

# DUP11

REC CONTR&INTERRUPT  
MD-11-DZDPC-A

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

NOV 1976  
digital  
MADE IN USA

This microfiche card contains a grid of frames, each displaying technical data. The frames are arranged in approximately 10 rows and 8 columns. Each frame contains a mix of text, tables, and diagrams. The text is small and difficult to read, but appears to be technical specifications or data tables. The diagrams are also small and appear to be flowcharts or block diagrams. The overall layout is dense and organized, typical of microfiche documentation.







46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100  
101

1.0 ABSTRACT

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

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

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

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

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

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

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

NOTE: ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.



001

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 4  
DZDPCA.CMB

102  
103



104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118

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.



119  
120  
121  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139  
140  
141  
142  
143  
144  
145  
146  
147  
148  
149  
150  
151  
152  
153  
154  
155  
156  
157  
158  
159  
160  
161  
162  
163  
164  
165  
166  
167

2.0 REQUIREMENTS

2.1 EQUIPMENT

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

2.2 STORAGE

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

3.0 LOADING PROCEDURE

3.1 METHOD

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

ABSOLUTE LOADER STARTING ADDRESS = \*+500

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

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

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

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



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  
 223

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.



H01

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 8  
DZDPCA.CMB

224  
225

SW 06 SET: RESERVED  
SW 05 SET: RESERVED



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  
281

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.



J01

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 10  
DZDPCA.CMB

282  
283

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

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

## 4.1.3 SWITCH REGISTER PRIORITIES

## A) ERROR SWITCHES

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

## B) SCOPE SWITCHES

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

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

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

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

## 4.2 STARTING ADDRESS

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

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

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  
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  
383

## 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.

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  
423

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.



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  
479

## 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

B02

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 15  
DZDPCA.CMB

480  
481

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



482  
483  
484  
485  
486  
487  
488  
489  
490  
491  
492  
493  
494  
495  
496  
497  
498  
499  
500  
501  
502  
503  
504  
505  
506  
507  
508  
509  
510  
511  
512  
513  
514  
515  
516  
517  
518  
519  
520  
521  
522  
523  
524  
525  
526  
527  
528  
529  
530  
531  
532  
533  
534  
535  
536  
537

THAT DUP11 NO. 00,04 WILL BE TESTED.

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

B.4 MORE ON THAT 'STATUS TABLE' (1500-1560)  
'MAP OF DUP11 STATUS'

1500 160050  
1502 000300  
1504 140000

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

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

1502 000300 THIS IS VECTOR 'A' FOR THE FIRST DUP11 IN THE SYSTEM.

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

THE BITS ARE AS FOLLOWS:

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

THE ABOVE IS REPEATED FOR EACH DUP11 IN THE SYSTEM. THE TABLE IS FILLED BY DEFAULT PARAMETERS OR BY THE MANUAL PARAMETER INPUT AS DESCRIBED PREVIOUSLY. ALSO, IF DESIRED BY THE USER - THE LOCATIONS MAY BE ALTERED BY HAND (TOGGLED IN) TO SUIT THE SPECIFIC CONFIGURATION, THUS MAKING EACH DEVICE MAP DIFFERENT. IT IS THE RESPONSIBILITY OF THE OPERATOR TO VERIFY THE DATA IN THE MAP.

B.5 METHOD OF DEVELOPING DEFAULT PARAMETERS

B.5.1 DEFAULT PARAMETER ASSUMPTIONS

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

002

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 17  
DZDPCA.CMB

538  
539

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



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.

3

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-DZDPC-A /<377>/DUP-11 OFFLINE SDLC RECEIVER, MODEM CONTROL AND INTERRUPT TE
; *COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754
; -----
  
```

```

; STARTING PROCEDURE
; LOAD PROGRAM
; LOAD ADDRESS 000200
; PRESS START
; PROGRAM WILL TYPE "MAINDEC-11-DZDPC-A /<377>/DUP-11 OFFLINE SDLC RECEIVER, MODE
; 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
  
```



```

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      .=1000
688      001000      005377      040515      047111      MTITLE: .ASCIZ <377><12>/MAINDEC-11-DZDPC-A /<377>/DUP-11 OFFLINE SDLC RECEIVER, MODEM
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      ;-----

```



DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 22  
DZDPCA.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      000004      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
  
```



DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 24  
DZDPCA.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

: MISC. PROGRAM DEFINITIONS

810			
811			
812			
813	001320	000000	PRTY: .WORD 0
814	001322	000001	TCNFG: .BLKB 1
815	001323	000001	OPCLRJ: .BLKB 1
816	001324	000000	DATA: .WORD 0
817	001326	000000	SHIFTS: .WORD 0
818	001330	000000	MIND: .WORD 0
819	001332	000000	FLAG: .WORD 0
820	001334	000001	STJMFL: .BLKW 1
821	001336	000001	SRJMFL: .BLKW 1
822			
823			

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

:PROGRAM CONTROL FLAGS

-----

001340 000  
001341 000  
001342 000  
001343 000

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

:PROGRAM INITIALIZATION FLAG  
:ERROR OCCURED FLAG  
:LOCK ON CURRENT TEST FLAG  
:QUICK VERIFY FLAG.  
:ON FIRST PASS OF EACH DUP11 ITERATIONS  
:WILL BE SUPPRESSED

000000

.EVEN  
\$Y=0

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

:\*\*\*\*\*

-----

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

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

:\*\*\*\*\*



DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 26  
 DZDPCA.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      HUPRCR: 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

```

899
900
901
902      001500      .=1500
903      001500      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      DUPO.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 * I V I E C I T O 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 S I Y I N I C * 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
966
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 MOV# DUPNUM,5AVNUM ;SAVE NUMBER OF DEVICES IN SYSTEM
973 001610 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
974 001614 105037 001341 CLR# ERRFLG ;CLEAR ERROR FLAG
975 001620 105037 001343 CLR# QV.FLG ;ZERO QUICK VERIFY FLAG
976 001624 012737 001500 001316 MOV #DUP.MAP,CREAM ;GET MAP POINTER.
977 001632 112737 000001 001314 MOV# #1,RUN ;POINT POINTER TO FIRST DEVICE.
978 001640 005037 001232 CLR ERRCNT ;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$,2#4 ;SETUP FOR TIMEOUT
986 001702 005777 177274 TST 2SWR ;REFERENCE HARDWARE SWITCH REG
987 001706 000407 BR 13$ ;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$: TSTB 2SWR ;BIT7=1??
1001 001772 100002 BPL 10$
1002 001774 000137 002520 JMP 1$
1003 002000 10$:
1004 002000 032777 000001 177174 BIT #SW00,2SWR ;ENTER PARAMETERS
1005 002006 001002 BNE +6 ;YES
1006 002010 000137 002360 JMP 21$ ;NO
1007 002014 105137 001332 COMB FLAG
1008 002020 112737 000001 001340 MOV# #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 005421 MCSR ;MESSAGE
1015 002046 104405 PARAM ;CONVERT STRING
1016 002050 160000 160000 ;LOW LIMIT

```

1017	002052	175500				175500	:HIGH LIMIT
1018	002054	001500				DUPCRO	: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	005440				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	005630				MPAR	: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,PRTY :SAVE PRIORITY
1038	002122	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1039	002124	005575				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	005453				MJMPR	: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	005526				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	71\$
1056	002166	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
1057	002170	005654				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	005707				MSRJM	:MESSAGE
1062	002202	104413				SETFLG	:SET FLAG BASED UPON INPUT STRING
1063	002204	001336				SRJMFL	:THIS FLAG
1064	002206	105737	001323			TSTB	OPCLRJ
1065	002212	001403				BEQ	69\$
1066	002214	052737	100000	001504		BIS	#BIT15,DUPO.A
1067	002222	105737	001322			TSTB	TCNFLG
1068	002226	001403				BEQ	70\$
1069	002230	052737	040000	001504		BIS	#BIT14,DUPO.A
1070	002236	112737	000001	001312		MOVB	#1,SAVACT
1071	002244	113737	001236	001311		MOVB	TEMP1,DUPNUM
1072	002252	113737	001236	001313		MOVB	TEMP1,SAVNUM



DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 31  
 DZDPCA.CMB PROGRAM INITIALIZATION AND START UP.

1073	002260	005337	001236	65\$:	DEC	TEMP1		
1074	002264	001404			BEQ	64\$		
1075	002266	000261			SEC			
1076	002270	106137	001312		ROLB	SAVACT		
1077	002274	000771			BR	65\$		
1078	002276	113737	001312	001240	64\$:	MOV	SAVACT,TEMP2	;# OF TIMES
1079	002304	113737	001312	001310		MOV	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	000005	001320	MOV	#5,PRIORITY	:LOAD PRIORITY	
1102	002426	012700	000001		MOV	#1,R0	:SAVE CORE THIS WAY	
1103	002432	110037	001310		MOV	R0,DUPACTV	:PRESET PROGRAM CONTROLS	
1104	002436	110037	001311		MOV	R0,DUPNUM	:DITTO	
1105	002442	110037	001312		MOV	R0,SAVACT	:DITTO	
1106	002446	110037	001313		MOV	R0,SAVNUM	:DITTO	
1107	002452	110037	001322		MOV	R0,TCNFLAG	:DITTO	
1108	002456	110037	001323		MOV	R0,OPCLRJ	:DITTO	
1109	002462			66\$:				
1110	002462	104402	005742	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	005770			XSTATQ			
1116	002510	062737	000002	001236	ADD	#2,TEMP1	:UPDATE POINTER	
1117	002516	000766			BR	5\$		
1118	002520	032777	000001	176454	1\$:	BIT	#SW00,JSWR	
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,JSWR	:SELECT SPECIFIC DEVICES??	
1127	002562	001424			BEQ	3\$	:BR IF NO.	
1128	002564	104402	005341		TYPE	,MNEW	:TYPE THE MESSAGE.	



DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 32  
 DZDPCA.CMB PROGRAM INITIALIZATION AND START UP.

```

1129 002570 005000          CLR      RO
1130 002572 000000          HALT
1131 002574 127737 176402 001312  CMPB    @SWR, SAVACT
1132 002602 101404          BLOS    2$
1133 002604 104402 005202    TYPE    ,MERR3
1134 002610 000000          HALT
1135 002612 000776          BR      -2
1136 002614 117737 176362 001310 2$:     MOVB    @SWR, DUPACTV
1137 002622 113700 001310    MOVB    DUPACTV, RO
1138 002626 042700 177400    BIC     #'C<377>, RO
1139 002632 000000          HALT
1140 002634 012700 000300 3$:     MOV     #300, RO
1141 002640 012701 000302    MOV     #302, R1
1142 002644 010120 4$:     MOV     R1, (RO)+
1143 002646 005021          CLR     (R1)+
1144 002650 022021          CMP     (RO)+, (R1)+
1145 002652 022700 001000    CMP     #1000, RO
1146 002656 001372          BNE     4$
1147
1148                                ; TEST START AND RESTART
1149                                -----
1150
1151 002660 012737 000340 177776 .BEGIN: MOV     #340, PS
1152 002666 012706 001150    MOV     #STACK, SP
1153 002672 005737 000042    TST     @#42
1154 002676 001023          BNE     2$
1155 002700 032777 000004 176274    BIT     #BIT2, @SWR
1156 002706 001411          BEQ     1$
1157 002710 104402 005240    TYPE    ,MLOCK
1158 002714 012737 000240 003174    MOV     #NOP, TTST
1159 002722 012737 000240 003176    MOV     #NOP, TTST+2
1160 002730 000406          BR      2$
1161 002732 013737 003306 003174 1$:     MOV     BRW, TTST
1162 002740 013737 003310 003176    MOV     BRX, TTST+2
1163 002746 012737 006152 001214 2$:     MOV     #CYCLE, RETURN
1164 002754 104402 005130    TYPE    MR
1165 002760 000177 176230    JMP     @RETURN

```

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

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

```

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          CLRB    ERRFLG          ;CLEAR ERROR FLAG
1174 002774 005237 001230          INC     PASCNT          ;UPDATE PASS COUNT
1175 003000 013777 001230 176172  MOV     PASCNT, @DISPLAY ;DISPLAY PASS COUNT
1176 003006 104402 005105          TYPE   ,MEPASS         ;TYPE END PASS
1177 003012 104402 005267          TYPE   ,MCSR           ;TYPE CSR
1178 003016 104411 003130          CNVRT  ,XCSR           ;SHOW IT
1179 003022 104402 005275          TYPE   ,MVEC          ;TYPE VECTOR
1180 003026 104411 003136          CNVRT  ,XVEC           ;SHOW IT
1181 003032 104402 005303          TYPE   ,MPASSX        ;TYPE PASSES
1182 003036 104411 003144          CNVRT  ,XPASS          ;SHOW IT
1183 003042 104402 005314          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                   BNE    RESTR           ;BR IF NO.
1187 003060 112737 000377 001343  MOVB   #377, QV.FLG    ;SET THE QUICK VERIFY FLAG.
1188 003066 113737 001311 001313  MOVB   DUPNUM, SAVNUM ;RESTORE THE COUNT
1189 003074 013701 000042          MOV     @#42, R1       ;CHECK FOR ACT-11 OR DDP
1190 003100 001406                   BEQ    RESTR           ;IF NOT, CONTINUE TESTING
1191 003102 000005                   RESET                    ;STOP THE SHOW--CLEAR THE WORLD
1192 003104
1193 003104 004711          SENDAD: JSR      PC, (R1)
1194 003106 000240          NOP
1195 003110 000240          NOP
1196 003112 000240          NOP
1197 003114 000240          NOP
1198 003116 012737 006152 001214  RESTRT: MOV     #CYCLE, RETURN
1199 003124 000137 006152          JMP     CYCLE
1200 003130 000001          XCSR:   1
1201 003132          006          002          .BYTE   6,2
1202 003134 001404          RXCSR
1203 003136 000001          XVEC:   1
1204 003140          003          002          .BYTE   3,2
1205 003142 001374          DUPRVC
1206 003144 000001          XPASS:  1
1207 003146          006          002          .BYTE   6,2
1208 003150 001230          PASCNT
1209 003152 000001          XERR:   1
1210 003154          006          002          .BYTE   6,2
1211 003156 001232          ERRCNT
1212
1213                                     ;SCOPE LOOP AND INTERATION HANDLER
1214
1215 003160 005037 001234          .SCOPE: CLR     LSTERR          ;CLEAR LAST ERROR PC
1216 003164 010016          MOV     RD, (SP)       ;SAVE RD ON STACK
1217 003166 032777 040000 176006  BIT     #BIT14, @SWR    ;LOOP ON TEST?
1218 003174 001407          TTST:  BEQ     1$        ;BR IF NO (IF LOCK SW01 = 1; THIS LOCATION = 240)
1219 003176 000437          BR      3$             ;GO TO 3$ (DITTO)
1220 003200 105777 176000          TSTB   @TKCSR         ;KYBD DONE?
1221 003204 100034          BPL     3$             ;BR IF NO (LOCK: HIT A KEY ON TTY TO GO TO NEXT TEST)

```

```

1222 003206 017700 175774      MOV      @TKDBR,R0      ;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      CLRB   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),R0     ;POP RD 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

;CHECK FOR FREEZE ON CURRENT DATA
-----
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.

;TELETYPE OUTPUT ROUTINE
-----
1255 003336 010546      .TYPE: MOV    R5,-(SP)  ;SAVE R5 ON THE STACK.
1256 003340 017605      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(BIT?))
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      MOVB  (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
-----
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 006106      MOV    #INBUF,R4

```



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	006106		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	LIMITS:	CMP	R5, HILIM
1335	003662	101373			BHI	PARERR
1336	003664	020537	003726		CMP	R5, LOLIM
1337	003670	103770			BLO	PARERR
1338	003672	133705	003734		BITB	LOBITS, R5
1339	003676	001365			BNE	PARERR
1340						
1341						
1342						
1343	003700	013704	003732			
1344	003704	010524		1\$:	MOV	DEVADR, R4
1345	003706	062705	000002		MOV	R5, (R4)+
1346	003712	105337	003735		ADD	#2, R5
1347	003716	001372			DECB	ADRCNT
1348	003720	012604			BNE	1\$
1349	003722	012605			MOV	(SP)+, R4
1350	003724	000002			MOV	(SP)+, R5
1351	003726	000000			RTI	
1352	003730	000000		LOLIM:	0	
1353	003732	000000		HILIM:	0	
1354	003734	000000		DEVADR:	0	
1355		003735		LOBITS:	0	
1356				ADRCNT=	LOBITS+1	

;STORE NUMBER AT SPECIFIED ADDRESS

1357  
1358  
1359  
1360  
1361  
1362  
1363  
1364  
1365  
1366  
1367  
1368  
1369  
1370  
1371  
1372  
1373

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

1360	003736	016637	000004	001266	.SAV05:	MOV	4(SP), SAVPC	;SAVE R7 (PC)
1362								;SAVE R0-R5
1364	003744	010537	001262		SV05:	MOV	R5, SAVR5	;SAVE R5
1365	003750	010437	001260			MOV	R4, SAVR4	;SAVE R4
1366	003754	010337	001256			MOV	R3, SAVR3	;SAVE R3
1367	003760	010237	001254			MOV	R2, SAVR2	;SAVE R2
1368	003764	010137	001252			MOV	R1, SAVR1	;SAVE R1
1369	003770	010037	001250			MOV	R0, SAVR0	;SAVE R0
1370	003774	000002				RTI		;LEAVE.

;RESTORE R0-R5

1374	003776	013700	001250		.RES05:	MOV	SAVR0, R0	;RESTORE R0
1375	004002	013701	001252			MOV	SAVR1, R1	;RESTORE R1
1376	004006	013702	001254			MOV	SAVR2, R2	;RESTORE R2
1377	004012	013703	001256			MOV	SAVR3, R3	;RESTORE R3
1378	004016	013704	001260			MOV	SAVR4, R4	;RESTORE R4
1379	004022	013705	001262			MOV	SAVR5, R5	;RESTORE R5
1380	004026	000002				RTI		;LEAVE

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

1386	004030	104402	005064	.CONVR:	TYPE	MCRLF
1387	004034	010046		.CNVRT:	MOV	R0, -(SP)
1388	004036	010146			MOV	R1, -(SP)
1389	004040	010346			MOV	R3, -(SP)

1390	004042	010446			MOV	R4,-(SP)
1391	004044	010546			MOV	R5,-(SP)
1392	004046	017601	000012		MOV	@12(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	@(R1)+,BINWRD
1398	004100	013704	004240	2\$:	MOV	BINWRD,R4
1399	004104	113705	004236		MOVB	CHRCNT,R5
1400	004110	012700	006002		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	006044		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			BEQ	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	006044		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	@(SP),R5
1444	004246	042737	000040	006106	BIC	#40,INBUF
1445	004254	122737	000116	006106	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 006106 1$:  CMPB    #'Y,INBUF      ;IS IT "Y" ?
1450 004276 001005      BNE      3$
1451 004300 112715 177777      MOVB     #-1,(R5)      ;377
1452 004304 062716 000002      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 023146      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	005337			TYPE	,MASTEK	
1507	004524	104402	005325		1\$:	TYPE	,MTSTN	
1508	004530	104411	004726			CNVRT	,XTSTN	;SHOW IT
1509	004534	104402	005414			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,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,@#42	;IF ACT-11 AUTO MODE--HALT!!
1528	004624	001403				BEQ	1\$	
1529	004626	005777	174350			TST	@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!NEXT,@TXCSR	
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,@SWR	;GOTO TOP OF TEST?
1538	004666	001007				BNE	1\$	;BR IF YES
1539	004670	032777	002000	174304		BIT	#SW10,@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	@RETURN	;GOTO SPECIFIED TEST
1544	004716	000002			2\$:	RTI		;RETURN
1545	004720	000001			ERTABO:	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	@(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,@TXCSR	;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,@TXCSR	;POKE CLOCK DOWN



```

1558 004764 005300          DEC      RO          ;ARE WE DONE?
1559 004766 001366          BNE      IS          ;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      IS          ;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 DZDPCA /
(2) 005130 051377 000          MR:     .ASCIZ  <377>/R/
(2) 005133 377 051120 043517          MERR2:  .ASCIZ  <377>/PROGRAM INDICATES NO DEVICES PRESENT./
(2) 005202 044777 051516 043125          MERR3:  .ASCIZ  <377>/INSUFFICIENT DATA!/
(2) 005226 052377 051505 020124          MTSTPC: .ASCIZ  <377>/TEST PC-/
(2) 005240 046377 041517 020113          MLOCK:  .ASCIZ  <377>/LOCK ON SELECTED TEST/
(2) 005267 103 051123 020072          MCSRX:  .ASCIZ  /CSR: /
(2) 005275 126 041505 020072          MVECX:  .ASCIZ  /VEC: /
(2) 005303 120 051501 042523          MPASSX: .ASCIZ  /PASSES: /
(2) 005314 051105 047522 051522          MERRX:  .ASCIZ  /ERRORS: /
(2) 005325 124 051505 020124          MTSTN:  .ASCIZ  /TEST NO: /
(2) 005337 052 000          MASTEK: .ASCIZ  /*/
(2) 005341 377 042523 020124          MNEW:   .ASCIZ  <377>/SET SWITCH REG TO DUP11'S DESIRED ACTIVE./
(2) 005414 041520 020072 000          MERRPC: .ASCIZ  /PC: /
(2) 005421 377 042522 020103          MCSR:   .ASCIZ  <377>/REC CSR ADRS /
(2) 005440 053377 041505 040440          MVEC:   .ASCIZ  <377>/VEC ADRS /
(2) 005453 377 051511 052040          MJMPR:  .ASCIZ  <377>/IS THE OPTIONAL CLR JMPR IN? (Y OR N) /
(2) 005526 044777 020123 044124          MTCN:   .ASCIZ  <377>/IS THE H325 CONNECTOR ON? (Y OR N) /
(2) 005575 377 020043 043117          MTOTAL: .ASCIZ  <377>/# OF DUP'S (IN OCTAL) /
(2) 005630 050377 044522 051117          MPAR:   .ASCIZ  <377>/PRIORITY (4 TO 7) /
(2) 005654 051777 041505 052040          MSTJM:  .ASCIZ  <377>/SEC TX JMPR IN? (Y OR N) /

```



DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 41  
DZDPCA.CMB END OF PASS ROUTINE

(2)	005707	377	042523	020103	MSRJM:	.ASCIZ	<377>/SEC RX JMPR IN? (Y OR N) /
(2)	005742	046777	050101	047440	XHEAD:	.ASCIZ	<377>/MAP OF DUPI1 STATUS/<377>
(2)					.EVEN		
(2)	005770	000002			XSTATQ:	2	
1599911	005772	006	003		.BYTE	6,3	
1599912	005774	001236			TEMP1		
1599913	005776	006	002		.BYTE	6,2	
1599914	006000	001240			TEMP2		
1599915					.EVEN		
1599916	006002	000000			TEMP:	0	
1599917		006044			. = +40		
1599918	006044	000000			MDATA:	0	
1600		006106			. = +40		
1601	006106	000000			INBUF:	0	
1602		006150			. = +40		
1603	006150	000001			TRP.PC:	.BLKW 1	
1604							

```

1605
1606
1607
1608
1609
1610
1611
1612
1613
1614 006152 105737 001310          CYCLE: TSTB  DUPACTV      ;ARE ANY DUP11'S TO BE TESTED?
1615 006156 001004                    BNE  1$          ;BR IF OK.
1616 006160 104402 005133          TYPE  ,MERR2    ;NO DUP11'S SELECTED!!
1617 006164 000000                    HALT             ;STOP THE SHOW.
1618 006166 000776                    BR           ;DISQUALIFY CONT. SW.
1619 006170 133737 001314 001310 1$: BITB  RUN,DUPACTV ;IS THIS ONE "ACTIVE"
1620 006176 001020                    BNE  2$          ;BR IF GOOD ONE FOUND.
1621 006200 000241                    CLC             ;CLEAR PROC. CARRY BIT.
1622 006202 106137 001314          ROLB  RUN      ;UPDATE POINTER
1623 006206 105537 001314          ADCB  RUN      ;CATCH CARRY FROM RUN
1624 006212 062737 000006 001316  ADD  #6,CREAM  ;UPDATE ADDRESS POINTER.
1625 006220 022737 001560 001316  CMP  #DUP.END,CREAM
1626 006226 001360                    BNE  1$          ;KEEP GOING; NOT ALL TESTED FOR.
1627 006230 012737 001500 001316  MOV  #DUP.MAP,CREAM ;RESET ADDRESS POINTER.
1628 006236 000754                    BR  1$          ;KEEP LOOKING FOR ACTIVE DUP11
1629 006240 000241                    CLC             ;CLEAR PROC. CARRY.
1630 006242 106137 001314          ROLB  RUN      ;UPDATE POINTER.
1631 006246 105537 001314          ADCB  RUN      ;CATCH CARRY.
1632 006252 013700 001316          MOV  CREAM,RO  ;GET ADDRESS POINTER.
1633 006256 062737 000006 001316  ADD  #6,CREAM  ;UPDATE.
1634 006264 022737 001560 001316  CMP  #DUP.END,CREAM
1635
1636 006272 001003                    BNE  3$          ;ALL DONE?
1637 006274 012737 001500 001316  MOV  #DUP.MAP,CREAM ;BR IF NO.
1638 006302 012037 001404          MOV  (RO)+,RXCSR ;RESTORE POINTER.
1639 006306 012037 001374          MOV  (RO)+,DUPRVC ;LOAD SYSTEM CTRL. REG
1640 006312 012037 001434          MOV  (RO)+,LOO.OO ;LOAD VECTOR
1641 006316 012700 000002          MOV  #2,RO     ;GET PARAMETERS
1642 006322 013737 001404 001424  MOV  RXCSR,HUPRCR ;SAVE CORE THIS WAY!
1643 006330 005237 001424          INC  HUPRCR    ;GET CONTROL REG HIGH BYTE
1644 006334 013737 001424 001406  MOV  HUPRCR,RXDBUF ;GOT IT
1645 006342 005237 001406          INC  RXDBUF   ;GET RX CONTROL REG BUFFER
1646 006346 013737 001406 001416  MOV  RXDBUF,DUPSEC ;GOT IT
1647 006354 013737 001406 001410  MOV  RXDBUF,PARCSR ;GOT SECONDARY REG SELECT REG
1648 006362 013737 001406 001422  MOV  RXDBUF,HUPRBF ;GOT PARAMETER STATUS REGISTER
1649 006370 005237 001422          INC  HUPRBF   ;GET RX BUFFER HIGH BYTE
1650 006374 013737 001422 001420  MOV  HUPRBF,HUPPSR ;GOT IT
1651 006402 013737 001420 001412  MOV  HUPPSR,TXCSR ;GOT PAR STATUS REG HIGH BYTE
1652 006410 005237 001412          INC  TXCSR    ;GET TX CONTROL REGISTER
1653 006414 013737 001412 001430  MOV  TXCSR,HUPTCR ;GOT IT
1654 006422 005237 001430          INC  HUPTCR   ;GET TX CONTROL REG HIGH BYTE
1655 006426 013737 001430 001414  MOV  HUPTCR,TXDBUF ;GOT IT
1656 006434 005237 001414          INC  TXDBUF  ;BET TX BUFFER
1657 006440 013737 001414 001426  MOV  TXDBUF,HUPTBF ;GOT IT
1658 006446 005237 001426          INC  HUPTBF  ;GET TX BUFFER HIGH BYTE
1659
1660 006452 013737 001374 001376  MOV  DUPRVC,DUPRPS ;RX VECTOR

```

1661	006460	060037	001376			ADD	RO, DUPRPS	;RX PRIORITY LEVEL
1662	006464	013737	001376	001400		MOV	DUPRPS, DUPTVC	
1663	006472	060037	001400			ADD	RO, DUPTVC	;TX VECTOR
1664	006476	013737	001400	001402		MOV	DUPTVC, DUPTPS	
1665	006504	060037	001402			ADD	RO, DUPTPS	;TX PRIORITY LEVEL
1666								
1667								
1668	006510	012700	001434			MOV	#LOO.00, RO	;LOAD STAU8 00-00
1669	006514	012701	001432			MOV	#MASK.A, R1	;PREPARE MASK.
1670	006520	012702	001433			MOV	#CLK.A, R2	;PREPARE CLOCKS
1671	006524	004737	006670			JSR	PC, FIX.00	;GO AND CALCULATE CONFIGURATION.
1672	006530	005737	000042			TST	#42	
1673	006534	001050				BNE	#42	
1674	006536	032777	000002	172436		BIT	#SW01, #SWR	;IF SW01=1, GET STARTING TEST #
1675	006544	001444				BEQ	#42	
1676	006546	104402	005064		7S:	TYPE	, MCRLF	
1677	006552	104403				INSTR	; OUTPUT MESSAGE & GET INPUT STRING	
1678	006554	005325				MTSTN	; MESSAGE	
1679	006556	104405				PARAM	; CONVERT STRING	
1680	006560	000001				1	; LOW LIMIT	
1681	006562	001000				1000	; HIGH LIMIT	
1682	006564	001226				TSTNO	; STORE AT THIS LOCATION	
1683	006566	000			.BYTE	0	; MASK	
1684	006567	001			.BYTE	1	; HOW MANY TIMES + 2	
1685	006570	012700	007304			MOV	#TST1, RO	
1686	006574	022710	012737		5S:	CMP	#12737, (RO)	
1687	006600	001017				BNE	#6S	
1688	006602	023760	001226	000002		CMP	TSTNO, 2(RO)	
1689	006610	001013				BNE	#6S	
1690	006612	022760	001226	000004		CMP	#TSTNO, 4(RO)	
1691	006620	001007				BNE	#6S	
1692	006622	010037	001214			MOV	RO, RETURN	;SAVE PC
1693	006626	104402	005064			TYPE	, MCRLF	
1694	006632	104402	005130			TYPE	, MR	
1695	006636	000412				BR	#8S	
1696	006640	005720			6S:	TST	(RO)+	
1697	006642	020027	021522			CMP	RO, #TLAST+10	
1698	006646	001352				BNE	#5S	
1699	006650	104402	005060			TYPE	, MQM	
1700	006654	000734				BR	#7S	
1701								
1702	006656	012737	007304	001214	4S:	MOV	#TST1, RETURN	;PREPARE RETURN ADDRESS
1703	006664	000177	172324		8S:	JMP	#RETURN	;GO START TESTING.
1704								
1705	006670	011003			FIX.00:	MOV	(RO), R3	;GET PARAMETERS.
1706	006672	000207			5S:	RTS	PC	



1707  
1708  
1709  
1710  
1711  
1712  
1713  
1714  
1715  
1716  
1717  
1718  
1719  
1720  
1721  
1722  
1723  
1724  
1725  
1726  
1727  
1728  
1729  
1730  
1731  
1732  
1733  
1734  
1735  
1736  
1737  
1738  
1739  
1740  
1741  
1742  
1743  
1744  
1745  
1746  
1747  
1748  
1749  
1750  
1751  
1752  
1753  
1754  
1755  
1756  
1757  
1758  
1759  
1760  
1761  
1762

006674 012737 000010 001326  
006702 013700 001324  
006706 012701 000200  
006712 006000  
006714 005537 001330  
006720 022737 000005 001330  
006726 003004  
006730 005037 001330  
006734 005237 001326  
006740 006001  
006742 103363  
006744 000207  
  
042777 002000 172436  
006754 006037 001324  
006760 103003  
006762 052777 002000 172422  
006770 000240  
006772 104412 000002  
006776 005337 001326  
007002 001361  
007004 000201  
  
012137 001244  
007012 005037 001246  
007016 042777 002000 172366  
007024 104412 000002  
007030 052777 002000 172354  
007036 104412 000002  
007042 005237 001246  
007046 022737 000006 001246  
007054 001365  
007056 042777 002000 172326  
007064 104412 000002  
007070 005337 001244  
007074 001346  
007076 000201  
  
012137 001244  
007104 005037 001246  
007110 052777 002000 172274  
007116 104412 000002

;ROUTINE TO DETERMINE IF BIT STUFFING IS REQUIRED  
-----  
STUFCK: MOV #8, SHIFTS  
MOV DATA, R0  
MOV #200, R1  
1S: ROR R0  
ADC MIND  
CMP #5, MIND  
BGT 2S  
CLR MIND  
INC SHIFTS  
2S: ROR R1  
BCC 1S  
RTS PC  
  
;THIS ROUTINE POKES THE RECEIVER BIT WINDOW  
;BASED ON THE INFORMATION IN DATA AND SHIFTS  
-----  
RPOKE: BIC #BITW, @TXCSR  
ROR DATA  
BCC 1S  
BIS #BITW, @TXCSR  
1S: NOP  
PKCLK ,2  
DEC SHIFTS  
BNE RPOKE  
RTS R1  
  
RFLG: MOV (R1)+, TEMP4 ;GET THE # OF FLAGS  
1S: CLR TEMPS ;CLEAR ONES COUNT  
BIC #BITW, @TXCSR ;SET FIRST BIT  
PKCLK ,2 ;PUSH OUT THE BIT  
2S: BIS #BITW, @TXCSR ;LOAD THE BIT  
PKCLK ,2 ;PUSH OUT THE BIT  
INC TEMPS ;INCREMENT 1'S COUNTER  
CMP #6, TEMPS ;DID WE PUSH OUT 6 ONES  
BNE 2S ;NO-GO BACK  
BIC #BITW, @TXCSR ;SET THE LAST BIT  
PKCLK ,2 ;PUSH OUT THE BIT  
DEC TEMP4 ;ARE WE DONE WITH FLAGS?  
BNE 1S ;BR IF NO  
RTS R1  
  
SFLG: MOV (R1)+, TEMP4 ;GET THE # OF FLAGS  
1S: CLR TEMPS ;CLEAR ONES COUNT  
2S: BIS #BITW, @TXCSR ;LOAD THE BIT  
PKCLK ,2 ;PUSH OUT THE BIT



1816  
1817  
1818  
1819  
1820  
1821  
1822  
1823  
1824  
1825  
1826  
1827  
1828  
1829  
1830  
1831  
1832  
1833  
1834  
1835  
1836  
1837  
1838  
1839  
1840  
1841  
1842  
1843  
1844  
1845  
1846  
1847  
1848  
1849  
1850  
1851  
1852  
1853  
1854  
1855  
1856  
1857  
1858  
1859  
1860  
1861  
1862  
1863  
1864  
1865  
1866  
1867  
1868  
1869  
1870  
1871

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

\*\*\*\*\*  
\*  
: TEST 1  
\*  
\*\*\*\*\*

```

TST1:  MOV    #1, @TSTNO
        MOV    #TST2, NEXT
        BIS    #MRESET, @TXCSR ; RESET THE DEVICE
        JSR    PC, SMALL ; WAIT FOR RESET TO FINISH
        MOV    RXCSR, R3 ; GET THE RECEIVER CONTROL REGISTER
        BIS    #RCREN, @PARCSR ; TURN OFF CRC
        BIS    #RCVEN, (R3) ; TURN ON THE RECEIVER
        BIS    #MMODE, @TXCSR ; ENTER M/MODE
        JSR    R1, RFLG ; PUSH OUT A FLAG
        BIT    #REACT, (R3) ; CHECK RECEIVER ACTIVE
        BEQ    1$ ; BRANCH IF BIT IS OFF
        HLT    6 ; BIT WAS SET - SHOULD BE CLEAR
1$:    CLR    DATA ; SETUP FOR CHAR
        MOV    #9, SHIFTS ; POKE CLOCK SETUP
        JSR    R1, @POKE ; PUSH DATA INTO RECEIVER
        BIT    #RXDONE, (R3) ; TEST DONE
        BNE    2$ ; BR IF SET
        MOV    #RXDONE, R5 ; ERROR MSG SETUP
        MOV    (R3), R4 ; SET UP FOUND
        HLT    3 ; RXDONE FAILED TO SET
2$:    BIT    #RSOM, @RXDBUF ; TEST FOR START OF MSG
        BNE    10$ ; BR IF OK
        MOV    #RSOM, R5 ; SET EXPECTED
        MOV    RXDBUF, R3 ; SET UP ERROR MSG
        MOV    (R3), R4 ; SET FOUND
        HLT    3 ; RSOM FAILED TO SET
10$:   MOV    RXCSR, R3
        BIT    #REACT, (R3)
        BNE    11$
11$:   HLT    14
        SCOPE

```

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

\*\*\*\*\*  
\*  
\*\*\*\*\*



```

1872 ; TEST 2
1873 ;*****
1874 ;*****
1875 ;*****
1876 007472 012737 000002 001226 TST2: MOV #2, @TSTNO
1877 007500 012737 007762 001216 MOV #TST3, NEXT
1878 007506 052777 000400 171676 BIS #MRESET, @TXCSR ; RESET THE DEVICE
1879 007514 004737 004772 JSR PC, SMALL ; WAIT FOR RESET TO FINISH
1880 007520 013703 001404 MOV RXCSR, R3 ; GRT THE RECEIVER CONTROL REGISTER
1881 007524 052777 001000 171656 BIS #CRCEN, @PARCSR ; TURN OFF CRC
1882 007532 052713 000020 BIS #RCVEN, (R3) ; TURN ON THE RECEIVER
1883 007536 052777 010000 171644 BIS #PRISEC, @PARCSR ; ENTER SECONDARY MODE
1884 007544 052777 014000 171640 BIS #MMODE, @TXCSR ; ENTER M/MODE
1885 007552 004137 007006 JSR R1, RFLG ; PUSH OUT A FLAG
1886 007556 000001 I
1887 007560 032713 004000 BIT #REACT, (R3) ; CHECK RECEIVER ACTIVE
1888
1889 007564 001403 BEQ 3$ ; BR IF OFF
1890 007566 005005 CLR R5 ; SET EXPECTED
1891 007570 011304 MOV (R3), R4 ; SET FOUND
1892 007572 104003 HLT 3 ; BIT WAS SET AND SHOULD BE CLEARED
1893 ; WE ARE IN SECONDARY MODE
1894 007574 032713 000200 3$: BIT #RXDONE, (R3) ; TEST DONE
1895 007600 001401 BEQ 4$ ; BR IF CLEAR
1896 007602 104003 HLT 3 ; DONE IS SET - SHOULD NOT BE BECAUSE
1897 ; WE ARE IN SECONDARY MODE AND HAVE NOT PUSHED DATA
1898 007604 005037 001324 4$: CLR DATA ; CLEAR DATA CHAR
1899 007610 012737 000011 001326 MOV #9, SHIFTS ; LOAD THE # OF BITS TO PUSH
1900 007616 004137 006746 JSR R1, RPOKE ; PUSH DATA TO RECEIVER
1901 007622 032713 004000 BIT #REACT, (R3) ; TEST ACTIVE
1902 007626 001004 BNE 5$ ; BR IF CLEAR
1903 007630 012705 004000 MOV #REACT, R5 ; SET EXPECTED
1904 007634 011304 MOV (R3), R4 ; SET FOUND
1905 007636 104003 HLT 3 ; ACTIVE FAILED TO SET
1906 007640 032713 000200 5$: BIT #RXDONE, (R3) ; TEST DONE
1907 007644 001404 BEQ 6$ ; BR IF NOT SET - THE CHAR WAS
1908 ; THE SECONDARY ADRS
1909 007646 012705 000200 MOV #RXDONE, R5 ; SET EXPECTED
1910 007652 011304 MOV (R3), R4 ; SET FOUND
1911 007654 104003 HLT 3 ; DONE WAS SET AND SHOULDN'T BE - THE FIRST
1912 ; DATA WAS SECONDARY ADRS, NOT DATA TO THE BUFFER
1913 007656 032777 000400 171522 6$: BIT #R50M, @RXDBUF ; CHECK START OF MSG
1914 007664 001401 BEQ 7$ ; BR IF NOT SET
1915 007666 104020 HLT 20 ; START OF MSG SHOULD BE CLEAR
1916 007670 112737 000252 001324 7$: MOV #252, DATA ; LOAD DATA CHAR
1917 007676 012737 000010 001326 MOV #8, SHIFTS ; LOAD CLOCK COUNT
1918 007704 004137 006746 JSR R1, RPOKE ; PUSH OUT DATA
1919 007710 104412 000002 PKCLK 2 ; POKE ANOTHER FULL CLOCK
1920 007714 032713 000200 BIT #RXDONE, (R3) ; TEST DONE
1921 007720 001011 BNE 10$ ; BR IF SET
1922 007722 012705 000200 MOV #RXDONE, R5 ; SET EXPECTED
1923 007726 011304 MOV (R3), R4 ; SET FOUND
1924 007730 104003 HLT 3 ; DONE FAILED TO SET AFTER PUSHING
1925 ; OUT A CHAR IN SEC. MODE
1926 007732 032777 000400 171446 BIT #R50M, @RXDBUF
1927 007740 001001 BNE 10$

```

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 48  
DZDPCA.CMB TEST OF REC ENABLE, RXACT, AND R50M IN SECONDARY MODE

1928 007742 104021  
1929 007744 013703 001404  
1930 007750 032713 004000  
1931 007754 001001  
1932 007756 104014  
1933 007760 104400

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

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

\*\*\*\*\*

\*\*\*\*\*  
\*  
TEST 3  
\*  
\*\*\*\*\*

1946 007762 012737 000003 001226  
1947 007770 012737 010210 001216  
1948 007776 052777 000400 171406  
1949 010004 004737 004772  
1950 010010 052777 000020 171366  
1951 010016 052777 001000 171364  
1952 010024 005037 001236  
1953 010030 052777 014000 171354  
1954 010036 004137 007006  
1955 010042 000001  
1956 010044 012737 000125 001324  
1957 010052 012737 000010 001326  
1958 010060 004137 006746  
1959 010064 042777 002000 171320  
1960 010072 104412 000002  
1961 010076 032777 000200 171300  
1962 010104 001001  
1963 010106 104000  
1964 010110 105777 171272  
1965 010114 052777 002000 171270  
1966 010122 104412 000002  
1967 010126 005237 001236  
1968 010132 022737 000006 001236  
1969 010140 001365  
1970 010142 042777 002000 171242  
1971 010150 104412 000004  
1972 010154 017737 171224 001240  
1973 010162 032777 001000 171216  
1974 010170 001001  
1975 010172 104001  
1976 010174 032737 000200 001240  
1977 010202 001001  
1978 010204 104002  
1980 010206 104400

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

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

1981  
1982  
1983



1984  
1985  
1986  
1987  
1988  
1989  
1990  
1991  
1992  
1993  
1994  
1995  
1996  
1997  
1998  
1999  
2000  
2001  
2002  
2003  
2004  
2005  
2006  
2007  
2008  
2009  
2010  
2011  
2012  
2013  
2014  
2015  
2016  
2017  
2018  
2019  
2020  
2021  
2022  
2023  
2024  
2025  
2026  
2027  
2028  
2029  
2030  
2031  
2032  
2033  
2034  
2035  
2036  
2037  
2038  
2039

;\*WITH DONE IN PRIMARY MODE  
;\*USING A COMMON ZERO BIT IN FLAGS.

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

```
TEST 4  
*****  
*****  
TST4:  MOV    #4,@TSTNO  
      MOV    #TSTS,NEXT  
      BIS    #MRESET,@TXCSR ;RESET THE DEVICE  
      JSR    PC,SMALL      ;WAIT FOR RESET TO FINISH  
      BIS    #RCVEN,@RXCSR ;TURN ON RECEIVER  
      BIS    #CRCEN,@PARCSR ;TURN OFF CRC  
      CLR    TEMP1        ;BIT COUNTER  
      BIS    #MMODE,@TXCSR ;ENTER MAINT MODE  
      JSR    R1,RFLG      ;PUSH OUT A FLAG  
      1  
      JSR    R1,SFLG      ;PUSH OUT SPECIAL FLAGS  
      4  
      MOV    #125,DATA    ;CLEAR OUT CHAR  
      MOV    #8,SHIFTS   ;LOAD THE # OF CLOCKS  
      JSR    R1,RPOKE    ;PUSH OUT THE DATA  
      BIC    #BITW,@TXCSR ;CLEAR THE DATA WINDOW  
      PKCLK  2           ;PUSH OUT THE BIT  
      BIT    #RXDONE,@RXCSR ;CHECK FOR DONE  
      BNE   5$          ;BR IF SET  
      HLT  
      5$:  MOVB   @RXDBUF,TEMP ;GET DATA  
      CMPB  #125,TEMP    ;CHECK IT  
      BEQ   64$         ;BR IF A MATCH  
      HLT   10          ;DATA COMPARE ERROR  
      64$: BIS    #BITW,@TXCSR ;PUT A 1 IN THE WINDOW  
      PKCLK  2           ;PUSH IT OUT  
      INC   TEMP1       ;INC THE # TO DO  
      CMP   #6,TEMP1    ;CHECK FOR FINISH  
      BNE   64$         ;BR IF MORE TO GO  
      BIC   #BITW,@TXCSR ;CLEAR THE WINDOW  
      PKCLK  4           ;PUSH OUT 2 BINTS  
      MOV   @RXCSR,TEMP2 ;GET THE CSR  
      BIT   #REOM,@RXDBUF ;TEST FOR END OF MSG  
      BNE   1$          ;BR IF SET  
      HLT   1           ;BIT FAILED TO SET  
      1$:  BIT   #RXDONE,TEMP2 ;TEST DONE  
      BNE   4$          ;BR IF SET  
      HLT   2           ;DONE FAILED TO SET  
      4$:  SCOPE ;SCOPE THIS TEST
```

```
***** TEST 5 *****  
;*TEST TO PROVE THE INTERACTION OF REOM  
;*WITH DONE IN SECONDARY MODE. TEST FOR REOM  
;*AT THE WRONG ADDRESS, THEN AT THE CORRECT  
;*SECONDARY ADDRESS.
```



# K04

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 50  
 DZDPCA.CMB TEST OF REOM AND DONE IN SECONDARY MODE

```

2040      ::*****
2041      ::*****
2042      *
2043      TEST 5
2044      *
2045      ::*****
2046      ::*****
2047 010460 012737 000005 001226 TSTS:  MOV    #5, @TSTNO
2048 010466 012737 011066 001216      MOV    #TST6, NEXT
2049 010474 052777 000400 170710      BIS    #MRESET, @TXCSR ; RESET THE DEVICE
2050 010502 004737 004772          JSR    PC, SMALL ; WAIT FOR RESET TO FINISH
2051 010506 052777 000020 170670      BIS    #RCVEN, @RXCSR ; TURN ON RECEIVER
2052 010514 052777 001000 170666      BIS    #CRCEN, @PARCSR ; TURN OFF CRC
2053 010522 005037 001236          CLR    TEMP1 ; BIT COUNTER
2054 010526 052777 010000 170654      BIS    #PRISEC, @PARCSR ; ENTER SECONDARY MODE
2055 010534 052777 014000 170650      BIS    #MMODE, @TXCSR ; ENTER MAINT MODE
2056 010542 004137 007006          JSR    R1, RFLG ; PUSH OUT A FLAG
2057 010546 000001          1 ; ONE FLAG
2058 010550 012737 000001 001324      MOV    #1, DATA ; LOAD A CHAR
2059 010556 012737 000010 001326      MOV    #8, SHIFTS ; LOAD THE # OF CLOCKS
2060 010564 004137 006746          JSR    R1, RPOKE ; PUSH OUT THE DATA
2061 010570 042777 002000 170614      BIC    #BITW, @TXCSR ; CLEAR THE DATA WINDOW
2062 010576 104412 000002          PKCLK 2 ; PUSH OUT THE BIT
2063 010602 032777 000200 170574      BIT    #RXDONE, @RXCSR ; CHECK FOR DONE
2064 010610 001401          BEQ    SS ; BR IF CLEAR
2065 010612 104000          HLT ; DONE BIT ERROR
2066 010614 105777 170566          SS:  TSTB  @RXDBUF ; READ THE BUFFER
2067 010620 052777 002000 170564      64$:  BIS    #BITW, @TXCSR ; PUT A 1 IN THE WINDOW
2068 010626 104412 000002          PKCLK 2 ; PUSH IT OUT
2069 010632 005237 001236          INC    TEMP1 ; INC THE # TO DO
2070 010636 022737 000006 001236      CMP    #6, TEMP1 ; CHECK FOR FINISH
2071 010644 001365          BNE    64$ ; BR IF MORE TO GO
2072 010646 042777 002000 170536      BIC    #BITW, @TXCSR ; CLEAR THE WINDOW
2073 010654 104412 000004          PKCLK, 4 ; PUSH OUT 2 BINTS
2074 010660 032777 001000 170520      BIT    #REOM, @RXDBUF ; TEST REC END OF MSG
2075 010666 001401          BEQ    2$ ; BR IF NOT SET
2076 010670 104004          HLT 4 ; BIT IS SET AND SHOULDN'T
2077          ; BE - THIS IS SECONDARY MODE
2078 010672 004137 007006          2$:  JSR    R1, RFLG ; OUTPUT A FLAG
2079 010676 000001          1
2080 010700 005037 001324          CLR    DATA ; CLEAR DATA
2081 010704 012737 000010 001326      MOV    #8, SHIFTS ; LOAD THE # OF CLOCKS
2082 010712 004137 006746          JSR    R1, RPOKE ; PUSHOUT SEC. ADRS
2083 010716 012737 000252 001324      MOV    #252, DATA ; LOAD DATA
2084 010724 012737 000010 001326      MOV    #8, SHIFTS ; LOAD # OF CLOCKS
2085 010732 004137 006746          JSR    R1, RPOKE ; PUSHOUT DATA
2086 010736 042777 002000 170446      BIC    #BITW, @TXCSR ; CLEAR OUT DATA WINDOW
2087 010744 104412 000002          PKCLK, 2 ; PUSH OUT A BIT
2088 010750 032777 000200 170426      BIT    #RXDONE, @RXCSR ; CHECK DONE
2089 010756 001001          BNE    6$ ; BR IF SET
2090 010760 104002          HLT 2 ; DONE FAILED TO SET
2091 010762 005037 001236          6$:  CLR    TEMP1 ; CLEAR TO KEEP TRACK OF THE BITS
2092 010766 105777 170414          TSTB  @RXDBUF ; READ THE BUFFER
2093 010772 052777 002000 170412      65$:  BIS    #BITW, @TXCSR ; SET THE WINDOW BIT
2094 011000 104412 000002          PKCLK, 2 ; PUSH IT OUT
2095 011004 005237 001236          INC    TEMP1 ; CHECK TO SEE IF
  
```

```

2096 011010 022737 000006 001236      CMP      #6,TEMP1      ;6 ARE DONE YET
2097 011016 001365                BNE      65$          ;BR IF NO
2098 011020 042777 002000 170364      BIC      #BITW,@TXCSR ;CLEAR THE WINDOW
2099 011026 104412 000004                PKCLK,   4           ;PUSH 2 BITS
2100 011032 017737 170346 006002      MOV      @RXCSR,TEMP  ;READ THE CSR
2101 011040 032777 001000 170340      BIT      #REOM,@RXDBUF ;TEST END OF MSG
2102 011046 001001                BNE      3$          ;BR IF SET
2103 011050 104001                HLT      1           ;EOM FAILED TO SET IN SEC. MODE
2104 011052 032737 000200 001240 3$:      BIT      #RXDONE,TEMP2 ;TEST DONE
2105 011060 001001                BNE      4$          ;BR IF SET
2106 011062 104002                HLT      2           ;BIT FAILED TO SET

```

```

2107
2108 011064 104400                4$:      SCOPE          ;SCOPE THIS TEST
2109

```

```

:***** TEST 6 *****
:*TEST TO PROVE THE INTERACTION OF ABORT
:*WITH A DONE AND RX ERROR IN PRIMARY MODE

```

```

:*****

```

```

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

```

```

2120
2121 011066 012737 000006 001226  TST6:  MOV      #6,@TSTNO
2122 011074 012737 011300 001216      MOV      #TST7,NEXT
2123 011102 052777 000400 170302      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2124 011110 004737 004772                JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
2125 011114 052777 001000 170266      BIS      #CRCEN,@PARCSR ;TURN OFF CRC
2126 011122 052777 000020 170254      BIS      #RCVEN,@RXCSR  ;TURN ON RECEIVER
2127 011130 052777 014000 170254      BIS      #MMODE,@TXCSR  ;ENTER MAINT MODE
2128 011136 004137 007006                JSR      R1,RFLG     ;PUSH OUT A FLAG
2129 011142 000001                1           ;ONE FLAG
2130 011144 005037 001324                CLR      DATA      ;CLEAR DATA
2131 011150 012737 000010 001326      MOV      #8,SHIFTS   ;LOAD CLOCKS
2132 011156 004137 006746                JSR      R1,RPOKE    ;PUSH OUT DATA CHAR
2133 011162 052777 002000 170222      BIS      #BITW,@TXCSR  ;SET THE WINDOW
2134 011170 104412 000002                PKCLK,   2           ;PUSH OUT A BIT
2135 011174 105777 170206                TSTB     @RXDBUF     ;READ A CHAR
2136 011200 112737 000177 001324      MOVB     #177,DATA    ;LOAD A SECOND CHAR
2137 011206 012737 000010 001326      MOV      #8,SHIFTS   ;LOAD CLOCKS
2138 011214 004137 006746                JSR      R1,RPOKE    ;PUSH OUT THE DATA
2139 011220 017737 170160 001236      MOV      @RXCSR,TEMP1 ;READ THE CSR
2140 011226 017700 170154                MOV      @RXDBUF,RO  ;SAVE THE BUFFER
2141 011232 032700 002000                BIT      #RABORT,RO  ;TEST ABORT BIT
2142 011236 001001                BNE      1$          ;BR IF SET
2143 011240 104005                HLT      5           ;ABORT BIT FAILED TO SET
2144 011242 032737 004000 001236 1$:      BIT      #REACT,TEMP1 ;TEST REC ACT
2145 011250 001401                BEQ      2$          ;BR IF CLEAR
2146 011252 104006                HLT      6           ;REC ACT. FAILED TO CLEAR
2147 011254 032737 000200 001236 2$:      BIT      #RXDONE,TEMP1 ;TEST DONE
2148 011262 001001                BNE      3$          ;BR IF SET
2149 011264 104002                HLT      2           ;DONE FAILED TO SET
2150 011266 032700 100000 3$:      BIT      #RXDERR,RO  ;TEST ERROR BIT
2151 011272 001001                BNE      12$         ;BR IF SET

```



```

2152 011274 104007          HLT      7          ;ERROR BIT FAILED TO SET AFTER ABORT
2153 011276 104400          12$: SCOPE          ;SCOPE THIS TEST
2154                                     ;***** TEST 7 *****
2155                                     ;*TEST TO PROVE THE INTERACTION OF ABORT IN SECONDARY MODE
2156                                     ;*TEST FOR ABORT AT THE WRONG SECONDARY ADDRESS, THEN TEST
2157                                     ;*AT THE CORRECT SECONDARY ADDRESS
2158                                     ;*****
2159                                     ;*****
2160                                     ;*
2161                                     ;TEST 7
2162                                     ;*
2163                                     ;*****
2164                                     ;*****
2165 011300 012737 000007 001226 TST7: MOV      #7, @#TSTNO
2166 011306 012737 011636 001216      MOV      #TST10, NEXT
2167 011314 052777 000400 170070      BIS      #MRESET, @TXCSR ;RESET THE DEVICE
2168 011322 004737 004772          JSR      PC, SMALL      ;WAIT FOR RESET TO FINISH
2169 011326 052777 001000 170054      BIS      #CRCEN, @PARCSR ;TURN OFF CRC
2170 011334 052777 000020 170042      BIS      #RCVEN, @RXCSR  ;TURN ON RECEIVER
2171 011342 052777 010000 170040      BIS      #PRISEC, @PARCSR ;ENTER SECONDARY MODE
2172 011350 052777 014000 170034      BIS      #MMODE, @TXCSR  ;ENTER MAINT MODE
2173 011356 004137 007006          JSR      R1, RFLG      ;PUSH OUT A FLAG
2174 011362 000001 000001          1          ;ONE FLAG
2175 011364 012737 000001 001324      MOV      #1, DATA ;LOAD WRONG ADRS
2176 011372 012737 000010 001326      MOV      #8, SHIFTS ;LOAD CLOCKS
2177 011400 004137 006746          JSR      R1, RPOKE     ;PUSH OUT DATA CHAR
2178 011404 052777 002000 170000      BIS      #BITW, @TXCSR  ;SET THE WINDOW
2179 011412 104412 000002          PKCLK, 2 ;PUSH OUT A BIT
2180 011416 105777 167764          TSTB   @RXDBUF      ;READ A CHAR
2181 011422 112737 000177 001324      MOVB   #177, DATA ;LOAD A SECOND CHAR
2182 011430 012737 000010 001326      MOV      #8, SHIFTS ;LOAD CLOCKS
2183 011436 004137 006746          JSR      R1, RPOKE     ;PUSH OUT THE DATA
2184 011442 032777 002000 167736      BIT     #RABORT, @RXDBUF ;TEST FOR ABORT
2185 011450 001401          BEQ     5$          ;BR IF NOT SET
2186 011452 104011          HLT     11          ;ABORT IS SET AND SHOULDN'T
2187                                     ;BE - THIS IS A SECONDARY STATION
2188 011454 004137 007006          5$: JSR      R1, RFLG     ;PUSH OUT A FLAG
2189 011460 000001 000001          1          ;ONE FLAG
2190 011462 005037 001324          CLR     DATA ;LOAD ADRS
2191 011466 012737 000010 001326      MOV      #8, SHIFTS ;LOAD CLOCKS
2192 011474 004137 006746          JSR      R1, RPOKE     ;PUSH OUT THE ADRS
2193 011500 012737 000252 001324      MOV      #252, DATA ;LOAD A CHAR
2194 011506 012737 000010 001326      MOV      #8, SHIFTS ;LOAD CLOCKS
2195 011514 004137 006746          JSR      R1, RPOKE     ;PUSH OUT THE CHAR
2196 011520 052777 002000 167664      BIS      #BITW, @TXCSR  ;SET THE WINDOW
2197 011526 104412 000002          PKCLK, 2 ;PUSH OUT A BIT
2198 011532 105777 167650          TSTB   @RXDBUF      ;READ THE CHAR
2199 011536 112737 000177 001324      MOVB   #177, DATA ;LOAD THE ABORT
2200 011544 012737 000010 001326      MOV      #8, SHIFTS ;LOAD CLOCKS
2201 011552 004137 006746          JSR      R1, RPOKE     ;PUSH OUT THE ABORT
2202 011556 017737 167622 001240      MOV      @RXCSR, TEMP2 ;READ THE CSR
2203 011564 017700 167616          MOV      @RXDBUF, RO ;SAVE THE BUFFER
2204 011570 032700 002000          BIT     #RABORT, RO ;TEST THE ABORT BIT
2205 011574 001001          SNE     6$          ;BR IF SET
2206 011576 104005          HLT     5          ;BIT IS CLEAR - SHOULD BE SET
2207 011600 032737 004000 001240      6$: BIT     #REACT, TEMP2 ;TEST ACTIVE

```



```

2208 011606 001401          BEQ      7$          ;BR IF CLEAR
2209 011610 104006          HLT      6          ;BIT FAILED TO CLEAR AFTER ABORT
2210 011612 032737 000200 001240 7$:  BIT      #RXDONE,TEMP2 ;TEST DONE
2211 011620 001001          BNE     10$         ;BR IF SET
2212 011622 104002          HLT      2          ;DONE FAILED TO SET
2213 011624 032700 100000          BIT      #RXDERR,RO   ;TEST FOR ERROR
2214 011630 001001          BNE     12$         ;BR IF SET
2215 011632 104007          HLT      7          ;ERROR WASN'T SET AFTER ABORT
2216
2217 011634 104400          12$:  SCOPE          ;SCOPE THIS TEST
2218          ;***** TEST 10 *****
2219          ;*DATA OVERRUN TEST IN PRIMARY MODE. TEST TO
2220          ;*PROVE OVERRUN ERROR AND RX ERROR WILL OCCUR
2221          ;*****
2222
2223          ;*****
2224          ;*
2225          ;* TEST 10
2226          ;*
2227          ;*****
2228          ;*****
2229 011636 012737 000010 001226 TST10: MOV      #10, @TSTNO
2230 011644 012737 012010 001216      MOV      #TST11, NEXT
2231 011652 052777 000400 167532      BIS      #MRESET, @TXCSR ;RESET THE DEVICE
2232 011660 004737 004772          JSR      PC, SMALL ;WAIT FOR RESET TO FINISH
2233 011664 013703 001404          MOV      RXCSR, R3 ;LOAD THE CONTROL REGISTER
2234 011670 052777 001000 167512      BIS      #CRCEN, @PARCSR ;TURN OFF CRC
2235 011676 052713 000020          BIS      #RCVEN, (R3) ;TURN ON THE RECEIVER
2236 011702 052777 014000 167502      BIS      #MMODE, @TXCSR ;ENTER MAINTENANCE MODE
2237 011710 004137 007006          JSR      R1, RFLG ;PUSH OUT A FLAG
2238 011714 000001          1 ;ONE FLAG
2239 011716 012737 000252 001324      MOV      #252, DATA ;LOAD CHAR TO BE OUTPUT
2240 011724 012737 000010 001326      MOV      #8, SHIFTS ;CLOCK SETUP
2241 011732 004137 006746          JSR      R1, RPOKE ;PUSH DATA INTO RECEIVER
2242 011736 012737 000070 001324      MOV      #70, DATA ;LOAD A SECOND CHARACTER
2243 011744 012737 000011 001326      MOV      #9, SHIFTS ;SETUP #OF CLOCKS
2244 011752 004137 006746          JSR      R1, RPOKE ;PUSH THE SECOND CHARACTER INTO RX
2245 011756 017700 167424          MOV      @RXDBUF, RO ;SAVE BUFFER
2246 011762 042700 037777          BIC      #!C<RXDERR!OVRUN>, RO ;CLEAR JUNK
2247 011766 022700 140000          CMP      #RXDERR!OVRUN, RO ;CHECK TO SEE IF BOTH ARE SET
2248 011772 001401          BEQ     2$          ;BR IF OK
2249 011774 104013          HLT     13         ;THEY DIDN'T MATCH
2250
2251 011776 032713 004000          2$:  BIT      #REACT, (R3) ;TEST REC. ACT
2252 012002 001001          BNE     3$          ;BR IF STILL SET
2253 012004 104014          HLT     14         ;REC. ACTIVE CLEARED AND SHOULD BE SET
2254
2255 012006 104400          3$:  SCOPE          ;SCOPE THIS TEST
2256
2257          ;***** TEST 11 *****
2258          ;*DATA OVERRUN TEST IN SECONDARY MODE. TEST TO PROVE
2259          ;*THAT OVERRUN DOES NOT OCCUR IF THIS STATION IS NOT
2260          ;*SELECTED. THEN SELECT THIS SECONDARY STATION AND
2261          ;*PROVE OVERRUN ERROR AND RX ERROR WILL OCCUR
2262          ;*****
2263

```

2264  
2265  
2266  
2267  
2268  
2269  
2270  
2271  
2272  
2273  
2274  
2275  
2276  
2277  
2278  
2279  
2280  
2281  
2282  
2283  
2284  
2285  
2286  
2287  
2288  
2289  
2290  
2291  
2292  
2293  
2294  
2295  
2296  
2297  
2298  
2299  
2300  
2301  
2302  
2303  
2304  
2305  
2306  
2307  
2308  
2309  
2310  
2311  
2312  
2313  
2314  
2315  
2316  
2317  
2318  
2319

```

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

2287 012126 104015 HLT 15 ; OVERRUN IS SET AND SHOULDN'T BE
; THIS IS A SECONDARY STATION WHICH
; DID NOT GET ITS ADDRESS
2290 012130 004137 007006 4$: JSR R1, RFLG ; PUSH OUT A FLAG
1 ; ONE FLAG
2292 012136 005037 001324 CLR DATA ; GET CORRECT SEC. STATION ADRS
2293 012142 012737 000010 001326 MOV #8, SHIFTS ; MOVE THE #OF CLOCKS TO PUSH
2294 012150 004137 006746 JSR R1, RPOKE ; PUSH OUT THE SEC. ADRS
2295 012154 012737 000252 001324 MOV #252, DATA ; PUSH OUT
2296 012162 012737 000010 001326 MOV #8, SHIFTS ; TWO
2297 012170 004137 006746 JSR R1, RPOKE ; DATA
2298 012174 012737 000070 001324 MOV #70, DATA ; CHARACTERS
2299 012202 012737 000011 001326 MOV #9, SHIFTS ; TO FORCE
2300 012210 004137 006746 JSR R1, RPOKE ; AN OVERRUN ERROR
2301 012214 017700 167166 MOV @RXDBUF, R0 ; SAVE
2302 012220 042700 037777 BIC #C<RXDERR!OVRUN>, R0 ; CLEAR JUNK
2303 012224 022700 140000 CMP #RXDERR!OVRUN, R0 ; ARE THE ERROR BITS THERE
2304 012230 001401 BEQ 6$ ; BR IF YES
2305 012232 104013 HLT 13 ; ERROR BITS MISSING

2307 012234 032713 004000 6$: BIT #REACT, (R3) ; TEST ACTIVE
2308 012240 001001 BNE 3$ ; BR IF SET
2309 012242 104005 HLT 5 ; ACTIVE DROPPED OUT
2310 012244 104400 3$: SCOPE ; SCOPE THIS TEST

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

```



```

2320
2321
2322 012246 012737 000012 001226
2323 012254 012737 012512 001216
2324 012262 052777 000400 167122
2325 012270 004737 004772
2326 012274 012703 012500
2327 012300 012704 012500
2328 012304 012702 000004
2329 012310 052777 001000 167072
2330 012316 052777 014000 167066
2331 012324 052777 000020 167052
2332 012332 052777 000020 167052
2333 012340 052777 000400 167046
2334 012346 104412 000010
2335 012352 012377 167036
2336 012356 005302
2337 012360 104412 000020
2338 012364 012377 167024
2339 012370 005302
2340 012372 104412 000020
2341 012376 105777 167002
2342 012402 100401
2343 012404 104000
2344 012406 017737 166774 001324
2345 012414 121437 001324
2346 012420 001401
2347 012422 104000
2348 012424 005724
2349 012426 005702
2350 012430 001355
2351 012432 104412 000020
2352 012436 105777 166742
2353 012442 100401
2354 012444 104000
2355 012446 017737 166734 001324
2356 012454 121437 001324
2357 012460 001401
2358 012462 104000
2359 012464
2360 012464 052777 000400 166720
2361 012472 004737 004772
2362 012476 104400
2363
2364
2365 012500 000125
2366 012502 000252
2367 012504 000000
2368 012506 000377
2369 012510 000000
2370
2371
2372
2373
2374
2375

```

```

:*****
:*****
TST12: MOV #12, @TSTNO
MOV #TST13, NEXT
BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
MOV #TBLA, R3 ; LOAD THE TABLE POINTER
MOV #TBLA, R4 ; DITTO
MOV #4, R2 ; LOAD THE # OF CHARS TO DO
BIS #CRCEN, @PARCSR
BIS #MMODE, @TXCSR ; ENTER M/MODE
BIS #RCVEN, @RXCSR ; TURN ON THE RECEIVER
BIS #SEND, @TXCSR ; TURN ON THE TRANSMITTER
BIS #TSOM, @TXDBUF ; START A FLAG
PKCLK 8 ; START A FLAG
MOV (R3)+, @TXDBUF ; LOAD FIRST DATA CHAR
DEC R2 ; LOWER THE # TO DO
PKCLK 16 ; PUSH OUT
MOV (R3)+, @TXDBUF ; LOAD DATA
DEC R2 ; LOWER THE # TO DO
PKCLK 16 ; PUSH OUT DATA
TST @RXCSR ; CHECK FOR RX DONE
BMI 25 ; BR IF DONE SET
HLT ; RX DONE FAILED TO SET
MOV @RXDBUF, DATA ; GET THE BUFFER
CMPB (R4), DATA ; CHECK IT
BEQ 35 ; BR IF OK
HLT ; DATA COMPARE ERROR
TST (R4)+
TST R2 ; CHECK FOR FINISH
BNE 15 ; BR IF MORE TO GO
PKCLK 16 ; PUSH OUT LAST CHAR
TSTB @RXCSR ; CHECK FOR DONE
BMI 45 ; BR IF DONE SET
HLT ; DONE FAILED TO SET
MOV @RXDBUF, DATA ; GET THE BUFFER
CMPB (R4), DATA ; CHECK IT
BEQ 55 ; BR IF A MATCH
HLT ; DATA COMPARE ERROR
BIS #MRESET, @TXCSR ; RESET THE DEVICE
JSR PC, SMALL ; WAIT FOR RESET TO FINISH
SCOPE

```

```

:DATA TABLE
TBLA: .WORD 125
      .WORD 252
      .WORD 000
      .WORD 377
      .WORD 000

```

```

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

```



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  
2425  
2426  
2427  
2428  
2429  
2430  
2431

012512 012737 000013 001226  
012520 012737 013006 001216  
012526 012737 012554 001220  
012534 005005  
012536 005037 001236  
012542 005037 001240  
012546 012737 011000 001236  
012554  
012554 052777 000400 166630  
012562 004737 004772  
012566 013777 001236 166614  
012574 052777 000020 166602  
012602 052777 004020 166602  
012610 105777 166576  
012614 100375  
012616 012777 000400 166570  
012624 105777 166562  
012630 100375  
012632 013777 001240 166554  
012640 105777 166546  
012644 100375  
012646 012777 000252 166540  
012654 032777 004000 166532  
012662 001374  
012664 032777 004000 166522  
012672 001774  
012674 105777 166504  
012700 100375  
012702 005777 166476  
012706 100001  
012710 104014  
012712 017704 166470  
012716 032704 000400  
012722 001001  
012724 104000  
012726 122704 000252  
012732 001401  
012734 104012  
012736 104401  
012740 052777 000400 166444  
012746 004737 004772  
012752 105205  
012754 001413  
012756 110537 001236  
012762 110537 001240  
012766 052737 011000 001236  
012774 013777 001236 166406  
013002 000674  
013004 104400

```

:*****
:
: TEST 13
:
:*****
:*****
TST13: MOV #13, @TSTNO
MOV @TST14, NEXT
MOV #20$, LOCK ;SW09 SETUP
CLR R5 ;CLEAR SEC ADRS HOLD
CLR TEMP1 ;CLEAR TEMP STORAGE
CLR TEMP2 ;DITTO
MOV #CRCEN!PRISEC, TEMP1 ;LOAD MODE AND NO CRC
20$: BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
MOV TEMP1, @PARCSR ;LOAD MODE AND NO CRC AND SEC STATION
1$: BIS #RCVEN, @RXCSR ;TURN ON RECEIVER
BIS #SYSTST!SEND, @TXCSR ;TURN ON TRANSMITTER AND CLOCK
2$: TSTB @TXCSR ;WAIT FOR
BPL 2$ ;DONE
MOV @TSOM, @TXDBUF ;LOAD START OF MSG
3$: TSTB @TXCSR ;WAIT FOR
BPL 3$ ;DONE AGAIN
MOV TEMP2, @TXDBUF ;LOAD SEC STATION ADRS AND
4$: TSTB @TXCSR ;WAIT FOR
BPL 4$ ;DONE AGAIN
MOV #252, @TXDBUF ;NOW LOAD DATA
64$: BIT @TIMER, @TXDBUF ;CHECK THE TIMER BIT
BNE 64$ ;BR IF SET
65$: BIT @TIMER, @TXDBUF ;CHECK THE TIMER BIT
BEQ 65$ ;BR IF CLEAR
5$: TSTB @RXCSR ;TEST FOR
BPL 5$ ;RX DONE
TST @RXCSR ;TEST FOR ERROR
BPL 6$ ;BR IF NO ERROR
HLT 14 ;ERROR FOUND!
6$: MOV @RXDBUF, R4 ;GET THE BUFFER
BIT #RSOM, R4 ;CHECK FOR START OF MSG
BNE 7$ ;BR IF SET
HLT ;START OF MSG FAILED TO SET
7$: CMPB #252, R4 ;CHECK FOR DATA
BEQ 10$ ;BR IF A MATCH
HLT ;FAILED TO RECEIVE DATA AS A SEC STATION
10$: SCOP1 ;SW09=1?
BIS #MRESET, @TXCSR ;RESET THE DEVICE
JSR PC, SMALL ;WAIT FOR RESET TO FINISH
INCB R5 ;GET NEXT SEC ADRS
BEQ 11$ ;BR IF ALL DONE
MOV R5, TEMP1 ;LOAD THE ADRS FOR PAR CSR
MOV R5, TEMP2 ;DITTO FOR TXD BUF
BIS #CRCEN!PRISEC, TEMP1 ;LOAD FOR PAR CSR
MOV TEMP1, @PARCSR ;DO IT
BR 1$ ;BR TO DO OVER
11$: SCOPE ;SCOPE THIS TEST

```

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

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

\*\*\*\*\*  
\*  
TEST 14  
\*  
\*\*\*\*\*

013006 012737 000014 001226  
013014 012737 013524 001216  
013022 012704 013332  
013026 012703 000004  
013032 012701 013332  
013036 012702 013332  
013042 005037 001324  
013046 012737 102010 013516  
013054 012737 177777 013522  
013062 013737 013522 013104  
013070 011437 013102  
013074 004537 013344  
013100 000010  
013102 000001  
013104 000001  
013106 052777 000400 166276  
013114 004737 004772  
013120 052777 014000 166264  
013126 052777 000020 166250  
013134 052777 000020 166250  
013142 052777 000400 166244  
013150 104412 000012  
013154 105777 166232  
013160 100401  
013162 104000  
013164  
013164 011177 166224  
013170 104412 000020  
013174 052777 001000 166212  
013202 104412 000020  
013206 105777 166172  
013212 100401  
013214 104000  
013216 017737 166164 001324  
013224 121237 001324  
013230 001401  
013232 104000  
013234 104412 000020  
013240 105777 166140  
013244 100401  
013246 104000  
013250 017737 166132 001324  
013256 104412 000032  
013262 105777 166116

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



2488	013266	100401			BMI	5\$		:BR IF SET
2489	013270	104000			HLT			:DONE FAILED TO SET ON SECOND OCTET OF BCC
2490	013272	117737	166110	001325	5\$:	MOVB	2RXDBUF,DATA +1	:GET IT
2491	013300	005137	013522		COM		CALBCC	:INVERT SOFTWARE
2492	013304	023737	013522	001324	CMP		CALBCC,DATA	:COMPARE
2493	013312	001401			BEQ	6\$		:BR IF A MATCH
2494	013314	104000			HLT			:BCC COMPARE FAILURE
2495								:USE CRC DEBUGGING ROUTINE TO
2496								:REPAIR CRC.SEE FRONT OF LISTING
2497	013316	005721			6\$:	TST	(R1)+	:POP POINTER
2498	013320	005722				TST	(R2)+	:DITTO
2499	013322	005724				TST	(R4)+	:DITTO
2500	013324	005303				DEC	R3	:DECREMENT THE # TO DO
2501	013326	001247				BNE	10\$	:BR IF MORE TOGO
2502	013330	104400				SCOPE		:SCOPE THIS TEST
2503								
2504								
2505	013332	000252			:DATA TABLE			
2506	013334	000000			TBLB:	.WORD	252	
2507	013336	000125				.WORD	0	
2508	013340	000377				.WORD	125	
2509	013342	000000				.WORD	377	
2510						.WORD	0	
2511								
2512	013344	010046			SIMBCC:	MOV	RO,-(SP)	
2513	013346	010146				MOV	R1,-(SP)	
2514	013350	010246				MOV	R2,-(SP)	
2515	013352	012537	001236			MOV	(R5)+,TEMP1	
2516	013356	012537	001240			MOV	(R5)+,TEMP2	
2517	013362	012537	001242			MOV	(R5)+,TEMP3	
2518	013366	005037	013520		1\$:	CLR	BCCFBK	
2519	013372	013700	001242			MOV	TEMP3,RO	
2520	013376	006037	001240			ROR	TEMP2	
2521	013402	005500				ADC	RO	
2522	013404	032700	000001			BIT	#BIT0,RO	
2523	013410	001402				BEQ	2\$	
2524	013412	005137	013520			COM	BCCFBK	
2525	013416	013700	013516		2\$:	MOV	XPOLY,RO	
2526	013422	005100				COM	RO	
2527	013424	040037	013520			BIC	RO,BCCFBK	
2528	013430	000241				CLC		
2529	013432	006037	001242			ROR	TEMP3	
2530	013436	013700	013520			MOV	BCCFBK,RO	
2531	013442	013701	001242			MOV	TEMP3,R1	
2532	013446	010102				MOV	R1,R2	
2533	013450	040100				BIC	R1,RO	
2534	013452	043702	013520			BIC	BCCFBK,R2	
2535	013456	050200				BIS	R2,RO	
2536	013460	043737	013516	001242		BIC	XPOLY,TEMP3	
2537	013466	050037	001242			BIS	RO,TEMP3	
2538	013472	005337	001236			DEC	TEMP1	
2539	013476	001333				BNE	1\$	
2540	013500	013737	001242	013522		MOV	TEMP3,CALBCC	
2541	013506	012602				MOV	(SP)+,R2	
2542	013510	012601				MOV	(SP)+,R1	
2543	013512	012600				MOV	(SP)+,RO	

2544 013514 000205  
2545 013516 000000  
2546 013520 000000  
2547 013522 000000  
2548 120001  
2549 102010  
2550  
2551  
2552  
2553  
2554  
2555  
2556  
2557  
2558  
2559  
2560  
2561  
2562  
2563

R15 R5  
XPOLY: 0  
BCCFBK: 0  
CALBCC: 0  
CRC16=120001  
CRC.CCITT=102010

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

\*\*\*\*\*  
: TEST 15  
\*\*\*\*\*

2564 013524 012737 000015 001226  
2565 013532 012737 014026 001216  
2566 013540 052777 000400 165644  
2567 013546 004737 004772  
2568 013552 005037 001236  
2569 013556 005037 001240  
2570 013562 005037 001242  
2571 013566 052777 014000 165616  
2572 013574 052777 001000 165606  
2573 013602 052777 000020 165602  
2574 013610 052777 000020 165566  
2575 013616 052777 001400 165570  
2576 013624 104412 000004  
2577 013630 005237 001236 1S:  
2578 013634 022737 000021 001236  
2579 013642 001407  
2580 013644 104412 000002  
2581 013650 032777 040000 165534  
2582 013656 001764  
2583 013660 104000  
2584 013662 104412 000002 2S:  
2585 013666 032777 040000 165516  
2586 013674 001401 3S:  
2587 013676 104000  
2588 013700 012777 000125 165506 4S:  
2589 013706 104412 000002  
2590 013712 032777 040000 165472  
2591 013720 001001  
2592 013722 104000  
2593 013724 005237 001240 5S:  
2594 013730 022737 000006 001240  
2595 013736 001363  
2596 013740 104412 000002  
2597 013744 032777 040000 165440  
2598 013752 001401  
2599 013754 104000

\*\*\*\*\*  
TST15: MOV #15, @TSTNO  
MOV @TST16, NEXT  
BIS #MRESET, @TXCSR ; RESET THE DEVICE  
JSR PC\_SMALL ; WAIT FOR RESET TO FINISH  
CLR TEMP1 ; CLEAR FOR SOFTWARE STORAGE  
CLR TEMP2  
CLR TEMP3  
BIS #MMODE, @TXCSR ; ENTER MAINT MODE  
BIS #CRCEN, @PARCSR ; TURN OFF CRC  
BIS #SEND, @TXCSR ; TURN ON TRANSMITTER  
BIS #RCVEN, @RXCSR ; TURN ON RECEIVER  
BIS #TEOM!TOM, @TXDBUF ; START SPECIAL SEQUENCE  
PKCLK 4 ; SYNC UP DUP  
1S: INC TEMP1 ; UPDATE COUNT  
CMP #17., TEMP1 ; CHECK FOR FINISH  
BEQ 2S ; BR IF YES  
PKCLK 2 ; POKE CLOCK  
BIT @MCDATA, @TXCSR ; CHECK THE DATA  
BEQ 1S ; BR IF OK  
HLT ; DATA SHOULD BE A 0, WAS A 1  
2S: PKCLK 2 ; POKE CLOCK  
BIT @MCDATA, @TXCSR ; CHECK THE DATA  
BEQ 3S ; BR IF OK  
HLT ; DATA SHOULD BE 0, WAS A ONE  
3S: MOV #125, @TXDBUF ; LOAD DATA CHAR  
4S: PKCLK 2 ; POKE CLOCK  
BIT @MCDATA, @TXCSR ; CHECK THE BIT  
BNE 5S ; BR IF OK  
HLT ; DATA SHOULD BE 1, WAS 0  
5S: INC TEMP2 ; UPDATE FOR FINISH  
CMP #6, TEMP2 ; CHECK FOR DONE  
BNE 4S ; BR IF MORE TO GO  
PKCLK 2 ; POKE CLOCK  
BIT @MCDATA, @TXCSR ; CHECK THE DATA  
BEQ 6S ; BR IF OK  
HLT ; DATA SHOULD BE 0, WAS A 1



```

2600 013756 104412 000006 6S: PKCLK 6 ; START OUT DATA CHAR
2601 013762 052777 001000 165422 BIS #TEOM, @TXCSR ; TURN OFF TRANSMITTER
2602 013770 104412 000016 PKCLK 14 ; FINISH
2603 013774 105777 165404 TSTB @RXCSR ; CHECK RECEIVER
2604 014000 100401 BMI 7S ; BR IF OK
2605 014002 104000 HLT ; RECEIVER FAILED TO ACCEPT SPECIAL CHAR
2606 014004 117737 165376 001242 7S: MOVB @RXDBUF, TEMP3 ; GET THE CHAR
2607 014012 122737 000125 001242 CMPB #125, TEMP3 ; CHECK IT
2608 014020 001401 BEQ 10S
2609 014022 104000 HLT ; DATA FAILED TO MATCH AFTER
2610 ; SPECIAL SPACES CHARACTER
2611 014024 104400 10S: SCOPE ; SCOPE THIS TEST

```

```

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

```

```

;*****
; TEST 16
;*****
;*****

```

```

2628 014026 012737 000016 001226 TST16: MOV #16, @TSTNO
2629 014034 012737 014474 001216 MOV #TST17, NEXT
2630 014042 012737 014122 001220 MOV #15, LOCK
2631 014050 105737 001322 TSTB TCNFLG
2632 014054 001002 BNE .+6
2633 014056 000137 014462 10S: JMP 6S
2634 014062 005077 165316 CLR @RXCSR ; CLEAR THE REGISTER
2635 014066 004137 007260 JSR R1, OJUMPER ; THIS CALL DETERMINES IF TURNAROUND CONNECTOR
2636 014072 014464 7S ; AND OPTIONAL JUMPER ARE USED
2637 ; AND LOADS R5 (EXPECTED) ACCORDINGLY.
2638 014074 052777 000400 165310 BIS #MRESET, @TXCSR ; RESET THE DEVICE
2639 014102 004737 004772 JSR PC, SMALL ; WAIT FOR RESET TO FINISH
2640 014106 013703 001404 MOV RXCSR, R3 ; LOAD THE RECEIVER CONTROL REGISTER TO R3.
2641 014112 005013 CLR (R3) ; CLEAR OUT EXTRA BITS
2642 014114 052777 010000 165270 BIS #MEXT, @TXCSR ; ENTER EXTERNAL MAINT. MODE
2643 014122 052713 000002 1S: BIS #DTR, (R3) ; TURN ON DTR
2644 014126 012737 003110 014156 MOV #110, 68S ; LOAD THE NUMBER
2645 014134 032777 004000 165252 66S: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
2646 014142 001374 BNE 66S ; BR IF SET
2647 014144 032777 004000 165242 67S: BIT #TIMER, @TXDBUF ; CHECK THE BIT
2648 014152 001774 BEQ 67S ; BR IF CLEAR
2649 014154 005327 DEC (PC)+ ; DECREMENT THE NUMBER
2650 014156 000110 68S: 110 ; OF TIMES TO REPEAT
2651 014160 001365 BNE 66S ; BR IF MORE TO GO
2652 014162 011304 MOV (R3), R4 ; GET THE BITS FROM THE RXCSR
2653 014164 020504 CMP R5, R4 ; R5=GOOD R4=?
2654 014166 001423 BEQ 2S ; BRANCH IF THEY MATCH
2655 014170 104003 HLT 3 ; NO MATCH - SHOW OPR.

```

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 61  
 DZDPCA.CMB MODEM CONTROL BITS DTR,RING,AND DSR INTERACTION TEST

2656	014172	104401				SCOP1		
2657	014174	012737	000005	014224		MOV	#5,73\$	;LOAD THE NUMBER
2658	014202	032777	004000	165204	71\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
2659	014210	001374				BNE	71\$	;BR IF SET
2660	014212	032777	004000	165174	72\$:	BIT	#TIMER,@TXDBUF	;CHECK THE BIT
2661	014220	001774				BEQ	72\$	;BR IF CLEAR
2662	014222	005327				DEC	(PC)+	;DECREMENT THE NUMBER
2663	014224	000005			73\$:	5		;OF TIMES TO REPEAT
2664	014226	001365				BNE	71\$	;BR IF MORE TO GO
2665	014230	032713	040000			BIT	#RING,(R3)	
2666	014234	001000				BNE	2\$	
2667	014236	012737	014250	001220	2\$:	MOV	#3\$,LOCK	;SW09 SETUP
2668	014244	042705	041002			BIC	#RING!DSR!DTR,R5	;CLEAR OUT UNWANTED BITS
2669	014250	005013			3\$:	CLR	(R3)	;CLEAR OUT THE REGISTER
2670	014252	012737	000005	014302		MOV	#5,78\$	;LOAD THE NUMBER
2671	014260	032777	004000	165126	76\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
2672	014266	001374				BNE	76\$	;BR IF SET
2673	014270	032777	004000	165116	77\$:	BIT	#TIMER,@TXDBUF	;CHECK THE BIT
2674	014276	001774				BEQ	77\$	;BR IF CLEAR
2675	014300	005327				DEC	(PC)+	;DECREMENT THE NUMBER
2676	014302	000005			78\$:	5		;OF TIMES TO REPEAT
2677	014304	001365				BNE	76\$	;BR IF MORE TO GO
2678	014306	011304				MOV	(R3),R4	;READ BACK THE REGISTER
2679	014310	020504				CMP	R5,R4	;R5=GOOD R4=?
2680	014312	001402				BEQ	4\$	;BRANCH IF ONLY THE DSC BITS ARE SET
2681	014314	104003				HLT	3	;NO-GO TELL OPR
2682	014316	104401				SCOP1		
2683	014320	012737	014326	001220	4\$:	MOV	#5\$,LOCK	;SW09 SETUP
2684	014326	052713	000002		5\$:	BIS	#DTR,(R3)	;TURN ON DTR
2685	014332	012737	000005	014362		MOV	#5,83\$	;LOAD THE NUMBER
2686	014340	032777	004000	165046	81\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
2687	014346	001374				BNE	81\$	;BR IF SET
2688	014350	032777	004000	165036	82\$:	BIT	#TIMER,@TXDBUF	;CHECK THE BIT
2689	014356	001774				BEQ	82\$	;BR IF CLEAR
2690	014360	005327				DEC	(PC)+	;DECREMENT THE NUMBER
2691	014362	000005			83\$:	5		;OF TIMES TO REPEAT
2692	014364	001365				BNE	81\$	;BR IF MORE TO GO
2693	014366	005005				CLR	R5	;CLEAR OUT EXPECTED
2694	014370	005013				CLR	(R3)	;CLEAR OUT THE REGISTER
2695	014372	052777	000400	165012		BIS	#MRESET,@TXCSR	;RESET THE DEVICE
2696	014400	004737	004772			JSR	PC,SMALL	;WAIT FOR RESET TO FINISH
2697	014404	052777	010000	165000		BIS	#MEXT,@TXCSR	;TURN ON EXTERNAL MODE
2698	014412	012737	000005	014442		MOV	#5,88\$	;LOAD THE NUMBER
2699	014420	032777	004000	164766	86\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
2700	014426	001374				BNE	86\$	;BR IF SET
2701	014430	032777	004000	164756	87\$:	BIT	#TIMER,@TXDBUF	;CHECK THE BIT
2702	014436	001774				BEQ	87\$	;BR IF CLEAR
2703	014440	005327				DEC	(PC)+	;DECREMENT THE NUMBER
2704	014442	000005			88\$:	5		;OF TIMES TO REPEAT
2705	014444	001365				BNE	86\$	;BR IF MORE TO GO
2706	014446	005713				TST	(R3)	;STRIP DSCA & DSCB FROM CSR
2707	014450	011304				MOV	(R3),R4	;GET THE REGISTER
2708	014452	020504				CMP	R5,R4	;R5=GOOD,R4=?
2709	014454	001402				BEQ	6\$	;BR IF OK
2710	014456	104003				HLT	3	;REPORT THE ERROR
2711	014460	104401				SCOP1		;SW09=1?



# J05

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 62  
 DZDPCA.CMB MODEM CONTROL BITS DTR,RING,AND DSR INTERACTION TEST

```

2712
2713 014462 104400
2714 014464 141003
2715 014466 141001
2716 014470 001002
2717 014472 000000
2718
2719
2720
2721
2722
2723
2724
2725
2726
2727
2728
2729
2730
2731
2732
2733
2734 014474 012737 000017 001226
2735 014502 012737 015142 001216
2736 014510 012737 014570 001220
2737 014516 105737 001322
2738 014522 001002
2739 014524 000137 015130
2740 014530 005077 164650
2741 014534 004137 007260
2742 014540 015132
2743
2744 014542 052777 000400 164642
2745 014550 004737 004772
2746 014554 013703 001404
2747 014560 005013
2748 014562 052777 010000 164622
2749 014570 052713 000004
2750 014574 012737 000110 014624
2751 014602 032777 004000 164604
2752 014610 001374
2753 014612 032777 004000 164574
2754 014620 001774
2755 014622 005327
2756 014624 000110
2757 014626 001365
2758 014630 011304
2759 014632 020504
2760 014634 001423
2761 014636 104003
2762 014640 104401
2763 014642 012737 000005 014672
2764 014650 032777 004000 164536
2765 014656 001374
2766 014660 032777 004000 164526
2767 014666 001774
  
```

```

6S: SCOPE ;SCOPE THE WHOLE TEST
7S: .WORD 141003
      .WORD 141001
      .WORD 1002
      .WORD 0
  
```

```

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

```

*****
:TEST 17
*****
  
```

```

*****
TST17: MOV #17,@TSTNO
        MOV #TST20,NEXT
        MOV #15,LOCK
        TSTB TCNFLG
        BNE .+6
10S:   JMP 6S
        CLR @RXCSR ;CLEAR THE REGISTER
        JSR R1,0JUMPER ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
                          ;AND OPTIONAL JUMPER ARE USED
                          ;AND LOADS R5 (EXPECTED) ACCORDINGLY.
        BIS #MRESET,@TXCSR ;RESET THE DEVICE
        JSR PC_SMALL ;WAIT FOR RESET TO FINISH
        MOV RXCSR,R3 ;LOAD THE RECEIVER CONTROL REGISTER TO R3.
        CLR (R3) ;CLEAR OUT EXTRA BITS
        BIS #NEXT,@TXCSR ;ENTER EXTERNAL MAINT. MODE
        BIS #RTS,(R3) ;TURN ON RTS
        MOV #110,68S ;LOAD THE NUMBER
66S:   BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
        BNE 66S ;BR IF SET
67S:   BIT #TIMER,@TXDBUF ;CHECK THE BIT
        BEQ 67S ;BR IF CLEAR
        DEC (PC)+ ;DECREMENT THE NUMBER
68S:   110 ;OF TIMES TO REPEAT
        BNE 66S ;BR IF MORE TO GO
        MOV (R3),R4 ;GET THE BITS FROM THE RXCSR
        CMP R5,R4 ;R5=GOOD R4=?
        BEQ 2S ;BRANCH IF THEY MATCH
        HLT 3 ;NO MATCH - SHOW OPR.
        SCOP1
71S:   MOV #5,73S ;LOAD THE NUMBER
        BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
        BNE 71S ;BR IF SET
72S:   BIT #TIMER,@TXDBUF ;CHECK THE BIT
        BEQ 72S ;BR IF CLEAR
  
```

# K05

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 63  
 DZDPCA.CMB MODEM CONTROL BITS RTS,CTS,AND CARDET INTERACTION TEST

2768	014670	005327				DEC	(PC)+	; DECREMENT THE NUMBER
2769	014672	000005			73\$:	5		; OF TIMES TO REPEAT
2770	014674	001365				BNE	71\$	; BR IF MORE TO GO
2771	014676	032713	040000			BIT	#RING, (R3)	
2772	014702	001000				BNE	2\$	
2773	014704	012737	014716	001220	2\$:	MOV	#3\$, LOCK	; SW09 SETUP
2774	014712	042705	030004			BIC	#CTS!CARDET!RTS, R5	; CLEAR OUT UNWANTED BITS
2775	014716	005013			3\$:	CLR	(R3)	; CLEAR OUT THE REGISTER
2776	014720	012737	000005	014750		MOV	#5, 78\$	; LOAD THE NUMBER
2777	014726	032777	004000	164460	76\$:	BIT	#TIMER, @TXDBUF	; CHECK THE TIMER BIT
2778	014734	001374				BNE	76\$	; BR IF SET
2779	014736	032777	004000	164450	77\$:	BIT	#TIMER, @TXDBUF	; CHECK THE BIT
2780	014744	001774				BEQ	77\$	; BR IF CLEAR
2781	014746	005327				DEC	(PC)+	; DECREMENT THE NUMBER
2782	014750	000005			78\$:	5		; OF TIMES TO REPEAT
2783	014752	001365				BNE	76\$	; BR IF MORE TO GO
2784	014754	011304				MOV	(R3), R4	; READ BACK THE REGISTER
2785	014756	020504				CMP	R5, R4	; R5=GOOD R4=?
2786	014760	001402				BEQ	4\$	; BRANCH IF ONLY THE DSC BITS ARE SET
2787	014762	104003				HLT	3	; NO-GO TELL OPR
2788	014764	104401				SCOPE1		
2789	014766	012737	014774	001220	4\$:	MOV	#5\$, LOCK	; SW09 SETUP
2790	014774	052713	000004		5\$:	BIS	#RTS, (R3)	; TURN ON RTS
2791	015000	012737	000005	015030		MOV	#5, 83\$	; LOAD THE NUMBER
2792	015006	032777	004000	164400	81\$:	BIT	#TIMER, @TXDBUF	; CHECK THE TIMER BIT
2793	015014	001374				BNE	81\$	; BR IF SET
2794	015016	032777	004000	164370	82\$:	BIT	#TIMER, @TXDBUF	; CHECK THE BIT
2795	015024	001774				BEQ	82\$	; BR IF CLEAR
2796	015026	005327				DEC	(PC)+	; DECREMENT THE NUMBER
2797	015030	000005			83\$:	5		; OF TIMES TO REPEAT
2798	015032	001365				BNE	81\$	; BR IF MORE TO GO
2799	015034	005005				CLR	R5	; CLEAR OUT EXPECTED
2800	015036	005013				CLR	(R3)	; CLEAR OUT THE REGISTER
2801	015040	052777	000400	164344		BIS	#MRESET, @TXCSR	; RESET THE DEVICE
2802	015046	004737	004772			JSR	PC, SMALL	; WAIT FOR RESET TO FINISH
2803	015052	052777	010000	164332		BIS	#MEXT, @TXCSR	; TURN ON EXTERNAL MODE
2804	015060	012737	000005	015110		MOV	#5, 88\$	; LOAD THE NUMBER
2805	015066	032777	004000	164320	86\$:	BIT	#TIMER, @TXDBUF	; CHECK THE TIMER BIT
2806	015074	001374				BNE	86\$	; BR IF SET
2807	015076	032777	004000	164310	87\$:	BIT	#TIMER, @TXDBUF	; CHECK THE BIT
2808	015104	001774				BEQ	87\$	; BR IF CLEAR
2809	015106	005327				DEC	(PC)+	; DECREMENT THE NUMBER
2810	015110	000005			88\$:	5		; OF TIMES TO REPEAT
2811	015112	001365				BNE	86\$	; BR IF MORE TO GO
2812	015114	005713				TST	(R3)	; STRIP DSCA & DSCB FROM CSR
2813	015116	011304				MOV	(R3), R4	; GET THE REGISTER
2814	015120	020504				CMP	R5, R4	; R5=GOOD, R4=?
2815	015122	001402				BEQ	6\$	; BR IF OK
2816	015124	104003				HLT	3	; REPORT THE ERROR
2817	015126	104401				SCOPE1		; SW09=1?
2818								
2819	015130	104400			6\$:	SCOPE		; SCOPE THE WHOLE TEST
2820	015132	130005			7\$:	.WORD	130005	
2821	015134	130001				.WORD	130001	
2822	015136	000004				.WORD	4	
2823	015140	000000				.WORD	0	



2824  
2825  
2826  
2827  
2828  
2829  
2830  
2831  
2832  
2833  
2834  
2835  
2836  
2837  
2838  
2839  
2840  
2841  
2842  
2843  
2844  
2845  
2846  
2847  
2848  
2849  
2850  
2851  
2852  
2853  
2854  
2855  
2856  
2857  
2858  
2859  
2860  
2861  
2862  
2863  
2864  
2865  
2866  
2867  
2868  
2869  
2870  
2871  
2872  
2873  
2874  
2875  
2876  
2877  
2878  
2879

015142 012737 000020 001226  
015150 012737 015624 001216  
015156 012737 015252 001220  
015164 105737 001322  
015170 001002  
015172 000137 015612  
015176 105737 001334  
015202 001773  
015204 105737 001336  
015210 001770  
015212 005077 164166  
015216 004137 007260  
015222 015614  
015224 052777 000400 164160  
015232 004737 004772  
015236 013703 001404  
015242 005013  
015244 052777 010000 164140  
015252 052713 000010  
015256 012737 000110 015306  
015264 032777 004000 164122  
015272 001374  
015274 032777 004000 164112  
015302 001774  
015304 005327  
015306 000110  
015310 001365  
015312 011304  
015314 020504  
015316 001423  
015320 104003  
015322 104401  
015324 012737 000005 015354  
015332 032777 004000 164054  
015340 001374  
015342 032777 004000 164044  
015350 001774  
015352 005327  
015354 000005

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

*****
*
* TEST 20
*
*****
*****
TST20: MOV #20, @TSTNO
MOV #TST21, NEXT
MOV #15, LOCK
TSTB TCNFLG
BNE .+6
10$: JMP 6$
TSTB STJNFL
BEQ 10$
TSTB SRJNFL
BEQ 10$
CLR @RXCSR ;CLEAR THE REGISTER
JSR R1, OJUMPER ;THIS CALL DETERMINES IF TURNAROUND CONNECTOR
;AND OPTIONAL JUMPER ARE USED
;AND LOADS R5 (EXPECTED) ACCORDINGLY.
;RESET THE DEVICE
;WAIT FOR RESET TO FINISH
;LOAD THE RECEIVER CONTROL REGISTER TO R3.
BIS #MRESET, @TXCSR
JSR PC, SMALL
MOV RXCSR, R3
CLR (R3) ;CLEAR OUT EXTRA BITS
BIS #MEXT, @TXCSR ;ENTER EXTERNAL MAINT. MODE
1$: BIS #STD, (R3) ;TURN ON STD
MOV #110, 68$ ;LOAD THE NUMBER
66$: BIT #TIMER, @TXDBUF ;CHECK THE TIMER BIT
BNE 66$ ;BR IF SET
67$: BIT #TIMER, @TXDBUF ;CHECK THE BIT
BEQ 67$ ;BR IF CLEAR
DEC (PC)+ ;DECREMENT THE NUMBER
68$: 110 ;OF TIMES TO REPEAT
BNE 66$ ;BR IF MORE TO GO
MOV (R3), R4 ;GET THE BITS FROM THE RXCSR
CMP R5, R4 ;R5=GOOD R4=?
BEQ 2$ ;BRANCH IF THEY MATCH
HLT 3 ;NO MATCH - SHOW OPR.
MOV #5, 73$ ;LOAD THE NUMBER
71$: BIT #TIMER, @TXDBUF ;CHECK THE TIMER BIT
BNE 71$ ;BR IF SET
72$: BIT #TIMER, @TXDBUF ;CHECK THE BIT
BEQ 72$ ;BR IF CLEAR
DEC (PC)+ ;DECREMENT THE NUMBER
73$: 5 ;OF TIMES TO REPEAT
```

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 65  
 DZDPCA.CMB MODEM CONTROL BITS STD AND SRD INTERACTION TEST

```

2880 015356 001365      BNE      71$      ;BR IF MORE TO GO
2881 015360 032713 040000      BIT      #RING, (R3)
2882 015364 001000      BNE      2$
2883 015366 012737 015400 001220 2$:      MOV      #3$, LOCK      ;SW09 SETUP
2884 015374 042705 002010      BIC      #STD!SRD, R5      ;CLEAR OUT UNWANTED BITS
2885 015400 005013      CLR      (R3)      ;CLEAR OUT THE REGISTER
2886 015402 012737 000005 015432      MOV      #5, 78$      ;LOAD THE NUMBER
2887 015410 032777 004000 163776 76$:      BIT      #TIMER, @TXDBUF      ;CHECK THE TIMER BIT
2888 015416 001374      BNE      76$      ;BR IF SET
2889 015420 032777 004000 163766 77$:      BIT      #TIMER, @TXDBUF      ;CHECK THE BIT
2890 015426 001774      BEQ      77$      ;BR IF CLEAR
2891 015430 005327      DEC      (PC)+      ;DECREMENT THE NUMBER
2892 015432 000005      5      ;OF TIMES TO REPEAT
2893 015434 001365      BNE      76$      ;BR IF MORE TO GO
2894 015436 011304      MOV      (R3), R4      ;READ BACK THE REGISTER
2895 015440 020504      CMP      R5, R4      ;R5=GOOD R4=?
2896 015442 001402      BEQ      4$      ;BRANCH IF ONLY THE DSC BITS ARE SET
2897 015444 104003      HLT      3      ;NO-GO TELL OPR
2898 015446 104401      SCOPE1
2899 015450 012737 015456 001220 4$:      MOV      #5$, LOCK      ;SW09 SETUP
2900 015456 052713 000010 5$:      BIS      #STD, (R3)      ;TURN ON STD
2901 015462 012737 000005 015512      MOV      #5, 83$      ;LOAD THE NUMBER
2902 015470 032777 004000 163716 81$:      BIT      #TIMER, @TXDBUF      ;CHECK THE TIMER BIT
2903 015476 001374      BNE      81$      ;BR IF SET
2904 015500 032777 004000 163706 82$:      BIT      #TIMER, @TXDBUF      ;CHECK THE BIT
2905 015506 001774      BEQ      82$      ;BR IF CLEAR
2906 015510 005327      DEC      (PC)+      ;DECREMENT THE NUMBER
2907 015512 000005      5      ;OF TIMES TO REPEAT
2908 015514 001365      BNE      81$      ;BR IF MORE TO GO
2909 015516 005005      CLR      R5      ;CLEAR OUT EXPECTED
2910 015520 005013      CLR      (R3)      ;CLEAR OUT THE REGISTER
2911 015522 052777 000400 163662      BIS      #MRESET, @TXCSR      ;RESET THE DEVICE
2912 015530 004737 004772      JSR      PC, SMALL      ;WAIT FOR RESET TO FINISH
2913 015534 052777 010000 163650      BIS      #MEXT, @TXCSR      ;TURN ON EXTERNAL MODE
2914 015542 012737 000005 015572      MOV      #5, 88$      ;LOAD THE NUMBER
2915 015550 032777 004000 163636 86$:      BIT      #TIMER, @TXDBUF      ;CHECK THE TIMER BIT
2916 015556 001374      BNE      86$      ;BR IF SET
2917 015560 032777 004000 163626 87$:      BIT      #TIMER, @TXDBUF      ;CHECK THE BIT
2918 015566 001774      BEQ      87$      ;BR IF CLEAR
2919 015570 005327      DEC      (PC)+      ;DECREMENT THE NUMBER
2920 015572 000005      5      ;OF TIMES TO REPEAT
2921 015574 001365      BNE      86$      ;BR IF MORE TO GO
2922 015576 005713      TST      (R3)      ;STRIP DSCA & DSCB FROM CSR
2923 015600 011304      MOV      (R3), R4      ;GET THE REGISTER
2924 015602 020504      CMP      R5, R4      ;R5=GOOD, R4=?
2925 015604 001402      BEQ      6$      ;BR IF OK
2926 015606 104003      HLT      3      ;REPORT THE ERROR
2927 015610 104401      SCOPE1      ;SW09=1?
2928
2929 015612 104400      6$:      SCOPE      ;SCOPE THE WHOLE TEST
2930 015614 002011      7$:      .WORD    2011
2931 015616 002001      .WORD    2001
2932 015620 000010      .WORD    10
2933 015622 000000      .WORD    0
2934
2935

```



2936  
2937  
2938  
2939  
2940  
2941  
2942  
2943  
2944  
2945  
2946  
2947  
2948  
2949  
2950  
2951  
2952  
2953  
2954  
2955  
2956  
2957  
2958  
2959  
2960  
2961  
2962  
2963  
2964  
2965  
2966  
2967  
2968  
2969  
2970  
2971  
2972  
2973  
2974  
2975  
2976  
2977  
2978  
2979  
2980  
2981  
2982  
2983  
2984  
2985  
2986  
2987  
2988  
2989  
2990  
2991

015624 012737 000021 001226  
 015632 012737 016306 001216  
 015640 012737 015734 001220  
 015646 105737 001322  
 015652 001002  
 015654 000137 016274  
 015660 105737 001334  
 015664 001773  
 015666 105737 001336  
 015672 001770  
 015674 005077 163504  
 015700 004137 007260  
 015704 016276  
 015706 052777 000400 163476  
 015714 004737 004772  
 015720 013703 001404  
 015724 005013  
 015726 052777 010000 163456  
 015734 052713 000016  
 015740 012737 000110 015770  
 015746 032777 004000 163440  
 015754 001374  
 015756 032777 004000 163430  
 015764 001774  
 015766 005327  
 015770 000110  
 015772 001365  
 015774 011304  
 015776 020504  
 016000 001423  
 016002 104003  
 016004 104401  
 016006 012737 000005 016036  
 016014 032777 004000 163372  
 016022 001374  
 016024 032777 004000 163362  
 016032 001774  
 016034 005327  
 016036 000005  
 016040 001365  
 016042 032713 040000

```

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

*****
TEST 21
*****
*****
TST21: MOV #21, @TSTNO
      MOV #TST22, NEXT
      MOV #15, LOCK
      TSTB TCNFLG
      BNE .+6
10$:  JMP 6$
      TSTB STJMFL
      BEQ 10$
      TSTB SRJMFL
      BEQ 10$
      CLR @RXCSR ; CLEAR THE REGISTER
      JSR R1, OJUMPER ; THIS CALL DETERMINES IF TURNAROUND CONNECTOR
                       ; AND OPTIONAL JUMPER ARE USED
                       ; AND LOADS R5 (EXPECTED) ACCORDINGLY.
      7$ ; RESET THE DEVICE
                       ; WAIT FOR RESET TO FINISH
                       ; LOAD THE RECEIVER CONTROL REGISTER TO R3.
      BIS #MRESET, @TXCSR
      JSR PC_SMALL
      MOV RXCSR, R3
      CLR (R3) ; CLEAR OUT EXTRA BITS
      BIS #MEXT, @TXCSR ; ENTER EXTERNAL MAINT. MODE
1$:  BIS #DTR!RTS!STD, (R3) ; TURN ON DTR!RTS!STD
      MOV #110, 68$ ; LOAD THE NUMBER
      BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
      BNE 66$ ; BR IF SET
      BIT #TIMER, @TXDBUF ; CHECK THE BIT
      BEQ 67$ ; BR IF CLEAR
      DEC (PC)+ ; DECREMENT THE NUMBER
      68$: 110 ; OF TIMES TO REPEAT
      BNE 66$ ; BR IF MORE TO GO
      MOV (R3), R4 ; GET THE BITS FROM THE RXCSR
      CMP R5, R4 ; R5=GOOD R4=?
      BEQ 25$ ; BRANCH IF THEY MATCH
      HLT 3 ; NO MATCH - SHOW OPR.
      SCOP1
      MOV #5, 73$ ; LOAD THE NUMBER
      BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
      BNE 71$ ; BR IF SET
      BIT #TIMER, @TXDBUF ; CHECK THE BIT
      BEQ 72$ ; BR IF CLEAR
      DEC (PC)+ ; DECREMENT THE NUMBER
      73$: 5 ; OF TIMES TO REPEAT
      BNE 71$ ; BR IF MORE TO GO
      BIT #RING, (R3)
  
```

```

2992 016046 001000      BNE      2$
2993 016050 012737 016062 001220 2$: MOV      #3$,LOCK      ;SW09 SETUP
2994 016056 042705 073016      BIC      #RING!CTS!CARDET!SRD!DSR!STD!RTS!DTR,R5      ;CLEAR OUT UNWANTED BITS
2995 016062 005013      CLR      (R3)      ;CLEAR OUT THE REGISTER
2996 016064 012737 000005 016114 3$: MOV      #5,78$      ;LOAD THE NUMBER
2997 016072 032777 004000 163314 76$: BIT      #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
2998 016100 001374      BNE      76$      ;BR IF SET
2999 016102 032777 004000 163304 77$: BIT      #TIMER,@TXDBUF      ;CHECK THE BIT
3000 016110 001774      BEQ      77$      ;BR IF CLEAR
3001 016112 005327      DEC      (PC)+      ;DECREMENT THE NUMBER
3002 016114 000005      5      ;OF TIMES TO REPEAT
3003 016116 001365      BNE      76$      ;BR IF MORE TO GO
3004 016120 011304      MOV      (R3),R4      ;READ BACK THE REGISTER
3005 016122 020504      CMP      R5,R4      ;R5=GOOD,R4=?
3006 016124 001402      BEQ      4$      ;BRANCH IF ONLY THE DSC BITS ARE SET
3007 016126 104003      HLT      3      ;NO-GO TELL OPR
3008 016130 104401      SCOPE1
3009 016132 012737 016140 001220 4$: MOV      #5$,LOCK      ;SW09 SETUP
3010 016140 052713 000016 55$: BIS      #DTR!RTS!STD,(R3)      ;TURN ON DTR!RTS!STD
3011 016144 012737 000005 016174 5$: MOV      #5,83$      ;LOAD THE NUMBER
3012 016152 032777 004000 163234 81$: BIT      #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
3013 016160 001374      BNE      81$      ;BR IF SET
3014 016162 032777 004000 163224 82$: BIT      #TIMER,@TXDBUF      ;CHECK THE BIT
3015 016170 001774      BEQ      82$      ;BR IF CLEAR
3016 016172 005327      DEC      (PC)+      ;DECREMENT THE NUMBER
3017 016174 000005      5      ;OF TIMES TO REPEAT
3018 016176 001365      BNE      81$      ;BR IF MORE TO GO
3019 016200 005005      CLR      R5      ;CLEAR OUT EXPECTED
3020 016202 005013      CLR      (R3)      ;CLEAR OUT THE REGISTER
3021 016204 052777 000400 163200 81$: BIS      #MRESET,@TXCSR      ;RESET THE DEVICE
3022 016212 004737 004772      JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
3023 016216 052777 010000 163166 81$: BIS      #NEXT,@TXCSR      ;TURN ON EXTERNAL MODE
3024 016224 012737 000005 016254 86$: MOV      #5,88$      ;LOAD THE NUMBER
3025 016232 032777 004000 163154 86$: BIT      #TIMER,@TXDBUF      ;CHECK THE TIMER BIT
3026 016240 001374      BNE      86$      ;BR IF SET
3027 016242 032777 004000 163144 87$: BIT      #TIMER,@TXDBUF      ;CHECK THE BIT
3028 016250 001774      BEQ      87$      ;BR IF CLEAR
3029 016252 005327      DEC      (PC)+      ;DECREMENT THE NUMBER
3030 016254 000005      5      ;OF TIMES TO REPEAT
3031 016256 001365      BNE      86$      ;BR IF MORE TO GO
3032 016260 005713      TST      (R3)      ;STRIP DSCA & DSCB FROM CSR
3033 016262 011304      MOV      (R3),R4      ;GET THE REGISTER
3034 016264 020504      CMP      R5,R4      ;R5=GOOD,R4=?
3035 016266 001402      BEQ      6$      ;BR IF OK
3036 016270 104003      HLT      3      ;REPORT THE ERROR
3037 016272 104401      SCOPE1      ;SW09=1?
3038
3039 016274 104400      6$: SCOPE      ;SCOPE THE WHOLE TEST
3040 016276 173017      7$: .WORD    173017
3041 016300 173001      .WORD    173001
3042 016302 001016      .WORD    1016
3043 016304 000000      .WORD    0
3044
3045
3046
3047

```

```

;***** TEST 22 *****
;*TEST THAT SETTING TRANSMIT INTERRUPT
;*ENABLE AND TRANSMITTER DONE PRODUCE

```



3048  
3049  
3050  
3051  
3052  
3053  
3054  
3055  
3056  
3057  
3058  
3059  
3060  
3061  
3062  
3063  
3064  
3065  
3066  
3067  
3068  
3069  
3070  
3071  
3072  
3073  
3074  
3075  
3076  
3077  
3078  
3079  
3080  
3081  
3082  
3083  
3084  
3085  
3086  
3087  
3088  
3089  
3090  
3091  
3092  
3093  
3094  
3095  
3096  
3097  
3098  
3099  
3100  
3101  
3102  
3103

016306 012737 000022 001226  
016314 012737 016424 001216  
016322 012706 001150  
016326 000005  
016330 012737 000340 177776  
016336 052777 000400 163046  
016344 004737 004772  
016350 004537 007160  
016354 007202  
016356 016412  
016360 340 340  
016362 052777 004000 163022  
016370 005037 177776  
016374 052777 000100 163010  
016402 000240  
016404 000240  
016406 000240  
016410 104027  
016412 012706 001150  
016416 005077 162770  
016422 104400

```

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

```

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

```

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

```

```

3104 016514 000240      NOP      ;DITTO
3105 016516 000240      NOP      ;DITTO
3106 016520 104027      HLT      27      ;DUP FAILED TO INTERRUPT
3107 016522 000411      BR       55      ;LEAVE TEST
3108 016524 005037 177776 25:  CLR      PS      ;LOWER PROCESSOR STATUS
3109 016530 000240      NOP      ;STALL
3110 016532 000240      NOP      ;DITTO
3111 016534 000240      NOP      ;DITTO
3112 016536 105777 162650 45:  TSTB    @TXCSR  ;CHECK THE DONE BIT
3113 016542 100401      BMI     55      ;BR IF SET
3114 016544 104024      HLT     24      ;DONE IS CLEARED AND SHOULDN'T BE
3115 016546 012737 000340 177776 55:  MOV     @340,PS ;RAISE PROCESSOR STATUS
3116 016554 005077 162632      CLR    @TXCSR  ;CLEAR OUT DUP
3117 016560 012706 001150      MOV    @STACK,SP ;RESET STACK
3118 016564 104400      SCOPE   ;SCOPE THIS TEST
3119
3120 016566 012716 016524 15:  MOV     @25,(SP) ;SET UP SECOND PART OF TEST
3121 016572 004537 007160      JSR    R5,SETVEC ;SETUP FOR SECOND INTERRUPT TRY
3122 016576 007202      NO.ATRAP ;RECEIVER
3123 016600 016606      35     ;TRANSMITTER
3124 016602      340    340     ;LEVEL
3125 016604 000002      RTI    ;RETURN
3126 016606
3127 016606 104026 35:  HLT     26      ;REPORT THE FACT YOU GOT HERE
3128 016610 012716 016536      MOV    @45,(SP) ;SETUP FOR END OF TEST
3129 016614 000002      RTI    ;RETURN
3130
3131      ;***** TEST 24 *****
3132      ;*TEST THAT SETTING DATA SET INTERRUPT
3133      ;*ENABLE AND RECEIVING A DATA SET
3134      ;*CHANGE 1 OR DATA SET CHANGE 2
3135      ;*PRODUCES AN INTERRUPT TO THE
3136      ;*RECEIVER VECTOR
3137      ;*****
3138
3139      ;*****
3140      ;*
3141      ;* TEST 24
3142      ;*
3143      ;*****
3144      ;*****
3145 016616 012737 000024 001226 TST24: MOV     @24,@TSTNO
3146 016624 012737 017134 001216      MOV    @TST25,NEXT
3147 016632 105737 001322      TSTB  TCNFLAG
3148 016636 001002      BNE    +6
3149 016640 000137 017122      JMP    55
3150 016644 012737 016712 001220      MOV    @15,LOCK ;SW09 SETUP
3151 016652 012737 000340 177776      MOV    @340,PS  ;LOCK OUT INTERRUPTS
3152 016660 052777 000400 162524      BIS    @MRESET,@TXCSR ;RESET THE DEVICE
3153 016666 004737 004772      JSR    PC,SMALL ;WAIT FOR RESET TO FINISH
3154 016672 052777 010000 162512      BIS    @NEXT,@TXCSR ;ENTER MAINT EXTERNAL MODE
3155 016700 004537 007160      JSR    R5,SETVEC ;SET UP VECTORS
3156 016704 016772      25     ;RECEIVER
3157 016706 007206      NO.BTRAP ;TRANSMITTER
3158 016710      340    340     ;PRIORITY AT 7
3159 016712 005037 177776 15:  CLR     PS      ;LOWER PS

```



DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 70  
 DZDPCA.CMB RECEIVER DATA SET CHANGE BITS INTERRUPT TEST

```

3160 016716 052777 000040 162450      BIS      #DSINTE,DRXCSR ;TURN ON INT. ENABLE
3161 016724 052777 000004 162452      BIS      #RTS,DRXCSR ;TURN ON INT. BIT
3162 016732 012737 000005 016762      MOV      #5,68$ ;LOAD THE NUMBER
3163 016740 032777 004000 162446 66$:      BIT      #TIMER,DTXDBUF ;CHECK THE TIMER BIT
3164 016746 001374      BNE      66$ ;BR IF SET
3165 016750 032777 004000 162436 67$:      BIT      #TIMER,DTXDBUF ;CHECK THE BIT
3166 016756 001774      BEQ      67$ ;BR IF CLEAR
3167 016760 005327      DEC      (PC)+ ;DECREMENT THE NUMBER
3168 016762 000005 68$:      5 ;OF TIMES TO REPEAT
3169 016764 001365      BNE      66$ ;BR IF MORE TO GO
3170 016766 104022      HLT      22 ;FAILED TO INTERRUPT
3171 016770 104401      SCOPE1 ;SW09=1
3172 016772 012706 001150 2$:      MOV      #STACK,SP ;RESET THE STACK
3173 016776 005077 162402      CLR      DRXCSR ;CLEAR OUT RECEIVER CONTRL REGISTER.
3174 017002 012737 017022 001220      MOV      #3$,LOCK ;SW09 SETUP
3175 017010 004537 007160      JSR      R5,SETVEC ;SET THE VECTORS
3176 017014 017110      4$ ;RECEIVER
3177 017016 007206      NO.BTRAP ;TRANSMITTERS
3178 017020 340 340 .BYTE 340,340 ;PRIOR 27
3179 017022 012737 000005 001236 3$:      MOV      #5,TEMP1 ;LOAD TEMP1
3180 017030 052777 000040 162346      BIS      #DSINTE,DRXCSR ;TURN ON INT. ENABLE
3181 017036 005037 177776      CLR      PS ;LOWER CPU STATUS
3182 017042 052777 000002 162334      BIS      #DTR,DRXCSR ;PUSH OUT INT. BITS
3183 017050 012737 000156 017100      MOV      #110,73$ ;LOAD THE NUMBER
3184 017056 032777 004000 162330 71$:      BIT      #TIMER,DTXDBUF ;CHECK THE TIMER BIT
3185 017064 001374      BNE      71$ ;BR IF SET
3186 017066 032777 004000 162320 72$:      BIT      #TIMER,DTXDBUF ;CHECK THE BIT
3187 017074 001774      BEQ      72$ ;BR IF CLEAR
3188 017076 005327      DEC      (PC)+ ;DECREMENT THE NUMBER
3189 017100 000156 73$:      110. ;OF TIMES TO REPEAT
3190 017102 001365      BNE      71$ ;BR IF MORE TO GO
3191 017104 104022      HLT      22 ;FAILED TO INTERRUPT
3192 017106 104401      SCOPE1 ;BIT09=1?
3193 017110 4$:
3194 017110 032777 100000 162266      BIT      #BIT15,DRXCSR ;CHECK FOR INTERRUPT FROM DSC1
3195 017116 001001      BNE      5$ ;IT CAME FROM DSC1
3196 017120 104023      HLT      23 ;BIT15 IS CLEARED - INTERRUPTED
3197 ;FROM THE WRONG DSC BIT.
3198 017122 005077 162256 5$:      CLR      DRXCSR ;CLEAR OUT RECEIVER CONTROL REG
3199 017126 012706 001150      MOV      #STACK,SP ;RESET THE STACK
3200 017132 104400      SCOPE ;SCOPE THIS TEST
3201 ;***** TEST 25 *****
3202 ;*TEST THAT SETTING RECEIVER INTERRUPT
3203 ;*ENABLE AND RECEIVER DONE CAUSES AN
3204 ;*INTERRUPT TO THE RECEIVER VECTOR
3205 ;*****
3206
3207 ;*****
3208 ;*
3209 ;* TEST 25
3210 ;*
3211 ;*****
3212 ;*****
3213 017134 012737 000025 001226 TST25: MOV #25,DTSTNO
3214 017142 012737 017414 001216      MOV      #TST26,NEXT
3215 017150 012737 000340 177776      MOV      #340,PS ;LOCK OUT INTERRUPTS

```

```

3216 017156 052777 000400 162226      BIS      #MRESET, @TXCSR      ;RESET THE DEVICE
3217 017164 004737 004772                JSR      PC, SMALL      ;WAIT FOR RESET TO FINISH
3218 017170 052777 004000 162214      BIS      #SYSTST, @TXCSR ;ENTER SYSTEM TEST MODE
3219 017176 004537 007160                JSR      RS, SETVEC     ;SET UP VECTORS
3220 017202 017376                3$      ;RECEIVER
3221 017204 007206                NO.BTRAP ;TRANSMITTER
3222 017206      340      340      .BYTE 340,340 ;PRIORITY AT 7
3223 017210 005037 177776                CLR      PS            ;LOWER PS
3224 017214 052777 000020 162162      BIS      #RCVEN, @RXCSR ;TURN ON RECEIVER
3225 017222 052777 000100 162154      BIS      #RINTEN, @RXCSR ;TURN ON INT. ENABLE
3226 017230 052777 000020 162154      BIS      #SEND, @TXCSR  ;TURN ON TRANSMITTER
3227 017236                1$:
3228 017236 012737 000005 017266      MOV      #5, 68$      ;LOAD THE NUMBER
3229 017244 032777 004000 162142 66$:      BIT      #TIMER, @TXDBUF ;CHECK THE TIMER BIT
3230 017252 001374                BNE     66$           ;BR IF SET
3231 017254 032777 004000 162132 67$:      BIT      #TIMER, @TXDBUF ;CHECK THE BIT
3232 017262 001774                BEQ     67$           ;BR IF CLEAR
3233 017264 005327                DEC     (PC)+        ;DECREMENT THE NUMBER
3234 017266 000005                5         ;OF TIMES TO REPEAT
3235 017270 001365                BNE     66$           ;BR IF MORE TO GO
3236 017272 032777 000200 162112      BIT      #TXDONE, @TXCSR ;TEST TXDONE
3237 017300 001001                BNE     2$           ;BR IF SET
3238 017302 104024                HLT     24           ;TXDONE FAILED TO SET
3239 017304                2$:
3240 017304 012777 000400 162102      MOV      #400, @TXDBUF ;LOAD TX BUFFER
3241 017312 105777 162074                TSTB   @TXCSR        ;CHECK FOR
3242 017316 100375                BPL     -4           ;DONE
3243 017320 005077 162070                CLR     @TXDBUF      ;CLEAR TX BUFFER
3244 017324 105777 162062                TSTB   @TXCSR        ;AND CHECK
3245 017330 100375                BPL     -4           ;FOR DONE
3246 017332 012777 001000 162054      MOV      #1000, @TXDBUF ;LOAD END OF MSG
3247 017340 012737 000050 017370      MOV      #40, 73$    ;LOAD THE NUMBER
3248 017346 032777 004000 162040 71$:      BIT      #TIMER, @TXDBUF ;CHECK THE TIMER BIT
3249 017354 001374                BNE     71$          ;BR IF SET
3250 017356 032777 004000 162030 72$:      BIT      #TIMER, @TXDBUF ;CHECK THE BIT
3251 017364 001774                BEQ     72$          ;BR IF CLEAR
3252 017366 005327                DEC     (PC)+        ;DECREMENT THE NUMBER
3253 017370 000050                40        ;OF TIMES TO REPEAT
3254 017372 001365                BNE     71$          ;BR IF MORE TO GO
3255 017374 104022                HLT     22           ;RECEIVER FAILED TO INTERRUPT
3256 017376 012706 001150 3$:      MOV      #STACK, SP  ;RESET STACK
3257 017402 005077 161776                CLR     @RXCSR       ;CLEAR OUT REGISTER
3258 017406 005077 162000                CLR     @TXCSR       ;DITTO
3259 017412 104400                SCOPE ;SCOPE THIS TEST
3260
3261
3262
3263
3264
3265
3266
3267
3268
3269
3270
3271

```

```

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

```



```

3272
3273 017414 012737 000026 001226 :*****
3274 017422 012737 017774 001216 TST26: MOV #26,@TSTNO
3275 017430 012737 000340 177776 MOV #TST27,NEXT
3276 017436 052777 000400 161746 MOV #340,PS ;LOCK OUT INTERRUPTS
3277 017444 004737 004772 JSR PC,SMALL ;RESET THE DEVICE
3278 017450 052777 004000 161734 BIS #MRESET,@TXCSR ;WAIT FOR RESET TO FINISH
3279 017456 004537 007160 JSR #SYSTST,@TXCSR ;ENTER SYSTST MODE
3280 017462 017744 6S JSR R5,SETVEC ;SETUP VECTORS
3281 017464 007206 NO.BTRAP ;RECEIVER VECTOR
3282 017466 340 340 .BYTE 340,340 ;TRANSMITTER VECTOR
3283 017470 052777 000020 161706 BIS #RCVEN,@RXCSR ;LEVEL
3284 017476 052777 000100 161700 BIS #RINTEN,@RXCSR ;TURN ON RECEIVER
3285 017504 052777 000020 161700 BIS #SEND,@TXCSR ;TURN ON INT. ENABLE
3286 017512 1S: MOV #5,68S ;TURN ON TRANSMITTER
3287 017512 012737 000005 017542 66S: BIT #TIMER,@TXDBUF ;LOAD THE NUMBER
3288 017520 032777 004000 161666 BNE 66S ;CHECK THE TIMER BIT
3289 017526 001374 67S: BIT #TIMER,@TXDBUF ;BR IF SET
3290 017530 032777 004000 161656 BEQ 67S ;CHECK THE BIT
3291 017536 001774 68S: DEC (PC)+ ;BR IF CLEAR
3292 017540 005327 5 ;DECREMENT THE NUMBER
3293 017542 000005 OF TIMES TO REPEAT
3294 017544 001365 BNE 66S ;BR IF MORE TO GO
3295 017546 032777 000200 161636 BIT #TXDONE,@TXCSR ;TEST TXDONE
3296 017554 001001 BNE 2S ;WAIT
3297 017556 104024 HLT 24 ;TXDONE FAILED TO SET
3298 017560 012777 000400 161626 2S: MOV #TSOM,@TXDBUF ;LOAD TX BUFFER
3299 017566 105777 161620 TSTB @TXCSR ;CHECK DONE
3300 017572 100375 BPL -4 ;AND THEN
3301 017574 005077 161614 CLR @TXDBUF ;LOAD BUFFER
3302 017600 105777 161606 TSTB @TXCSR ;AND CHECK
3303 017604 100375 BPL -4 ;DONE AGAIN, THEN
3304 017606 012777 001000 161600 MOV #TEOM,@TXDBUF ;SET END OF MSG
3305 017614 005037 177776 CLR PS ;LOWER PS
3306 017620 10S:
3307 017620 012737 000050 017650 MOV #40..73S ;LOAD THE NUMBER
3308 017626 032777 004000 161560 71S: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3309 017634 001374 BNE 71S ;BR IF SET
3310 017636 032777 004000 161550 72S: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3311 017644 001774 BEQ 72S ;BR IF CLEAR
3312 017646 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3313 017650 000050 73S: 40. ;OF TIMES TO REPEAT
3314 017652 001365 BNE 71S ;BR IF MORE TO GO
3315 017654 104022 HLT 22 ;RECEIVER FAILED TO INTERRUPT
3316 017656 000420 BR 5S ;LEAVE
3317 017660 3S:
3318 017660 032777 004000 161526 74S: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3319 017666 001374 BNE 74S ;BR IF SET
3320 017670 032777 004000 161516 75S: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3321 017676 001774 BEQ 75S ;BR IF CLEAR
3322 017700 012737 000340 177776 MOV #340,PS ;RAISE PS
3323 017706 033777 000200 161470 BIT RXDONE,@RXCSR ;TEST RXDONE
3324 017714 001001 BNE 5S ;BR IF SET
3325 017716 104024 HLT 24 ;RXDONE IS NOT SET AND SHOULD BE
3326 017720 012737 000340 177776 5S: MOV #340,PS ;LOCKOUT INTERRUPTS
3327 017726 005077 161452 CLR @RXCSR ;CLEAR OUT DEVICE

```

```

3328 017732 005077 161450 CLR @RXDBUF ;DITTO
3329 017736 012706 001150 MOV #STACK,SP ;RESET THE STACK
3330 017742 104400 SCOPE ;SCOPE THIS TEST
3331 017744 65:
3332 017744 012716 017660 MOV #3$, (SP) ;2ND PART SETUP
3333 017750 004537 007160 JSR R5,SETVEC ;SETUP FOR SECOND INTERRUPT TRY
3334 017754 017764 7$ ;RECEIVER VECTOR
3335 017756 007206 NO.BTRAP ;TRANSMITTER VECTOR
3336 017760 340 340 .BYTE 340,340 ;LEVEL
3337 017762 000002 RTI ;RETURN
3338 017764 7$:
3339 017764 104023 HLT 23 ;REPORT THE FACT YOU GOT HERE
3340 ;YOU TOOK A SECOND INTERRUPT AND SHOULDN'T HAVE
3341 017766 012716 017720 MOV #5$, (SP) ;SETUP TO LEAVE TEST
3342 017772 000002 RTI ;LEAVE
3343
3344
3345
3346
3347
3348
3349
3350
3351
3352
3353
3354
3355
3356

```

```

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

```

```

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

```

```

3357 017774 012737 000027 001226 TST27: MOV #27,@TSTNO
3358 020002 012737 020334 001216 MOV #TST30,NEXT
3359 020010 012737 000340 177776 MOV #340,P5 ;SET PRIORITY TO 7
3360 020016 052777 000400 161366 BIS #MRESET,@TXCSR ;RESET THE DEVICE
3361 020024 004737 004772 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
3362 020030 004537 007160 JSR R5,SETVEC ;SET UP THE VECTORS
3363 020034 020314 4$ ;RECEIVER VECTOR
3364 020036 007206 NO.BTRAP ;TRANSMITTER VECTOR
3365 020040 340 340 .BYTE 340,340 ;LEVEL
3366 020042 052777 004000 161342 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
3367 020050 052777 000020 161326 BIS #RCVEN,@RXCSR ;TURN ON RECEIVER
3368 020056 052777 000100 161320 BIS #RINTEN,@RXCSR ;TURN ON REC. DONE INT. ENABLE
3369 020064 052777 001000 161316 BIS #CRCEN,@PARCSR ;TURN OFF CRC
3370 020072 052777 000020 161312 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
3371 020100 1$:
3372 020100 012737 000005 020130 MOV #5,68$ ;LOAD THE NUMBER
3373 020106 032777 004000 161300 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3374 020114 001374 BNE 66$ ;BR IF SET
3375 020116 032777 004000 161270 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
3376 020124 001774 BEQ 67$ ;BR IF CLEAR
3377 020126 005327 DEC (PC)+ ;DECREMENT THE NUMBER
3378 020130 000005 68$: 5 ;OF TIMES TO REPEAT
3379 020132 001365 BNE 66$ ;BR IF MORE TO GO
3380 020134 032777 000200 161250 BIT #TXDONE,@TXCSR ;TEST TXDONE
3381 020142 001001 BNE 2$ ;BR IF SET
3382 020144 104024 HLT 24 ;TXDONE FAILED TO SET
3383 020146 2$:

```



```

3384 020146 012777 000400 161240      MOV      #400, @TXDBUF      ;LOAD TX BUFFER
3385 020154 105777 161232      TSTB    @TXCSR            ;CHECK FOR
3386 020160 100375      BPL     .-4                ;DONE
3387 020162 005077 161226      CLR     @TXDBUF           ;LOAD THE BUFFER
3388 020166 105777 161220      TSTB    @TXCSR            ;AND CHECK
3389 020172 100375      BPL     .-4                ;DONE AGAIN
3390 020174 012777 001000 161212      MOV     #1000, @TXDBUF    ;LOAD TEOM
3391 020202 012737 000144 020232      MOV     #100, 73$        ;LOAD THE NUMBER
3392 020210 032777 004000 161176 71$:    BIT     #TIMER, @TXDBUF   ;CHECK THE TIMER BIT
3393 020216 001374      BNE     71$                ;BR IF SET
3394 020220 032777 004000 161166 72$:    BIT     #TIMER, @TXDBUF   ;CHECK THE BIT
3395 020226 001774      BEQ     72$                ;BR IF CLEAR
3396 020230 005327      DEC     (PC)+             ;DECREMENT THE NUMBER
3397 020232 000144      73$:    100.                    ;OF TIMES TO REPEAT
3398 020234 001365      BNE     71$                ;BR IF MORE TO GO
3399 020236 105777 161142      TSTB    @RXCSR            ;CHECK DONE
3400 020242 100401      BMI     5$                ;BR IF SET
3401 020244 104024      HLT     24                ;DONE FAILED TO SET
3402 020246 032777 000200 161136 5$:    BIT     #TXDONE, @TXCSR   ;TEST TXDONE
3403 020254 001001      BNE     3$                ;BR IF SET
3404 020256 104024      HLT     24                ;TXDONE NOT SET SHOULD BE
3405 020260 052777 000100 161124 3$:    BIS     #TXINTE, @TXCSR   ;TURN ON TRANSMITTER INT. ENABLE
3406 020266 005037 177776      CLR     PS                ;LOWER PROCESSOR STATUS
3407 020272 032777 004000 161114 74$:    BIT     #TIMER, @TXDBUF   ;CHECK THE TIMER BIT
3408 020300 001374      BNE     74$                ;BR IF SET
3409 020302 032777 004000 161104 75$:    BIT     #TIMER, @TXDBUF   ;CHECK THE TIMER BIT
3410 020310 001774      BEQ     75$                ;BR IF CLEAR
3411 020312 104027      HLT     27                ;DUP FAILED TO INTERRUPT
3412 020314 012706 001150 4$:    MOV     #STACK, SP        ;RESET THE STACK
3413 020320 052777 000400 161064      BIS     #MRESET, @TXCSR   ;RESET THE DEVICE
3414 020326 004737 004772      JSR     PC, SMALL         ;WAIT FOR RESET TO FINISH
3415 020332 104400      SCOPE                      ;SCOPE THIS TEST
3416
3417
3418
3419
3420
3421
3422
3423
3424
3425
3426
3427
3428
3429 020334 012737 000030 001226      TST30: MOV     #30, @TSTNO
3430 020342 012737 020642 001216      MOV     #TST31, NEXT
3431 020350 012737 000340 177776      MOV     #340, PS          ;LOCK OUT INTERRUPTS
3432 020356 052777 000400 161026      BIS     #MRESET, @TXCSR   ;RESET THE DEVICE
3433 020364 004737 004772      JSR     PC, SMALL         ;WAIT FOR RESET TO FINISH
3434 020370 004537 007160      JSR     R5, SETVEC        ;INTERRUPT VECTOR SETUP
3435 020374 007202      NO. ATRAP                ;RECEIVER VECTOR
3436 020376 020554      4$                        ;TRANSMITTER VECTOR
3437 020400 340 340      .BYTE 340, 340           ;LEVEL
3438 020402 052777 004120 161002      BIS     #SYSTST!SEND!TXINTE, @TXCSR ;TURN ON TRANSMITTER, CLOCK
3439

```

```

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

```

```

*****
*
* TEST 30
*
*****
*****

```

3440	020410	052777	000400	160776		BIS	#TSOM, @TXDBUF	;LOAD START OF MSG
3441	020416	005037	177776			CLR	PS	;LOWER PROCESSOR STATUS
3442	020422	012737	000005	020452		MOV	#5, 68\$	;LOAD THE NUMBER
3443	020430	032777	004000	160756	66\$:	BIT	#TIMER, @TXDBUF	;CHECK THE TIMER BIT
3444	020436	001374				BNE	66\$	;BR IF SET
3445	020440	032777	004000	160746	67\$:	BIT	#TIMER, @TXDBUF	;CHECK THE BIT
3446	020446	001774				BEQ	67\$	;BR IF CLEAR
3447	020450	005327				DEC	(PC)+	;DECREMENT THE NUMBER
3448	020452	000005			68\$:	5		;OF TIMES TO REPEAT
3449	020454	001365				BNE	66\$	;BR IF MORE TO GO
3450	020456	104027				HLT	27	;DUP FAILED TO INTERRUPT THE FIRST TIME
3451	020460	000427				BR	3\$	;LEAVE THE TEST
3452	020462	005037	177776		1\$:	CLR	PS	;LOWER PROCESSOR STATUS
3453	020466	012737	000005	020516		MOV	#5, 73\$	;LOAD THE NUMBER
3454	020474	032777	004000	160712	71\$:	BIT	#TIMER, @TXDBUF	;CHECK THE TIMER BIT
3455	020502	001374				BNE	71\$	;BR IF SET
3456	020504	032777	004000	160702	72\$:	BIT	#TIMER, @TXDBUF	;CHECK THE BIT
3457	020512	001774				BEQ	72\$	;BR IF CLEAR
3458	020514	005327				DEC	(PC)+	;DECREMENT THE NUMBER
3459	020516	000005			73\$:	5		;OF TIMES TO REPEAT
3460	020520	001365				BNE	71\$	;BR IF MORE TO GO
3461	020522	104027				HLT	27	;DUP FAILED TO INTERRUPT AFTER SERVICING DONE
3462	020524	000405				BR	3\$	;LEAVE
3463	020526				2\$:			
3464	020526	005077	160660			CLR	@TXCSR	;SHUT DOWN THE DUP
3465	020532	012716	020540			MOV	#3\$, (SP)	;SETUP TO END TEST
3466	020536	000002				RTI		;RETURN
3467	020540	012737	000340	177776	3\$:	MOV	#340, PS	;RAISE PROCESSOR STATUS
3468	020546	012706	001150			MOV	#STACK, SP	;RESET STACK
3469	020552	104400				SCOPE		;SCOPE THIS TEST
3470	020554	032777	000200	160630	4\$:	BIT	#TXDONE, @TXCSR	;CLEAR DONE AND RE-ARM INTERRUPT
3471	020562	005077	160626			CLR	@TXDBUF	;LOAD BUFFER
3472	020566	012737	000005	020616		MOV	#5, 78\$	;LOAD THE NUMBER
3473	020574	032777	004000	160612	76\$:	BIT	#TIMER, @TXDBUF	;CHECK THE TIMER BIT
3474	020602	001374				BNE	76\$	;BR IF SET
3475	020604	032777	004000	160602	77\$:	BIT	#TIMER, @TXDBUF	;CHECK THE BIT
3476	020612	001774				BEQ	77\$	;BR IF CLEAR
3477	020614	005327				DEC	(PC)+	;DECREMENT THE NUMBER
3478	020616	000005			78\$:	5		;OF TIMES TO REPEAT
3479	020620	001365				BNE	76\$	;BR IF MORE TO GO
3480	020622	012716	020462			MOV	#1\$, (SP)	;SETUP TO FINISH TEST
3481	020626	004537	007160			JSR	R5, SETVEC	;SETUP VECTORS FOR NEXT PART OF TEST
3482	020632	007202				NO. ATRAP		;RECEIVER VECTOR
3483	020634	020526				2\$		;TRANSMITTER VECTOR
3484	020636	340	340			.BYTE	340, 340	;LEVEL
3485	020640	000002				RTI		;RETURN

```

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

```

```

;*****
;

```

3486  
3487  
3488  
3489  
3490  
3491  
3492  
3493  
3494  
3495



```

3496 ; TEST 31
3497 ;
3498 ;*****
3499 ;*****
3500 020642 012737 000031 001226 TST31: MOV #31, @TSTNO
3501 020650 012737 021234 001216 MOV #TST32, NEXT
3502 020656 012737 000340 177776 MOV #340, PS ; LOCK OUT INTERRUPTS
3503 020664 052777 000400 160520 BIS #MRESET, @TXCSR ; RESET THE DEVICE
3504 020672 004737 004772 JSR PC, SMALL ; WAIT FOR RESET TO FINISH
3505 020676 052777 004000 160506 BIS #SYSTST, @TXCSR ; ENTER SYSTST MODE
3506 020704 004537 007160 JSR R5, SETVEC ; SETUP VECTORS
3507 020710 021152 4$ ; RECEIVER VECTOR
3508 020712 007206 NO.BTRAP ; TRANSMITTER VECTOR
3509 020714 340 340 .BYTE 340, 340 ; LEVEL
3510 020716 052777 000020 160460 BIS #RCVEN, @RXCSR ; TURN ON RECEIVER
3511 020724 052777 000100 160452 BIS #RINTEN, @RXCSR ; TURN ON INT. ENABLE
3512 020732 052777 000020 160452 BIS #SEND, @TXCSR ; TURN ON TRANSMITTER
3513 020740 1$:
3514 020740 012737 000005 020770 MOV #5, 68$ ; LOAD THE NUMBER
3515 020746 032777 004000 160440 66$: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
3516 020754 001374 BNE 66$ ; BR IF SET
3517 020756 032777 004000 160430 67$: BIT #TIMER, @TXDBUF ; CHECK THE BIT
3518 020764 001774 BEQ 67$ ; BR IF CLEAR
3519 020766 005327 DEC (PC)+ ; DECREMENT THE NUMBER
3520 020770 000005 5 ; OF TIMES TO REPEAT
3521 020772 001365 BNE 66$ ; BR IF MORE TO GO
3522 020774 032777 000200 160410 BIT #TXDONE, @TXCSR ; TEST TXDONE
3523 021002 001001 BNE 2$ ; BR IF SET
3524 021004 104024 HLT 24 ; TXDONE FAILED TO SET
3525 021006 005037 177776 2$: CLR PS ; LOWER PROCESSOR STATUS
3526 021012 012777 000400 160374 MOV #400, @TXDBUF ; LOAD TX BUFFER
3527 021020 105777 160366 TSTB @TXCSR
3528 021024 100375 BPL -4
3529 021026 005077 160362 CLR @TXDBUF
3530 021032 105777 160354 TSTB @TXCSR
3531 021036 100375 BPL -4
3532 021040 012777 001000 160346 MOV #1000, @TXDBUF
3533 021046 7$:
3534 021046 012737 000050 021076 MOV #40, 73$ ; LOAD THE NUMBER
3535 021054 032777 004000 160332 71$: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
3536 021062 001374 BNE 71$ ; BR IF SET
3537 021064 032777 004000 160322 72$: BIT #TIMER, @TXDBUF ; CHECK THE BIT
3538 021072 001774 BEQ 72$ ; BR IF CLEAR
3539 021074 005327 DEC (PC)+ ; DECREMENT THE NUMBER
3540 021076 000050 40. ; OF TIMES TO REPEAT
3541 021100 001365 BNE 71$ ; BR IF MORE TO GO
3542 021102 104022 HLT 22 ; RECEIVER FAILED TO INTERRUPT
3543 021104 000445 BR 6$ ; GET OUT OF TEST
3544 021106 005037 177776 3$: CLR PS ; LOWER STATUS
3545 021112 012737 000005 021142 MOV #5, 78$ ; LOAD THE NUMBER
3546 021120 032777 004000 160266 76$: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
3547 021126 001374 BNE 76$ ; BR IF SET
3548 021130 032777 004000 160256 77$: BIT #TIMER, @TXDBUF ; CHECK THE BIT
3549 021136 001774 BEQ 77$ ; BR IF CLEAR
3550 021140 005327 DEC (PC)+ ; DECREMENT THE NUMBER
3551 021142 000005 5 ; OF TIMES TO REPEAT
  
```

```

3552 021144 001365      BNE      76$      ;BR IF MORE TO GO
3553 021146 104022      HLT      22      ;DUP FAILED TO INTERRUPT AFTER SERVICING DONE
3554 021150 000423      BR       6$      ;LEAVE IT
3555
3556 021152 032777 000200 160224 4$:  BIT      #RXDONE,@RXCSR ;SERVICE THE DONE BIT
3557 021160 012716 021106      MOV      #3$,(SP) ;SETUP FOR 2ND PART OF TEST
3558 021164 004537 007160      JSR      R5,SETVEC ;SETUP NEW VECTORS
3559 021170 021200      5$      ;RECEIVER VECTOR
3560 021172 007206      NO.BTRAP ;TRANSMITTER VECTOR
3561 021174      340      340      .BYTE   340,340 ;LEVEL
3562 021176 000002      RTI     ;GO FINISH TEST
3563 021200
3564 021200 052777 000400 160204 5$:  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3565 021206 004737 004772      JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
3566 021212 012716 021220      MOV      #6$,(SP) ;SETUP TO FINISH TEST
3567 021216 000002      RTI     ;LEAVE
3568 021220 012737 000340 177776 6$:  MOV      #340,PS ;RAISE PROCESSOR STATUS
3569 021226 012706 001150      MOV      #STACK,SP ;RESET THE STACK
3570 021232 104400      SCOPE   ;SCOPE THIS TEST
3571
3572
3573
3574
3575
3576
3577
3578
3579
3580
3581
3582 021234 012737 000032 001226 TST32: MOV      #32,@TSTNO
3583 021242 012737 021512 001216      MOV      #TST33,NEXT
3584 021250 012737 000340 177776      MOV      #340,PS ;LOCK OUT INTERRUPTS
3585 021256 052777 000400 160126      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3586 021264 004737 004772      JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
3587 021270 052777 004000 160114      BIS      #SYSTST,@TXCSR ;ENTER SYS TST MODE
3588 021276 004537 007160      JSR      R5,SETVEC ;SET UP VECTORS
3589 021302 021464      2$      ;RECEIVER
3590 021304 007206      NO.BTRAP ;TRANSMITTER
3591 021306      340      340      .BYTE   340,340 ;LEVEL
3592 021310 005037 177776      CLR      PS ;LOWER PROCESSOR STATUS
3593 021314 052777 000120 160062      BIS      #RCVEN!RINTEN,@RXCSR ;TURN ON RECEIVER AND INTERRUPT ENABLE
3594 021322 052777 000020 160062      BIS      #SEND,@TXCSR ;TURN ON TRANSMITTER
3595 021330 012737 000005 021360      MOV      #5,68$ ;LOAD THE NUMBER
3596 021336 032777 004000 160050 66$:  BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3597 021344 001374      BNE     66$      ;BR IF SET
3598 021346 032777 004000 160040 67$:  BIT      #TIMER,@TXDBUF ;CHECK THE BIT
3599 021354 001774      BEQ     67$      ;BR IF CLEAR
3600 021356 005327      DEC     (PC)+ ;DECREMENT THE NUMBER
3601 021360 000005      5 ;OF TIMES TO REPEAT
3602 021362 001365      BNE     66$      ;BR IF MORE TO GO
3603 021364 105777 160022      TSTB   @TXCSR ;TEST DONE
3604 021370 100401      BMI     1$      ;BR IF SET
3605 021372 104024      HLT     24 ;DONE FAILED TO SET
3606 021374 012777 000400 160012 1$:  MOV      #T$OM,@TXDBUF ;LOAD TX BUFFER
3607 021402 105777 160004      TSTB   @TXCSR ;CHECK FOR DONE

```

```

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

```

```

:*****
:TEST 32
:*****
:*****

```



```

3608 021406 100375          BPL      -4          ;BR IF NOT SET
3609 021410 005077 160000   CLR      @TXDBUF     ;PUSH OUT A CHARACTER
3610 021414 105777 157772   TSTB    @TXCSR      ;WAIT FOR DONE
3611 021420 052777 002000 157766   BIS      #TABORT,@TXDBUF ;SET ABORT
3612 021426 012737 000113 021456   MOV      #75,73$     ;LOAD THE NUMBER
3613 021434 032777 004000 157752 71$:   BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
3614 021442 001374          BNE      71$         ;BR IF SET
3615 021444 032777 004000 157742 72$:   BIT      #TIMER,@TXDBUF ;CHECK THE BIT
3616 021452 001774          BEQ      72$         ;BR IF CLEAR
3617 021454 005327          DEC      (PC)+       ;DECREMENT THE NUMBER
3618 021456 000113          75.        ;OF TIMES TO REPEAT
3619 021460 001365          BNE      71$         ;BR IF MORE TO GO
3620 021462 104022          HLT                ;RECEIVER FAILED TO INTERRUPT
3621                                     ;ON AN ABORTED MESSAGE
3622 021464 012737 000340 177776 2$:   MOV      #340,PS     ;SET STATUS TO 7
3623 021472 012706 001150          MOV      #STACK,SP   ;RESET THE STACK
3624 021476 052777 000400 157706   BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3625 021504 004737 004772          JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
3626 021510 104400          SCOPE
3627
3628                                     ;***** TEST 33 *****
3629                                     ;*THIS TEST IS AN AID FOR DEBUGGING CRC
3630                                     ;*ERRORS. A CHARACTER IS LOADED INTO THE
3631                                     ;*DUP AND PUSHED OUT BIT BY BIT WHILE
3632                                     ;*ALLOWING THE OPERATOR TO MONITOR THE CRC
3633                                     ;*CHARACTER AS IT IS GENERATED. THE DATA CHARACTER
3634                                     ;*CAN ALSO BE CHANGED BY THE OPERATOR.
3635                                     ;*PUT SW09=1 TO LOCK ON BITS. TO CONTINUE HIT
3636                                     ;*ANY KEY ON THE TTY. AFTER 16 TIMES PUT DOWN SW09 TO LEAVE
3637                                     ;NOTE: REMEMBER--IN SDLC A ONE IS A LOGIC LOW IN THE
3638                                     ;CRC GENERATOR.
3639                                     ;*****
3640                                     ;*****
3641                                     ;*
3642                                     ;* TEST 33
3643                                     ;*
3644                                     ;*****
3645                                     ;*****
3646 021512 012737 000033 001226 TST33: MOV      #33,@TSTNO
3647 021520 012737 002764 001216   MOV      #.EOP,NEXT
3648 021526 052777 000400 157656   BIS      #MRESET,@TXCSR ;RESET THE DEVICE
3649 021534 004737 004772          JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
3650 021540 012737 102010 013516   MOV      #CRC.CCITT,XPOLY ;LOAD THE POLYNOMIAL
3651 021546 012737 000125 021710   MOV      #125,3$     ;LOAD DATA TO SOFTWARE BCC-CHANGE CHARACTER HERE
3652 021554 013737 021710 001252   MOV      3$,SAVR1
3653 021562 012737 177777 013522   MOV      #-1,CALBCC  ;CLEAR FOR SOFTWARE BCC
3654 021570 013737 013522 021712   MOV      CALBCC,4$
3655 021576 005037 001242          CLR      TEMP3
3656 021602 005037 001244          CLR      TEMP4      ;CLEAR BIT COUNTER
3657 021606 005037 001246          CLR      TEMP5
3658 021612 052777 014000 157572   BIS      #MMODE,@TXCSR ;ENTER MAINT MODE-PROGRAM CLOCKING
3659 021620 052777 000020 157556   BIS      #RCVEN,@RXCSR ;TURN ON RECEIVER
3660 021626 052777 000020 157556   BIS      #SEND,@TXCSR ;TURN ON TRANSMITTER
3661 021634 012777 000400 157552   MOV      #TSOM,@TXDBUF ;
3662 021642 104412 000044          PKCLK    36.        ;PUSH OUT 2
3663 021646 013777 021710 157540   MOV      3$,@TXDBUF ;LOAD DATA

```





3713	022110	051377	047505	020115	EM1:	.ASCIZ	<377>/REOM BIT /		
(1)	022123	377	054122	047504	EM2:	.ASCIZ	<377>/RXDONE BIT /		
(1)	022140	051377	040530	047502	EM3:	.ASCIZ	<377>/RXABORT BIT /		
(1)	022156	051377	040530	052103	EM4:	.ASCIZ	<377>/RXACTIVE BIT /		
(1)	022175	377	054122	042504	EM5:	.ASCIZ	<377>/RXDERR BIT /		
(1)	022212	051377	043505	051511	EM6:	.ASCIZ	<377>/REGISTER ERROR /		
(1)	022233	377	054122	042504	EM7:	.ASCIZ	<377>/RXDERR OR OVERRUN /		
(1)	022257	377	053117	051105	EM10:	.ASCIZ	<377>/OVERRUN BIT /		
(1)	022275	377	040504	040524	EM11:	.ASCIZ	<377>/DATA COMPARE ERROR /		
(1)	022322	042377	052101	020101	EM12:	.ASCIZ	<377>/DATA COMPARE ERROR IN SECONDARY MODE /		
(1)	022371	377	051103	020103	EM13:	.ASCIZ	<377>/CRC CALCULATION ERROR /		
(1)	022421	377	051522	046517	EM14:	.ASCIZ	<377>/RSOM BIT /		
(1)	022434	040506	046111	042105	DH1:	.ASCIZ	/FAILED TO CLEAR /		
(1)	022455	106	044501	042514	DH2:	.ASCIZ	/FAILED TO SET /		
(1)	022474	052777	042516	050130	DH3:	.ASCIZ	<377>/UNEXPECTED /		
(1)	022511	377	054105	042520	DH6:	.ASCIZ	<377>/EXPECTED FOUND REGISTER /		
(1)	022551	377	051103	020103	EM17:	.ASCIZ	<377>/CRC GENERATOR STATUS /		
(1)	022600	042377	052101	020101	MH1:	.ASCIZ	<377>/DATA CHAR DATA BIT IN CRC GEN.		CRC FOR THIS BIT /
(1)	022673	015	052012	040522	EM15:	.ASCIZ	<15><12>/TRANSMITTER /		
(1)	022713	015	051012	041505	EM16:	.ASCIZ	<15><12>/RECEIVER /		
(1)	022730	047504	042516	041040	EM23:	.ASCIZ	/DONE BIT /		
(1)	022743	106	044501	042514	EM22:	.ASCIZ	/FAILED TO SET /		
(1)	022762	052777	042516	050130	EM20:	.ASCIZ	<377>/UNEXPECTED RECEIVER INTERRUPT /		
(1)	023023	377	047125	054105	EM21:	.ASCIZ	<377>/UNEXPECTED TRANSMITTER INTERRUPT /		
(1)	023067	106	044501	042514	DH4:	.ASCIZ	/FAILED TO INTERUPT. /		
(1)	023114	047111	042524	052522	DH5:	.ASCIZ	/INTERUPTED UNEXPECTEDLY. /		
(1)		023146							
(1)									
(1)									
(1)									
(1)	023146								
(1)	023146	000000							
(1)	023150	000000							
(1)	023152	000000							
(1)	023154	022110							
(1)	023156	022455							
(1)	023160	000000							
(1)									
(1)	023162	022123							
(1)	023164	022455							
(1)	023166	000000							
(1)									
(1)	023170	022212							
(1)	023172	022511							
(1)	023174	023404							
(1)									
(1)	023176	022110							
(1)	023200	022434							
(1)	023202	000000							
(1)									
(1)	023204	022140							
(1)	023206	022455							
(1)	023210	000000							
(1)									
(1)	023212	022156							
(1)	023214	022434							

.ERRTAB:

0		
0		
0		
EM1		
DH2	;HALT 1	
0		
EM2		
DH2	;HALT 2	
0		
EM6		
DH6	;HALT 3	
DT6		
EM1		
DH1	;HALT 4	
0		
EM3		
DH2	;HALT 5	
0		
EM4		
DH1	;HALT 6	

(1)	023216	000000	0	
(1)				
(1)	023220	022175	EM5	
(1)	023222	022455	DH2	:HALT 7
(1)	023224	000000	0	
(1)				
(1)	023226	022275	EM11	
(1)	023230	000000	0	;HALT 10
(1)	023232	000000	0	
(1)				
(1)	023234	022140	EM3	
(1)	023236	022434	DH1	;HALT 11
(1)	023240	000000	0	
(1)				
(1)	023242	022322	EM12	
(1)	023244	000000	0	;HALT 12
(1)	023246	000000	0	
(1)				
(1)	023250	022233	EM7	
(1)	023252	022455	DH2	;HALT 13
(1)	023254	000000	0	
(1)				
(1)	023256	022156	EM4	
(1)	023260	022455	DH2	;HALT 14
(1)	023262	000000	0	
(1)				
(1)	023264	022257	EM10	
(1)	023266	022434	DH1	;HALT 15
(1)	023270	000000	0	
(1)				
(1)	023272	022474	DH3	
(1)	023274	022212	EM6	;HALT 16
(1)	023276	000000	0	
(1)				
(1)	023300	022371	EM13	
(1)	023302	000000	0	;HALT 17
(1)	023304	000000	0	
(1)				
(1)	023306	022421	EM14	
(1)	023310	022434	DH1	;HALT 20
(1)	023312	000000	0	
(1)				
(1)	023314	022421	EM14	
(1)	023316	022455	DH2	;HALT 21
(1)	023320	000000	0	
(1)				
(1)	023322	022713	EM16	
(1)	023324	023067	DH4	;HALT 22
(1)	023326	000000	0	
(1)				
(1)	023330	022713	EM16	
(1)	023332	023114	DH5	;HALT 23
(1)	023334	000000	0	
(1)				
(1)	023336	022730	EM23	
(1)	023340	022743	EM22	;HALT 24



```

(1) 023342 000000      0
(1)
(1) 023344 022762      EM20
(1) 023346 000000      0      ;HALT 25
(1) 023350 000000      0
(1)
(1) 023352 023023      EM21
(1) 023354 000000      0      ;HALT 26
(1) 023356 000000      0
(1)
(1) 023360 022673      EM15
(1) 023362 023067      DM4      ;HALT 27
(1) 023364 000000      0
(1)
(1) 023366 000003      DT1: 3
(1) 023370 006      021      .BYTE 6,17.
(1) 023372 001252      SAVR1
(1) 023374 006      017      .BYTE 6,15.
(1) 023376 001254      SAVR2
(1) 023400 006      002      .BYTE 6,2
(1) 023402 001256      SAVR3
(1)
(1) 023404 000003      DT6: 3
(1) 023406 006      004      .BYTE 6,4
(1) 023410 001262      SAVR5
(1) 023412 006      002      .BYTE 6,2
(1) 023414 001260      SAVR4
(1) 023416 006      002      .BYTE 6,2
(1) 023420 001256      SAVR3
(1) 023422 000001      CORMAX:
      .END

```













GV.FLG 001343	831#	975*	1187*	1226										
RABORT= 002000	774#	2141	2184	2204										
RCRCIN= 040000	803#													
RCRC7T= 100000	802#													
RCVEN = 000020	765#	1835	1882	1950	1998	2051	2126	2170	2235	2276	2331	2394	2462	
REACT= 004000	2574	3224	3283	3367	3510	3593	3659							
REOM = 001000	758#	1839	1859	1887	1901	1903	1930	2144	2207	2251	2307			
RESREG 004614	775#	1973	2026	2074	2101									
RESTRT 003116	1523	1526#												
RESOS = 104407	1186	1190	1198#											
RETURN 001214	858#	1526												
RFLG 007006	707#	981*	1163*	1165	1198*	1235*	1238	1541*	1543	1576	1692*	1702*	1703	
RING = 040000	1743#	1837	1885	1954	2002	2056	2078	2128	2173	2188	2237	2279	2290	
RINTEN= 000100	755#	2665	2668	2771	2881	2991	2994							
RPOKE 006746	763#	3225	3284	3368	3511	3593								
	1732#	1739	1845	1900	1918	1958	2008	2060	2082	2085	2132	2138	2177	
	2183	2192	2195	2201	2241	2244	2283	2294	2297	2300				
RSOM = 000400	776#	1851	1853	1913	1926	2415								
RTS = 000004	767#	2749	2774	2790	2969	2994	3010	3161						
RUN 001314	747#	977*	1619	1622*	1623*	1630*	1631*							
RXCSR 001404	877#	1202	1638*	1642	1833	1858	1880	1929	1950*	1961	1972	1998*	2011	
	2025	2051*	2063	2088	2100	2126*	2139	2170*	2202	2233	2274	2331*	2341	
	2352	2394*	2409	2411	2462*	2474	2482	2487	2574*	2603	2634*	2640	2740*	
	2746	2850*	2856	2960*	2966	3160*	3161*	3173*	3180*	3182*	3194	3198*	3224*	
	3225*	3257*	3283*	3284*	3323	3327*	3367*	3368*	3399	3510*	3511*	3556	3593*	
	3659*													
RXDBUF 001406	878#	1644*	1645*	1646	1647	1648	1851	1854	1913	1926	1964	1973	2014	
	2026	2066	2074	2092	2101	2135	2140	2180	2184	2198	2203	2245	2284	
	2301	2344	2355	2414	2477	2485	2490	2606	3328*					
	771#	2150	2213	2246	2247	2302	2303							
RXDERR= 100000	762#	1846	1848	1894	1906	1909	1920	1922	1961	1976	2011	2029	2063	
RXDONE= 000200	2088	2104	2147	2210	3323	3556								
RO =%000000	607#	1009*	1010*	1011	1082*	1087	1089	1091	1093*	1094*	1095	1097*	1098*	
	1099*	1100*	1102*	1103	1104	1105	1106	1107	1108	1129*	1137*	1138*	1140*	
	1142*	1144	1145	1216	1222*	1236*	1369	1374*	1387	1400*	1404*	1414	1430*	
	1532*	1551*	1555*	1558*	1632*	1638	1639	1640	1641*	1661	1663	1665	1668*	
	1685*	1686	1688	1690	1692	1696	1697	1705	1715*	1717*	1791*	1796	1799	
	1802	1804	1807*	1810	1812	1813	2140*	2141	2150	2203*	2204	2213	2245*	
	2246*	2247	2301*	2302*	2303	2512	2519*	2521*	2522	2525*	2526*	2527	2530*	
	2533*	2535*	2537	2543*										
R1 =%000001	608#	1083*	1087*	1088*	1089*	1090*	1091*	1141*	1142	1143*	1144	1189*	1193	
	1368	1375*	1388	1392*	1394	1395	1396	1397	1429*	1580*	1581	1583	1584	
	1669*	1716*	1723*	1740*	1743	1756*	1759	1770*	1791	1805*	1807	1814*	1837*	
	1845*	1885*	1900*	1918*	1954*	1958*	2002*	2004*	2008*	2056*	2060*	2078*	2082*	
	2085*	2128*	2132*	2138*	2173*	2177*	2183*	2188*	2192*	2195*	2201*	2237*	2241*	
	2244*	2279*	2283*	2290*	2294*	2297*	2300*	2448*	2470	2497	2513	2531*	2532	
	2533	2542*	2635*	2741*	2851*	2961*	3709*							
R2 =%000002	609#	1367	1376*	1579*	1581*	1582*	1670*	2328*	2336*	2339*	2349	2449*	2478	
	2498	2514	2532*	2534*	2535	2541*								
R3 =%000003	610#	1271	1278*	1288*	1291*	1292	1297*	1366	1377*	1389	1401*	1402*	1403*	
	1404	1413*	1414*	1419*	1422*	1428*	1705*	1833*	1835*	1839	1846	1849	1854*	
	1855	1858*	1859	1880*	1882*	1887	1891	1894	1901	1904	1906	1910	1920	
	1923	1929*	1930	2233*	2235*	2251	2274*	2276*	2307	2326*	2335	2338	2447*	
	2500*	2640*	2641*	2643*	2652	2665	2669*	2678	2684*	2694*	2706	2707	2746*	
	2747*	2749*	2758	2771	2775*	2784	2790*	2800*	2612	2813	2856*	2857*	2859*	
	2868	2881	2885*	2894	2900*	2910*	2922	2923	2966*	2967*	2969*	2978	2991	









TKCSR	001204	699#	1220	1279	3706															
TKDBR	001206	700#	1222	1281	1287	3709														
TLAST =	021512	1697	3713#																	
TPCSR	001210	701#	1263	1285	1477															
TPDBR	001212	702#	1265*	1287*	1479*															
TRPOK	004330	1466#																		
TRP.PC	006150	1603#																		
TSOM =	000400	809#	2333	2398	2464	2575	3298	3440	3606	3661										
TSTNO	001226	712#	980*	1550	1682	1688	1690	1829*	1876*	1946*	1994*	2047*	2121*	2165*						
		2229*	2270*	2322*	2383*	2444*	2564*	2628*	2734*	2840*	2950*	3057*	3091*	3145*						
		3213*	3273*	3357*	3429*	3500*	3582*	3646*												
TST1	007304	1685	1702	1829#																
TST10	011636	2166	2229#																	
TST11	012010	2230	2270#																	
TST12	012246	2271	2322#																	
TST13	012512	2323	2383#																	
TST14	013006	2384	2444#																	
TST15	013524	2445	2564#																	
TST16	014026	2565	2628#																	
TST17	014474	2629	2734#																	
TST2	007472	1830	1876#																	
TST20	015142	2735	2840#																	
TST21	015624	2841	2950#																	
TST22	016306	2951	3057#																	
TST23	016424	3058	3091#																	
TST24	016616	3092	3145#																	
TST25	017134	3146	3213#																	
TST26	017414	3214	3273#																	
TST27	017774	3274	3357#																	
TST3	007762	1877	1946#																	
TST30	020334	3358	3429#																	
TST31	020642	3430	3500#																	
TST32	021234	3501	3582#																	
TST33	021512	3583	3646#	3713																
TST34 =	***** U	3647																		
TST4	010210	1947	1994#																	
TST5	010460	1995	2047#																	
TST6	011066	2048	2121#																	
TST7	011300	2122	2165#																	
TTST	003174	1158#	1159#	1161*	1162*	1218#														
TWOSYN=	000000	650#																		
TXACT =	001000	788#																		
TXCSR	001412	880#	1533*	1554*	1557*	1651*	1652*	1653	1732*	1735*	1745*	1747*	1752*	1761*						
		1766#	1831*	1836*	1878*	1884*	1948*	1953*	1959*	1965*	1970*	1996*	2001*	2009*						
		2018#	2023*	2049*	2055*	2061*	2067*	2072*	2086*	2093*	2098*	2123*	2127*	2133*						
		2167#	2172*	2178*	2196*	2231*	2236*	2272*	2278*	2324*	2330*	2332*	2360*	2391*						
		2395#	2396	2399	2402	2422*	2459*	2461*	2463*	2466	2566*	2571*	2573*	2581						
		2585	2590	2597	2601*	2638*	2642*	2695*	2697*	2744*	2748*	2801*	2803*	2854*						
		2858#	2911*	2913*	2964*	2968*	3021*	3023*	3062*	3068*	3070*	3076*	3094*	3100*						
		3101#	3112	3116*	3152*	3154*	3216*	3218*	3226*	3236	3241	3244	3258*	3276*						
		3278#	3285*	3295	3299	3302	3360*	3366*	3370*	3380	3385	3388	3402	3405*						
		3413#	3432*	3438*	3464*	3470	3503*	3505*	3512*	3522	3527	3530	3564*	3585*						
		3587#	3594*	3603	3607	3610	3624*	3648*	3658*	3660*	3687*									
TXDBUF	001414	881#	1655*	1656*	1657	2333*	2335*	2338*	2398*	2401*	2404*	2405	2407	2464*						
		2470#	2472*	2575*	2588*	2645	2647	2658	2660	2671	2673	2686	2688	2699						
		2701	2751	2753	2764	2766	2777	2779	2792	2794	2805	2807	2861	2863						





DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 93  
 DZDPCA.CMB CROSS REFERENCE TABLE -- USER SYMBOLS

.CONVR	004030	861	1386#		
.EOP	002764	1172#	3647		
.ERRTA	023146	1494	3713#		
.HLT	004350	668	1475#		
.INSTE	003516	853	1292#		
.INSTR	003412	851	1271#		
.INST1	003432	1275#	1295		
.MSG	003434	1273*	1276#		
.PARAM	003536	855	1303#		
.PFAIL	004776	666	971	1569#	1573
.PKCLK	004734	865	1551#		
.RESOS	003776	859	1374#		
.SAVOS	003736	857	1360#		
.SCOPE	003160	845	1215#		
.SCOPI	003312	847	1245#		
.SETFL	004242	867	1443#	1455	
.START	001562	684	969#	981	
.TRPSR	004316	670	1463#		
.TRPTA	001344	843#	1468		
.TYPE	003336	849	1255#		

DUPEND	18	1166													
DUPFRN	18	568													
HLT	5318	1778	1782	1842	1850	1856	1861	1892	1896	1905	1911	1915	1924	1928	1932
	1963	1975	1978	2013	2017	2028	2031	2065	2076	2090	2103	2106	2143	2146	2149
	2152	2186	2206	2209	2212	2215	2249	2253	2287	2305	2309	2343	2347	2354	2358
	2413	2417	2420	2468	2476	2480	2484	2489	2494	2583	2587	2592	2599	2605	2609
	2655	2681	2710	2761	2787	2816	2871	2897	2926	2981	3007	3036	3074	3106	3114
	3127	3170	3191	3196	3238	3255	3297	3315	3325	3339	3382	3401	3404	3411	3450
	3461	3524	3542	3553	3605	3620									
DOO	18														
SABORT	5688	3571													
SALTOS	18														
SBT	5688	2613	2719	2825	2935										
SBUFE	18	1595													
SCLKDN	18	1557													
SCLKUP	18	1554													
SCLRVE	18	1577													
SCYCLE	18	1605													
SDEC23	5688	3627													
SDEFAU	18	1093													
SDELY	18														
SEOP	18	1166													
SERRTA	18														
SFINI	18	3713													
SGETFL	18	1046	1050	1056	1060										
SGETPA	18	1013	1021	1029	1038	1677									
SHEADE	18	568													
SINTA	5688	3344													
SJMPR	5688	1785													
SMSG	18	1590													
SMSGA	37138														
SPFAIL	18	1567													
SPKCK	18	1551													
SQUEST	18	1004													
SRCRC	5688	2432													
SRESET	18	1831	1878	1948	1996	2049	2123	2167	2231	2272	2324	2359	2390	2422	2459
	2566	2638	2695	2744	2801	2854	2911	2964	3021	3062	3094	3152	3216	3276	3360
	3413	3432	3503	3564	3585	3624	3648	3687							
SRFLAG	5688	1742													
SRITX	5688	3261													
SRNT	5688	3487													
SRPOKE	5688	1727													
SRT1	5688	1816	1864												
SRT2	5688	1935	1982	2035											
SRT3	5688	2110	2154												
SRT4	5688	2312													
SRT5	5688	2218	2257												
SRT6	5688	2370													
SRXINT	5688	3131													
SRXITB	5688	3201													
SSCOPE	18	1212													
SSCOP2	18	5688	3691												
SSETFL	18	1437													
SSETVE	5688	1772													
SSFLAG	5688	1758													
SSIMBC	25118	2512													



SSMALL	18	1562													
SSPACE	5688	2551													
SSTUF	5688	1710													
STINTX	5688	3079													
STNT	5688	3417													
STRPDE	18	844	846	848	850	852	854	856	858	860	862	864	866		
STSTN	18	1823	1870	1940	1988	2041	2115	2159	2223	2264	2316	2377	2438	2558	2622
	2728	2834	2944	3051	3085	3139	3207	3267	3351	3423	3494	3576	3640		
STXINT	5688	3045													
SVARIA	18	686													
SWAIT	18	2405	2644	2657	2670	2685	2698	2750	2763	2776	2791	2804	2860	2873	2886
	2901	2914	2970	2983	2996	3011	3024	3162	3183	3227	3247	3286	3306	3318	3371
	3391	3407	3442	3453	3472	3513	3533	3545	3595	3612					
SXZ	18	1816	1820	1864	1868	1935	1939	1982	1987	2035	2040	2110	2114	2154	2158
	2218	2221	2257	2262	2312	2315	2371	2375	2433	2436	2552	2556	2614	2620	2720
	2726	2826	2832	2936	2942	3045	3049	3079	3083	3131	3137	3201	3205	3261	3265
	3344	3349	3417	3421	3488	3492	3571	3574	3628	3639					

ADC	1718	2521													
ADCB	1623	1631													
ADD	1088	1090	1116	1257	1274	1345	1393	1403	1452	1468	1491	1494	1552	1624	1633
	1661	1663	1665												
ASL	1324	1325	1326	1466	1490	1492									
BCC	1086	1724	1734	3698											
BEG	994	997	1055	1065	1068	1074	1113	1119	1127	1156	1190	1218	1227	1246	1248
	1284	1315	1323	1418	1476	1483	1499	1505	1514	1519	1523	1528	1540	1556	1675
	1793	1795	1801	1809	1841	1889	1895	1907	1914	2016	2064	2075	2145	2185	2208
	2248	2285	2304	2346	2357	2408	2419	2425	2479	2493	2523	2579	2582	2586	2598
	2608	2648	2654	2661	2674	2680	2689	2702	2709	2754	2760	2767	2780	2786	2795
	2808	2815	2847	2849	2864	2870	2877	2890	2896	2905	2918	2925	2957	2959	2974
	2980	2987	3000	3006	3015	3028	3035	3166	3187	3232	3251	3291	3311	3321	3376
	3395	3410	3446	3457	3476	3518	3538	3549	3599	3616	3692				
BGT	1319	1720													
BHI	1335														
BIC	1138	1402	1444	1467	1493	1533	1557	1732	1745	1752	1766	1959	1970	2009	2023
	2061	2072	2086	2098	2246	2302	2527	2533	2534	2536	2668	2774	2884	2994	
BICB	1282	1320													
BIS	1066	1069	1554	1735	1747	1761	1831	1834	1835	1836	1878	1881	1882	1883	1884
	1948	1950	1951	1953	1965	1996	1998	1999	2001	2018	2049	2051	2052	2054	2055
	2067	2093	2123	2125	2126	2127	2133	2167	2169	2170	2171	2172	2178	2196	2231
	2234	2235	2236	2272	2275	2276	2277	2278	2324	2329	2330	2331	2332	2333	2360
	2391	2394	2395	2422	2428	2459	2461	2462	2463	2464	2472	2535	2537	2566	2571
	2572	2573	2574	2575	2601	2638	2642	2643	2684	2695	2697	2744	2748	2749	2790
	2801	2803	2854	2858	2859	2900	2911	2913	2964	2968	2969	3010	3021	3023	3062
	3068	3070	3094	3100	3101	3152	3154	3160	3161	3180	3182	3216	3218	3224	3225
	3226	3276	3278	3283	3284	3285	3360	3366	3367	3368	3369	3370	3405	3413	3432
	3438	3440	3503	3505	3510	3511	3512	3564	3585	3587	3593	3594	3611	3624	3648
	3658	3659	3660	3687											
BISB	1321														
BIT	1004	1118	1126	1155	1217	1224	1245	1258	1475	1480	1537	1539	1674	1839	1846
	1851	1859	1887	1894	1901	1906	1913	1920	1926	1930	1961	1973	1976	2011	2026
	2029	2063	2074	2088	2101	2104	2141	2144	2147	2150	2184	2204	2207	2210	2213
	2251	2284	2307	2405	2407	2415	2522	2581	2585	2590	2597	2645	2647	2658	2660
	2665	2671	2673	2686	2688	2699	2701	2751	2753	2764	2766	2771	2777	2779	2792
	2794	2805	2807	2861	2863	2874	2876	2881	2887	2889	2902	2904	2915	2917	2971
	2973	2984	2986	2991	2997	2999	3012	3014	3025	3027	3163	3165	3184	3186	3194
	3229	3231	3236	3248	3250	3288	3290	3295	3308	3310	3318	3320	3323	3373	3375
	3380	3392	3394	3402	3407	3409	3443	3445	3454	3456	3470	3473	3475	3515	3517
	3522	3535	3537	3546	3548	3556	3596	3598	3613	3615	3691				
BITB	1338	1619													
BLO	1337														
BLOS	1132														
BLT	1317														
BMI	2342	2353	2467	2475	2483	2488	2604	3113	3400	3604					
BNE	1005	1012	1096	1121	1125	1146	1154	1186	1225	1230	1259	1266	1289	1339	1347
	1412	1416	1421	1425	1446	1450	1481	1501	1538	1559	1585	1615	1620	1626	1636
	1673	1687	1689	1691	1698	1739	1751	1755	1765	1769	1847	1852	1860	1902	1921
	1927	1931	1962	1964	1974	1977	2012	2022	2027	2030	2071	2089	2097	2102	2105
	2142	2148	2151	2205	2211	2214	2252	2308	2350	2406	2416	2501	2539	2591	2595
	2632	2646	2651	2659	2664	2666	2672	2677	2687	2692	2700	2705	2738	2752	2757
	2765	2770	2772	2778	2783	2793	2798	2806	2811	2844	2862	2867	2875	2880	2882
	2888	2893	2903	2908	2916	2921	2954	2972	2977	2985	2990	2992	2998	3003	3013
	3018	3026	3031	3148	3164	3169	3185	3190	3195	3230	3235	3237	3249	3254	3289
	3294	3296	3309	3314	3319	3324	3374	3379	3381	3393	3398	3403	3408	3444	3449



	3455	3460	3474	3479	3516	3521	3523	3536	3541	3547	3552	3597	3602	3614	3619
	3678	3681	3686	3700											
BPL	1001	1221	1261	1264	1280	1286	1478	1530	2397	2400	2403	2410	2412	3242	3245
	3300	3303	3386	3389	3528	3531	3608	3708							
BR	987	995	1077	1092	1117	1135	1160	1219	1223	1295	1327	1329	1448	1455	1618
	1628	1695	1700	2430	3107	3316	3451	3462	3543	3554					
CLC	1080	1084	1405	1407	1409	1621	1629	2528	3674	3695					
CLRC	973	978	979	1010	1094	1123	1129	1143	1172	1215	1232	1233	1312	1582	1721
	1744	1760	1843	1890	1898	1952	2000	2053	2080	2091	2130	2190	2292	2386	2387
	2388	2450	2518	2568	2569	2570	2634	2641	2669	2693	2694	2740	2747	2775	2799
	2800	2850	2857	2885	2909	2910	2960	2967	2995	3019	3020	3069	3076	3102	3108
	3116	3159	3173	3181	3198	3223	3243	3257	3258	3301	3305	3327	3328	3387	3406
	3441	3452	3464	3471	3525	3529	3544	3592	3609	3655	3656	3657	3679		
CLRB	974	975	1173	1231	1422	1447	1485								
CMP	990	996	1011	1095	1144	1145	1229	1334	1336	1482	1527	1583	1584	1625	1634
	1686	1688	1690	1697	1719	1750	1764	1799	1812	1968	2021	2070	2096	2247	2303
	2492	2578	2594	2653	2679	2708	2759	2785	2814	2869	2895	2924	2979	3005	3034
	3677	3680	3683												
CMPB	1131	1283	1314	1316	1318	1322	1445	1449	2015	2345	2356	2418	2478	2607	
COM	2491	2524	2526												
COMB	999	1007	3703												
DEC	1073	1288	1411	1424	1555	1558	1738	1754	1768	2336	2339	2500	2538	2649	2662
	2675	2690	2703	2755	2768	2781	2796	2809	2865	2878	2891	2906	2919	2975	2988
	3001	3016	3029	3167	3188	3233	3252	3292	3312	3377	3396	3447	3458	3477	3519
	3539	3550	3600	3617											
DECB	1185	1346	1415	1420											
EMT	631														
HALT	662	1130	1134	1139	1534	1570	1617								
INC	1174	1228	1536	1643	1645	1649	1652	1654	1656	1658	1722	1749	1763	1967	2020
	2069	2095	2577	2593	3667	3668									
INCB	2424														
JMP	684	1002	1006	1122	1165	1199	1238	1470	1543	1576	1703	1798	1803	1811	2633
	2739	2845	2955	3149											
JSR	1193	1671	1832	1837	1845	1879	1885	1900	1918	1949	1954	1958	1997	2002	2004
	2008	2050	2056	2060	2078	2082	2085	2124	2128	2132	2138	2168	2173	2177	2183
	2188	2192	2195	2201	2232	2237	2241	2244	2273	2279	2283	2290	2294	2297	2300
	2325	2361	2392	2423	2455	2460	2567	2635	2639	2696	2741	2745	2802	2851	2855
	2912	2961	2965	3022	3063	3064	3095	3096	3121	3153	3155	3175	3217	3219	3277
	3279	3333	3361	3362	3414	3433	3434	3481	3504	3506	3558	3565	3586	3588	3625
	3649	3669	3673	3688											
MOV	969	970	971	976	980	981	983	984	985	988	989	991	992	1009	1037
	1082	1083	1087	1089	1091	1093	1097	1098	1099	1100	1102	1111	1112	1140	1141
	1142	1151	1152	1158	1159	1161	1162	1163	1175	1189	1198	1216	1222	1234	1235
	1236	1249	1255	1256	1267	1271	1272	1273	1277	1278	1287	1290	1291	1292	1293
	1296	1297	1303	1304	1305	1306	1307	1308	1311	1313	1343	1344	1348	1349	1360
	1364	1365	1366	1367	1368	1369	1374	1375	1376	1377	1378	1379	1387	1388	1389
	1390	1391	1392	1394	1397	1398	1400	1401	1413	1426	1427	1428	1429	1430	1443
	1463	1465	1469	1484	1487	1489	1495	1496	1497	1532	1541	1542	1551	1569	1572
	1573	1579	1580	1581	1627	1632	1637	1638	1639	1640	1641	1642	1644	1646	1647
	1648	1650	1651	1653	1655	1657	1660	1662	1664	1668	1669	1670	1685	1692	1702
	1705	1714	1715	1716	1743	1759	1772	1773	1791	1796	1802	1804	1807	1810	1813
	1829	1830	1833	1844	1848	1849	1853	1854	1855	1858	1876	1877	1880	1891	1899
	1903	1904	1909	1910	1917	1922	1923	1929	1946	1947	1956	1957	1972	1994	1995
	2006	2007	2025	2047	2048	2058	2059	2081	2083	2084	2100	2121	2122	2131	2137
	2139	2140	2165	2166	2175	2176	2182	2191	2193	2194	2200	2202	2203	2229	2230
	2233	2239	2240	2242	2243	2245	2270	2271	2274	2281	2282	2293	2295	2296	2298

	2299	2301	2322	2323	2326	2327	2328	2335	2338	2344	2355	2383	2384	2385	2389
	2393	2398	2401	2404	2414	2429	2444	2445	2446	2447	2448	2449	2451	2452	2453
	2454	2470	2477	2485	2512	2513	2514	2515	2516	2517	2519	2525	2530	2531	2532
	2540	2541	2542	2543	2564	2565	2588	2628	2629	2630	2640	2644	2652	2657	2667
	2670	2678	2683	2685	2698	2707	2734	2735	2736	2746	2750	2758	2763	2773	2776
	2784	2789	2791	2804	2813	2840	2841	2842	2856	2860	2868	2673	2883	2886	2894
	2899	2901	2914	2923	2950	2951	2952	2966	2970	2978	2983	2993	2996	3004	3009
	3011	3024	3033	3057	3058	3059	3061	3075	3091	3092	3093	3115	3117	3120	3128
	3145	3146	3150	3151	3162	3172	3174	3179	3183	3199	3213	3214	3215	3228	3240
	3246	3247	3256	3273	3274	3275	3287	3298	3304	3307	3322	3326	3329	3332	3341
	3357	3358	3359	3372	3384	3390	3391	3412	3429	3430	3431	3442	3453	3465	3467
	3468	3472	3480	3500	3501	3502	3514	3526	3532	3534	3545	3557	3566	3568	3569
	3582	3583	3584	3595	3606	3612	3622	3623	3646	3647	3650	3651	3652	3653	3654
	3661	3663	3666	3676	3682	3693	3694	3709							
MOV8	972	977	1008	1070	1071	1072	1078	1079	1101	1103	1104	1105	1106	1107	1108
	1136	1137	1187	1188	1265	1281	1309	1310	1395	1396	1399	1404	1414	1419	1451
NOP	1479	1512	1774	1775	1916	2014	2136	2181	2199	2426	2427	2490	2606		
	1158	1159	1194	1195	1196	1197	1564	1736	3071	3072	3073	3103	3104	3105	3109
	3110	3111													
RESET	1191	1571	3060												
ROL	3697														
ROLB	1076	1622	1630												
ROR	1406	1408	1410	1717	1723	1733	2520	2529	3696						
RORB	1081	1085	3675												
RTI	1250	1268	1298	1350	1370	1380	1431	1453	1544	1560	1779	1783	3125	3129	3337
	3342	3466	3485	3562	3567										
RTS	1565	1586	1706	1725	1740	1756	1770	1776	1805	1814	2544	3710			
SEC	1075														
SUB	1464	1488													
TRAP	844	846	848	850	852	854	856	858	860	862	864	866			
TST	986	1120	1124	1153	1247	1500	1504	1513	1518	1522	1529	1672	1696	2348	2349
	2411	2497	2498	2499	2706	2812	2922	3032							
TSTB	993	1000	1054	1064	1067	1220	1226	1260	1263	1279	1285	1417	1477	1498	1614
	1792	1794	1800	1808	1964	2066	2092	2135	2180	2198	2341	2352	2396	2399	2402
	2409	2466	2474	2482	2487	2603	2631	2737	2843	2846	2848	2953	2956	2958	3112
	3147	3241	3244	3299	3302	3385	3388	3399	3527	3530	3603	3607	3610	3699	3706
.ASCIZ	688	1590	3713												
.BLKB	743	744	745	746	747	814	815								
.BLKW	820	821	904	905	906	908	909	910	912	913	914	916	917	918	920
	921	922	924	925	926	928	929	930	932	933	934	1603	2457	2458	
.BYTE	828	829	830	831	893	895	1019	1020	1027	1028	1035	1036	1044	1045	1201
	1204	1207	1210	1546	1549	1591	1593	1683	1684	3067	3099	3124	3158	3178	3222
	3282	3336	3365	3437	3484	3509	3561	3591	3713						
.ENABL	1														
.END	3713														
.ENDC	1817	1820	1821	1822	1831	1858	1865	1868	1869	1878	1929	1936	1938	1939	1940
	1948	1956	1965	1980	1983	1986	1987	1988	1996	2006	2018	2033	2035	2036	2040
	2041	2049	2058	2067	2108	2111	2114	2115	2123	2153	2155	2158	2159	2167	2217
	2219	2221	2222	2231	2255	2258	2262	2263	2272	2310	2313	2315	2316	2324	2335
	2372	2375	2376	2385	2409	2434	2436	2437	2446	2470	2553	2556	2557	2566	2615
	2620	2621	2630	2634	2652	2665	2678	2693	2706	2721	2726	2727	2736	2740	2758
	2771	2784	2799	2812	2827	2832	2833	2842	2850	2868	2881	2894	2909	2922	2937
	2942	2943	2952	2960	2978	2991	3004	3019	3032	3046	3049	3050	3059	3080	3083
	3084	3093	3132	3137	3138	3147	3170	3191	3202	3205	3206	3215	3236	3255	3262
	3265	3266	3275	3295	3315	3322	3345	3349	3350	3359	3380	3399	3411	3418	3421
	3422	3431	3450	3461	3480	3489	3492	3493	3502	3522	3542	3553	3572	3574	3575



	3584	3603	3620	3629	3639	3640	3648	3713							
.EQUIV	631														
.EVEN	748	834	1590	1595	3713										
.IF	1816	1817	1820	1830	1841	1864	1865	1868	1877	1889	1935	1936	1938	1939	1947
	1956	1964	1972	1982	1983	1985	1987	1995	2004	2014	2025	2035	2036	2040	2048
	2058	2066	2074	2110	2111	2114	2122	2139	2154	2155	2158	2156	2184	2218	2219
	2221	2230	2239	2257	2258	2262	2271	2281	2312	2313	2315	2323	2335	2371	2372
	2375	2384	2405	2433	2434	2436	2445	2470	2552	2553	2556	2565	2614	2615	2620
	2629	2634	2644	2657	2670	2685	2698	2720	2721	2726	2735	2740	2750	2763	2776
	2791	2804	2826	2827	2832	2841	2846	2860	2873	2886	2901	2914	2936	2937	2942
	2951	2956	2970	2983	2996	3011	3024	3045	3046	3049	3058	3079	3080	3083	3092
	3131	3132	3137	3146	3162	3183	3201	3202	3205	3214	3228	3247	3261	3262	3265
	3274	3287	3307	3318	3344	3345	3349	3358	3372	3391	3407	3417	3418	3421	3430
	3442	3453	3472	3488	3489	3492	3501	3514	3534	3545	3571	3572	3574	3583	3595
	3612	3628	3629	3639	3647	3713									
.IFF	1822	1830	1831	1858	1864	1877	1878	1889	1940	1947	1948	1964	1980	1988	1995
	1996	2018	2033	2035	2048	2049	2066	2074	2115	2122	2123	2153	2154	2166	2167
	2184	2222	2230	2231	2255	2257	2271	2272	2281	2316	2323	2324	2384	2385	2409
	2445	2446	2565	2566	2629	2630	2644	2657	2670	2685	2698	2735	2736	2750	2763
	2776	2791	2804	2841	2842	2860	2873	2886	2901	2914	2951	2952	2970	2983	2996
	3011	3024	3058	3059	3092	3093	3146	3147	3162	3183	3214	3215	3228	3247	3274
	3275	3287	3307	3322	3358	3359	3372	3391	3411	3430	3431	3442	3453	3472	3501
	3502	3514	3534	3545	3583	3584	3595	3612	3647						
.IFT	1816	1841	1864	1889	1935	1938	1956	1964	1972	1982	1985	2004	2014	2025	2035
	2058	2066	2074	2110	2139	2154	2184	2218	2239	2257	2281	2312	2335	2405	2470
	2634	2644	2657	2670	2685	2698	2740	2750	2763	2776	2791	2804	2846	2860	2873
	2886	2901	2914	2956	2970	2983	2996	3011	3024	3162	3183	3228	3247	3287	3307
	3318	3372	3391	3407	3442	3453	3472	3514	3534	3545	3595	3612			
.IIF	1830	1831	1836	1877	1878	1883	1947	1948	1953	1956	1957	1962	1963	1995	1996
	2001	2006	2007	2012	2013	2048	2049	2054	2058	2064	2122	2123	2127	2130	2131
	2166	2167	2171	2175	2230	2231	2236	2271	2272	2277	2323	2324	2329	2330	2384
	2385	2445	2446	2463	2565	2566	2629	2630	2735	2736	2841	2842	2951	2952	3058
	3059	3092	3093	3146	3147	3214	3215	3274	3275	3358	3359	3430	3431	3501	3502
	3583	3584	3647	3648											
.IRP	844	846	848	850	852	854	856	858	860	862	864	866	904	1816	1823
	1830	1864	1870	1877	1935	1940	1947	1982	1988	1995	2035	2041	2048	2110	2115
	2122	2154	2159	2166	2218	2223	2230	2257	2264	2271	2312	2316	2323	2371	2377
	2384	2433	2438	2445	2552	2558	2565	2614	2622	2629	2720	2728	2735	2826	2834
	2841	2936	2944	2951	3045	3051	3058	3079	3085	3092	3131	3139	3146	3201	3207
	3214	3261	3267	3274	3344	3351	3358	3417	3423	3430	3488	3494	3501	3571	3576
	3583	3628	3640	3647	3713										
.LIST	1	568	581	602	650	652	662	686	688	846	848	850	852	854	856
	858	860	862	864	866	868	942	961	1110	1166	1590	1707	1816	1820	1831
	1864	1868	1878	1935	1939	1948	1982	1987	1996	2035	2040	2049	2110	2114	2123
	2154	2158	2167	2218	2221	2231	2257	2262	2272	2312	2315	2324	2370	2371	2375
	2385	2432	2433	2436	2446	2551	2552	2556	2566	2613	2614	2620	2631	2719	2720
	2726	2737	2825	2826	2832	2843	2935	2936	2942	2953	3045	3049	3059	3079	3083
	3093	3131	3137	3147	3201	3205	3215	3261	3265	3275	3344	3349	3359	3417	3421
	3431	3487	3488	3492	3502	3571	3574	3584	3627	3628	3639	3648	3713		
.MACRO	1	568	581	602	650	652	662	686	688	846	848	850	852	854	856
.NLIST	1	568	581	602	650	652	662	686	688	846	848	850	852	854	856
	858	860	862	864	866	868	942	961	1110	1166	1590	1707	1816	1820	1831
	1864	1868	1878	1935	1939	1948	1982	1987	1996	2035	2040	2049	2110	2114	2123
	2154	2158	2167	2218	2221	2231	2257	2262	2272	2312	2315	2324	2370	2371	2375
	2385	2432	2433	2436	2446	2551	2552	2556	2566	2613	2614	2620	2631	2719	2720
	2726	2737	2825	2826	2832	2843	2935	2936	2942	2953	3045	3049	3059	3079	3083

DZDPCA MACY11 27(732) 21-OCT-76 15:43 PAGE 102  
 DZDPCA.CMB CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	3093	3131	3137	3147	3201	3205	3215	3261	3265	3275	3344	3349	3359	3417	3421
	3431	3487	3488	3492	3502	3571	3574	3594	3627	3628	3639	3648	3713		
.PAGE	1	568	602	652	750	824	871	899	942	961	1166	1605	1707	3713	
.REM	1	942													
.REPT	662														
.SBTTL	568	602	652	686	961	1166	1816	1864	1935	1982	2035	2110	2154	2218	2257
	2312	2370	2432	2551	2613	2719	2825	2935	3045	3079	3131	3201	3261	3344	3417
	3487	3571	3627												
.TITLE	581														
.WORD	813	816	817	818	819	2365	2366	2367	2368	2369	2505	2506	2507	2508	2509
	2714	2715	2716	2717	2820	2821	2822	2823	2930	2931	2932	2933	3040	3041	3042
	3043	3671	3672												

ERRORS DETECTED: 0  
 DEFAULT GLOBALS GENERATED: 0

\*,DZDPCA.SEQ/SOL/CRF/PAGNUM/NL:TOC=DZDPCX.MAC.DZDPCA.CMB  
 RUN-TIME: 32 45 7 SECONDS  
 RUN-TIME RATIO: 164/84=1.9  
 CORE USED: 27K (53 PAGES)



