY BIT6 WAS CLEARED.
;*****

```

```

;*****
; TEST 125

```

```

4710
4711
4712
4713
4714
4715
4716
4717
4718
4719
4720

```



```

4721
4722
4723
4724 021272 012737 000125 001226
4725 021300 012737 021346 001216
4726 021306 013703 001404
4727 021312 012705 000100
4728 021316 010513
4729 021320 011304
4730 021322 020504
4731 021324 001401
4732 021326 104003
4733 021330 040513
4734 021332 011304
4735 021334 005005
4736 021336 020504
4737 021340 001401
4738 021342 104003
4739 021344 104400

```

```

:*****
:*****
:*****
TST125: MOV #125, @TSTNO
MOV #TST126, NEXT
MOV RXCSR, R3 ;SET REGISTER TO BE TESTED.
MOV #BIT6, R5 ;SET "EXPECTED"
MOV R5, (R3) ;WRITE THE REGISTER.
MOV (R3), R4 ;READ THE REGISTER.
CMP R5, R4 ;R5=GOOD; R4=UNKNOWN.
BEQ 1$ ;ARE THEY THE SAME?
HLT 3 ;COMPARISON ERROR.
1$: BIC R5, (R3) ;CLEAR BIT6
MOV (R3), R4 ;READ THE REGISTER.
CLR R5 ;SET "EXPECTED"
CMP R5, R4 ;R5=GOOD; R4=?
BEQ 2$ ;BR IF OK
HLT 3 ;COMPARISON ERROR
2$: SCOPE ;SCOPE THIS TEST

```

```

:***** TEST 126 *****
:RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
:SET BITS, VERIFY BITS WAS SET.
:CLEAR BITS, VERIFY BITS WAS CLEARED.
:*****

```

```

4740
4741
4742
4743
4744
4745
4746
4747
4748
4749
4750
4751
4752
4753
4754 021346 012737 000126 001226
4755 021354 012737 021422 001216
4756 021362 013703 001404
4757 021366 012705 000040
4758 021372 010513
4759 021374 011304
4760 021376 020504
4761 021400 001401
4762 021402 104003
4763 021404 040513
4764 021406 011304
4765 021410 005005
4766 021412 020504
4767 021414 001401
4768 021416 104003
4769 021420 104400

```

```

:*****
:*****
TEST 126
:*****
:*****
TST126: MOV #126, @TSTNO
MOV #TST127, NEXT
MOV RXCSR, R3 ;SET REGISTER TO BE TESTED.
MOV #BIT5, R5 ;SET "EXPECTED"
MOV R5, (R3) ;WRITE THE REGISTER.
MOV (R3), R4 ;READ THE REGISTER.
CMP R5, R4 ;R5=GOOD; R4=UNKNOWN.
BEQ 1$ ;ARE THEY THE SAME?
HLT 3 ;COMPARISON ERROR.
1$: BIC R5, (R3) ;CLEAR BITS
MOV (R3), R4 ;READ THE REGISTER.
CLR R5 ;SET "EXPECTED"
CMP R5, R4 ;R5=GOOD; R4=?
BEQ 2$ ;BR IF OK
HLT 3 ;COMPARISON ERROR
2$: SCOPE ;SCOPE THIS TEST

```

```

:***** TEST 127 *****
:RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
:SET BIT4, VERIFY BIT4 WAS SET.
:CLEAR BIT4, VERIFY BIT4 WAS CLEARED.
:*****

```

```

4770
4771
4772
4773
4774
4775
4776

```

4777  
4778  
4779  
4780  
4781  
4782  
4783  
4784  
4785  
4786  
4787  
4788  
4789  
4790  
4791  
4792  
4793  
4794  
4795  
4796  
4797  
4798  
4799  
4800  
4801  
4802  
4803  
4804  
4805  
4806  
4807  
4808  
4809  
4810  
4811  
4812  
4813  
4814  
4815  
4816  
4817  
4818  
4819  
4820  
4821  
4822  
4823  
4824  
4825  
4826  
4827  
4828  
4829  
4830  
4831  
4832

021422 012737 000127 001226  
021430 012737 021476 001216  
021436 013703 001404  
021442 012705 000020  
021446 010513  
021450 011304  
021452 020504  
021454 001401  
021456 104003  
021460 040513  
021462 011304  
021464 005005  
021466 020504  
021470 001401  
021472 104003  
021474 104400

```
*****  
: TEST 127  
*****  
*****  
TST127: MOV #127, #TSTNO  
MOV #TST130, NEXT  
MOV RXCSR, R3 ; SET REGISTER TO BE TESTED.  
MOV #BIT4, R5 ; SET "EXPECTED"  
MOV R5, (R3) ; WRITE THE REGISTER.  
MOV (R3), R4 ; READ THE REGISTER.  
CMP R5, R4 ; R5=GOOD; R4=UNKNOWN.  
BEQ 1$ ; ARE THEY THE SAME?  
HLT 3 ; COMPARISON ERROR.  
1$: BIC R5, (R3) ; CLEAR BIT4  
MOV (R3), R4 ; READ THE REGISTER.  
CLR R5 ; SET "EXPECTED"  
CMP R5, R4 ; R5=GOOD; R4=?  
BEQ 2$ ; BR IF OK  
HLT 3 ; COMPARISON ERROR  
2$: SCOPE ; SCOPE THIS TEST
```

\*\*\*\*\* TEST 130 \*\*\*\*\*  
\*TRANSMITTER CONTROL REGISTER READ/WRITE BIT TEST.  
\*SET BIT6, VERIFY BIT6 WAS SET.  
\*CLEAR BIT6, VERIFY BIT6 WAS CLEARED.  
\*\*\*\*\*

```
*****  
: TEST 130  
*****  
*****  
TST130: MOV #130, #TSTNO  
MOV #TST131, NEXT  
MOV TXCSR, R3 ; SET REGISTER TO BE TESTED.  
MOV #BIT6, R5 ; SET "EXPECTED"  
MOV R5, (R3) ; WRITE THE REGISTER.  
MOV (R3), R4 ; READ THE REGISTER.  
BIC #BIT7, R4 ; CLEAR UNWANTED BITS  
CMP R5, R4 ; R5=GOOD; R4=UNKNOWN.  
BEQ 1$ ; ARE THEY THE SAME?  
HLT 3 ; COMPARISON ERROR.  
1$: BIC R5, (R3) ; CLEAR BIT6  
MOV (R3), R4 ; READ THE REGISTER.  
BIC #BIT7, R4 ; CLEAR UNWANTED BITS  
CLR R5 ; SET "EXPECTED"  
CMP R5, R4 ; R5=GOOD; R4=?  
BEQ 2$ ; BR IF OK  
HLT 3 ; COMPARISON ERROR  
2$: SCOPE ; SCOPE THIS TEST
```

4833  
4834  
4835  
4836  
4837  
4838  
4839  
4840  
4841  
4842  
4843  
4844  
4845  
4846  
4847  
4848  
4849  
4850  
4851  
4852  
4853  
4854  
4855  
4856  
4857  
4858  
4859  
4860  
4861  
4862  
4863  
4864  
4865  
4866  
4867  
4868  
4869  
4870  
4871  
4872  
4873  
4874  
4875  
4876  
4877  
4878  
4879  
4880  
4881  
4882  
4883  
4884  
4885  
4886  
4887  
4888

021562 012737 000131 001226  
021570 012737 021646 001216  
021576 013703 001404  
021602 012705 000002  
021606 010513  
021610 011304  
021612 042704 177001  
021616 020504  
021620 001401  
021622 104003  
021624 040513  
021626 011304  
021630 042704 177001  
021634 005005  
021636 020504  
021640 001401  
021642 104003  
021644 104400

```

***** TEST 131 *****
*RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
*SET BIT1, VERIFY BIT1 WAS SET.
*CLEAR BIT1, VERIFY BIT1 WAS CLEARED.
*****

*****
*
* TEST 131
*
*****
*****
TST131: MOV      #131, @TSTNO
        MOV      @TST132, NEXT
        MOV      RXCSR, R3          ; SET REGISTER TO BE TESTED.
        MOV      #BIT1, R5        ; SET "EXPECTED "
        MOV      R5, (R3)         ; WRITE THE REGISTER.
        MOV      (R3), R4        ; READ THE REGISTER.
        BIC      #177001, R4     ; CLEAR UNWANTED BITS
        CMP      R5, R4         ; R5=GOOD; R4=UNKNOWN.
        BEQ      1$             ; ARE THEY THE SAME?
        HLT      3              ; COMPARISON ERROR.
1$:     BIC      R5, (R3)        ; CLEAR BIT1
        MOV      (R3), R4        ; READ THE REGISTER.
        BIC      #177001, R4     ; CLEAR UNWANTED BITS
        CLR      R5             ; SET "EXPECTED"
        CMP      R5, R4         ; R5=GOOD; R4=?
        BEQ      2$             ; BR IF OK
        HLT      3              ; COMPARISON ERROR
2$:     SCOPE                    ; SCOPE THIS TEST

```

\*\*\*\*\* TEST 132 \*\*\*\*\*  
\*RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.  
\*SET BIT2, VERIFY BIT2 WAS SET.  
\*CLEAR BIT2, VERIFY BIT2 WAS CLEARED.  
\*\*\*\*\*

021646 012737 000132 001226  
021654 012737 021732 001216  
021662 013703 001404  
021666 012705 000004  
021672 010513  
021674 011304  
021676 042704 177001  
021702 020504  
021704 001401  
021706 104003  
021710 040513

```

*****
*
* TEST 132
*
*****
*****
TST132: MOV      #132, @TSTNO
        MOV      @TST133, NEXT
        MOV      RXCSR, R3          ; SET REGISTER TO BE TESTED.
        MOV      #BIT2, R5        ; SET "EXPECTED "
        MOV      R5, (R3)         ; WRITE THE REGISTER.
        MOV      (R3), R4        ; READ THE REGISTER.
        BIC      #177001, R4     ; CLEAR UNWANTED BITS
        CMP      R5, R4         ; R5=GOOD; R4=UNKNOWN.
        BEQ      1$             ; ARE THEY THE SAME?
        HLT      3              ; COMPARISON ERROR.
1$:     BIC      R5, (R3)        ; CLEAR BIT2

```

```

4889 021712 011304          MOV      (R3),R4      ;READ THE REGISTER.
4890 021714 042704 177001  BIC      #177001,R4   ;CLEAR UNWANTED BITS
4891 021720 005005          CLR      R5           ;SET "EXPECTED"
4892 021722 020504          CMP      R5,R4       ;R5=GOOD; R4=?
4893 021724 001401          BEQ     2$           ;BR IF OK
4894 021726 104003          HLT     3           ;COMPARISON ERROR
4895 021730 104400          SCOPE  3           ;SCOPE THIS TEST

```

```

***** TEST 133 *****
*RECEIVER CONTROL REGISTER READ/WRITE BIT TEST.
*SET BIT3, VERIFY BIT3 WAS SET.
*CLEAR BIT3, VERIFY BIT3 WAS CLEARED.
*****

```

```

*****
*
* TEST 133
*
*****

```

```

4909
4910 021732 012737 000133 001226 TST133: MOV      #133,2#TSTNO
4911 021740 012737 022016 001216  MOV      #TST134,NEXT
4912 021746 013703 001404          MOV      RXCSR,R3    ;SET REGISTER TO BE TESTED.
4913 021752 012705 000010          MOV      #BIT3,R5    ;SET "EXPECTED"
4914 021756 010513          MOV      R5,(R3)     ;WRITE THE REGISTER.
4915 021760 011304          MOV      (R3),R4     ;READ THE REGISTER.
4916 021762 042704 177001  BIC      #177001,R4   ;CLEAR UNWANTED BITS
4917 021766 020504          CMP      R5,R4       ;R5=GOOD; R4=UNKNOWN.
4918 021770 001401          BEQ     1$           ;ARE THEY THE SAME?
4919 021772 104003          HLT     3           ;COMPARISON ERROR.
4920 021774 040513          BIC     R5,(R3)     ;CLEAR BIT3
4921 021776 011304          MOV      (R3),R4     ;READ THE REGISTER.
4922 022000 042704 177001  BIC      #177001,R4   ;CLEAR UNWANTED BITS
4923 022004 005005          CLR      R5           ;SET "EXPECTED"
4924 022006 020504          CMP      R5,R4       ;R5=GOOD; R4=?
4925 022010 001401          BEQ     2$           ;BR IF OK
4926 022012 104003          HLT     3           ;COMPARISON ERROR
4927 022014 104400          SCOPE  3           ;SCOPE THIS TEST

```

```

***** TEST 134 *****
*RECEIVER CONTROL REGISTER READ/WRITE BIT 6 RESET AND CLEAR TEST
*WRITE BIT 6, AND TEST THAT IT WILL BE CLEARED AFTER A
*DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
*****

```

```

*****
*
* TEST 134
*
*****

```

```

4941
4942 022016 012737 000134 001226 TST134: MOV      #134,2#TSTNO
4943 022024 012737 022112 001216  MOV      #TST135,NEXT
4944 022032 013703 001404          MOV      RXCSR,R3    ;GET THE RECEIVER CONTROL REGISTER

```

```

4945 022036 052713 000100      BIS      #BIT6,(R3)      ;SET BIT 6 AT RECEIVER CONTROL REGISTER
4946 022042 005005          CLR      R5        ;SET "EXPECTED"
4947 022044 052777 000400 157340  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
4948 022052 004737 004772          JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
4949 022056 032713 000100      BIT      #BIT6,(R3) ;TEST BIT 6
4950 022062 001402          BEQ     1$        ;BIT 6 IS CLEARED
4951 022064 011304          MOV     (R3),R4   ;LOAD "FOUND"
4952 022066 104003          HLT     3        ;BIT 6 IS SET AND SHOULDN'T BE
4953 022070 052713 000100 1$:  BIS      #BIT6,(R3) ;SET BIT 6 AGAIN
4954 022074 005013          CLR      (R3)    ;CLEAR THE RECEIVER CONTROL REGISTER
4955 022076 032713 000100      BIT      #BIT6,(R3) ;TEST TO SEE IF BIT 6 CLEARED
4956 022102 001402          BEQ     2$        ;BIT 6 IS OK
4957 022104 011304          MOV     (R3),R4   ;LOAD "FOUND"
4958 022106 104003          HLT     3        ;BIT 6 FAILED TO CLEAR
4959 022110 104400 2$:  SCOPE ;SCOPE THIS TEST

```

```

;***** TEST 135 *****
;RECEIVER CONTROL REGISTER READ/WRITE BIT 5 RESET AND CLEAR TEST
;WRITE BIT 5 AND TEST THAT IT WILL BE CLEARED AFTER A
;DEVICE RESET AND RECEIVER CONTROL REGISTER CLR INSTRUCTION
;*****

```

```

;*****
;TEST 135
;*****

```

```

4974 022112 012737 000135 001226 TST135: MOV     #135,@TSTNO
4975 022120 012737 022206 001216  MOV     #TST136,NEXT
4976 022126 013703 001404          MOV     RXCSR,R3 ;GET THE RECEIVER CONTROL REGISTER
4977 022132 052713 000040      BIS      #BIT5,(R3) ;SET BIT 5 AT RECEIVER CONTROL REGISTER
4978 022136 005005          CLR      R5        ;SET "EXPECTED"
4979 022140 052777 000400 157244  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
4980 022146 004737 004772          JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
4981 022152 032713 000040      BIT      #BIT5,(R3) ;TEST BIT 5
4982 022156 001402          BEQ     1$        ;BIT 5 IS CLEARED
4983 022160 011304          MOV     (R3),R4   ;LOAD "FOUND"
4984 022162 104003          HLT     3        ;BIT 5 IS SET AND SHOULDN'T BE
4985 022164 052713 000040 1$:  BIS      #BIT5,(R3) ;SET BIT 5 AGAIN
4986 022170 005013          CLR      (R3)    ;CLEAR THE RECEIVER CONTROL REGISTER
4987 022172 032713 000040      BIT      #BIT5,(R3) ;TEST TO SEE IF BIT 5 CLEARED
4988 022176 001402          BEQ     2$        ;BIT 5 IS OK
4989 022200 011304          MOV     (R3),R4   ;LOAD "FOUND"
4990 022202 104003          HLT     3        ;BIT 5 FAILED TO CLEAR
4991 022204 104400 2$:  SCOPE ;SCOPE THIS TEST

```

```

;***** TEST 136 *****
;TRANSMITTER CONTROL REGISTER READ/WRITE BIT 6 RESET AND CLEAR TEST
;WRITE BIT 6 AND TEST THAT IT WILL BE CLEARED AFTER A
;DEVICE RESET AND TRANSMITTER CONTROL REGISTER CLR INSTRUCTION
;*****

```

```

;*****

```

# K08

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 102  
 DZDPBA.CMB TRANSMITTER CONTROL REGISTER RESET AND CLEAR TEST BIT 6

```

5001      :
5002      : TEST 136 *
5003      :
5004      : *****
5005      : *****
5006 022206 012737 000136 001226 TST136: MOV #136, @TSTNO
5007 022214 012737 022302 001216      MOV #TST137, NEXT
5008 022222 013703 001412      MOV TXCSR, R3 ; GET THE TRANSMITTER CONTROL REGISTER
5009 022226 052713 000100      BIS #BIT6, (R3) ; SET BIT 6 AT TRANSMITTER CONTROL REGISTER
5010 022232 005005      CLR R5 ; SET "EXPECTED"
5011 022234 052777 000400 157150      BIS #MRESET, @TXCSR ; RESET THE DEVICE
5012 022242 004737 004772      JSR PC, SMALL ; WAIT FOR RESET TO FINISH
5013 022246 032713 000100      BIT #BIT6, (R3) ; TEST BIT 6
5014 022252 001402      BEQ 1$ ; BIT 6 IS CLEARED
5015 022254 011304      MOV (R3), R4 ; LOAD "FOUND"
5016 022256 104003      HLT 3 ; BIT 6 IS SET AND SHOULDN'T BE
5017 022260 052713 000100 1$: BIS #BIT6, (R3) ; SET BIT 6 AGAIN
5018 022264 005013      CLR (R3) ; CLEAR THE TRANSMITTER CONTROL REGISTER
5019 022266 032713 000100      BIT #BIT6, (R3) ; TEST TO SEE IF BIT 6 CLEARED
5020 022272 001402      BEQ 2$ ; BIT 6 IS OK
5021 022274 011304      MOV (R3), R4 ; LOAD "FOUND"
5022 022276 104003      HLT 3 ; BIT 6 FAILED TO CLEAR
5023 022300 104400 2$: SCOPE ; SCOPE THIS TEST
5024
5025
5026      : ***** TEST 137 *****
5027      : *THIS TEST CHECKS THE MAINTENANCE CLOCK
5028      : *USED THROUGHOUT THE REMAINING DIAGNOSTICS
5029      : *****
5030
5031      : *****
5032      : TEST 137 *
5033      :
5034      : *****
5035      : *****
5036      : *****
5037 022302 012737 000137 001226 TST137: MOV #137, @TSTNO
5038 022310 012737 022436 001216      MOV #TST140, NEXT
5039 022316 052777 000400 157066      BIS #MRESET, @TXCSR ; RESET THE DEVICE
5040 022324 004737 004772      JSR PC, SMALL ; WAIT FOR RESET TO FINISH
5041 022330 005037 001236      CLR TEMP1 ; TEST TIM SETUP
5042 022334 052777 004000 157050      BIS #SYSTST, @TXCSR ; ENTER SYSTEM TST MODE TO TURN ON CLOCK
5043 022342 032777 004000 157044 1$: BIT #TIMER, @TXDBUF ; CHECK THE CLOCK BIT
5044 022350 001407      BEQ 2$ ; BR IF OFF
5045 022352 005237 001236      INC TEMP1 ; INC WAIT LOOP
5046 022356 022737 177777 001236      CMP #-1, TEMP1 ; CHECK FOR LOOP TO BE DONE
5047 022364 001366      BNE 1$ ; BR IF MORE TIME TO WAIT
5048 022366 104000      HLT ; TIMER CLOCK BIT FAILED TO CLEAR
5049 022370 005037 001236 2$: CLR TEMP1 ; SECOND HALF SETUP
5050 022374 032777 004000 157012 3$: BIT #TIMER, @TXDBUF ; CHECK THE CLOCK BIT
5051 022402 001007      BNE 4$ ; BR IF ON
5052 022404 005237 001236      INC TEMP1 ; INC THE WAIT LOOP
5053 022410 022737 177777 001236      CMP #-1, TEMP1 ; CHECK FOR LOPP DONE
5054 022416 001366      BNE 3$ ; BR IF MORE TIME TO WAIT
5055 022420 104000      HLT ; TIMER BIT FAILED TO SET
5056 022422 4$:

```

5057 022422 052777 000400 156762  
5058 022430 004737 004772  
5059 022434 104400  
5060  
5061  
5062  
5063  
5064  
5065  
5066  
5067  
5068  
5069  
5070  
5071  
5072  
5073  
5074  
5075 022436 012737 000140 001226  
5076 022444 012737 022612 001216  
5077 022452 013702 001414  
5078 022456 013703 001412  
5079 022462 005012  
5080 022464 052777 000400 156720  
5081 022472 004737 004772  
5082 022476 052777 001000 156704  
5083 022504 052713 014000  
5084 022510 052713 000020  
5085 022514 052705 014220  
5086 022520 011304  
5087 022522 020504  
5088 022524 001401  
5089 022526 104003  
5090 022530 005005  
5091 022532 011204  
5092 022534 042704 170000  
5093 022540 020504  
5094 022542 001405  
5095 022544 013703 001414  
5096 022550 104003  
5097 022552 013703 001412  
5098 022556 010512  
5099 022560 032713 000200  
5100 022564 001401  
5101 022566 104005  
5102 022570 052712 000400  
5103 022574 104412 000017  
5104 022600 032713 000200  
5105 022604 001001  
5106 022606 104006  
5107 022610 104400  
5108  
5109  
5110  
5111  
5112

BIS #MRESET,@TXCSR ;RESET THE DEVICE  
JSR PC,SMALL ;WAIT FOR RESET TO FINISH  
SCOPE ;SCOPE THIS TEST

\*\*\*\*\* TEST 140 \*\*\*\*\*  
;THIS TEST WILL PERFORM STATIC TRANSMITTER FUNCTIONS  
;IN MAINTENANCE MODE. IT WILL PROVE THE INTERACTION  
;OF SEND, DONE AND TSOM.  
\*\*\*\*\*

\*\*\*\*\*  
TEST 140  
\*\*\*\*\*

\*\*\*\*\*

```
TST140: MOV #140,@TSTNO
MOV #TST141,NEXT
MOV TXDBUF,R2 ;LOAD TX BUFFER
MOV TXCSR,R3 ;LOAD TX CONTROL REGISTER
CLR (R2) ;CLEAR BUFFER
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC,SMALL ;WAIT FOR RESET TO FINISH
BIS #CRCEN,@PARCSR ;TURN OFF CRC
BIS #MMODE,(R3) ;ENTER M/M - PROGRAM NOW CLOCKING
BIS #SEND,(R3) ;ASSERT SEND
BIS #SEND!TXDONE!MMODE,R5 ;SET "EXPECTED"
MOV (R3),R4 ;READ TX CONTROL REGISTER
CMP R5,R4 ;ARE THEY EQUAL?
BEQ 1$ ;BR IF YES
HLT 3 ;NO
1$: CLR R5 ;"SET EXPECTED"
MOV (R2),R4 ;READ TX BUFFER
BIC #170000,R4 ;MASK
CMP R5,R4 ;R5=GOOD, R4=?
BEQ 2$ ;ARE THEY EQUAL?
MOV TXDBUF,R3 ;ERROR MESSAGE SETUP
HLT 3 ;NO
2$: MOV TXCSR,R3 ;RETURN TO NORMAL
MOV R5,(R2) ;LOAD BUFFER
BIT #TXDONE,(R3) ;TEST DONE
BEQ 3$
HLT 5 ;BIT FAILED TO CLR
3$: BIS #TSOM,(R2) ;SET TSOM
PKCLK 15
BIT #TXDONE,(R3) ;TEST DONE
BNE 4$
HLT 6 ;BIT FAILED TO SET
4$: SCOPE ;SCOPE THIS TEST
```

\*\*\*\*\* TEST 141 \*\*\*\*\*  
;THIS TEST CHECKS THE STATIC FUNCTIONS OF THE TRANSMITTER  
;IN MAINTENANCE MODE. THIS TEST PROVES THE INTERACTION OF

```

5113                                     ;*TXDBUF, TXACT, TSOM, TRANSMITTED DATA AND DONE.
5114                                     ;:*****
5115
5116                                     ;:*****
5117                                     ;:
5118                                     ;: TEST 141
5119                                     ;:
5120                                     ;:*****
5121                                     ;:*****
5122 022612 012737 000141 001226 1ST141: MOV #141, @#TSTNO
5123 022620 012737 023124 001216 MOV #TST142, NEXT
5124 022626 013703 001412 MOV TXCSR, R3 ;LOAD CONTROL REGISTER
5125 022632 013702 001414 MOV TXDBUF, R2 ;LOAD BUFFER
5126 022636 052777 000400 156546 BIS #MRESET, @TXCSR ;RESET THE DEVICE
5127 022644 004737 004772 JSR PC, SMALL ;WAIT FOR RESET TO FINISH
5128 022650 052713 014000 BIS #MMODE, (R3) ;ENTER M/M--PROGRAM CLOCKING
5129 022654 052777 001000 156526 BIS #CRCEN, @PARCSR ;TURN OFF CRC
5130 022662 052713 000020 BIS #SEND, (R3) ;SET SEND
5131 022666 005005 CLR R5 ;SET "EXPECTED".
5132 022670 010512 MOV R5, (R2) ;LOAD TX BUFFER
5133 022672 012712 000400 MOV #TSOM, (R2) ;TURN ON TSOM
5134 022676 104412 000006 PKCLK 6 ;SYNC UP DUP
5135 022702 052777 020000 156502 BIS #CLK, @TXCSR ;POKE CLOCK UP
5136 022710 032713 001000 BIT #TXACT, (R3) ;IS TXACT HIGH?
5137 022714 001001 BNE 1$ ;BR IF SET
5138 022716 104007 HLT 7 ;TXACT FAILED TO SET
5139 022720
5140 022720 042777 020000 156464 1$: BIC #CLK, @TXCSR ;POKE CLOCK DOWN
5141 022726 104412 000015 PKCLK 13 ;PUSH OUT DATA
5142 022732 032713 000020 BIT #SEND, (R3) ;CHECK SEND
5143 022736 001001 BNE 2$ ;BR IF YES
5144 022740 104017 HLT 17 ;BIT FAILED TO SET
5145 022742 012737 022742 001220 2$: MOV #2$, LOCK ;SETUP FOR SW 09
5146 022750 012705 000400 MOV #TSOM, R5 ;SET "EXPECTED".
5147 022754 011204 MOV (R2), R4 ;GET THE BUFFER REG
5148 022756 042704 170000 BIC #170000, R4 ;MASK CRC BITS
5149 022762 020504 CMP R5, R4 ;R5=GOOD, R4=?
5150 022764 001406 BEQ 3$ ;BR IF A MATCH
5151 022766 013703 001414 MOV TXDBUF, R3 ;ERROR MESSAGE SETUP
5152 022772 104003 HLT 3 ;BIT FAILED TO SET
5153 022774 104401 SCOP1 ;SW09=1?
5154 022776 012703 001412 MOV #TXCSR, R3 ;LOAD TRANSMITTER CSR
5155 023002 012737 023002 001220 3$: MOV #3$, LOCK ;SW09 SETUP
5156 023010 042712 000400 BIC #TSOM, (R2) ;CLR TSOM
5157 023014 032713 000200 BIT #TXDONE, (R3) ;TEST DONE
5158 023020 001402 BEQ 4$ ;BR IF CLEAR
5159 023022 104005 HLT 5 ;DONE BIT IS SET AND SHOULD BE CLEARED
5160 023024 104401 SCOP1 ;SW09=1?
5161 023026 012737 023026 001220 4$: MOV #4$, LOCK ;SW09 SETUP
5162 023034 032712 000400 BIT #TSOM, (R2) ;TEST TSOM
5163 023040 001402 BEQ 5$ ;BR IF CLEAR
5164 023042 104010 HLT 10 ;BIT FAILED TO CLR
5165 023044 104401 SCOP1 ;SW09=1?
5166 023046 042713 000020 5$: BIC #SEND, (R3) ;TURN OFF SEND
5167 023052 104412 000020 PKCLK 16 ;POKE 8 BITS
5168 023056 032713 000200 BIT #TXDONE, (R3) ;CHECK DONE
  
```



5169	023062	001401		BEQ	.+4		;BR IF OFF
5170	023064	104017		HLT	17		;DONE SET AND SHOULDN'T BE
5171	023066	104412	000002	PKCLK	2		;POKE ONE FULL CLOCK
5172	023072	032713	001000	BIT	#TXACT, (R3)		;CHECK ACTIVE
5173	023076	001401		BEQ	6\$		;BR IF OFF
5174	023100	104011		HLT	11		;ACTIVE SETS AND SHOULDN'T BE
5175	023102	032713	000200	6\$: BIT	#TXDONE, (R3)		;IS DONE UP?
5176	023106	001001		BNE	7\$		;BR IF YES
5177	023110	104006		HLT	6		;NO-REPORT ERROR
5178	023112	032713	040000	7\$: BIT	#MTDATA, (R3)		;CHECK DATA OUT
5179	023116	001401		BEQ	10\$		;BR IF OFF
5180	023120	104012		HLT	12		;DATA SET SHOULD BE CLEAR
5181	023122	104400		10\$: SCOPE			;SCOPE THIS TEST

```

:***** TEST 142 *****
:*THIS TEST VERIFIES THAT THE DEVICE IDLES FLAGS
:*IDLE A MINIMUM OF 64. FLAGS.
:*****

```

```

:*****
: TEST 142
:*****
:*****

```

5195	023124	012737	000142	001226	TST142: MOV	#142, #TSTNO	
5196	023132	012737	023212	001216	MOV	#TST143, NEXT	
5197	023140	013702	001414		MOV	TXDBUF, R2	;LOAD TX BUFFER
5198	023144	013703	001412		MOV	TXCSR, R3	;LOAD TX CONTROL REGISTER
5199	023150	052777	000400	156234	BIS	#MRESET, TXCSR	;RESET THE DEVICE
5200	023156	004737	004772		JSR	PC, SMALL	;WAIT FOR RESET TO FINISH
5201	023162	052713	014000		BIS	#MMODE, (R3)	;ENTER M/MODE
5202	023166	052713	000020		BIS	#SEND, (R3)	;SET SEND
5203	023172	052712	000400		BIS	#TSOM, (R2)	;TURN ON START OF MSG
5204	023176	104412	000004		PKCLK	4	;SYNC UP DUP
5205	023202	004137	007003		JSR	R1, FLG	;SEND 64. FLAGS
5206	023206	000100			64.		;64. FLAGS
5207	023210	104400			SCOPE		;SCOPE THIS TEST

```

:***** TEST 143 *****
:*THIS TEST PUSHES DATA THRU THE TRANSMITTER
:*IN MAINTENANCE MODE. THE TEST SENDS A FLAG,
:*AND TWO ALTERNATING ONES AND ZEROES CHARACTERS,
:*AN ALL ZEROES CHARACTER AND AN ALL ONES
:*CHARACTER TO VERIFY THE BIT STUFF CAPABILITY OF
:*THE DUP WITHOUT A CRC CHECK. THE TEST ROTATES
:*THE BITS THRU, SAMPLING THE DATA ON A BIT-BY-BIT
:*BASIS, LSB FIRST. IT STORES THE BIT IN THE MSB OF
:*THE SAVE LOCATION, COMPARES AND ROTATES RIGHT UNTIL
:*THE CHARACTER IS ASSEMBLED.
:*****

```

```

:*****
: TEST 143

```

5208  
5209  
5210  
5211  
5212  
5213  
5214  
5215  
5216  
5217  
5218  
5219  
5220  
5221  
5222  
5223  
5224

```

5225
5226
5227
5228 023212 012737 000143 001226
5229 023220 012737 023564 001216
5230 023226 013703 001412
5231 023232 013702 001414
5232 023236 052777 000400 156146
5233 023244 004737 004772
5234 023250 052713 014000
5235 023254 052777 001000 156126
5236 023262 052713 000020
5237 023266 052712 000400
5238 023272 005037 001236
5239 023276 005037 001240
5240 023302 005037 001242
5241 023306 005037 001244
5242 023312 012704 000004
5243 023316 012705 023552
5244 023322 012701 023552
5245 023326 011137 001246
5246 023332 104412 000010
5247 023336 012512
5248 023340 104412 000014
5249 023344 104412 000002 15:
5250 023350 000241
5251 023352 032713 040000
5252 023356 001401
5253 023360 000261
5254 023362 106037 001236 25:
5255 023366 000241
5256 023370 106037 001246
5257 023374 103006
5258 023376 000261
5259 023400 106037 001242
5260 023404 005237 001240
5261 023410 000404
5262 023412 106037 001242 35:
5263 023416 005037 001240
5264 023422 023737 001236 001242 45:
5265 023430 001401
5266 023432 104004
5267
5268
5269
5270
5271
5272
5273 023434 005704 55:
5274 023436 001003
5275 023440 052712 001000
5276 023444 000405
5277 023446 105777 155740 65:
5278 023452 100002
5279 023454 012512
5280 023456 005304

```

```

:*****
:*****
:*****
TST143: MOV #143, @TSTNO
MOV #TST144, NEXT
MOV TXCSR, R3 ;LOAD TRANSMITTER CONTROL REGISTER
MOV TXDBUF, R2 ;LOAD TRANSMITTER BUFFER
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
BIS #MODE, (R3) ;ENTER MINT MODE--PROGRAM CLOCKING
BIS #CRCEN, @PARCSR ;TURN OFF CRC
BIS #SEND, (R3) ;ASSERT SEND
BIS #TSOM, (R2)
CLR TEMP1 ;CLEAR
CLR TEMP2 ;CLEAR
CLR TEMP3 ;CLEAR
CLR TEMP4 ;CLEAR
MOV #4, R4 ;LOAD THE # OF CHARACTERS TO DO
MOV #TBL1, R5 ;LOAD THE TABLE POINTER
MOV #TBL1, R1 ;DITTO
MOV (R1), TEMPS ;DITTO
PKCLK # ;START A FLAG
MOV (R5)+, (R2) ;LOAD THE FIRST CHARACTER AND CLR TSOM
PKCLK #12 ;FINISH THE FLAG
PKCLK #2 ;PUSH OUT A BIT
CLC ;PUT CARRY IN A KNOWN STATE
BIT #MTDATA, (R3) ;TEST THE BIT
BEQ #25 ;BR IF CLEAR
SEC ;SET CARRY FOR SOFTWARE
RORB TEMP1 ;STORE THE BIT
CLC ;PUT CARRY IN A KNOWN STATE
RORB TEMPS ;CHECK TO SEE WHAT THE BIT SHOULD BE
BCC #35 ;BR IF CLEAR
SEC ;SET CARRY FOR SOFTWARE
RORB TEMP3 ;SHIFT IN A ONE
INC TEMP2 ;INC ONES COUNT
BR #45 ;JUMP OVER
RORB TEMP3 ;LOAD A ZERO
CLR TEMP2 ;CLEAR ONES COUNT
CMP TEMP1, TEMP3 ;DOES HARDWARE = SOFTWARE
BEQ #55 ;BR IF YES
HLT #4 ;HARDWARE AND SOFTWARE DON'T MATCH
;R1 HOLDS THE ADRS OF THE OUTPUT CHAR
;TEMP1, BIT7 = HARDWARE FOUND
;TEMP3, BIT7 = SOFTWARE CALCULATED
;TEMP4 GIVE THE BIT POSITION OUTPUT

```

```

55: TST R4 ;CHECK FOR LAST CHAR
BNE #65 ;BR IF NOT DONE
BIS #TEOM, (R2) ;LOAD END OF MESSAGE
BR #75 ;FINISH TEST
65: TSTB @TXCSR ;CHECK TO SEE IF EREADY FOR NEXT CHAR
BPL #75 ;BR IF NO
MOV (R5)+, (R2) ;LOAD NEXT CHAR
DEC R4 ;LOWER CHAR COUNT

```

```

5281 023460 022737 000005 001240 75:  CMP #5,TEMP2 ;CHECK FOR STUFFED ZERO
5282 023466 001006  BNE 10$ ;BR IF NO
5283 023470 104412 000002  PKCLK 2 ;PUSH OUT THE STUFFED ZERO
5284 023474 032713 040000  BIT #MTDATA,(R3) ;CHECK IT
5285 023500 001401  BEQ 10$ ;BR IF OK
5286 023502 104021  HLT 21 ;FAILED TO BIT-STUFF!!
5287 023504 005237 001244 10$: INC TEMP4 ;INC BIT COUNTER
5288 023510 022737 000010 001244  CMP #8.,TEMP4 ;ARE WE DONE WITH THIS CHAR?
5289 023516 001312  BNE 1$ ;BR IF MORE TO GO
5290 023520 005037 001236  CLR TEMP1 ;CLEAR OUT HARDWARE SAVE
5291 023524 005037 001242  CLR TEMP3 ;CLEAR OUT SOFTWARE SAVE
5292 023530 005037 001244  CLR TEMP4 ;CLEAR OU BIT COUNTER
5293 023534 005721  TST (R1)+ ;POP TBL POINTER
5294 023536 011137 001246  MOV (R1),TEMP5 ;
5295 023542 032712 001000  BIT #TEOM,(R2) ;CHECK FOR END OF TEST
5296 023546 001676  BEQ 1$ ;BR IF MORE TO GO
5297 023550 104400  SCOPE ;SCOPE THIS TEST

```

```

TBL1: .WORD 252 ;THE FIRST FOUR CHARACTERS
      .WORD 000 ;OF THIS TABLE ARE OUTPUT.
      .WORD 125 ;THE LAST CHARACTER IS
      .WORD 377 ;A PAD.
      .WORD 000

```

```

;R1 = SOFTWARE TABLE POINTER
;R2 = TXDBUF
;R3 = TXCSR
;R4 = CHAR COUNTER
;R5 = HARDWARE TABLE POINTER
;TEMP1 = HARDWARE BIT
;TEMP2 = 1'S COUNT
;TEMP3 = SOFTWARE BIT
;TEMP4 = CHARACTER BIT COUNTER
;TEMP5 = SOFTWARE BYTE

```

```

***** TEST 144 *****
;THIS TEST VERIFIES THE ABORT SEQUENCE AND
;NORMAL DATA SEQUENCE OF FLAG, DATA, FLAG
;FOLLOWED BY IDLE LINE. THIS TEST ALSO PROVES
;THE FUNCTIONING OF ACTIVE, TSON, TEOM, SEND AND DONE.
;WITHOUT USING A CRC CHECK.
*****

```

```

*****
TEST 144
*****

```

```

5331 023564 012737 000144 001226  TS144: MOV #144, @TSTNO
5332 023572 012737 024050 001216  MOV @TST145, NEXT
5333 023600 013703 001412  MOV TXCSR, R3 ;LOAD CONTROL REGISTER
5334 023604 013702 001414  MOV TXDBUF, R2 ;LOAD TRANSMITTER BUFFER
5335 023610 052777 000400 155574  BIS #MRESET, @TXCSR ;RESET THE DEVICE
5336 023616 004737 004772  JSR PC, SMALL ;WAIT FOR RESET TO FINISH

```

5337	023622	052713	014000		BIS	#MMODE, (R3)	:ENTER M/MODE
5338	023626	052777	001000	155554	BIS	#CRCEN, 2PARCSR	:SHUT OFF CRC
5339	023634	052713	000020		BIS	#SEND, (R3)	:SET SEND
5340	023640	052712	000400		BIS	#TSOM, (R2)	:TURN ON START OF MSG
5341	023644	104412	000010		PKCLK	8.	:SYNC UP DUP AND START PUSHING OUT A FLAG
5342	023650	052712	002000		BIS	#TABORT, (R2)	:SET ABORT
5343	023654	104412	000014		PKCLK	12.	:PUSH OUT THE ABORT
5344	023660	005037	001236		CLR	TEMP1	:CLEAR HOLD
5345	023664	104412	000002	15:	PKCLK	2	:PUSH A BIT
5346	023670	032713	040000		BIT	#MTDATA, (R3)	:CHECK DATA
5347	023674	001001			BNE	25	:BR IF SET
5348	023676	104013			HLT	13	:DATA FAILED TO SET
5349	023700	005237	001236	25:	INC	TEMP1	:INC HOLD
5350	023704	022737	000002	001236	CMP	#2, TEMP1	:CHECK FOR FINISH
5351	023712	001364			BNE	15	:BR IF NO
5352	023714	032713	001000		BIT	#TXACT, (R3)	:TEST TRANSMITTER ACTIVE
5353	023720	001001			BNE	645	:BR IF ACTIVE SET
5354	023722	104007			HLT	7	:ACTIVE IS RESET AND SHOULDN'T BE
5355	023724			645:			
5356	023724	042712	002000		BIC	#TABORT, (R2)	:CLEAR ABORT
5357	023730	005037	001236		CLR	TEMP1	:CLEAR HOLD
5358	023734	104412	000002	35:	PKCLK, 2		:PUSH A BIT
5359	023740	032713	040000		BIT	#MTDATA, (R3)	:CHECK DATA
5360	023744	001001			BNE	45	:BR IF SET
5361	023746	104013			HLT	13	:DATA OUT FAILED TO SET
5362	023750	005237	001236	45:	INC	TEMP1	:INC # TO DO
5363	023754	022737	000006	001236	CMP	#6, TEMP1	:AND CHECK IT
5364	023762	001364			BNE	35	:BR IF MORE TO GO
5365	023764	104412	000006		PKCLK	6.	:POKE CLOCK
5366	023770	012712	000252		MOV	#252, (R2)	:CLEAR TSOM AND LOAD DATA
5367	023774	104412	000024		PKCLK	20.	:FINISH THE FLAG AND DATA
5368	024000	052712	001000		BIS	#TEOM, (R2)	:SET TEOM
5369	024004	104412	000006		PKCLK	6	:POKE CLOCK
5370	024010	004137	007000		JSR	R1, FLG	:SEND THREE FLAGS
5371	024014	000003			3		:DITTO
5372	024016	042713	000020		BIC	#SEND, (R3)	:CLEAR SEND
5373	024022	104412	000004		PKCLK	4	:POKE CLOCK
5374	024026	032713	040000		BIT	#MTDATA, (R3)	:TEST TXDAT
5375	024032	001001			BNE	655	:MARK OUT
5376	024034	104013			HLT	13	:TXDAT A SPACE - SHOULD BE 1
5377	024036	032713	000200	655:	BIT	#TXDONE, (R3)	:IS DONE UP
5378	024042	001001			BNE	665	:YES
5379	024044	104006			HLT	6	:NO - BUT IT SHOULD BE.
5380	024046	104400		665:	SCOPE		:SCOPE THIS TEST

5381  
5382  
5383  
5384  
5385  
5386  
5387  
5388  
5389  
5390  
5391  
5392

```

:***** TEST 145 *****
:THIS TEST VERIFIES THAT A DATA
:UNDER RUN CONDITION WILL CAUSE
:THE TRANSMITTER DATA LATE BIT TO SET
:AND THAT THE DEVICE WILL ABORT
:UNTIL SEND IS CLEARED WHEN THE OUTPUT GOES TO A SPACE
:*****

```

```

:*****
:

```

```

5393
5394
5395
5396
5397 024050 012737 000145 001226
5398 024056 012737 024250 001216
5399 024064 013703 001412
5400 024070 013702 001414
5401 024074 052777 000400 155310
5402 024102 004737 004772
5403 024106 052713 014000
5404 024112 052777 001000 155270
5405 024120 052713 000020
5406 024124 052712 000400
5407 024130 104412 000006
5408 024134 012777 000000 155252
5409 024142 104412 000016
5410 024146 104412 000017
5411 024152 032777 100000 155232
5412 024160 001401
5413 024162 104014
5414 024164 104412 000002 15:
5415 024170 032777 100000 155214
5416 024176 001001
5417 024200 104015
5418 024202
5419 024202 042777 000020 155202
5420 024210 104412 000020
5421 024214 012737 000010 001236
5422 024222 104412 000002 35:
5423 024226 032777 040000 155156
5424 024234 001401
5425 024236 104012
5426 024240 005337 001236 45:
5427 024244 001366
5428 024246 104400

```

```

: TEST 145
: *****
: *****
TST145: MOV #145, @TSTNO
MOV #TST146, NEXT
MOV TXCSR, R3 ;LOAD TRANSMITTER CONTROL REGISTER
MOV TXDBUF, R2 ;LOAD TRANSMITTER BUFFER
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
BIS #MMODE, (R3) ;ENTER MINT MODE--PROGRAM CLOCKING
BIS #CRCEN, @PARCSR ;TURN OFF CRC
BIS #SEND, (R3) ;ASSERT SEND
BIS #TSOM, (R2)
PKCLK 6 ;START OUTPUTING FLAG
MOV #0, @TXDBUF ;CLEAR TSOM-LOAD BUFFER WITH ZEROES
PKCLK 14 ;FINISH FLAG
PKCLK 15 ;OUTPUT UP TO 7 1/2 CLOCK TICKS
BIT #TXDLAT, @TXCSR ;MAKE SURE DRA IS NOT SET
BEQ 15 ;BR IF CLEARED
HLT 14 ;BIT IS SET TOO SOON
PKCLK 2 ;FINISH LAST CLOCK
BIT #TXDLAT, @TXCSR ;NOW CHECK DRA
BNE 25 ;BRANCH IF SET
HLT 15 ;BIT IS CLEARED AND SHOULD BE SET
25: BIC #SEND, @TXCSR ;TURN OFF SEND
PKCLK 16 ;PUSH 8 BITS
MOV #8, TEMP1 ;SETUP FOR IDLE LINE
35: PKCLK 2 ;OUTPUT BIT
BIT #MTDATA, @TXCSR ;CHECK IT
BEQ 45 ;BRANCH IF ZERO
HLT 12 ;BIT IS A 1
45: DEC TEMP1 ;LOWER COUNT
BNE 35 ;NOT DONE? - GO BACK
SCOPE ;SCOPE THIS TEST

```

```

: ***** TEST 146 *****
: *THIS TEST VERIFIES THAT DROPPING OF
: *SEND BEFORE SETTING TEOM CAUSES
: *A SPACE TO BE OUTPUT AFTER COMPLETION OF
: *A CHARACTER WITH AND WITHOUT BIT STUFF.
: *****

```

```

5429
5430
5431
5432
5433
5434
5435
5436
5437
5438
5439
5440
5441
5442
5443
5444 024250 012737 000146 001226
5445 024256 012737 024616 001216
5446 024264 013703 001412
5447 024270 013702 001414
5448 024274 052777 000400 155110

```

```

: *****
: *****
: TEST 146
: *****
: *****
TST146: MOV #146, @TSTNO
MOV #TST147, NEXT
MOV TXCSR, R3 ;LOAD TRANSMITTER CONTROL REGISTER
MOV TXDBUF, R2 ;LOAD TRANSMITTER BUFFER
BIS #MRESET, @TYCSR ;RESET THE DEVICE

```

5449	024302	004737	004772		JSR	PC, SMALL	:WAIT FOR RESET TO FINISH
5450	024306	052713	014000		BIS	#MMODE, (R3)	:ENTER MINT MODE--PROGRAM CLOCKING
5451	024312	052777	001000	155070	BIS	#CRCEN, @PARCSR	:TURN OFF CRC
5452	024320	052713	000020		BIS	#SEND, (R3)	:ASSERT SEND
5453	024324	052712	000400		BIS	#TSOM, (R2)	
5454	024330	104412	000006		PKCLK	6	:PUSH 2 BITS
5455	024334	012777	000252	155052	MOV	#252, @TXDBUF	:LOAD DATA
5456	024342	104412	000014		PKCLK	12	:PUSH 6 BITS
5457	024346	042777	000020	155036	BIC	#SEND, @TXCSR	:TURN OFF TRANSMITTER
5458	024354	104412	000002		PKCLK	2	:POKE A FULL CLOCK
5459	024360	012737	000010	001236	MOV	#8., TEMP1	:LOAD TEMP1
5460	024366	104412	000012		PKCLK	10	:PUSH 5 BITS
5461	024372	032777	040000	155012	BIT	#MTDATA, @TXCSR	:CHECK DATA
5462	024400	001401			BEQ	2\$	:BR IF CLEAR
5463	024402	104012			HLT	12	:DATA IS SET - SHOULD BE CLEAR
5464	024404	005337	001236		DEC	TEMP1	:LOWER THE # OF TIMES TO REPEAT
5465	024410	001366			BNE	1\$	:BR IF TO GOE MSG
5466	024412	032777	001000	154772	BIT	#TXACT, @TXCSR	:CHECK ACTIVE
5467	024420	001401			BEQ	7\$	:BR IF OFF
5468	024422	104011			HLT	11	:ACTIVE FPILED TO CLEAR
5469	024424						
5470	024427	052777	000400	154760	BIS	#MRESET, @TXCSR	:RESET THE DEVICE
5471	024432	004737	004772		JSR	PC, SMALL	:WAIT FOR RESET TO FINISH
5472	024436	013703	001412		MOV	TXCSR, R3	:LOAD TRANSMITTER CONTROL REGISTER
5473	024442	013702	001414		MOV	TXDBUF, R2	:LOAD TRANSMITTER BUFFER
5474	024446	052777	000400	154736	BIS	#MRESET, @TXCSR	:RESET THE DEVICE
5475	024454	004737	004772		JSR	PC, SMALL	:WAIT FOR RESET TO FINISH
5476	024460	052713	014000		BIS	#MMODE, (R3)	:ENTER MINT MODE--PROGRAM CLOCKING
5477	024464	052777	001000	154716	BIS	#CRCEN, @PARCSR	:TURN OFF CRC
5478	024472	052713	000020		BIS	#SEND, (R3)	:ASSERT SEND
5479	024476	052712	000400		BIS	#TSOM, (R2)	
5480	024502	104412	000006		PKCLK	6	:PUSH TWO BITS
5481	024506	012777	000177	154700	MOV	#177, @TXDBUF	:LOAD DATA
5482	024514	104412	000020		PKCLK	16	:PUSH 8 BITS
5483	024520	042777	000020	154664	BIC	#SEND, @TXCSR	:TURN OFF TRANSMITTER
5484	024526	104412	000014		PKCLK	12	:PUSH 6 BITS
5485	024532	032777	040000	154652	BIT	#MTDATA, @TXCSR	:CHECK DATA OUT FOR BIT STUFF FUNCTION
5486	024540	001001			BNE	3\$	:BR IF SET
5487	024542	104013			HLT	13	:BIT IS A 0, SHOULD BE A 1 - DEVICE
5488							:FAILED TO BIT STUFF
5489	024544	104412	000004		PKCLK	4	:PUSH 2 BITS
5490	024550	012737	000010	001236	MOV	#8., TEMP1	:LOAD TEMP1
5491	024556	104412	000002		PKCLK	2	:PUSH A BIT
5492	024562	032777	040000	154622	BIT	#MTDATA, @TXCSR	:CHECK DATA
5493	024570	001401			BEQ	5\$	:BR IF OFF
5494	024572	104012			HLT	12	:DATA WINDOW SET - SHOULD BE CLEAR
5495	024574	005337	001236		DEC	TEMP1	:LOWER THE # OF TIMES TO CHECK
5496	024600	001366			BNE	4\$	:BR IF MORE TOGO
5497	024602	032777	001000	154602	BIT	#TXACT, @TXCSR	:CHECK ACTIVE
5498	024610	001401			BEQ	6\$	:BR IF OFF
5499	024612	104011			HLT	11	:ACTIVE FAILED TO CLEAR
5500	024614	104400			SCOPE		:SCOPE THIS TEST

5501  
 5502 :\*\*\*\*\* TEST 147 \*\*\*\*\*  
 5503 :\*THIS TEST VERIFIES THAT A DATA UNDERRUN  
 5504 :\*CONDITION WILL CAUSE THE TRANSMITTER DATA

5505  
5506  
5507  
5508  
5509  
5510  
5511  
5512  
5513  
5514  
5515  
5516  
5517  
5518  
5519  
5520  
5521  
5522  
5523  
5524  
5525  
5526  
5527  
5528  
5529  
5530  
5531  
5532  
5533  
5534  
5535  
5536  
5537  
5538  
5539  
5540  
5541  
5542  
5543  
5544  
5545  
5546  
5547  
5548  
5549  
5550  
5551  
5552  
5553  
5554  
5555  
5556  
5557  
5558  
5559  
5560

024616 012737 000147 001226  
024624 012737 025010 001216  
024632 013703 001412  
024636 013702 001414  
024642 052777 000400 154542  
024650 004737 004772  
024654 052713 014000  
024660 052777 001000 154522  
024666 052713 000020  
024672 052712 000400  
024676 104412 000006  
024702 112777 000176 154504  
024710 042777 000400 154476  
024716 104412 000016  
024722 104412 000017  
024726 032777 100000 154456  
024734 001401  
024736 104014  
024740 104412 000002 15:  
024744 032777 100000 154440  
024752 001401  
024754 104014  
024756 104412 000004 25:  
024762 032777 100000 154422  
024770 001001  
024772 104016  
024774 032777 040000 154410 35:  
025002 001001  
025004 104013  
025006 104400 45:

;\*LATE BIT TO SET AND THAT THE DUP WILL GO TO  
;\*A MARK OUTPUT  
:\*\*\*\*\*

:\*\*\*\*\*  
: \*  
: TEST 147  
: \*  
:\*\*\*\*\*

:\*\*\*\*\*

TST147: MOV #147,@TSTNO  
MOV #TST150,NEXT  
MOV TXCSR,R3 ;LOAD TRANSMITTER CONTROL REGISTER  
MOV TXDBUF,R2 ;LOAD TRANSMITTER BUFFER  
BIS #MRESET,@TXCSR ;RESET THE DEVICE  
JSR PC,SMALL ;WAIT FOR RESET TO FINISH  
BIS #MMODE,(R3) ;ENTER MINT MODE--PROGRAM CLOCKING  
BIS #CRCEN,@PARCSR ;TURN OFF CRC  
BIS #SEND,(R3) ;ASSERT SEND  
BIS #TSOM,(R2)  
PKCLK 6 ;START FLAG  
MOVB #176,@TXDBUF ;LOAD CHARACTER TO BE BIT-STUFFED  
BIC #TSOM,@TXDBUF ;SHUT OFF TSOM  
PKCLK 14. ;FINISH CLOCKING FLAG  
PKCLK 15. ;CLOCK TO WITHIN 1 1/2 CLOCKS  
BIT #TXDLAT,@TXCSR ;CHECK DATA LATE  
BEQ 15 ;BRANCH IF CLEAR  
HLT 14 ;BIT IS SET TOO SOON  
15: PKCLK 2 ;CLOCK TO WITHIN A HALF-CLOCK OF DNA  
BIT #TXDLAT,@TXCSR ;CHECK DNA  
BEQ 25 ;BR IF OFF  
HLT 14 ;BIT SET TOO SOON, DEVICE FAILED TO BIT-STUFF  
25: PKCLK 4 ;FINISH CHARACTER  
BIT #TXDLAT,@TXCSR ;CHECK DNA  
BNE 35 ;BRANCH IF SET  
HLT 16 ;BIT SHOULD BE SET AND IS CLEARED  
35: BIT #MTDATA,@TXCSR ;CHECK DATA  
BNE 45 ;BR IF SET  
HLT 13 ;DATA WAS CLEAR - SHOULD BE SET  
45: SCOPE ;SCOPE THIS TEST

:\*\*\*\*\* TEST 150 \*\*\*\*\*  
: \*THIS TEST VERIFIES THAT SETTING TEOM  
: \*AND TSOM AT THE SAME TIME WILL HOLD  
: \*ACTIVE UP AND SEND A FLAG  
:\*\*\*\*\*

:\*\*\*\*\*  
: \*  
: TEST 150  
: \*  
:\*\*\*\*\*

:\*\*\*\*\*

025010 012737 000150 001226  
025016 012737 025154 001216

TST150: MOV #150,@TSTNO  
MOV #TST151,NEXT

```

5561 025024 013703 001412      MOV      TXCSR,R3      ;LOAD TRANSMITTER CONTROL REGISTER
5562 025030 013702 001414      MOV      TXDBUF,R2     ;LOAD TRANSMITTER BUFFER
5563 025034 052777 000400 154350  BIS      #MRESET, @TXCSR ;RESET THE DEVICE
5564 025042 004737 004772      JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
5565 025046 052713 014000      BIS      #MMODE,(R3)   ;ENTER MINT MODE--PROGRAM CLOCKING
5566 025052 052777 001000 154330  BIS      #CRCEN,@PARCSR ;TURN OFF CRC
5567 025060 052713 000020      BIS      #SEND,(R3)   ;ASSERT SEND
5568 025064 052712 000400      BIS      #TSOM,(R2)
5569 025070 104412 000006      PKCLK   6             ;SYNC UP DUP
5570 025074 012777 000252 154312  MOV      #252,@TXDBUF  ;LOAD DATA
5571 025102 104412 000032      PKCLK   26           ;FINISH THE FLAG AND DATA
5572 025106 052777 001000 154300  BIS      #TEOM,@TXDBUF ;TURN ON START AND END OF MSG
5573 025114 104412 000002      PKCLK   2            ;PUSH A BIT
5574 025120 052777 000400 154266  BIS      #TSOM,@TXDBUF ;TURN ON START OF MSG
5575 025126 104412 000002      PKCLK   2            ;PUSH A BIT
5576 025132 032777 001000 154252  BIT      #TXACT,@TXCSR ;TEST ACTIVE
5577 025140 001001      BNE     1$           ;BR IF SET
5578 025142 104007      HLT     7            ;ACTIVE SHOULD BE SET AND IS CLEAR
5579 025144 004137 007000 1$:     JSR      R1,FLG   ;PUSH OUT A FLAG
5580 025150 000001      1       ;ONE FLAG
5581 025152 104400      SCOPE   ;SCOPE THIS TEST

```

```

***** TEST 151 *****
*TEST OF THE BCC OPERATION USING
*CRC.CCITT FOR THE POLYNOMIAL. SPECIFIC
*DATA PATTERNS ARE USED TO ISOLATE FAULTS.
*****

```

```

*****
*
TEST 151
*
*****

```

```

5595 025154 012737 000151 001226 1$:     TST151: MOV      #151,@TSTNO
5596 025162 012737 026204 001216      MOV      #TST152,NEXT
5597 025170 012737 102010 026176      MOV      #CRC.CCITT,XPOLY ;SET THE POLYNOMIAL
5598 025176 012737 026014 001244      MOV      #5$,TEMP4      ;GET THE TABLE POINTER
5599 025204 012737 025212 001220      MOV      #1$,LOCK      ;SETUP FOR SW09=1
5600 025212
5601 025212 052777 000400 154172 1$:     BIS      #MRESET,@TXCSR ;RESET THE DEVICE
5602 025220 004737 004772      JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
5603 025224 052777 014000 154160  BIS      #MMODE,@TXCSR  ;ENTER M/MODE
5604 025232 052777 000020 154152  BIS      #SEND,@TXCSR   ;TURN ON DUP
5605 025240 052777 000400 154146  BIS      #TSOM,@TXDBUF  ;TURN ON START OF MSG
5606 025246 104412 000006      PKCLK   6             ;SYNC UP DUP
5607 025252 017777 153766 154134  MOV      @TEMP4,@TXDBUF ;LOAD DATA
5608 025260 104412 000020      PKCLK   16           ;PUSH 8 BITS
5609 025264 005037 026202      CLR     CALBCC        ;CLEAR OUT OLD BCC
5610 025270 004537 026024      JSR      R5,SIMBCC    ;CALCULATE A SOFTWARE BCC
5611 025274 000010      8.
5612 025276 000000      0
5613 025300 177777      -1
5614 025302 012777 001000 154104  MOV      #TEOM,@TXDBUF ;CLEAR TSOM, SET TEOM
5615 025310 104412 000016      PKCLK   14           ;PUSHOUT DATA
5616 025314 004137 006676      JSR      R1,ACC      ;DO A BCC CALCULATION (HDWR)

```



DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 113  
 DZDPBA.CMB SDLC CRC CALCULATION TEST

5617	025320	100000				BIT15			:FOR 16 BITS
5618	025322	013705	026202			MOV	CALBCC,R5		:MOV SOFTWARE BCC TO EXPECTED
5619	025326	005105				COM	R5		:INVERT IT
5620	025330	020504				CMP	R5,R4		:DOES EXPECTED=FOUND?
5621	025332	001401				BEQ	64\$		:BR IF OK
5622	025334	104020				HLT	20		:BCC CALCULATION ERROR
5623									:TO DEBUG CRC USE THE CRC
5624									:DEBUG AID TEST. SEE FRONT OF THE
5625									:LISTING FOR TEST LOCATION
5626	025336	104401			64\$:	SCOP1			:SW09=1?
5627	025340	062737	000002	001244		ADD	#2,TEMP4		:MOVE THE TABLE POINTER
5628	025346	012737	025354	001220		MOV	#2\$,LOCK		:SETUP FOR SW09=1
5629	025354				2\$:				
5630	025354	052777	000400	154030		BIS	#MRESET,@TXCSR		:RESET THE DEVICE
5631	025362	004737	004772			JSR	PC,SMALL		:WAIT FOR RESET TO FINISH
5632	025366	052777	014000	154016		BIS	#MMODE,@TXCSR		:ENTER M/MODE
5633	025374	052777	000020	154010		BIS	#SEND,@TXCSR		:TURN ON DUP
5634	025402	052777	000400	154004		BIS	#TSOM,@TXDBUF		:TURN ON START OF MSG
5635	025410	104412	000006			PKCLK	6		:SYNC UP DUP
5636	025414	017777	153624	153772		MOV	@TEMP4,@TXDBUF		:LOAD DATA
5637	025422	104412	000020			PKCLK	16		:PUSH 8 BITS
5638	025426	005037	026202			CLR	CALBCC		:CLEAR OUT OLD BCC
5639	025432	004537	026024			JSR	R5,SIMBCC		:CALCULATE A SOFTWARE BCC
5640	025436	000010				B.			:BASED
5641	025440	000252				↑B<10101010>			:ON THESE
5642	025442	177777				-1			:PARAMETERS
5643	025444	012777	001000	153742		MOV	#TEOM,@TXDBUF		:CLEAR TSOM, SET TEOM
5644	025452	104412	000016			PKCLK	14		:PUSHOUT DATA
5645	025456	004137	006676			JSR	R1,ACC		:DO A BCC CALCULATION (HDWR)
5646	025462	100000				BIT15			:FOR 16 BITS
5647	025464	013705	026202			MOV	CALBCC,R5		:MOV SOFTWARE BCC TO EXPECTED
5648	025470	005105				COM	R5		:INVERT IT
5649	025472	020504				CMP	R5,R4		:DOES EXPECTED=FOUND?
5650	025474	001401				BEQ	65\$		:BR IF OK
5651	025476	104020				HLT	20		:BCC CALCULATION ERROR
5652									:TO DEBUG CRC USE THE CRC
5653									:DEBUG AID TEST. SEE FRONT OF THE
5654									:LISTING FOR TEST LOCATION
5655	025500	104401			65\$:	SCOP1			:SW09=1?
5656	025502	062737	000002	001244		ADD	#2,TEMP4		:MOVE THE TABLE POINTER
5657	025510	012737	025516	001220		MOV	#3\$,LOCK		:SETUP FOR SW09=1
5658	025516				3\$:				
5659	025516	052777	000400	153666		BIS	#MRESET,@TXCSR		:RESET THE DEVICE
5660	025524	004737	004772			JSR	PC,SMALL		:WAIT FOR RESET TO FINISH
5661	025530	052777	014000	153654		BIS	#MMODE,@TXCSR		:ENTER M/MODE
5662	025536	052777	000020	153646		BIS	#SEND,@TXCSR		:TURN ON DUP
5663	025544	052777	000400	153642		BIS	#TSOM,@TXDBUF		:TURN ON START OF MSG
5664	025552	104412	000006			PKCLK	6		:SYNC UP DUP
5665	025556	017777	153462	153630		MOV	@TEMP4,@TXDBUF		:LOAD DATA
5666	025564	104412	000020			PKCLK	16		:PUSH 8 BITS
5667	025570	005037	026202			CLR	CALBCC		:CLEAR OUT OLD BCC
5668	025574	004537	026024			JSR	R5,SIMBCC		:CALCULATE A SOFTWARE BCC
5669	025600	000010				B.			:BASED
5670	025602	000125				↑B<01010101>			:ON THESE
5671	025604	177777				-1			:PARAMETERS
5672	025606	012777	001000	153600		MOV	#TEOM,@TXDBUF		:CLEAR TSOM, SET TEOM

5673	025614	104412	000016		PKCLK	14.	: PUSHOUT DATA
5674	025620	004137	006676		JSR	R1,ACC	: DO A BCC CALCULATION (HDWR)
5675	025624	100000			BIT15		: FOR 16 BITS
5676	025626	013705	026202		MOV	CALBCC,R5	: MOV SOFTWARE BCC TO EXPECTED
5677	025632	005105			COM	R5	: INVERT IT
5678	025634	020504			CMP	R5,R4	: DOES EXPECTED=FOUND?
5679	025636	001401			BEQ	66\$	: BR IF OK
5680	025640	104020			HLT	20	: BCC CALCULATION ERROR
5681							: TO DEBUG CRC USE THE CRC
5682							: DEBUG AID TEST. SEE FRONT OF THE
5683							: LISTING FOR TEST LOCATION
5684	025642	104401		66\$:	SCOPI		: SW09=1?
5685	025644	062737	000002	001244	ADD	#2,TEMP4	: MOVE THE TABLE POINTER
5686	025652	012737	025660	001220	MOV	#4\$,LOCK	: SETUP FOR SW09=1
5687	025660			4\$:			
5688	025660	052777	000400	153524	BIS	#MRESET,@TXCSR	: RESET THE DEVICE
5689	025666	004737	004772		JSR	PC,SMALL	: WAIT FOR RESET TO FINISH
5690	025672	052777	014000	153512	BIS	#MMODE,@TXCSR	: ENTER M/MODE
5691	025700	052777	000020	153504	BIS	#SEND,@TXCSR	: TURN ON DUP
5692	025706	052777	000400	153500	BIS	#TSOM,@TXDBUF	: TURN ON START OF MSG
5693	025714	104412	000006		PKCLK	6	: SYNC UP DUP
5694	025720	017777	153320	153466	MOV	@TEMP4,@TXDBUF	: LOAD DATA
5695	025726	104412	000020		PKCLK	16.	: PUSH 8 BITS
5696	025732	005037	026202		CLR	CALBCC	: CLEAR OUT OLD BCC
5697	025736	004537	026024		JSR	R5,SIMBCC	: CALCULATE A SOFTWARE BCC
5698	025742	000010			B.		: BASED
5699	025744	000377			#B<11111111>		: ON THESE
5700	025746	177777			-1		: PARAMETERS
5701	025750	012777	001000	153436	MOV	#TEOM,@TXDBUF	: CLEAR TSOM, SET TEOM
5702	025756	104412	000020		PKCLK	16.	: PUSH 8 BITS
5703	025762	004137	006676		JSR	R1,ACC	: DO A BCC CALCULATION (HDWR)
5704	025766	100000			BIT15		: FOR 16 BITS
5705	025770	012705	157400		MOV	#157400,R5	: LOAD THE BCC
5706	025774	020504			CMP	R5,R4	: DOES EXPECTED=FOUND?
5707	025776	001401			BEQ	67\$	: BR IF OK
5708	026000	104020			HLT	20	: BCC CALCULATION ERROR
5709							: TO DEBUG CRC USE THE CRC
5710							: DEBUG AID TEST. SEE FRONT OF THE
5711							: LISTING FOR TEST LOCATION
5712	026002	104401		67\$:	SCOPI		: SW09=1?
5713	026004	062737	000002	001244	ADD	#2,TEMP4	: MOVE THE TABLE POINTER
5714	026012	104400			SCOPE		: SCOPE THIS TEST
5715	026014	000000		5\$:	.WORD	0	
5716	026016	000252			.WORD	252	
5717	026020	000125			.WORD	125	
5718	026022	000377			.WORD	377	
5719	026024	010046		SIMBCC:	MOV	RO,-(SP)	
5720	026026	010146			MOV	R1,-(SP)	
5721	026030	010246			MOV	R2,-(SP)	
5722	026032	012537	001236		MOV	(R5)+,TEMP1	
5723	026036	012537	001240		MOV	(R5)+,TEMP2	
5724	026042	012537	001242		MOV	(R5)+,TEMP3	
5725	026046	005037	026200	1\$:	CLR	BCCFBK	
5726	026052	013700	001242		MOV	TEMP3,RO	
5727	026056	006037	001240		ROR	TEMP2	
5728	026062	005500			ADC	RO	



```

5729 026064 032700 000001
5730 026070 001402
5731 026072 005137 026200
5732 026076 013700 026176
5733 026102 005100
5734 026104 040037 026200
5735 026110 000241
5736 026112 006037 001242
5737 026116 013700 026200
5738 026122 013701 001242
5739 026126 010102
5740 026130 040100
5741 026132 043702 026200
5742 026136 050200
5743 026140 043737 026176 001242
5744 026146 050037 001242
5745 026152 005337 001236
5746 026156 001333
5747 026160 013737 001242 026202
5748 026166 012602
5749 026170 012601
5750 026172 012600
5751 026174 000205
5752 026176 000000
5753 026200 000000
5754 026202 000000
5755 120001
5756 102010

```

```

2$:
BIT #BIT0,RO
BEQ 2$
COM BCCFBK
MOV XPOLY,RO
COM RO
BIC RO,BCCFBK
CLC
ROR TEMP3
MOV BCCFBK,RO
MOV TEMP3,R1
MOV R1,R2
BIC R1,RO
BIC BCCFBK,R2
BIS R2,RO
BIC XPOLY,TEMP3
BIS RO,TEMP3
DEC TEMP1
BNE 1$
MOV TEMP3,CALBCC
MOV (SP)+,R2
MOV (SP)+,R1
MOV (SP)+,RO
RTS R5

```

```

XPOLY: 0
BCCFBK: 0
CALBCC: 0
CRC16=120001
CRC.CCITT=102010

```

```

***** TEST 152 *****
*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
*NOTE: REMEMBER--IN SDLC A ONE IS A LOGIC LOW IN
*THE CRC GENERATOR.
*****

```

```

*****
*
TEST 152
*
*****

```

```

5777 026204 012737 000152 001226
5778 026212 012737 002764 001216
5779 026220 052777 000400 153164
5780 026226 004737 004772
5781 026232 012737 102010 026176
5782 026240 012737 000125 026402
5783 026246 013737 026402 001252
5784 026254 012737 177777 026202

```

```

TST152: MOV #152,@TSTNO
MOV #.EOP,NEXT
BIS #MRESET,@TXCSR ;RESET THE DEVICE
JSR PC.SMALL ;WAIT FOR RESET TO FINISH
MOV #CRC.CCITT,XPOLY ;LOAD THE POLYNOMIAL
MOV #125,3$ ;LOAD DATA TO SOFTWARE BCC-CHANGE CHARACTER HERE
MOV 3$,SAVR1
MOV #-1,CALBCC ;CLEAR FOR SOFTWARE BCC

```

```

5785 026262 013737 026202 026404      MOV      CALBCC,4$
5786 026270 005037 001242      CLR      TEMP3          ;CLR SOFTWARE MEMORY
5787 026274 005037 001244      CLR      TEMP4          ;CLEAR BIT COUNTER
5788 026300 005037 001246      CLR      TEMP5
5789 026304 052777 014000 153100      BIS      #MMODE,@TXCSR  ;ENTER MAINT MODE-PROGRAM CLOCKING
5790 026312 052777 000020 153064      BIS      #RCVEN,@RXCSR ;TURN ON RECEIVER
5791 026320 052777 000020 153064      BIS      #SEND,@TXCSR  ;TURN ON TRANSMITTER
5792 026326 012777 000400 153060      MOV      #TSOM,@TXDBUF ;
5793 026334 104412 000044      PKCLK    ,36.          ;PUSH OUT 2
5794 026340 013777 026402 153046      MOV      3$,@TXDBUF   ;LOAD DATA
5795 026346 104412 000020      PKCLK    ,16.          ;PUSH OUT ANOTHER
5796 026352 104412 000002      PKCLK    ,2           ;PUSH OUT A BIT
5797 026356 013737 001244 001254      MOV      TEMP4,SAVR2  ;SET UP TO TYPE
5798 026364 005237 001242      INC      TEMP3        ;UPDATE THE COUNT
5799 026370 005237 001244      INC      TEMP4        ;UPDATE BIT COUNTER
5800 026374 004537 026024      JSR      R5,SIMBCC    ;CALCULATE SOFTWARE BCC BASED ON THESE PARAMETERS
5801 026400 000001      1          ;SHIFTS
5802 026402 000000      3$: .WORD 0          ;DATA
5803 026404 000000      4$: .WORD 0          ;PREVIOUS BCC
5804 026406 004737 026504      JSR      PC,5$        ;CHECK TO SEE IF WE SHOULD WAIT FOR SCOPING
5805 026412 000241      CLC          ;CLEAR FOR NEXT ROTATE
5806 026414 106037 026402      RORB     3$          ;SET UP THE NEXT BIT
5807 026420 013737 026202 026404      MOV      CALBCC,4$   ;FOR THE SOFTWARE BCC
5808 026426 022737 000006 001244      CMP      #6,TEMP4
5809 026434 001002      BNE     .+6
5810 026436 005077 152752      CLR      @TXDBUF
5811 026442 022737 000014 001242      CMP      #12.,TEMP3
5812 026450 001003      BNE     12$
5813 026452 012777 001000 152734      MOV      #TEOM,@TXDBUF
5814 026460 022737 000020 001244      CMP      #16.,TEMP4  ;ALL DONE WITH THE CHARACTER?
5815      ;INCREASE THE COMPARE NUMBER TO
5816      ;ALLOW CRC TO BE OUTPUT
5817 026466 001331      BNE     1$          ;BR IF MORE TO GO
5818 026470 052777 000400 152714      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
5819 026476 004737 004772      JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
5820 026502 104400      SCOPE          ;SCOPE THIS TEST
5821
5822 026504 032777 001000 152470 5$: BIT      #SW09,@SWR    ;SW09=1?
5823 026512 001432      BEQ     6$          ;BR IF NO
5824 026514 013704 026202      MOV      CALBCC,R4   ;THE DATA CHARACTER IS
5825 026520 012737 000001 001256      MOV      #1,SAVR3   ;FOLLOWED BY A ZERO CHARACTER.THE
5826 026526 000241      CLC          ;DATA BIT IN CRC SHOWS WHICH BIT
5827 026530 006004      ROR     R4          ;OF THE TWO CHARACTERS IS BEING
5828 026532 006137 001256      ROL     SAVR3       ;GENERATED
5829 026536 103374      BCC     11$
5830 026540 105737 001246      TSTB   TEMP5
5831 026544 001006      BNE     10$
5832 026546 104402 027301      TYPE   ,EM1        ;TYPE MSG
5833 026552 104402 027330      TYPE   ,MH1        ;TYPE HEADER
5834 026556 105137 001246      COMB   TEMP5
5835 026562 104410      10$: CONVRT
5836 026564 027772      DT1
5837 026566 105777 152412      7$: TSTB   @TKCSR     ;CHECK TTY DONE--GO SCOPE THE CRC GENERATOR
5838 026572 100375      BPL     7$          ;BR IF NOT YET
5839 026574 017701 152406      MOV      @TKDBR,R1  ;READ THE BUFFER
5840 026600 000207      6$: RTS      PC      ;RETURN

```

M09

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 117  
DZDPBA.CMB CRC DEBUGGING AID TEST

5841  
5842

```

5843
(1) 026602 005015 051124 047101 EM2: .ASCIZ <15><12>/TRANSMITTER /
(1) 026622 005015 042522 042503 EM3: .ASCIZ <15><12>/RECEIVER /
(1) 026637 015 052412 042516 EM4: .ASCIZ <15><12>/UNEXPECTED INTERRUPT/
(1) 026666 005015 051120 046511 EM6: .ASCIZ <15><12>↑PRIMARY REGISTER TEST ↑
(1) 026717 377 051124 047101 EM10: .ASCIZ <377>/TRANSMITTER DATA ERROR. /
(1) 026751 377 044103 041505 EM11: .ASCIZ <377>/CHECK CARRY AND OUTBIT. /
(1) 027005 104 047117 020105 EM12: .ASCIZ /DONE BIT /
(1) 027020 040506 046111 042105 EM13: .ASCIZ /FAILED TO CLEAR /
(1) 027041 106 044501 042514 EM14: .ASCIZ /FAILED TO SET /
(1) 027060 041501 044524 042526 EM15: .ASCIZ /ACTIVE BIT /
(1) 027075 124 047523 020115 EM16: .ASCIZ /TSOM BIT /
(1) 027110 052517 050124 052125 EM17: .ASCIZ /OUTPUT DATA /
(1) 027126 042523 020124 047524 EM22: .ASCIZ /SET TOO SOON /
(1) 027144 052377 042130 040514 EM23: .ASCIZ <377>/TXDLAT BIT /
(1) 027162 041777 041522 041440 EM24: .ASCIZ <377>/CRC CALCULATION ERROR /
(1) 027212 047111 042524 052522 DH2: .ASCIZ /INTERUPTED UNEXPECTEDLY./
(1) 027243 377 054105 042520 DH6: .ASCIZ <377>/EXPECTED FOUND PRI REG /
(1) 027301 377 051103 020103 EM1: .ASCIZ <377>/CRC GENERATOR STATUS /
(1) 027330 042377 052101 020101 MH1: .ASCIZ <377>/DATA CHAR DATA BIT IN CRC GEN. CRC FOR THIS BIT /
(1) 027423 106 044501 042514 DH7: .ASCIZ /FAILED TO BIT-STUFF. /
(1) 027451 015 050012 044522 EMS: .ASCIZ <15><12>/PRIMARY REGISTER ADDRESSING TIME-OUT /
(1) 027521 015 050012 044522 EM7: .ASCIZ <15><12>/PRIMARY REGISTER RESET TEST /
(1) 027561 377 042522 044507 DHS: .ASCIZ <377>/REGISTER REFERENCED TRAPPED FROM /
(1) .EVEN
(1) 027624 .ERRTAB:
(1) 027624 000000 0
(1)
(1) 027626 000000 0
(1) 027630 000000 0
(1) 027632 027451 EMS
(1) 027634 027561 DHS ;HALT 1
(1) 027636 030010 DT5
(1)
(1)
(1) 027640 027521 EM7
(1) 027642 027243 DH6 ;HALT 2
(1) 027644 030022 DT6
(1)
(1) 027646 026666 EM6
(1) 027650 027243 DH6 ;HALT 3
(1) 027652 030022 DT6
(1)
(1) 027654 026717 EM10
(1) 027656 000000 0 ;HALT 4
(1) 027660 000000 0
(1)
(1)
(1) 027662 027005 EM12
(1) 027664 027020 EM13 ;HALT 5
(1) 027666 000000 0
(1)
(1) 027670 027005 EM12
(1) 027672 027041 EM14 ;HALT 6
(1) 027674 000000 0
(1)
  
```

(1)	027676	027060		EM15	
(1)	027700	027041		EM14	;HALT 7
(1)	027702	000000		0	
(1)					
(1)	027704	027075		EM16	
(1)	027706	027020		EM13	;HALT 10
(1)	027710	000000		0	
(1)					
(1)	027712	027060		EM15	
(1)	027714	027020		EM13	;HALT 11
(1)	027716	000000		0	
(1)					
(1)	027720	027110		EM17	
(1)	027722	027020		EM13	;HALT 12
(1)	027724	000000		0	
(1)					
(1)	027726	027110		EM17	
(1)	027730	027041		EM14	;HALT 13
(1)	027732	000000		0	
(1)					
(1)	027734	027144		EM23	
(1)	027736	027126		EM22	;HALT 14
(1)	027740	000000		0	
(1)					
(1)	027742	027144		EM23	
(1)	027744	027041		EM14	;HALT 15
(1)	027746	000000		0	
(1)					
(1)	027750	027144		EM23	
(1)	027752	027020		EM13	;HALT 16
(1)	027754	000000		0	
(1)					
(1)	027756	027005		EM12	
(1)	027760	027126		EM22	;HALT 17
(1)	027762	000000		0	
(1)					
(1)	027764	027162		EM24	
(1)	027766	027243		DH6	;HALT 20
(1)	027770	030022		DT6	
(1)					
(1)	027772	000003		DT1:	3
(1)	027774	006	021	.BYTE	6,17.
(1)	027776	001252		SAVR1	
(1)	030000	006	017	.BYTE	6,15.
(1)	030002	001254		SAVR2	
(1)	030004	006	002	.BYTE	6,2
(1)	030006	001256		SAVR3	
(1)	030010	000002		DT5:	2
(1)	030012	006	017	.BYTE	6,15.
(1)	030014	001252		SAVR1	
(1)	030016	006	002	.BYTE	6,2
(1)	030020	001254		SAVR2	
(1)	030022	000003		DT6:	3
(1)	030024	006	004	.BYTE	6,4
(1)	030026	001262		SAVR5	
(1)	030030	006	002	.BYTE	6,2

(1)	030032	001260		SHVR4	
(1)	030034	006	002	.BYTE	6,2
(1)	030036	001256		SAVR3	
(1)					
(1)					
(1)					
(1)					
(1)					
(1)					
(1)	030040	000001		CORMAX:	
(1)				.END	



# D10

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 122  
 DZDPBA.CMB CROSS REFERENCE TABLE -- USER SYMBOLS

ABRT	006726	1719#												
ACC	006676	1710#	5616	5645	5674	5703								
ADRCNT=	003735	1310*	1346*	1355#										
BCCFBK	026200	5725*	5731*	5734*	5737	5741	5753#							
BINWRD	004240	1397*	1398	1435#										
BITW =	002000	787#												
BIT0 =	000001	649#	769	2169	2851	2855	2859	2861	3259	3266	3271	3619	3626	3631
		5729												
BIT1 =	000002	648#	768	2199	2499	2503	2507	2509	2883	2887	2891	2893	3659	3666
		3671	4849											
BIT10 =	002000	639#	759	774	787	807	2041	2469	2723	2727	2731	2733	3171	3175
		3179	3181	3379	3386	3391	4019	4026	4031					
BIT11 =	004000	638#	758	786	806	2073	2755	2759	2763	2765	3419	3426	3431	
BIT12 =	010000	637#	757	773	780	785	805	2105	2787	2791	2795	2797	3459	3466
		3471	4059	4066	4071	4344	4351	4356	4555	4558	4567	4572		
BIT13 =	020000	636#	756	784	804	2137	2819	2823	2827	2829	3499	3506	3511	4384
		4391	4396											
BIT14 =	040000	635#	755	772	783	803	1069	1217	3539	3546	3551	4099	4106	4111
		4260	4268	4273	4424	4431	4436							
BIT15 =	100000	634#	754	771	778	782	802	1066	3579	3586	3591	4139	4146	4151
		4302	4310	4315	4464	4471	4476	4603	4606	4615	4620	5617	5646	5675
		5704												
BIT2 =	000004	647#	767	1155	2229	2531	2535	2539	2541	2915	2919	2923	2925	3699
		3706	3711	4881										
BIT3 =	000010	646#	766	793	1977	2259	2563	2567	2571	2573	2659	2663	2667	2669
		2947	2951	2955	2957	3739	3746	3751	4913					
BIT4 =	000020	645#	765	792	2009	2289	2595	2599	2603	2605	2691	2695	2699	2701
		2979	2983	2987	2989	3779	3786	3791	4787					
BIT5 =	000040	644#	764	2319	3011	3015	3019	3021	3819	3826	3831	4757	4977	4981
		4985	4987											
BIT6 =	000100	643#	763	791	2349	3043	3047	3051	3053	3859	3866	3871	4727	4817
		4945	4949	4953	4955	5009	5013	5017	5019					
BIT7 =	000200	642#	762	790	1806	1807	1823	1930	1980	1986	2012	2018	2044	2050
		2076	2082	2108	2114	2140	2146	2379	3075	3079	3083	3085	3299	3306
		3311	3899	3906	3911	4180	4192	4222	4238	4264	4280	4306	4322	4675
		4820	4826											
BIT8 =	000400	641#	761	776	789	809	2409	2627	2631	2635	2637	3107	3111	3115
		3117	3939	3946	3951	4651	4654	4663	4668	4697				
BIT9 =	001000	640#	760	775	779	788	808	2439	3139	3143	3147	3149	3339	3346
		3351	3979	3986	3991	4218	4226	4231	4507	4510	4519	4524		
BRW	003306	1161	1239#											
BRX	003310	1162	1240#											
CALBCC	026202	5609*	5618	5638*	5647	5667*	5676	5696*	5747*	5754#	5784*	5785	5807	5824
CARDET=	010000	757#												
CHRCNT	004236	1395*	1399	1415*	1433#	1434								
CLK =	020000	784#	1554	1557	5135	5140								
CLK.A	001433	895#	1670											
CLRVEC	005032	1579#												
CNVRT =	104411	862#	1178	1180	1182	1184	1508	1510						
CONVRT=	104410	860#	1114	1524	5835									
CORMAX	030040	5843#												
CRCEN =	001000	779#	5082	5129	5235	5338	5404	5451	5477	5522	5566			
CRCERR=	010000	773#												
CRC.CC=	102010	5597	5756#	5781										
CRC16 =	120001	5755#												
CREAM	001316	749#	976*	1624*	1625	1627*	1632	1633*	1634	1637*				





MERRX	005315	1183	1590#															
MERR2	005134	1590#	1616															
MERR3	005203	1133	1590#															
MEXT =	010000	797#	1533															
MHI	027330	5833	5843#															
MIND	001330	818#																
MJMPR	005454	1047	1590#															
MLOCK	005241	1157	1590#															
MMODE =	014000	796#	5083	5085	5128	5201	5234	5337	5403	5450	5476	5521	5565	5603				
		5632	5661	5690	5789													
MMODEA=	004000	786#																
MPCDEB=	010000	785#																
MNEW	005342	1128	1590#															
MPAR	005631	1030	1590#															
MPASSX	005304	1181	1590#															
MPOWER	005067	1575	1590#															
MQM	005060	1294	1590#	1699														
MR	005131	1164	1590#	1694														
MRESET=	000400	650#	789#	2501	2533	2565	2597	2629	2661	2693	2725	2757	2789	2821				
		2853	2885	2917	2949	2981	3013	3045	3077	3109	3141	3173	3268	3308				
		3348	3388	3428	3468	3508	3548	3588	3628	3668	3708	3748	3788	3828				
		3868	3908	3948	3988	4028	4068	4108	4148	4193	4228	4270	4312	4353				
		4393	4433	4473	4521	4569	4617	4665	4947	4979	5011	5039	5057	5080				
		5126	5199	5232	5335	5401	5448	5470	5474	5519	5563	5601	5530	5659				
		5688	5779	5818														
MSPJM	005710	1061	1590#															
MSTJM	005655	1057	1590#															
MTCN	005527	1051	1590#															
MTDATA=	040000	783#	1713	1722	1734	1739	1746	5178	5251	5284	5346	5359	5374	5423				
		5461	5485	5492	5541													
		688#	998															
MTITLE	001000	1039	1590#															
MTOTAL	005576	1507	1590#	1678														
MTSTN	005326	1590#																
MTSTPC	005227	1022	1590#															
MVEC	005441	1179	1590#															
MVECX	005276	708#	1235	1541	1765*	1797*	1851*	1875*	1899*	1924*	1950*	1975*	2007*	2039*				
NEXT	001216	2071*	2103*	2135*	2167*	2197*	2227*	2257*	2287*	2317*	2347*	2377*	2407*	2437*				
		2467*	2497*	2529*	2561*	2593*	2625*	2657*	2689*	2721*	2753*	2785*	2817*	2849*				
		2881*	2913*	2945*	2977*	3009*	3041*	3073*	3105*	3137*	3169*	3202*	3230*	3257*				
		3297*	3337*	3377*	3417*	3457*	3497*	3537*	3577*	3617*	3657*	3697*	3737*	3777*				
		3817*	3857*	3897*	3937*	3977*	4017*	4057*	4097*	4137*	4177*	4216*	4258*	4300*				
		4342*	4382*	4422*	4462*	4505*	4553*	4601*	4649*	4695*	4725*	4755*	4785*	4815*				
		4847*	4879*	4911*	4943*	4975*	5007*	5038*	5076*	5123*	5196*	5229*	5332*	5398*				
		5445*	5516*	5560*	5596*	5778*												
OPCLRJ	001323	815#	1049	1064	1108*													
OVRUN=	040000	772#																
PARAM =	104405	854#	1015	1023	1031	1040	1679											
PARAM1	003576	1312#	1329															
PARBIT=	000000	650#																
PARCSR	001410	879#	1647*	1901	3231	4506	4554	4602	5082*	5129*	5235*	5338*	5404*	5451*				
		5477*	5522*	5566*														
PARERR	003652	1315	1317	1319	1328#	1335	1337	1339										
PASCNT	001230	713#	973*	1174*	1175	1208												
PC =%	000007	614#	1193*	1565*	1586*	1671*	1706*	2502*	2534*	2566*	2598*	2630*	2662*	2694*				
		2726*	2758*	2790*	2822*	2854*	2886*	2918*	2950*	2982*	3014*	3046*	3078*	3110*				





R4 =%000004

5128*	5130*	5136	5142	5151*	5154*	5157	5166*	5168	5172	5175	5178	5198*
5201*	5202*	5230*	5234*	5236*	5251	5284	5333*	5337*	5339*	5346	5352	5359
5372*	5374	5377	5399*	5403*	5405*	5446*	5450*	5452*	5472*	5476*	5478*	5517*
5521*	5523*	5561*	5565*	5567*								
611#	1272	1277*	1281*	1282*	1283	1290*	1293	1296*	1304	1313*	1314	1316
1318	1320*	1321	1322	1343*	1344*	1348*	1365	1378*	1390	1398*	1401	1406*
1408*	1410*	1427*	1489*	1490*	1491*	1492*	1493*	1494*	1495	1496	1497	1710*
1716*	1805*	1806*	1807*	1808	1822*	1823*	1824	1856*	1857	1879*	1880	1904*
1905	1929*	1930*	1931	1955*	1956	1979*	1980*	1981	1985*	1986*	1988	2011*
2012*	2013	2017*	2018*	2020	2043*	2044*	2045	2049*	2050*	2052	2075*	2076*
2077	2081*	2082*	2084	2107*	2108*	2109	2113*	2114*	2116	2139*	2140*	2141
2145*	2146*	2148	2171*	2172	2176*	2178	2201*	2202	2206*	2208	2231*	2232
2236*	2238	2261*	2262	2266*	2268	2291*	2292	2296*	2298	2321*	2322	2326*
2328	2351*	2352	2356*	2358	2381*	2382	2386*	2388	2411*	2412	2416*	2418
2441*	2442	2446*	2448	2471*	2472	2476*	2478	2505*	2511*	2537*	2543*	2569*
2575*	2601*	2607*	2633*	2639*	2665*	2671*	2697*	2703*	2729*	2735*	2761*	2767*
2793*	2799*	2825*	2831*	2857*	2863*	2889*	2895*	2921*	2927*	2953*	2959*	2985*
2991*	3017*	3023*	3049*	3055*	3081*	3087*	3113*	3119*	3145*	3151*	3177*	3183*
3207*	3208	3234*	3235*	3236	3237*	3238	3261*	3263	3270*	3271	3276*	3278
3301*	3303	3310*	3311	3316*	3318	3341*	3343	3350*	3351	3356*	3358	3381*
3383	3390*	3391	3396*	3398	3421*	3423	3430*	3431	3436*	3438	3461*	3463
3470*	3471	3476*	3478	3501*	3503	3510*	3511	3516*	3518	3541*	3543	3550*
3551	3556*	3558	3581*	3583	3590*	3591	3596*	3598	3621*	3623	3630*	3631
3636*	3638	3661*	3663	3670*	3671	3676*	3678	3701*	3703	3710*	3711	3716*
3718	3741*	3743	3750*	3751	3756*	3758	3781*	3783	3790*	3791	3796*	3798
3821*	3823	3830*	3831	3836*	3838	3861*	3863	3870*	3871	3876*	3878	3901*
3903	3910*	3911	3916*	3918	3941*	3943	3950*	3951	3956*	3958	3981*	3983
3990*	3991	3996*	3998	4021*	4023	4030*	4031	4036*	4038	4061*	4063	4070*
4071	4076*	4078	4101*	4103	4110*	4111	4116*	4118	4141*	4143	4150*	4151
4156*	4158	4182*	4183	4188*	4189	4195*	4196	4220*	4222*	4223	4230*	4231
4236*	4238*	4239	4262*	4264*	4265	4272*	4273	4278*	4280*	4281	4304*	4306*
4307	4314*	4315	4320*	4322*	4323	4346*	4348	4355*	4356	4361*	4363	4386*
4388	4395*	4396	4401*	4403	4426*	4428	4435*	4436	4441*	4443	4466*	4468
4475*	4476	4481*	4483	4509*	4510	4513*	4514	4515*	4516	4523*	4524	4529*
4531	4557*	4558	4561*	4562	4563*	4564	4571*	4572	4577*	4579	4605*	4606
4609*	4610	4611*	4612	4619*	4620	4625*	4627	4653*	4654	4657*	4658	4659*
4660	4667*	4668	4673*	4675*	4676	4699*	4700	4704*	4706	4729*	4730	4734*
4736	4759*	4760	4764*	4766	4789*	4790	4794*	4796	4819*	4820*	4821	4825*
4826*	4828	4851*	4852*	4853	4857*	4858*	4860	4883*	4884*	4885	4889*	4890*
4892	4915*	4916*	4917	4921*	4922*	4924	4951*	4957*	4983*	4989*	5015*	5021*
5086*	5087	5091*	5092*	5093	5147*	5148*	5149	5242*	5273	5280*	5620	5649
5678	5706	5824*	5827*									

R5 =%000005

612#	1255	1256*	1260	1265	1267*	1303	1305*	1306	1307	1308	1309	1310
1311	1312*	1321*	1324*	1325*	1326*	1334	1336	1338	1344	1345*	1349*	1364
1379*	1391	1399*	1411*	1426*	1443*	1447*	1451*	1487*	1488*	1489	1491	1803*
1804	1808	1821*	1824	1852*	1857	1876*	1880	1900*	1905	1925*	1931	1951*
1956	1977*	1978	1981	1984	1987*	1988	2009*	2010	2013	2016	2019*	2020
2041*	2042	2045	2048	2051*	2052	2073*	2074	2077	2080	2083*	2084	2105*
2106	2109	2112	2115*	2116	2137*	2138	2141	2144	2147*	2148	2169*	2170
2172	2175	2177*	2178	2199*	2200	2202	2205	2207*	2208	2229*	2230	2232
2235	2237*	2238	2259*	2260	2262	2265	2267*	2268	2289*	2290	2292	2295
2297*	2298	2319*	2320	2322	2325	2327*	2328	2349*	2350	2352	2355	2357*
2358	2379*	2380	2382	2385	2387*	2388	2409*	2410	2412	2415	2417*	2418
2439*	2440	2442	2445	2447*	2448	2469*	2470	2472	2475	2477*	2478	2500*
2532*	2564*	2596*	2628*	2660*	2692*	2724*	2756*	2788*	2820*	2852*	2884*	2916*
2948*	2960*	3012*	3044*	3076*	3108*	3140*	3172*	3205*	3208	3233*	3234	3238

# K10

DZDPBA MACY11 27(732) 21-OCT-76 15:33 PAGE 129  
DZDPBA.CMB CROSS REFERENCE TABLE -- USER SYMBOLS

3259*	3260	3262*	3263	3266*	3267	3274	3277*	3278	3299*	3300	3302*	3303		
3306*	3307	3314	3317*	3318	3339*	3340	3342*	3343	3346*	3347	3354	3357*		
3358	3379*	3380	3382*	3383	3386*	3387	3394	3397*	3398	3419*	3420	3422*		
3423	3426*	3427	3434	3437*	3438	3459*	3460	3462*	3463	3466*	3467	3474		
3477*	3478	3499*	3500	3502*	3503	3506*	3507	3514	3517*	3518	3539*	3540		
3542*	3543	3546*	3547	3554	3557*	3558	3579*	3580	3592*	3583	3586*	3587		
3594	3597*	3598	3619*	3620	3622*	3623	3626*	3627	3634	3637*	3638	3659*		
3660	3662*	3663	3666*	3667	3674	3677*	3678	3699*	3700	3702*	3703	3706*		
3707	3714	3717*	3718	3739*	3740	3742*	3743	3746*	3747	3754	3757*	3758		
3779*	3780	3782*	3783	3786*	3787	3794	3797*	3798	3819*	3820	3822*	3823		
3826*	3827	3834	3837*	3838	3859*	3860	3862*	3863	3866*	3867	3874	3877*		
3878	3899*	3900	3902*	3903	3906*	3907	3914	3917*	3918	3939*	3940	3942*		
3943	3946*	3947	3954	3957*	3958	3979*	3980	3982*	3983	3986*	3987	3994		
3997*	3998	4019*	4020	4022*	4023	4026*	4027	4034	4037*	4038	4059*	4060		
4062*	4063	4066*	4067	4074	4077*	4078	4099*	4100	4102*	4103	4106*	4107		
4114	4117*	4118	4139*	4140	4142*	4143	4146*	4147	4154	4157*	4158	4180*		
4183	4186*	4189	4192*	4196	4218*	4219	4221*	4223	4226*	4227	4234	4237*		
4239	4260*	4261	4263*	4265	4268*	4269	4276	4279*	4281	4302*	4303	4305*		
4307	4310*	4311	4318	4321*	4323	4344*	4345	4347*	4348	4351*	4352	4359		
4362*	4363	4384*	4385	4387*	4388	4391*	4392	4399	4402*	4403	4424*	4425		
4427*	4428	4431*	4432	4439	4442*	4443	4464*	4465	4467*	4468	4471*	4472		
4479	4482*	4483	4507*	4508*	4509	4516	4519*	4520	4527	4530*	4531	4555*		
4556*	4557	4564	4567*	4568	4575	4578*	4579	4603*	4604*	4605	4612	4615*		
4616	4623	4626*	4627	4651*	4652*	4653	4660	4663*	4664	4671	4674*	4676		
4697*	4698	4700	4703	4705*	4706	4727*	4728	4730	4733	4735*	4736	4757*		
4758	4760	4763	4765*	4766	4787*	4788	4790	4793	4795*	4796	4817*	4818		
4821	4824	4827*	4828	4849*	4850	4853	4856	4859*	4860	4881*	4882	4885		
4888	4891*	4892	4913*	4914	4917	4920	4923*	4924	4946*	4978*	5010*	5085*		
5087	5090*	5093	5098	5131*	5132	5146*	5149	5243*	5247	5279	5610*	5618*		
5619*	5620	5639*	5647*	5648*	5649	5668*	5676*	5677*	5678	5697*	5705*	5706		
5722	5723	5724	5751*	5800*										
SAVACT	001312	745#	1070*	1076*	1078	1105*	1131							
SAVNUM	001313	746#	972*	1072*	1106*	1185*	1188*							
SAVPC	001266	732#	1360*	1547										
SAVPCA	001306	741#												
SAVRD	001250	725#	1369*	1374										
SAVRDA	001270	734#												
SAVR1	001252	726#	1368*	1375	5783*	5843								
SAVR1A	001272	735#												
SAVR2	001254	727#	1367*	1376	5797*	5843								
SAVR2A	001274	736#												
SAVR3	001256	728#	1366*	1377	5825*	5828*	5843							
SAVR3A	001276	737#												
SAVR4	001260	729#	1365*	1378	5843									
SAVR4A	001300	738#												
SAVR5	001262	730#	1364*	1379	5843									
SAVR5A	001302	739#												
SAVSP	001264	731#												
SAVSPA	001304	740#												
SAVOS =	104406	856#	1486											
SCOPE =	104400	844#	1778	1831	1860	1883	1908	1934	1959	1991	2023	2055	2087	2119
		2151	2181	2211	2241	2271	2301	2331	2361	2391	2421	2451	2481	2513
		2545	2577	2609	2641	2673	2705	2737	2769	2801	2833	2865	2897	2929
		2961	2993	3025	3057	3089	3121	3153	3185	3211	3241	3281	3321	3361
		3401	3441	3481	3521	3561	3601	3641	3681	3721	3761	3801	3841	3881
		3921	3961	4001	4041	4081	4121	4161	4199	4242	4284	4326	4366	4406

SAVACT 001312  
SAVNUM 001313  
SAVPC 001266  
SAVPCA 001306  
SAVRD 001250  
SAVRDA 001270  
SAVR1 001252  
SAVR1A 001272  
SAVR2 001254  
SAVR2A 001274  
SAVR3 001256  
SAVR3A 001276  
SAVR4 001260  
SAVR4A 001300  
SAVR5 001262  
SAVR5A 001302  
SAVSP 001264  
SAVSPA 001304  
SAVOS = 104406  
SCOPE = 104400







TST126	021346	4725	4754#
TST127	021422	4755	4784#
TST13	010076	2039	2070#
TST130	021476	4785	4814#
TST131	021562	4815	4846#
TST132	021646	4847	4878#
TST133	021732	4879	4910#
TST134	022016	4911	4942#
TST135	022112	4943	4974#
TST136	022206	4975	5006#
TST137	022302	5007	5037#
TST14	010162	2071	2102#
TST140	022436	5038	5075#
TST141	022612	5076	5122#
TST142	023124	5123	5195#
TST143	023212	5196	5228#
TST144	023564	5229	5331#
TST145	024050	5332	5397#
TST146	024250	5398	5444#
TST147	024616	5445	5515#
TST15	010246	2103	2134#
TST150	025010	5516	5559#
TST151	025154	5560	5595#
TST152	026204	5596	5777#
TST153=	***** U	5778	5843
TST16	010332	2135	2166#
TST17	010406	2167	2196#
TST2	007212	1765	1796#
TST20	010462	2197	2226#
TST21	010536	2227	2256#
TST22	010612	2257	2286#
TST23	010666	2287	2316#
TST24	010742	2317	2346#
TST25	011016	2347	2376#
TST26	011072	2377	2406#
TST27	011146	2407	2436#
TST3	007370	1797	1850#
TST30	011222	2437	2466#
TST31	011276	2467	2496#
TST32	011372	2497	2528#
TST33	011466	2529	2560#
TST34	011562	2561	2592#
TST35	011656	2593	2624#
TST36	011752	2625	2656#
TST37	012046	2657	2688#
TST4	007432	1851	1874#
TST40	012142	2689	2720#
TST41	012236	2721	2752#
TST42	012332	2753	2784#
TST43	012426	2785	2816#
TST44	012522	2817	2848#
TST45	012616	2849	2880#
TST46	012712	2881	2912#
TST47	013006	2913	2944#
TST5	007470	1875	1898#
TST50	013102	2945	2976#



























2468	2483	2484	2488	2498	2515	2516	2520	2530	2547	2548	2552	2562	2579	2580	
2584	2594	2611	2612	2616	2626	2643	2644	2648	2658	2675	2676	2680	2690	2707	
2708	2712	2722	2739	2740	2744	2754	2771	2772	2776	2786	2803	2804	2808	2818	
2835	2836	2840	2850	2867	2868	2877	2882	2899	2900	2904	2914	2931	2932	2936	
2946	2963	2964	2968	2978	2995	2996	3000	3010	3027	3028	3032	3042	3059	3060	
3064	3074	3091	3092	3096	3106	3123	3124	3128	3138	3155	3156	3160	3170	3187	
3188	3193	3203	3213	3214	3221	3231	3243	3244	3248	3258	3283	3284	3288	3298	
3323	3324	3328	3338	3363	3364	3368	3378	3403	3404	3408	3418	3443	3444	3448	
3458	3483	3484	3488	3498	3523	3524	3528	3538	3563	3564	3568	3578	3603	3604	
3608	3618	3643	3644	3648	3658	3683	3684	3688	3698	3723	3724	3728	3738	3763	
3764	3768	3778	3803	3804	3808	3818	3843	3844	3848	3858	3883	3884	3888	3898	
3923	3924	3928	3938	3963	3964	3968	3978	4003	4004	4008	4018	4043	4044	4048	
4058	4083	4084	4088	4098	4123	4124	4128	4138	4163	4168	4178	4202	4203	4207	
4217	4244	4245	4249	4259	4286	4287	4291	4301	4328	4329	4333	4343	4368	4369	
4373	4383	4408	4409	4413	4423	4448	4449	4453	4463	4488	4489	4496	4506	4536	
4537	4544	4554	4584	4585	4592	4602	4632	4633	4640	4650	4681	4682	4686	4696	
4711	4712	4716	4726	4741	4742	4746	4756	4771	4772	4776	4786	4801	4802	4806	
4816	4833	4834	4838	4848	4865	4866	4870	4880	4897	4898	4902	4912	4929	4930	
4934	4944	4961	4962	4966	4976	4993	4994	4998	5008	5026	5029	5039	5062	5063	
5067	5077	5109	5110	5114	5124	5183	5184	5187	5197	5208	5209	5220	5230	5316	
5317	5323	5333	5383	5384	5390	5399	5430	5431	5436	5446	5501	5502	5507	5517	
5546	5547	5551	5561	5583	5587	5597	5758	5759	5770	5779	5843	5843	5843	5843	
.PAGE	1	568	602	652	750	824	871	899	942	961	1166	1605	1707	1753	5843
.REM	1	942													
.REPT	562														
.SBTTL	568	602	652	686	961	1166	1753	1783	1837	1862	1885	1910	1936	1961	1993
	2025	2057	2089	2121	2153	2183	2213	2243	2273	2303	2333	2363	2393	2423	2453
	2483	2515	2547	2579	2611	2643	2675	2707	2739	2771	2803	2835	2867	2899	2931
	2963	2995	3027	3059	3091	3123	3155	3187	3213	3243	3283	3323	3363	3403	3443
	3483	3523	3563	3603	3643	3683	3723	3763	3803	3843	3883	3923	3963	4003	4043
	4083	4123	4163	4202	4244	4286	4328	4368	4408	4448	4488	4536	4584	4632	4681
	4711	4741	4771	4801	4833	4865	4897	4929	4961	4993	5026	5062	5109	5183	5208
	5316	5383	5430	5501	5546	5583	5758								
.TITLE	581														
.WORD	813	816	817	818	819	1832	1833	1834	1835	5299	5300	5301	5302	5303	5715
	5716	5717	5718	5802	5803										

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

\*.DZDPBA.SEQ/SOL/CRF/PAGNUM/NL:TOC=DZDPXX.MAC,DZDPBA.CMB  
 RUN-TIME: 53 76 12 SECONDS  
 RUN-TIME RATIO: 193/141=1.3  
 CORE USED: 25K (49 PAGES)



B12

Spooler routine 22 Seconds, 96 KCS, 651 disk reads, 5 disk writes, 143 pages

~~\_\_\_\_\_ Data 00-Apr-76 16:11:02 Hostler JPC-D 0079 (100) output~~

```

0000000000000000000000000000000000000000000000000000000000000000000011111111111111111111111110
00000001111111122222222223333333334444444444555555555666666666777777777788888888899999999900000000011111111122222222233312
_____
00000000000000000000000000000000000000000000000000000000000000000000111111111111111111111111110
0000000111111112222222222333333333444444444555555555666666666777777777788888888899999999900000000011111111122222222233312

```

567890123456789012 \*