

# DUP11

DATA&FUNC&DECMODE TEST  
MD-11-DZDPD-A

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

NOV 1976  
**digital**  
MADE IN USA

This microfiche card contains a grid of frames. The left side of the card features a vertical column of frames, each containing a small table or data set. The right side of the card is mostly blank, with a few frames containing data or test results. The frames are arranged in a regular grid pattern, typical of microfiche cards used for data storage and retrieval.







000000  
000001  
000002  
000003  
000004  
000005  
000006  
000007  
  
177776  
001150  
  
005746  
005726  
010046  
012600  
024646  
022626  
  
100000  
040000  
020000  
010000  
004000  
002000  
001000  
000400  
000200  
000100  
000040  
000020  
000010  
000004  
000002  
000001

:REGISTER DEFINITIONS  
:-----

000000 R0=%0 :GENERAL REGISTER  
 000001 R1=%1 :GENERAL REGISTER  
 000002 R2=%2 :GENERAL REGISTER  
 000003 R3=%3 :GENERAL REGISTER  
 000004 R4=%4 :GENERAL REGISTER  
 000005 R5=%5 :GENERAL REGISTER  
 000006 SP=%6 :PROCESSOR STACK POINTER  
 000007 PC=%7 :PROGRAM COUNTER

:LOCATION EQUIVALENCIES  
:-----

177776 PS=177776 :PROCESSOR STATUS WORD  
 001150 STACK=1150 :START OF PROCESSOR STACK

:INSTRUCTION DEFINITIONS  
:-----

005746 PUSH1SP=5746 :DECREMENT PROCESSOR STACK 1 WORD  
 005726 POP1SP=5726 :INCREMENT PROCESSOR STACK 1 WORD  
 010046 PUSHRO=10046 :SAVE R0 ON STACK  
 012600 POPRO=12600 :RESTORE R0 FROM STACK  
 024646 PUSH2SP=24646 :DECREMENT STACK TWICE  
 022626 POP2SP=22626 :INCREMENT STACK TWICE  
 .EQUIV EMT,HLT :BASIC DEFINITION OF ERROR CALL

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



095  
096  
097  
098  
099  
100  
101  
102  
103  
104  
105  
106  
107  
108  
109  
110  
111  
112  
113  
114  
115  
116  
117  
118  
119  
120  
121  
(2)  
122  
123  
124  
125  
126  
127  
128  
129  
130  
131  
132  
133  
134  
135  
136  
137  
138  
139

```

:*****
:-----
: TRAPCATCHER FOR ILLEGAL INTERRUPTS
: THE STANDARD "TRAP CATCHER" IS PLACED
: BETWEEN ADDRESS 0 TO ADDRESS 776.
: IT LOOKS LIKE "PC+2 HALT".
:-----
:*****

      000000      .=0
:-----
: STANDARD INTERRUPT VECTORS
:-----

      000024      .=24
000024 004776      .PFAIL          ;POWER FAIL HANDLER
000026 000340      340             ;SERVICE AT LEVEL 7
000030 004350      .HLT           ;ERROR HANDLER
000032 000340      340             ;SERVICE AT LEVEL 7
000034 004316      .TRPSRV        ;GENERAL HANDLER DISPATCH SERVICE
000036 000340      340             ;SERVICE AT LEVEL 7

      000040      .=40
000040 000000      0               ;SAVE FOR ACT-11 OR DDP2
000042 000000      0               ;RETURN ADDRESS IF UNDER ACT-11 OR DDP2
000044 000000      0               ;SAVE FOR ACT-11 OR DDP2
000046 003104      $ENDAD         ;FOR USE WITH ACT-11 OR DDP2

      000052      .=52
000052 000000      0               ;ACT-11 PROGRAM CHARACTERISTICS

      000174      .=174
000174 000000      DISPREG:0       ;SOFTWARE DISPLAY REGISTER
000176 000000      SWREG: 0        ;SOFTWARE SWITCH REGISTER

      000200      .=200
000200 000137 001562      JMP      .START      ;GO TO START OF PROGRAM

      001000      .=1000
001000 005377 040515 047111      MTITLE: .ASCIZ <377><12>/MAINDEC-11-DZDPD-A /<377>/DUP-11 OFFLINE SDLC AND DEC MODE DAT

      001200      .=1200
:-----
: SWR AND LIGHTS
:-----

001200 177570      DISPLAY: 177570      ;11/45 CONSOLE LIGHTS
001202 177570      SWR: 177570         ;INDIRECT POINTER TO SWITCH REGISTER

:-----
: INDIRECT POINTERS TO TELETYPE VECTORS AND REGISTERS
:-----

001204 177560      TKCSR: 177560       ;TELETYPE KEYBOARD CONTROL REGISTER
001206 177562      TKDBR: 177562       ;TELETYPE KEYBOARD DATA BUFFER
001210 177564      TPCSR: 177564       ;TELEPRINTER CONTROL REGISTER
001212 177566      TPDBR: 177566       ;TELEPRINTER DATA BUFFER

:-----
: PROGRAM CONTROL PARAMETERS
:-----

```



140	001214	000000	RETURN:	0	: SCOPE ADDRESS FOR LOOP ON TEST
141	001216	000000	NEXT:	0	: ADDRESS OF NEXT TEST TO BE EXECUTED
142	001220	000000	LOCK:	0	: ADDRESS FOR LOCK ON CURRENT DATA
143	001222	000001	ICOUNT:	1	: NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
144	001224	000000	LPCNT:	0	: NUMBER OF ITERATIONS COMPLETED
145	001226	000000	TSTNO:	0	: NUMBER OF TEST IN PROGRESS
146	001230	000000	PASCNT:	0	: NUMBER OF PASSES COMPLETED
147	001232	000000	ERRCNT:	0	: TOTAL NUMBER OF ERRORS
148	001234	000000	LSTERR:	0	: PC OF LAST ERROR CALL
149					
150					
151					
152					
153	001236	000000	TEMP1:	0	: TEMPORARY STORAGE
154	001240	000000	TEMP2:	0	: TEMPORARY STORAGE
155	001242	000000	TEMP3:	0	: TEMPORARY STORAGE
156	001244	000000	TEMP4:	0	: TEMPORARY STORAGE
157	001246	000000	TEMP5:	0	: TEMPORARY STORAGE
158	001250	000000	SAVR0:	0	: R0 STORAGE
159	001252	000000	SAVR1:	0	: R1 STORAGE
160	001254	000000	SAVR2:	0	: R2 STORAGE
161	001256	000000	SAVR3:	0	: R3 STORAGE
162	001260	000000	SAVR4:	0	: R4 STORAGE
163	001262	000000	SAVR5:	0	: R5 STORAGE
164	001264	000000	SAVSP:	0	: STACK POINTER STORAGE
165	001266	000000	SAVPC:	0	: PROGRAM COUNTER STORAGE
166					
167	001270	000000	SAVR0A:	0	: R0 STORAGE
168	001272	000000	SAVR1A:	0	: R1 STORAGE
169	001274	000000	SAVR2A:	0	: R2 STORAGE
170	001276	000000	SAVR3A:	0	: R3 STORAGE
171	001300	000000	SAVR4A:	0	: R4 STORAGE
172	001302	000000	SAVR5A:	0	: R5 STORAGE
173	001304	000000	SAVSPA:	0	: STACK POINTER STORAGE
174	001306	000000	SAVPCA:	0	: PROGRAM COUNTER STORAGE
175					
176	001310	000001	DUPACTV:	.BLKB 1	: DUPII'S SELECTED ACTIVE.
177	001311	000001	DUPNUM:	.BLKB 1	: OCTAL NUMBER OF DUPII'S.
178	001312	000001	SAVACT:	.BLKB 1	: ORIGINAL ACTV. DEVICES.
179	001313	000001	SAVNUM:	.BLKB 1	: WORKABLE NUMBER.
180	001314	000001	RUN:	.BLKB 1	: POINTER ONE PAST RUNNING DEVICE.
181		001316	.EVEN		
182	001316	001500	CREAM:	DUP.MAP	: TABLE POINTER.



# F01

DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 6  
 DZDPDA.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

183
184                                     ;CONTROL REGISTER DEFINITIONS
185                                     -----
186                                     ;RXCSR BIT DEFINITIONS
187         100000      DSCA=BIT15      ;DATA SET CHANGE A
188         040000      RING=BIT14     ;RING
189         020000      CTS=BIT13      ;CLR TO SEND
190         010000      CARDET=BIT12   ;CARRIER DETECT
191         004000      RECACT=BIT11   ;REC ACTIVE
192         002000      SRD=BIT10      ;SEC REC DATA
193         001000      DSR=BIT9       ;DATA SET RDY
194         000400      STPSYN=BIT8    ;STRIP SYNC
195         000200      RXDONE=BIT7    ;REC DONE
196         000100      RINTEN=BIT6    ;REC INTR ENABLE
197         000040      DSINTE=BIT5    ;DSC INTR ENABLE
198         000020      RCVEN=BIT4     ;REC ENABLE
199         000010      STD=BIT3       ;SEC XMIT DATA
200         000004      RTS=BIT2       ;REQ TO SEND
201         000002      DTR=BIT1      ;DATA TERM RDY
202         000001      DSCB=BIT0     ;DATA SET CHANGE B
203                                     ;RXDBUF BIT DEFINITIONS
204         100000      RXDERR=BIT15   ;REC DATA ERROR
205         040000      OVRUN=BIT14   ;OVERRUN ERROR
206         010000      CRCERR=BIT12  ;CRC ERROR
207         002000      RABORT=BIT10  ;REC ABORT
208         001000      REOM=BIT9     ;REC END OF MESSAGE
209         000400      RSOM=BIT8     ;REC START OF MESSAGE
210                                     ;PARCSR BIT DEFINITIONS
211         100000      DECMOD=BIT15   ;DEC MODE (DDCMP)
212         001000      CRCEN=BIT9     ;CRC ENABLE
213         010000      PRISEC=BIT12  ;PRI/SEC SELECT
214                                     ;TXCSR BIT DEFINITIONS
215         100000      TXDLAT=BIT15   ;TX DATA LATE
216         040000      MTDATA=BIT14  ;MAINT DATA OUT
217         020000      CLK=BIT13     ;CLK
218         010000      MMODEB=BIT12  ;MAINT MODE B
219         004000      MMODEA=BIT11  ;MAINT MODE A
220         002000      BITW=BIT10   ;BIT WINDOW INPUT
221         001000      TXACT=BIT9    ;TX ACTIVE
222         000400      MRESET=BIT8   ;MASTER RESET
223         000200      TXDONE=BIT7   ;XMIT DONE
224         000100      TXINTE=BIT6  ;XMIT DONE INTR ENABLE
225         000020      SEND=BIT4    ;SEND
226         000010      HDXEN=BIT3    ;HDX/FDX
227                                     ;TXCSR WRD DEFINITIONS
228         000000      USER=0        ;USER MODE
229         014000      MMODE=14000   ;MAINT INT MODE
230         010000      MEXT=10000    ;MAINT EXT MODE
231         004000      SYSTST=4000   ;SYSTEM TEST MODE
232
233                                     ;TXDBUF BIT DEFINITIONS
234                                     -----
235         100000      RCRC7T=BIT15
236         040000      RCRCIN=BIT14
237         020000      TCRC7T=BIT13
238         010000      TCRCIN=BIT12
  
```



239	004000
240	002000
241	001000
242	000400
243	
244	
245	
246	
247	001320 000000
248	001322 000001
249	001323 000001
250	001324 000000
251	001326 000000
252	001330 000000
253	001332 000000
254	001334 000001
255	001336 000001
256	

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

;MISC. PROGRAM DEFINITIONS

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



# H01

DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 8  
DZDPDA.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

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  
282  
283  
284  
285  
286  
287  
288  
289  
290  
291  
292  
293  
294  
295  
296  
297  
298  
299  
300  
301  
302  
303

## ;PROGRAM CONTROL FLAGS

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

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

\*\*\*\*\*

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

\*\*\*\*\*



DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 9  
 DZDPDA.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

```

304                                     ;DUP11 VECTOR AND REGISTER INDIRECT POINTERS
305
306 001374 000000 DUPRVC: 0 ; POINTER TO DUP11 RECEIVER INTERRUPT VECTOR
307 001376 000000 DUPRPS: 0 ; POINTER TO DUP11 RECEIVER INTERRUPT SERVICE PS
308 001400 000000 DUPTVC: 0 ; POINTER TO DUP11 TRANSMITTER INTERRUPT VECTOR
309 001402 000000 DUPTPS: 0 ; POINTER TO DUP11 TRANSMITTER INTERRUPT SERVICE PS
310 001404 000000 RXCSR: 0 ; POINTER TO DUP11 RECEIVER STATUS REGISTER
311 001406 000000 RXDBUF: 0 ; POINTER TO DUP11 RECEIVER DATA BUFFER
312 001410 000000 PARCSR: 0 ; POINTER TO DUP11 PARAMETER STATUS REGISTER
313 001412 000000 TXCSR: 0 ; POINTER TO DUP11 TRANSMITTER STATUS REGISTER
314 001414 000000 TXDBUF: 0 ; POINTER TO DUP11 TRANSMITTER DATA BUFFER
315 001416 000000 DUPSEC: 0 ; POINTER TO DUP11 SECONDARY REGISTER SELECT REGISTER
316 001420 000000 HUPPSR: 0 ; POINTER TO PARAMETER STATUS HIGH BYTE
317 001422 000000 HUPRBF: 0 ; POINTER TO RECEIVER BUFFER HIGH BYTE
318 001424 000000 HUPRCR: 0 ; POINTER TO RECEIVER CONTROL REG HIGH BYTE
319 001426 000000 HUPTBF: 0 ; POINTER TO TRANSMITTER BUFFER HIGH BYTE
320 001430 000000 HUPTCR: 0 ; POINTER TO TRANSMITTER CONTROL REG HIGH BYTE
321
322
323                                     ;DUP11 CONTROL INDICATORS FOR CURRENT DUP11 UNDER TEST
324 -----
325
326 001432 000 MASK.A: .BYTE 000 ; LAST CHAR TO TEST AND PARITY MASK
327
328 001433 010 CLK.A: .BYTE 8. ; NUMBER OF CLOCKS NEEDED FOR ONE CHAR
329
330 001434 000000 LOO.OO: 000000 ; PARAMETERS
331

```



-----  
 ; DUP11 STATUS TABLE AND ADDRESS ASSIGNMENTS  
 -----

```

332
333
334
335      001500      001500      .=1500
336      001500      000001      DUP.MAP:
337      001500      000001      DUPCR0: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 0
338      001502      000001      DUPTR0: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 0
339      001504      000001      DUPO.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 0
340
341      001506      000001      DUPCR1: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 1
342      001510      000001      DUPTR1: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 1
343      001512      000001      DUP1.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 1
344
345      001514      000001      DUPCR2: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 2
346      001516      000001      DUPTR2: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 2
347      001520      000001      DUP2.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 2
348
349      001522      000001      DUPCR3: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 3
350      001524      000001      DUPTR3: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 3
351      001526      000001      DUP3.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 3
352
353      001530      000001      DUPCR4: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 4
354      001532      000001      DUPTR4: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 4
355      001534      000001      DUP4.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 4
356
357      001536      000001      DUPCR5: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 5
358      001540      000001      DUPTR5: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 5
359      001542      000001      DUP5.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 5
360
361      001544      000001      DUPCR6: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 6
362      001546      000001      DUPTR6: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 6
363      001550      000001      DUP6.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 6
364
365      001552      000001      DUPCR7: .BLKW 1      ;CONTROL STATUS REGISTER FOR DUP11 NUMBER 7
366      001554      000001      DUPTR7: .BLKW 1      ;VECTOR "A" FOR DUP11 NUMBER 7
367      001556      000001      DUP7.A: .BLKW 1      ;PARAMETER FOR DUP11 NUMBER 7
368
369      001560      000000      DUP.END:      000000
370
371
372
373
374
  
```



# K01

DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 11  
DZDPDA.P11 PROGRAM PARAMETERS, VARIABLES, AND TRAP CALLS.

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

## DEFINITIONS

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



```

394
395
396
397
398
399
400
401
402 001562 012737 000340 177776 .START: MOV #340,PS ;LOCK OUT INTERRUPTS
403 001570 012706 001150 MOV #STACK,SP ;SET UP STACK
404 001574 012737 004776 000024 MOV #.PFAIL,2#24 ;SET UP POWER FAIL VECTOR
405 001602 113737 001311 001313 MOV# DUPNUM,SAVNUM ;SAVE NUMBER OF DEVICES IN SYSTEM
406 001610 005037 001230 CLR PASCNT ;CLEAR PASS COUNT
407 001614 105037 001341 CLRB ERRFLG ;CLEAR ERROR FLAG
408 001620 105037 001343 CLRB QV.FLG ;ZERO QUICK VERIFY FLAG
409 001624 012737 001500 001316 MOV #DUP.MAP,CREAM ;GET MAP POINTER.
410 001632 112737 000001 001314 MOV# #1,RUN ;POINT POINTER TO FIRST DEVICE.
411 001640 005037 001232 CLR ERRCNT ;CLEAR ERROR COUNT
412 001644 005037 001234 CLR LSTERR ;CLEAR LAST ERROR POINTER
413 001650 012737 000001 001226 MOV #1,TSTNO ;SET UP FOR TEST 1
414 001656 012737 001562 001214 MOV #.START,RETURN ;SET UP FOR POWER FAIL BEFORE
415 ;TESTING STARTS
416 001664 013746 000006 MOV 2#6,-(SP) ;SAVE CURRENT VECTORS
417 001670 013746 000004 MOV 2#4,-(SP)
418 001674 012737 001710 000004 MOV #12$,2#4 ;SETUP FOR TIMEOUT
419 001702 005777 177274 TST 2SWR ;REFERENCE HARDWARE SWITCH REG
420 001706 000407 BR 13$ ;BR IF IT EXISTS
421 001710 012737 000176 001202 12$: MOV #SWREG,SWR ;POINT TO SOFT SWR
422 001716 012737 000174 001200 MOV #DISPREG,DISPLAY ;POINT TO SOFT DISPLAY REG
423 001724 022626 CMP (SP)+,(SP)+ ;ADJUST STACK
424 001726 012637 000004 13$: MOV (SP)+,2#4 ;RESTORE VECTORS
425 001732 012637 000006 MOV (SP)+,2#6
426 001736 105737 001340 TSTB INIFLG ;HAS INITIALIZATION BEEN PERFORMED
427 001742 001401 BEQ 11$
428 001744 000410 BR 6$
429 001746 022737 003104 000042 11$: CMP #SENDAD,2#42 ;IF ACT-11 AUTO MODE,
430 001754 001404 BEQ 6$ ;DON'T TYPE ID
431 001756 104402 001000 TYPE #MTITLE ;TYPE TITLE MESSAGE
432 001762 105137 001340 COMB INIFLG ;IF NOT SET FLAG AND DO
433 001766 105777 177210 6$: TSTB 2SWR ;BIT7=1??
434 001772 100002 BPL 10$
435 001774 000137 002520 JMP 1$
436 002000 10$:
437 002000 032777 000001 177174 BIT #SW00,2SWR ;ENTER PARAMETERS
438 002006 001002 BNE +6 ;YES
439 002010 000137 002360 JMP 21$ ;NO
440 002014 105137 001332 COMB FLAG
441 002020 112737 000001 001340 MOV# #1,INIFLG ;SET TO MANUAL ENTRY
442 002026 012700 001500 MOV #DUP.MAP,RO ;CLR MAP
443 002032 005020 68$: CLR (RO)+
444 002034 020027 001560 CMP RO,#DUP.END ;DONE WITH MAP?
445 002040 001374 BNE 68$ ;BR IF NO
446 002042 104403 INSTR ;OUTPUT MESSAGE & GET INPUT STRING
447 002044 005421 MCSR ;MESSAGE
448 002046 104405 PARAM ;CONVERT STRING
449 002050 160000 160000 ;LOW LIMIT

```



# MO1

DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 13  
 DZDPDA.P11 PROGRAM INITIALIZATION AND START UP.

450	002052	175500				175500	:HIGH LIMIT
451	002054	001500				DUPCRD	:STORE AT THIS LOCATION
452	002056	001			.BYTE	1	:MASK
453	002057	001			.BYTE	1	:HOW MANY TIMES + 2
454	002060	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
455	002062	005440				MVEC	:MESSAGE
456	002064	104405				PARAM	:CONVERT STRING
457	002066	000300				300	:LOW LIMIT
458	002070	000770				770	:HIGH LIMIT
459	002072	001502				DUPTRD	:STORE AT THIS LOCATION
460	002074	001			.BYTE	1	:MASK
461	002075	001			.BYTE	1	:HOW MANY TIMES + 2
462	002076	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
463	002100	005630				MPAR	:MESSAGE
464	002102	104405				PARAM	:CONVERT STRING
465	002104	000004				4	:LOW LIMIT
466	002106	000007				7	:HIGH LIMIT
467	002110	001240				TEMP2	:STORE AT THIS LOCATION
468	002112	000			.BYTE	0	:MASK
469	002113	001			.BYTE	1	:HOW MANY TIMES + 2
470	002114	013737	001240	001320		MOV	TEMP2,PRIITY :SAVE PRIORITY
471	002122	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
472	002124	005575				MTOTAL	:MESSAGE
473	002126	104405				PARAM	:CONVERT STRING
474	002130	000001				1	:LOW LIMIT
475	002132	000010				8	:HIGH LIMIT
476	002134	001236				TEMP1	:STORE AT THIS LOCATION
477	002136	000			.BYTE	0	:MASK
478	002137	001			.BYTE	1	:HOW MANY TIMES + 2
479	002140	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
480	002142	005453				MJMPR	:MESSAGE
481	002144	104413				SETFLG	:SET FLAG BASED UPON INPUT STRING
482	002146	001323				OPCLRJ	:THIS FLAG
483	002150	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
484	002152	005526				MTCN	:MESSAGE
485	002154	104413				SETFLG	:SET FLAG BASED UPON INPUT STRING
486	002156	001322				TCNFLG	:THIS FLAG
487	002160	105737	001322			TSTB	TCNFLG
488	002164	001410				BEQ	71\$
489	002166	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
490	002170	005654				MSTJM	:MESSAGE
491	002172	104413				SETFLG	:SET FLAG BASED UPON INPUT STRING
492	002174	001334				STJMFL	:THIS FLAG
493	002176	104403				INSTR	:OUTPUT MESSAGE & GET INPUT STRING
494	002200	005707				MSRJM	:MESSAGE
495	002202	104413				SETFLG	:SET FLAG BASED UPON INPUT STRING
496	002204	001336				SRJMFL	:THIS FLAG
497	002206	105737	001323		71\$:	TSTB	OPCLRJ
498	002212	001403				BEQ	69\$
499	002214	052737	100000	001504		BIS	#BIT15,DUPO.A
500	002222	105737	001322		69\$:	TSTB	TCNFLG
501	002226	001403				BEQ	70\$
502	002230	052737	040000	001504		BIS	#BIT14,DUPO.A
503	002236	112737	000001	001312	70\$:	MOVB	#1,SAVACT
504	002244	113737	001236	001311		MOVB	TEMP1,DUPNUM
505	002252	113737	001236	001313		MOVB	TEMP1,SAVNUM



506	002260	005337	001236		65\$:	DEC	TEMP1		
507	002264	001404				BEQ	64\$		
508	002266	000261				SEC			
509	002270	106137	001312			ROLB	SAVACT		
510	002274	000771				BR	65\$		
511	002276	113737	001312	001240	64\$:	MOVB	SAVACT,TEMP2	;	# OF TIMES
512	002304	113737	001312	001310		MOVB	SAVACT,DUPACTV		
513	002312	000241				CLC			
514	002314	106037	001240			RORB	TEMP2		
515	002320	012700	001500			MOV	#DUPCRO,RO		
516	002324	012701	001506			MOV	#DUPCRI,R1		
517	002330	000241			67\$:	CLC			
518	002332	106037	001240			RORB	TEMP2		
519	002336	103051				BCC	66\$		
520	002340	012011				MOV	(RO)+(R1)		
521	002342	062721	000010			ADD	#10,(R1)+	;	CSR
522	002346	012011				MOV	(RO)+(R1)		
523	002350	062721	000010			ADD	#10,(R1)+	;	VECTOR
524	002354	012021				MOV	(RO)+(R1)+	;	PARAMETERS
525	002356	000764				BR	67\$		
526	002360	012700	001500		21\$:	MOV	#DUP.MAP,RO	;	SETUP TO CLEAR MAP
527	002364	005020			20\$:	CLR	(RO)+	;	CLEAR
528	002366	020027	001560			CMP	RO,#DUP.END	;	CHECK FOR FINISH
529	002372	001374				BNE	20\$	;	BR IF MORE TO GO
530	002374	012700	001500			MOV	#DUP.MAP,RO	;	SETUP TO DEFAULT
531	002400	012710	160050			MOV	#160050,(RO)	;	LOAD CSR
532	002404	012760	000770	000002		MOV	#770,2(RO)	;	LOAD VECTOR
533	002412	012760	140026	000004		MOV	#140026,4(RO)	;	LOAD PARAMETERS AND SYNC
534	002420	112737	000005	001320		MOVB	#5,PRTY	;	LOAD PRIORITY
535	002426	012700	000001			MOV	#1,RO	;	SAVE CORE THIS WAY
536	002432	110037	001310			MOVB	RO,DUPACTV	;	PRESET PROGRAM CONTROLS
537	002436	110037	001311			MOVB	RO,DUPNUM	;	DITTO
538	002442	110037	001312			MOVB	RO,SAVACT	;	DITTO
539	002446	110037	001313			MOVB	RO,SAVNUM	;	DITTO
540	002452	110037	001322			MOVB	RO,TCNFLAG	;	DITTO
541	002456	110037	001323			MOVB	RO,OPCLRJ	;	DITTO
542	002462				66\$:				
543	002462	104402	005742		16\$:	TYPE	,XHEAD	;	TYPE HEADER
544	002466	012737	001500	001236		MOV	#DUP.MAP,TEMP1	;	SET POINTER
545	002474	017737	176536	001240	5\$:	MOV	@TEMP1,TEMP2	;	SET DATA
546	002502	001406				BEQ	1\$	;	ALL DONE WITH DATA
547	002504	104410				CONVRT		;	
548	002506	005770				XSTATQ		;	
549	002510	062737	000002	001236		ADD	#2,TEMP1	;	UPDATE POINTER
550	002516	000766				BR	5\$		
551	002520	032777	000001	176454	1\$:	BIT	#SW00,@SWR		
552	002526	001405				BEQ	7\$		
553	002530	005737	001332			TST	FLAG		
554	002534	001002				BNE	7\$		
555	002536	000137	002000			JMP	10\$		
556	002542	005037	001332		7\$:	CLR	FLAG		
557	002546	005737	000042			TST	@#42	;	IS PROGRAM RUNNING UNDER MONITOR
558	002552	001030				BNE	3\$	;	BR IF YES
559	002554	032777	000010	176420		BIT	#SW03,@SWR	;	SELECT SPECIFIC DEVICES??
560	002562	001424				BEQ	3\$	;	BR IF NO.
561	002564	104402	005341			TYPE	,MNEW	;	TYPE THE MESSAGE.

DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 15  
 DZDPDA.P11 PROGRAM INITIALIZATION AND START UP.

562	002570	005000			CLR	RO		:ZERO DATA LIGHTS
563	002572	000000			HALT			:WAIT FOR USER TO TELL WHAT DEVICES TO RUN
564	002574	127737	176402	001312	CMPB	DSWR, SAVACT		:IS THE NUMBER VALID?
565	002602	101404			BLOS	2\$		:BR IF NUMBER IS OK.
566	002604	104402	005202		TYPE	,MERR3		:TELL USER OF INVALID NUMBER.
567	002610	000000			HALT			:STOP EVERY THING.
568	002612	000776			BR	.-2		:RESTART THE PROGRAM AGAIN.
569	002614	117737	176362	001310	2\$:	MOV B DSWR, DUPACTV		:GET NEW DEVICE PATTERN
570	002622	113700	001310		MOV B	DUPACTV, RO		:SHOW THE USER WHAT HE SELECTED.
571	002626	042700	177400		BIC	#1<377>, RO		:USE ONLY LOW BYTE.
572	002632	000000			HALT			:CONTINUE DYNAMIC SWITCHES.
573	002634	012700	000300		3\$:	MOV #300, RO		:PREPARE TO CLEAR THE FLOATING
574	002640	012701	000302		MOV	#302, R1		:VECTOR AREA. 300-776
575	002644	010120			4\$:	MOV R1, (RO)+		:START PUTTING "PC+2 - HALT"
576	002646	005021			CLR	(R1)+		:IN VECTOR AREA.
577	002650	022021			CMP	(RO)+, (R1)+		:POP POINTERS
578	002652	022700	001000		CMP	#1000, RO		:ALL DONE??
579	002656	001372			BNE	4\$		:BR IF NO.
580								
581								
582								
583								
584	002660	012737	000340	177776	.BEGIN:	MOV #340, PS		:LOCK OUT INTERRUPTS
585	002666	012706	001150		MOV	#STACK, SP		:SET UP STACK
586	002672	005737	000042		TST	DS42		:IS PROGRAM UNDER MONITOR CONTROL
587	002676	001023			BNE	2\$		:BR IF YES
588	002700	032777	000004	176274	BIT	#BIT2, DSWR		:CHECK FOR LOCK ON TEST
589	002706	001411			BEQ	1\$		:BR IF NO LOCK DESIRED.
590	002710	104402	005240		TYPE	, MLOCK		:TYPE LOCK SELECTED.
591	002714	012737	000240	003174	MOV	#NOP, TTST		:ADJUST SCOPE ROUTINE.
592	002722	012737	000240	003176	MOV	#NOP, TTST+2		:SET UP TO LOCK
593	002730	000406			BR	2\$		:CONTINUE ALONG.
594	002732	013737	003306	003174	1\$:	MOV BRW, TTST		:PREPARE NORMAL SCOPE ROUTINE
595	002740	013737	003310	003176	MOV	BRX, TTST+2		:LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
596	002746	012737	006152	001214	2\$:	MOV #CYCLE, RETURN		:START AT "CYCLE" FIND WHICH DEVICE TO TEST
597	002754	104402	005130		TYPE	MR		:TYPE R
598	002760	000177	176230		JMP	RETURN		:START TESTING

:TEST START AND RESTART

-----



```

599                                     ;END OF PASS
600                                     ;TYPE NAME OF TEST
601                                     ;UPDATE PASS COUNT
602                                     ;CHECK FOR EXIT TO ACT-11
603                                     ;RESTART TEST
604
605 002764 005037 001234      .EOP: CLR      LSTERR      ;CLEAR LAST ERROR PC
606 002770 105037 001341      CLRAB   ERRFLG     ;CLEAR ERROR FLAG
607 002774 005237 001230      INC     PASCNT     ;UPDATE PASS COUNT
608 003000 013777 001230 176172 MOV     PASCNT, @DISPLAY ;DISPLAY PASS COUNT
609 003006 104402 005105      TYPE   ,MEPASS    ;TYPE END PASS
610 003012 104402 005267      TYPE   ,MCSRX     ;TYPE CSR
611 003016 104411 003130      CNVRT  ,XCSR      ;SHOW IT
612 003022 104402 005275      TYPE   ,MVECX     ;TYPE VECTOR
613 003026 104411 003136      CNVRT  ,XVEC      ;SHOW IT
614 003032 104402 005303      TYPE   ,MPASSX    ;TYPE PASSES
615 003036 104411 003144      CNVRT  ,XPASS     ;SHOW IT
616 003042 104402 005314      TYPE   ,MERRX     ;TYPE ERRORS
617 003046 104411 003152      CNVRT  ,XERR      ;SHOW IT
618 003052 105337 001313      DECB   $AVNUM     ;ARE ALL DEVICES TESTED?
619 003056 001017              BNE    RESTRT     ;BR IF NO.
620 003060 112737 000377 001343 MOVB   #377, QV.FLG ;SET THE QUICK VERIFY FLAG.
621 003066 113737 001311 001313 MOVB   DUPNUM, SAVNUM ;RESTORE THE COUNT
622 003074 013701 000042      MOV    @#42, R1  ;CHECK FOR ACT-11 OR DDP
623 003100 001406              BEQ    RESTRT     ;IF NOT, CONTINUE TESTING
624 003102 000005              RESET          ;STOP THE SHOW--CLEAR THE WORLD
625 003104
626 003104 004711      SENDAD: JSR    PC, (R1)
627 003106 000240      NOP
628 003110 000240      NOP
629 003112 000240      NOP
630 003114 000240      NOP
631 003116 012737 006152 001214 RESTRT: MOV    #CYCLE, RETURN
632 003124 000137 006152      JMP    CYCLE
633 003130 000001      XCSR:  1
634 003132 006 002      .BYTE  6,2
635 003134 001404      RXCSR
636 003136 000001      XVEC:  1
637 003140 003 002      .BYTE  3,2
638 003142 001374      DUPRVC
639 003144 000001      XPASS: 1
640 003146 006 002      .BYTE  6,2
641 003150 001230      PASCNT
642 003152 000001      XERR:  1
643 003154 006 002      .BYTE  6,2
644 003156 001232      ERRCNT
645
646                                     ;SCOPE LOOP AND INTERATION HANDLER
647
648 003160 005037 001234      .SCOPE: CLR     LSTERR     ;CLEAR LAST ERROR PC
649 003164 010016 040000 176006 MOV     RD, (SP)   ;SAVE RD ON STACK
650 003166 032777 040000 176006 BIT     #BIT14, @SWR ;LOOP ON TEST?
651 003174 001407      TTST: BEQ    1$      ;BR IF NO (IF LOCK SWD1 = 1; THIS LOCATION = 240)
652 003176 000437      BR     3$        ;GO TO 3$ (DITTO)
653 003200 105777 176000      TSTB  @TKCSR     ;KYBD DONE?
654 003204 100034      BPL   3$        ;BR IF NO (LOCK: HIT A KEY ON TTY TO GO TO NEXT TEST)

```

```

655 003206 017700 175774      MOV      @TKDBR,R0      ;CLR DONE BIT
656 003212 000415      BR       2$           ;CONTINUE
657 003214 032777 004000 175760 1$:  BIT      #SW11,@SWR    ;DELETE ITERATION (QUICK PASS)?
658 003222 001011      BNE     2$           ;BR IF YES
659 003224 105737 001343      TSTB   QV.FLG        ;HAS FIRST PASS BEEN COMPLETED?
660 003230 001406      BEQ    2$           ;BR IF QUICK VERIFY
661 003232 005237 001224      INC    LPCNT          ;UPDATE ITERATION COUNTER
662 003236 023737 001224 001222  CMP     LPCNT,ICOUNT  ;ALL ITERATIONS DONE?
663 003244 001014      BNE     3$           ;BR IF NOT YET
664 003246 105037 001341 2$:  CLRB   ERRFLG        ;PREPARE FOR NEW TEST
665 003252 005037 001224      CLR    LPCNT         ;START ICOUNT AT ZERO
666 003256 005037 001220      CLR    LOCK          ;
667 003262 012737 000050 001222  MOV     #50,ICOUNT    ;RESET ITERATIONS
668 003270 013737 001216 001214  MOV     NEXT,RETURN   ;GET NEXT TEST
669 003276 011600      MOV     (SP),R0      ;POP R0 OFF STACK
670 003300 022626      POP2SP              ;FAKE AN RTI
671 003302 000177 175706      JMP     @RETURN      ;GO DO THE TEST
672 003306 001407      BRW:   1407
673 003310 000437      BRX:   437
674
675
676
677
678 003312 032777 001000 175662 .SCOPI: BIT      #SW09,@SWR    ;IS SW09=1(SET)?
679 003320 001405      BEQ    1$           ;BR IF NOT SET.
680 003322 005737 001220      TST   LOCK
681 003326 001402      BEQ    1$
682 003330 013716 001220      MOV   LOCK,(SP)    ;GOTO THE ADDRESS IN LOCK.
683 003334 000002 1$:  RTI              ;GO BACK.
684
685
686
687
688 003336 010546      .TYPE: MOV     R5,-(SP)  ;SAVE R5 ON THE STACK.
689 003340 017605      MOV     @2(SP),R5   ;GET ADDRESS OF MESSAGE.
690 003344 062766 000002 000002  ADD     #2,2(SP)    ;POP OVER ADDRESS.
691 003352 032777 010000 175622 1$:  BIT     #SW12,@SWR  ;INHIBIT ALL PRINT OUT??
692 003360 001012      BNE     3$         ;BR IF NO PRINT OUT WANTED (SW12=1)
693 003362 105715      TSTB   (R5)        ;IS NUMBER MINUS? (MSB=1(BIT7))
694 003364 100002      BPL     2$         ;BR IF NUMBER IS PLUS
695 003366 104402 005064      TYPE   ,MCRLF      ;TYPE A CR/LF!
696 003372 105777 175612 2$:  TSTB   @TPCSR      ;TTY READY?
697 003376 100375      BPL     2$         ;BR IF NO.
698 003400 112577 175606      MOVB   (R5)+,@TPDBR ;PRINT CURRENT CHAR.
699 003404 001362      BNE     1$         ;IF NOT ZERO KEEP PRINTING!
700 003406 012605 3$:  MOV     (SP)+,R5    ;END OF OUTPUT. RESTORE R5
701 003410 000002      RTI              ;GO HOME
702
703
704 003412 010346      .INSTR: MOV     R3,-(SP) ;SAVE R3 ON STACK
705 003414 010446      MOV     R4,-(SP)   ;SAVE R4 ON STACK
706 003416 017637 000004 003434      MOV     @4(SP),MSG
707 003424 062766 000002 000004      ADD     #2,4(SP)
708 003432 104402      .INST1: TYPE
709 003434 000000      .MSG:   0
710 003436 012704 006106      MOV     #INBUF,R4

```



DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 18  
 DZDPDA.P11 END OF PASS ROUTINE

```

711 003442 012703 000007
712 003446 105777 175532
713 003452 100375
714 003454 117714 175526
715 003460 142714 000200
716 003464 122427 000015
717 003470 001417
718 003472 105777 175512
719 003476 100375
720 003500 017777 175502 175504
721 003506 005303
722 003510 001356
723 003512 012604
724 003514 012603
725 003516 010346
726 003520 010446
727 003522 104402 005060
728 003526 000741
729 003530 012604
730 003532 012603
731 003534 000002
732
733
734
735
736 003536 010546
737 003540 010446
738 003542 016605 000004
739 003546 012537 003726
740 003552 012537 003730
741 003556 012537 003732
742 003562 112537 003734
743 003566 112537 003735
744 003572 010566 000004
745 003576 005005
746 003600 012704 006106
747 003604 122714 000015
748 003610 001420
749 003612 121427 000060
750 003616 002415
751 003620 121427 000067
752 003624 003012
753 003626 142714 000060
754 003632 152405
755 003634 122714 000015
756 003640 001406
757 003642 006305
758 003644 006305
759 003646 006305
760 003650 000760
761 003652 104404
762 003654 000750
763
764
765
766

```

```

1$: MOV #7,R3
TSTB @TKCSR
BPL 1$
MOVB @TKDBR,(R4)
BICB #200,(R4)
CMPB (R4)+,#15
BEQ INSTR2
2$: TSTB @TPCSR
BPL 2$
MOV @TKDBR,@TPDBR
DEC R3
BNE 1$
MOV (SP)+,R4
MOV (SP)+,R3
.INSTE: MOV R3,-(SP)
MOV R4,-(SP)
TYPE ,MQM
BR .INST1
INSTR2: MOV (SP)+,R4 ;RESTORE R4
MOV (SP)+,R3 ;RESTORE R3
RTI

;CONVERT ASCII STRING TO OCTAL
-----
.PARAM: MOV R5,-(SP)
MOV R4,-(SP)
MOV 4(SP),R5
MOV (R5)+,LOLIM
MOV (R5)+,HILIM
MOV (R5)+,DEVADR
MOVB (R5)+,LOBITS
MOVB (R5)+,ADRCNT
PARAM1: MOV R5,4(SP)
CLR R5
MOV #INBUF,R4
CMPB #15,(R4)
BEQ PARERR
1$: CMPB (R4),#60
BLT PARERR
CMPB (R4),#67
BGT PARERR
BICB #60,(R4)
BISB (R4)+,R5
CMPB #15,(R4)
BEQ LIMITS
ASL R5
ASL R5
ASL R5
BR 1$
PARERR: INSTER
BR PARAM1

;TEST TO SEE IF NUMBER IS WITHIN LIMITS
-----

```

DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 19  
 DZDPDA.P11 END OF PASS ROUTINE

767	003656	020537	003730	LIMITS:	CMP	R5, HILIM	
768	003662	101373			BHI	PARERR	
769	003664	020537	003726		CMP	R5, LOLIM	
770	003670	103770			BLO	PARERR	
771	003672	133705	003734		BITB	LOBITS, R5	
772	003676	001365			BNE	PARERR	
773							
774							
775							
776	003700	013704	003732				
777	003704	010524		IS:	MOV	DEVADR, R4	
778	003706	062705	000002		MOV	R5, (R4)+	
779	003712	105337	003735		ADD	#2, R5	
780	003716	001372			DECB	ADRCNT	
781	003720	012604			BNE	IS	
782	003722	012605			MOV	(SP)+, R4	
783	003724	000002			MOV	(SP)+, R5	
784	003726	000000			RTI		
785	003730	000000		LOLIM:	0		
786	003732	000000		HILIM:	0		
787	003734	000000		DEVADR:	0		
788		003735		LOBITS:	0		
789				ADRCNT=	LOBITS+1		
790							
791							
792							
793	003736	016637	000004	001266	.SAV05:	MOV	4(SP), SAVPC ;SAVE R7 (PC)
794							
795							
796							
797	003744	010537	001262		SV05:	MOV	R5, SAVR5 ;SAVE R5
798	003750	010437	001260			MOV	R4, SAVR4 ;SAVE R4
799	003754	010337	001256			MOV	R3, SAVR3 ;SAVE R3
800	003760	010237	001254			MOV	R2, SAVR2 ;SAVE R2
801	003764	010137	001252			MOV	R1, SAVR1 ;SAVE R1
802	003770	010037	001250			MOV	R0, SAVR0 ;SAVE R0
803	003774	000002				RTI	;LEAVE.
804							
805							
806							
807	003776	013700	001250		.RES05:	MOV	SAVR0, R0 ;RESTORE R0
808	004002	013701	001252			MOV	SAVR1, R1 ;RESTORE R1
809	004006	013702	001254			MOV	SAVR2, R2 ;RESTORE R2
810	004012	013703	001256			MOV	SAVR3, R3 ;RESTORE R3
811	004016	013704	001260			MOV	SAVR4, R4 ;RESTORE R4
812	004022	013705	001262			MOV	SAVR5, R5 ;RESTORE R5
813	004026	000002				RTI	;LEAVE
814							
815							
816							
817							
818							
819	004030	104402	005064		.CONVR:	TYPE	MCRLF
820	004034	010046			.CNVRT:	MOV	R0, -(SP)
821	004036	010146				MOV	R1, -(SP)
822	004040	010346				MOV	R3, -(SP)

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



823	004042	010446			MOV	R4,-(SP)
824	004044	010546			MOV	R5,-(SP)
825	004046	017601	000012		MOV	2(12(SP),R1
826	004052	062766	000002	000012	ADD	#2,12(SP)
827	004060	012137	004234		MOV	(R1)+,WRDCNT
828	004064	112137	004236	1\$:	MOVB	(R1)+,CHRCNT
829	004070	112137	004237		MOVB	(R1)+,SPACNT
830	004074	013137	004240		MOV	2(R1)+,BINWRD
831	004100	013704	004240	2\$:	MOV	BINWRD,R4
832	004104	113705	004236		MOVB	CHRCNT,R5
833	004110	012700	006002		MOV	#TEMP,R0
834	004114	010403		3\$:	MOV	R4,R3
835	004116	042703	177770		BIC	#177770,R3
836	004122	062703	000060		ADD	#060,R3
837	004126	110320			MOVB	R3,(R0)+
838	004130	000241			CLC	
839	004132	006004			ROR	R4
840	004134	000241			CLC	
841	004136	006004			ROR	R4
842	004140	000241			CLC	
843	004142	006004			ROR	R4
844	004144	005305			DEC	R5
845	004146	001362			BNE	3\$
846	004150	012703	006044		MOV	#MDATA,R3
847	004154	114023		4\$:	MOVB	-(R0),(R3)+
848	004156	105337	004236		DECB	CHRCNT
849	004162	001374			BNE	4\$
850	004164	105737	004237		TSTB	SPACNT
851	004170	001405			BEQ	6\$
852	004172	112723	000040	5\$:	MOVB	#040,(R3)+
853	004176	105337	004237		DECB	SPACNT
854	004202	001373			BNE	5\$
855	004204	105013		6\$:	CLRB	(R3)
856	004206	104402	006044		TYPE	,MDATA
857	004212	005337	004234		DEC	WRDCNT
858	004216	001322			BNE	1\$
859	004220	012605			MOV	(SP)+,R5
860	004222	012604			MOV	(SP)+,R4
861	004224	012603			MOV	(SP)+,R3
862	004226	012601			MOV	(SP)+,R1
863	004230	012600			MOV	(SP)+,R0
864	004232	000002			RTI	
865	004234	000000			WRDCNT:	0
866	004236	000000			CHRCNT:	0
867		004237			SPACNT=	CHRCNT+1
868	004240	000000			BINWRD:	0
869						
870						
871						
872						
873						
874						
875						
876	004242	017605	000000		.SETFLG:MOV	2(SP),R5
877	004246	042737	000040	006106	BIC	#40,INBUF
878	004254	122737	000116	006106	CMPB	#'N,INBUF ; IS IT "N" ?

```

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

```

```

879 004262 001002      BNE      1$
880 004264 105015      CLRB     (R5)      ;000
881 004266 000406      BR      2$
882 004270 122737 000131 006106 1$:  CMPB    #'Y, INBUF      ;IS IT "Y" ?
883 004276 001005      BNE     3$
884 004300 112715 177777      MOVB    #-1, (R5)      ;377
885 004304 062716 000002      2$:  ADD     #2, (SP)
886 004310 000002      RTI
887 004312 104404      3$:  INSTER ;RETRY
888 004314 000752      BR      .SETFLG
889
890
891
892
893
894
895
896 004316 011646      .TRPSR: MOV     (SP), -(SP)      ;GET PC OF RETURN
897 004320 162716 000002      SUB     #2, (SP)      ;=PC OF TRAP
898 004324 017616 000000      MOV     @ (SP), (SP)      ;GET TRP
899 004330 006316      TRPOK: ASL     (SP)      ;MULTIPLY TRAP ARG BY 2
900 004332 042716 177001      BIC     #177001, (SP)      ;CLEAR UNWANTED BITS
901 004336 062716 001344      ADD     #.TRPTAB, (SP)      ;POINTER TO SUBROUTINE ADDRESS
902 004342 017616 000000      MOV     @ (SP), (SP)      ;SUBROUTINE ADDRESS
903 004346 000136      JMP     @ (SP)+          ;GO TO SUBROUTINE
904
905
906
907
908 004350 032777 010000 174624 .HLT:  BIT     #SW12, @SWR      ;BELL ON ERROR?
909 004356 001406      BEQ     XBX              ;BR IF NO BELL
910 004360 105777 174624      TSTB   @TPCSR           ;TTY READY.
911 004364 100003      BPL     XBX              ;DON'T WAIT IF TTY NOT READY.
912 004366 112777 000207 174616      MOVB   #207, @TPDBR     ;PUSH A BELL AT THE TTY.
913 004374 032777 020000 174600 XBX:  BIT     #SW13, @SWR      ;DELETE ERROR PRINT OUT?
914 004402 001105      BNE     HALTS           ;BR IF NO PRINT OUT WANTED.
915 004404 021637 001234      CMP     (SP), LSTERR     ;WAS THIS ERROR FOUND LAST TIME?
916 004410 001404      BEQ     1$              ;BR IF YES
917 004412 011637 001234      MOV     (SP), LSTERR     ;RECORD BEING HERE
918 004416 105037 001341      CLRB   ERRFLG          ;PREPARE HEADER
919 004422 104406      1$:  SAVO5
920 004424 011605      MOV     (SP), R5         ;SAVE ALL PROC REGISTERS
921 004426 162705 000002      SUB     #2, R5           ;GET THE PC OF ERROR
922 004432 011504      MOV     (R5), R4         ;GET ADDRESS OF TRAP CALL
923 004434 006304      ASL     R4               ;GET HLT INSTRUCTION
924 004436 061504      ADD     (R5), R4         ;MULT BY TWO
925 004440 006304      ASL     R4               ;DOUBLE IT
926 004442 042704 177001      BIC     #177001, R4      ;MULT AGAIN
927 004446 062704 023222      ADD     #.ERRTAB, R4     ;CLEAR JUNK
928 004452 012437 004566      MOV     (R4)+, ERRMSG    ;GET POINTER
929 004456 012437 004600      MOV     (R4)+, DATAHD   ;GET ERROR MESSAGE
930 004462 011437 004612      MOV     (R4), DATABP     ;GET DATA HEADRER
931 004466 105737 001341      TSTB   ERRFLG          ;GET DATA TABLE
932 004472 001403      BEQ     TYPMSG          ;TYPE HEADREER
933 004474 005737 004612      TST    DATABP          ;BR IF YES
934 004500 001040      BNE     TYPDAT          ;DOES DATA TABLE EXIST?
                        ;BR IF YES.
;-----
;ERROR HANDLER
;-----
;TRAP DISPATCH SERVICE
;ARGUMENT OF TRAP IS EXTRACTED
;AND USED AS OFFSET TO OBTAIN POINTER
;TO SELECTED SUBROUTINE

```



935	004502	104402	005064		TYPMSG:	TYPE	,MCRLF	
936	004506	104402	005064			TYPE	,MCRLF	
937	004512	005737	001220			TST	LOCK	
938	004516	001402				BEQ	1\$	
939	004520	104402	005337			TYPE	,MASTEK	
940	004524	104402	005325		1\$:	TYPE	,MTSTN	
941	004530	104411	004726			CONVRT	,XTSTN	;SHOW IT
942	004534	104402	005414			TYPE	,MERRPC	;TYPE PC.
943	004540	104411	004720			CONVRT	,ERTABO	;SHOW IT
944	004544	104402	005064			TYPE	,MCRLF	;GIVE A CR/LF
945	004550	112737	177777	001341		MOVB	#-1,ERRFLG	;NO MORE HEADER UNLESS NO DATA TABLE.
946	004556	005737	004566			TST	ERRMSG	;IS THERE AN ERROR MESSAGE?
947	004562	001402				BEQ	WRKO.FM	;BR IF NO.
948	004564	104402				TYPE		;TYPE
949	004566	000000			ERRMSG:	0		ERROR MESSAGE
950	004570				WRKO.FM:			
951	004570	005737	004600			TST	DATAHD	;DATA HEADER?
952	004574	001402				BEQ	TYPDAT	;BR IF NO
953	004576	104402				TYPE		;TYPE
954	004600	000000			DATAHD:	0		DATA HEADER
955	004602	005737	004612		TYPDAT:	TST	DATABP	DATA TABLE?
956	004606	001402				BEQ	RESREG	;BR IF NO.
957	004610	104410				CONVRT		;SHOW
958	004612	000000			DATABP:	0		DATA TABLE
959	004614	104407			RESREG:	RESOS		RESTORE PROC REGISTERS
960	004616	022737	003104	000042	HALTS:	CMP	#\$ENDAD,2#42	;IF ACT-11 AUTO MODE--HALT!!
961	004624	001403				BEQ	1\$	
962	004626	005777	174350			TST	2\$WR	;HALT ON ERROR?
963	004632	100010				BPL	EXITER	;BR IF NO HALT ON ERROR
964	004634	010046			1\$:	PUSHRO		;SAVE RO
965	004636	016600	000002			MOV	2(SP),RO	;SHOW ERROR PC IN DATA LIGHTS
966	004642	042777	014000	174542		BIC	#\$SYSTST!MEXT,2TXCSR	
967	004650	000000				HALT		;HALT
968	004652	012600				POPPO		;GET RO
969	004654	005237	001232		EXITER:	INC	ERRCNT	;UPDATE ERROR COUNT
970	004660	032777	000400	174314		BIT	#\$SW08,2\$WR	;GOTO TOP OF TEST?
971	004666	001007				BNE	1\$	;BR IF YES
972	004670	032777	002000	174304		BIT	#\$SW10,2\$WR	;GOTO NEXT TEST?
973	004676	001407				BEQ	2\$	;BR IF NO
974	004700	013737	001216	001214		MOV	NEXT,RETURN	;SET FOR NEXT TEST
975	004706	012706	001150		1\$:	MOV	#\$STACK,SP	;RESET SP
976	004712	000177	174276			JMP	2\$RETURN	;GOTO SPECIFIED TEST
977	004716	000002			2\$:	RTI		;RETURN
978	004720	000001			ERTABO:	1		
979	004722	006	002			.BYTE	6,2	
980	004724	001266				SAVPC		
981	004726	000001			XTSTN:	1		
982	004730	003	002			.BYTE	3,2	
983	004732	001226				TSTNO		
984	004734	017600	000000		.PKCLK:	MOV	2(SP),RO	;GET THE # OF TICKS TO POKE
985	004740	062716	000002			ADD	#2,(SP)	;POP OVER THE #
986	004744				1\$:			
987	004744	052777	020000	174440		BIS	#\$CLK,2TXCSR	;POKE CLOCK UP
988	004752	005300				DEC	RO	;ARE WE DONE?
989	004754	001405				BEQ	2\$	;YES-GO TO 2\$
990	004756	042777	020000	174426		BIC	#\$CLK,2TXCSR	;POKE CLOCK DOWN

```

991 004764 005300          DEC      RO          ;ARE WE DONE?
992 004766 001366          BNE     1$          ;NO-REPEAT
993 004770 000002          2$:    RTI          ;RETURN
994
995
996          ;WAIT ROUTINE
997 004772 000240          SMALL: NOP          ;STALL
998 004774 000207          RTS     PC          ;RETURN
999
1000          ;POWER FAIL ROUTINE
1001
1002 004776 012737 005006 000024 .PFAIL: MOV     #PWRUP,24 ;LOAD PFAIL VECTOR FOR POWER UP
1003 005004 000000          HALT          ;
1004 005006 000005          PWRUP: RESET          ;WAIT TTY TO COME UP
1005 005010 012706 001150          MOV     #STACK,SP ;REINIT STACK POINTER
1006 005014 012737 004776 000024          MOV     #.PFAIL,24 ;LOAD PFAIL VECTOR FOR POWER DOWN
1007 005022 104402          TYPE
1008 005024 005067          MPOWER
1009 005026 000177 174162          JMP     @RETURN
1010          ;CLRVEC,ROUTINE TO FILL COMMUNICATION VECTOR AREA WITH .+2,HALT
1011
1012 005032 012702 000300          CLRVEC: MOV     #300,R2 ;R2 COMM VECTOR AREA ADRS
1013 005036 012701 000302          MOV     #302,R1 ;INIT R1 WITH ADRS OF HALT
1014 005042 010122          1$:    MOV     R1,(R2)+ ;MOV .+2 TO PC
1015 005044 005022          CLR     (R2)+ ;MOV HALT TO PC
1016 005046 022121          CMP     (R1)+,(R1)+ ;INC TO NEXT VECTOR AREA
1017 005050 022701 000776          CMP     #776,R1 ;END OF VECTOR AREA
1018 005054 001372          BNE     1$          ;NO
1019 005056 000207          RTS     PC          ;RETURN
1020
1021
1022
1023 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 DZDPDA /
(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          MCSRX: .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) /

```





1038  
 1039  
 1040  
 1041  
 1042  
 1043  
 1044  
 1045  
 1046  
 1047  
 1048  
 1049  
 1050  
 1051  
 1052  
 1053  
 1054  
 1055  
 1056  
 1057  
 1058  
 1059  
 1060  
 1061  
 1062  
 1063  
 1064  
 1065  
 1066  
 1067  
 1068  
 1069  
 1070  
 1071  
 1072  
 1073  
 1074  
 1075  
 1076  
 1077  
 1078  
 1079  
 1080  
 1081  
 1082  
 1083  
 1084  
 1085  
 1086  
 1087  
 1088  
 1089  
 1090  
 1091  
 1092  
 1093

006152 105737 001310  
 006156 001004  
 006160 104402 005133  
 006164 000000  
 006166 000776  
 006170 133737 001314 001310  
 006176 001020  
 006200 000241  
 006202 106137 001314  
 006206 105537 001314  
 006212 062737 000006 001316  
 006220 022737 001560 001316  
 006226 001360  
 006230 012737 001500 001316  
 006236 000754  
 006240 000241  
 006242 106137 001314  
 006246 105537 001314  
 006252 013700 001316  
 006256 062737 000006 001316  
 006264 022737 001560 001316  
 006272 001003  
 006274 012737 001500 001316  
 006302 012037 001404  
 006306 012037 001374  
 006312 012037 001434  
 006316 012700 000002  
 006322 013737 001404 001424  
 006330 005237 001424  
 006334 013737 001424 001406  
 006342 005237 001406  
 006346 013737 001406 001416  
 006354 013737 001406 001410  
 006362 013737 001406 001422  
 006370 005237 001422  
 006374 013737 001422 001420  
 006402 013737 001420 001412  
 006410 005237 001412  
 006414 013737 001412 001430  
 006422 005237 001430  
 006426 013737 001430 001414  
 006434 005237 001414  
 006440 013737 001414 001426  
 006446 005237 001426  
 006452 013737 001374 001376

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

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



1094	006460	060037	001376			ADD	RO, DUPRPS	;RX PRIORITY LEVEL
1095	006464	013737	001376	001400		MOV	DUPRPS, DUPTVC	
1096	006472	060037	001400			ADD	RO, DUPTVC	;TX VECTOR
1097	006476	013737	001400	001402		MOV	DUPTVC, DUPTPS	
1098	006504	060037	001402			ADD	RO, DUPTPS	;TX PRIORITY LEVEL
1099								
1100								
1101	006510	012700	001434			MOV	#LOO.00, RO	;LOAD STAUS 00-00
1102	006514	012701	001432			MOV	#MASK.A, R1	;PREPARE MASK.
1103	006520	012702	001433			MOV	#CLK.A, R2	;PREPARE CLOCKS
1104	006524	004737	006670			JSR	PC, FIX.00	;GO AND CALCULATE CONFIGURATION.
1105	006530	005737	000042			TST	2#42	
1106	006534	001050				BNE	4\$	
1107	006536	032777	000002	172436		BIT	#SW01, 2SWR	;IF SW01=1, GET STARTING TEST #
1108	006544	001444				BEQ	4\$	
1109	006546	104402	005064		7\$:	TYPE	, MCRLF	
1110	006552	104403				INSTR	;OUTPUT MESSAGE & GET INPUT STRING	
1111	006554	005325				MTSTN	;MESSAGE	
1112	006556	104405				PARAM	;CONVERT STRING	
1113	006560	000001				1	;LOW LIMIT	
1114	006562	001000				1000	;HIGH LIMIT	
1115	006564	001226				TSTNO	;STORE AT THIS LOCATION	
1116	006566	000			.BYTE	0	;MASK	
1117	006567	001			.BYTE	1	;HOW MANY TIMES + 2	
1118	006570	012700	007106			MOV	#TST1, RO	
1119	006574	022710	012737		5\$:	CMP	#12737, (RO)	
1120	006600	001017				BNE	6\$	
1121	006602	023760	001226	000002		CMP	TSTNO, 2(RO)	
1122	006610	001013				BNE	6\$	
1123	006612	022760	001226	000004		CMP	#TSTNO, 4(RO)	
1124	006620	001007				BNE	6\$	
1125	006622	010037	001214			MOV	RO, RETURN	;SAVE PC
1126	006626	104402	005064			TYPE	, MCRLF	
1127	006632	104402	005130			TYPE	, MR	
1128	006636	000412				BR	8\$	
1129	006640	005720			6\$:	TST	(RO)+	
1130	006642	020027	021412			CMP	RO, #TLAST+10	
1131	006646	001352				BNE	5\$	
1132	006650	104402	005060			TYPE	, MQM	
1133	006654	000734				BR	7\$	
1134								
1135	006656	012737	007106	001214	4\$:	MOV	#TST1, RETURN	;PREPARE RETURN ADDRESS
1136	006664	000177	172324		8\$:	JMP	2RETURN	;GO START TESTING.
1137								
1138	006670	011003			FIX.00:	MOV	(RO), R3	;GET PARAMETERS.
1139	006672	000207			5\$:	RTS	PC	

1140					
1141					
1142					
1143	006674	012577	172474	SETVEC:	MOV (R5)+, @DUPRVC
1144	006700	012577	172474		MOV (R5)+, @DUPTVC
1145	006704	112577	172466		MOVB (R5)+, @DUPRPS
1146	006710	112577	172466		MOVB (R5)+, @DUPTPS
1147	006714	000205			RTS R5
1148	006716			NO.ATRAP:	
1149	006716	104012			HLT 12
1150	006720	000002			RTI
1151					
1152	006722			NO.BTRAP:	
1153	006722	104013			HLT 13
1154	006724	000002			RTI
1155					
1156	006726	010046		SIMBCC:	MOV R0, -(SP)
1157	006730	010146			MOV R1, -(SP)
1158	006732	010246			MOV R2, -(SP)
1159	006734	012537	001236		MOV (R5)+, TEMP1
1160	006740	012537	001240		MOV (R5)+, TEMP2
1161	006744	012537	001242		MOV (R5)+, TEMP3
1162	006750	005037	007102	1\$:	CLR BCCFBK
1163	006754	013700	001242		MOV TEMP3, R0
1164	006760	006037	001240		ROR TEMP2
1165	006764	005500			ADC R0
1166	006766	032700	000001		BIT #BIT0, R0
1167	006772	001402			BEQ 2\$
1168	006774	005137	007102		COM BCCFBK
1169	007000	013700	007100	2\$:	MOV XPOLY, R0
1170	007004	005100			COM R0
1171	007006	040037	007102		BIC R0, BCCFBK
1172	007012	000241			CLC
1173	007014	006037	001242		ROR TEMP3
1174	007020	013700	007102		MOV BCCFBK, R0
1175	007024	013701	001242		MOV TEMP3, R1
1176	007030	010102			MOV R1, R2
1177	007032	040100			BIC R1, R0
1178	007034	043702	007102		BIC BCCFBK, R2
1179	007040	050200			BIS R2, R0
1180	007042	043737	007100 001242		BIC XPOLY, TEMP3
1181	007050	050037	001242		BIS R0, TEMP3
1182	007054	005337	001236		DEC TEMP1
1183	007060	001333			BNE 1\$
1184	007062	013737	001242 007104		MOV TEMP3, CALBCC
1185	007070	012602			MOV (SP)+, R2
1186	007072	012601			MOV (SP)+, R1
1187	007074	012600			MOV (SP)+, R0
1188	007076	000205			RTS R5
1189	007100	000000		XPOLY:	0
1190	007102	000000		BCCFBK:	0
1191	007104	000000		CALBCC:	0
1192		120001		CRC16=	120001
1193		102010		CRC.CCITT=	102010
1194					
1195					



1196  
1197  
1198  
1199  
1200  
1201  
1202  
1203  
1204  
1205  
1206  
1207  
1208  
1209  
1210  
1211  
1212  
1213  
1214  
1215  
1216  
1217  
1218  
1219  
1220  
1221  
1222  
1223  
1224  
1225  
1226  
1227  
1228  
1229  
1230  
1231  
1232  
1233  
1234  
1235  
1236  
1237  
1238  
1239  
1240  
1241  
1242  
1243  
1244  
1245  
1246  
1247  
1248  
1249  
1250  
1251

007106 012737 000001 001226  
007114 012737 007212 001216  
007122 012737 000340 177776  
007130 052777 000400 172254  
007136 004737 004772  
007142 004537 006674  
007146 006716  
007150 006722  
007152 340 340  
007154 012777 004100 172230  
007162 012737 000340 177776  
007170 000240  
007172 000240  
007174 000240  
007176 005077 172210  
007202 104400  
007204 012716 007176  
007210 000002  
007220 012737 007326 001216  
007226 122737 000005 001320  
007234 001026  
007236 012737 000340 177776  
007244 052777 000400 172140  
007252 004737 004772  
007256 004537 006674  
007262 006716  
007264 006722  
007266 340 340  
007270 012777 004100 172114

\*\*\*\*\* TEST 1 \*\*\*\*\*  
\*PRIORITY INTERRUPT TEST.  
\*SET PROCESSOR STATUS TO PRIORITY 7  
\*AND VERIFY THAT THE DUP11 WILL NOT INTERRUPT.  
\*\*\*\*\*

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

\*\*\*\*\*  
TST1: MOV #1, @TSTNO  
MOV #TST2, NEXT  
MOV #340, PS ; LOCK OUT INTERRUPTS  
BIS #MRESET, @TXCSR ; RESET THE DEVICE  
JSR PC, SMALL ; WAIT FOR RESET TO FINISH  
JSR RS, SETVEC ; SET UP VECTORS  
NO. ATRAP ; VECTOR "A"  
NO. BTRAP ; VECTOR "B"  
.BYTE 340, 340 ; LEVEL  
MOV #TXINTE!SYSTST, @TXCSR ; TURN ON DUP TX INT. ENABLE AND ENTER SYSTST MODE  
MOV #340, PS ; SET CPU PRIORITY--CHANGE HERE IF NOT = 5  
NOP ; STALL  
NOP ; DITTO  
NOP ; DITTO  
15: CLR @TXCSR ; DISABLE THE DUP11  
SCOPE ; SCOPE THIS TEST  
25: MOV #15, (SP) ; SETUP FOR RETURN  
RTI ; RETURN

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

\*\*\*\*\*  
\*  
TEST 2  
\*  
\*\*\*\*\*

\*\*\*\*\*  
TST2: MOV #2, @TSTNO  
MOV #TST3, NEXT  
CMPB #5, PIRTY ; COMPARE REAL WITH NORMAL  
BNE 15 ; BR IF NOT A MATCH  
MOV #340, PS ; LOCK OUT INTERRUPTS  
BIS #MRESET, @TXCSR ; RESET THE DEVICE  
JSR PC, SMALL ; WAIT FOR RESET TO FINISH  
JSR RS, SETVEC ; SET UP VECTORS  
NO. ATRAP ; VECTOR "A"  
NO. BTRAP ; VECTOR "B"  
.BYTE 340, 340 ; LEVEL  
MOV #TXINTE!SYSTST, @TXCSR ; TURN ON DUP TX INT. ENABLE AND ENTER SYSTST MODE

```

1252 007276 012737 000300 177776      MOV      #300,PS      ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
1253 007304 000240                    NOP                    ;STALL
1254 007306 000240                    NOP                    ;DITTO
1255 007310 000240                    NOP                    ;DITTO
1256 007312 005077 172074      1S:    CLR      @TXCSR  ;DISABLE THE DUP11
1257 007316 104400                    SCOPE                 ;SCOPE THIS TEST
1258 007320 012716 007312      2S:    MOV      #1S,(SP) ;SETUP FOR RETURN
1259 007324 000002                    RTI                    ;RETURN

```

1260  
1261  
1262  
1263  
1264  
1265  
1266  
1267  
1268  
1269  
1270  
1271  
1272  
1273  
1274  
1275  
1276  
1277  
1278  
1279  
1280  
1281  
1282  
1283  
1284  
1285  
1286  
1287  
1288  
1289  
1290  
1291  
1292  
1293  
1294  
1295  
1296  
1297  
1298  
1299  
1300  
1301  
1302  
1303  
1304  
1305  
1306  
1307

```

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

```

```

*****
*
* TEST 3
*
*****

```

```

*****
TST3: MOV      #3,@TSTNO
      MOV      #TST4,NEXT
      CMPB    #5,PRIRTY ;COMPARE REAL WITH NORMAL
      BNE     1S      ;BR IF NOT A MATCH
      MOV     #340,PS  ;LOCK OUT INTERRUPTS
      BIS     #MRESET,@TXCSR ;RESET THE DEVICE
      JSR    PC,SMALL ;WAIT FOR RESET TO FINISH
      JSR    RS,SETVEC ;SET UP VECTORS
      NO.ATRAP ;VECTOR "A"
      NO.BTRAP ;VECTOR "B"
      .BYTE 340,340 ;LEVEL
      MOV     #TXINTE!SYSTST,@TXCSR ;TURN ON DUP TX INT. ENABLEAND ENTER SYSTST MODE
      MOV     #240,PS  ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
      NOP
      NOP
      NOP
      NOP
      1S:    CLR      @TXCSR ;DISABLE THE DUP11
      SCOPE
      2S:    MOV      #1S,(SP) ;SCOPE THIS TEST
      RTI                    ;SETUP FOR RETURN
      ;RETURN

```

```

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

```

```

*****
*
* TEST 4
*
*****

```



```

1308 007442 012737 000004 001226 TST4:  MOV      #4, @TSTNO
1309 007450 012737 007560 001216      MOV      #TST5, NEXT
1310 007456 122737 000005 001320      CMPB    #5, PRIRTY      ;COMPARE REAL WITH NORMAL
1311 007464 001027                BNE     1$              ;BR IF NOT A MATCH
1312 007466 012737 000340 177776      MOV      #340, PS      ;LOCK OUT INTERRUPTS
1313 007474 052777 000400 171710      BIS      #MRESET, @TXCSR ;RESET THE DEVICE
1314 007502 004737 004772                JSR     PC, SMALL      ;WAIT FOR RESET TO FINISH
1315 007506 004537 006674                JSR     RS, SETVEC     ;SET UP VECTORS
1316 007512 006716                NO. ATRAP              ;VECTOR "A"
1317 007514 007552                2$                    ;VECTOR "B"
1318 007516                340 340                ;LEVEL
1319 007520 012777 004100 171664      MOV      #TXINTE!SYSTST, @TXCSR ;TURN ON DUP TX INT. ENABLE AND ENTER SYSTST MODE
1320 007526 012737 000200 177776      MOV      #200, PS      ;SET CPU PRIORITY--CHANGE HERE IF NOT = 5
1321 007534 000240                NOP                      ;STALL
1322 007536 000240                NOP                      ;DITTO
1323 007540 000240                NOP                      ;DITTO
1324 007542 104013                HLT     13              ;DUP FAILED TO INTERRUPT-POSSBILY WRONG PRIORITY-CHANGE IF NOT 5
1325 007544 005077 171642        1$:  CLR      @TXCSR        ;DISABLE THE DUP11
1326 007550 104400                SCOPE                    ;SCOPE THIS TEST
1327 007552 012716 007544        2$:  MOV      #1$, (SP)     ;SETUP FOR RETURN
1328 007556 000002                RTI                      ;RETURN

```

```

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

```

```

*****
*
* TEST 5
*
*****

```

```

1341
1342 007560 012737 000005 001226 TST5:  MOV      #5, @TSTNO
1343 007566 012737 010120 001216      MOV      #TST6, NEXT
1344 007574 105737 001322                TSTB    TCNFLG
1345 007600 001510                BEQ     1$
1346 007602 012737 000340 177776      MOV      #340, PS      ;LOCK OUT INTERRUPTS
1347 007610 052777 000400 171574      BIS      #MRESET, @TXCSR ;RESET THE DEVICE
1348 007616 004737 004772                JSR     PC, SMALL      ;WAIT FOR RESET TO FINISH
1349 007622 052777 010010 171562      BIS      #MEXT!HDEN, @TXCSR ;ENTER MAINT EXT AND HALF-DUPLEX MODES
1350 007630 004537 006674                JSR     RS, SETVEC     ;SET UP VECTORS
1351 007634 010110                2$                    ;RECEIVER
1352 007636 006722                NO. BTRAP              ;TRANSMITTER
1353 007640                340 340                ;LEVEL
1354 007642 005037 177776      CLR     PS              ;LOWER PROC. STATUS
1355 007646 052777 000020 171530      BIS      #RCVEN, @RXCSR ;TURN ON RECEIVER
1356 007654 052777 000100 171522      BIS      #RINTEN, @RXCSR ;TURN ON INT. ENABLE
1357 007662 052777 000020 171522      BIS      #SEND, @TXCSR  ;TURN ON TRANSMITTER
1358 007670 012737 000005 007720      MOV      #5, 68$       ;LOAD THE NUMBER
1359 007676 032777 004000 171510        66$: BIT      #TIMER, @TXDBUF ;CHECK THE TIMER BIT
1360 007704 001374                BNE     66$            ;BR IF SET
1361 007706 032777 004000 171500        67$: BIT      #TIMER, @TXDBUF ;CHECK THE BIT
1362 007714 001774                BEQ     67$            ;BR IF CLEAR
1363 007716 005327                DEC     (PC)+           ;DECREMENT THE NUMBER

```

```

1364 007720 000005          68$: 5          ;OF TIMES TO REPEAT
1365 007722 001365          ;BNE          66$          ;BR IF MORE TO GO
1366 007724 105777 171462  TSTB      @TXCSR
1367 007730 100401          BMI        3$
1368 007732 104005          HLT        5          ;TXDONE FAILED TO SET
1369 007734 012777 000400 171452 3$: MOV      @TSOM,@TXDBUF ;LOAD TX BUFFER
1370 007742 012737 000005 007772 MOV      @5,73$ ;LOAD THE NUMBER
1371 007750 032777 004000 171436 71$: BIT      @TIMER,@TXDBUF ;CHECK THE TIMER BIT
1372 007756 001374          BNE        71$ ;BR IF SET
1373 007760 032777 004000 171426 72$: BIT      @TIMER,@TXDBUF ;CHECK THE BIT
1374 007766 001774          BEQ        72$ ;BR IF CLEAR
1375 007770 005327          DEC      (PC)+ ;DECREMENT THE NUMBER
1376 007772 000005          73$: 5          ;OF TIMES TO REPEAT
1377 007774 001365          BNE        71$ ;BR IF MORE TO GO
1378 007776 105777 171410  TSTB      @TXCSR ;CHECK FOR DONE
1379 010002 100401          BMI        4$ ;BR IF SET
1380 010004 104000          HLT        ;EXTERNAL CLOCKING STOPPED
1381 ;OR DATA WAS NOT RECEIVED.CHECK
1382 ;EIA DATA AND CLOCK PATHS
1383 010006 005077 171402 4$: CLR      @TXDBUF ;LOAD A CHARACTER
1384 010012 105777 171374  TSTB      @TXCSR ;CHECK FOR DONE
1385 010016 100375          BPL        .-4 ;BR IF NOT SET
1386 010020 012777 001000 171366 MOV      @TEOM,@TXDBUF ;END THE MESSAGE
1387 010026 012737 000050 010056 MOV      @40,78$ ;LOAD THE NUMBER
1388 010034 032777 004000 171352 76$: BIT      @TIMER,@TXDBUF ;CHECK THE TIMER BIT
1389 010042 001374          BNE        76$ ;BR IF SET
1390 010044 032777 004000 171342 77$: BIT      @TIMER,@TXDBUF ;CHECK THE BIT
1391 010052 001774          BEQ        77$ ;BR IF CLEAR
1392 010054 005327          DEC      (PC)+ ;DECREMENT THE NUMBER
1393 010056 000050          78$: 40. ;OF TIMES TO REPEAT
1394 010060 001365          BNE        76$ ;BR IF MORE TO GO
1395 010062 012737 000340 177776 1$: MOV      @340,PS ;RAISE PROCESSOR STATUS
1396 010070 012706 001150          MOV      @STACK,SP ;RESET STACK
1397 010074 052777 000400 171310 BIS      @MRESET,@TXCSR ;RESET THE DEVICE
1398 010102 004737 004772 JSR      PC,SMALL ;WAIT FOR RESET TO FINISH
1399 010106 104400          SCOPE ;SCOPE THIS TEST
1400
1401 010110 104007          2$: HLT      7          ;RECEIVER INTERRUPTED AND SHOULD
1402 010112 012716 010062  MOV      @1$, (SP) ;NOT HAVE--THIS IS HALF
1403 010116 000002          RTI        ;DUPLEX.

```

```

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

```

```

;*****
; TEST 6
;*****
;*****

```

```

1416 010120 012737 000006 001226 TST6: MOV      @6,@TSTNO
1417 010126 012737 010560 001216 MOV      @TST7,NEXT
1418 010134 052777 000400 171250 BIS      @MRESET,@TXCSR ;RESET THE DEVICE
1419 010142 004737 004772 JSR      PC,SMALL ;WAIT FOR RESET TO FINISH

```



1420	010146	012737	000001	001236	MOV	#1,TEMP1	;LOAD DATA
1421	010154	005037	001240		CLR	TEMP2	;CLEAR EXPECTED
1422	010160	012737	000340	177776	MOV	#340,PS	;PS = 7
1423	010166	052777	004000	171216	BIS	#SYSTST, @TXCSR	;ENTER SYSTEM TEST MODE
1424	010174	004537	006674		JSR	RS,SETVEC	;LOAD INTERRUPT VECTORS
1425	010200	010374			11\$		;RECEIVER
1426	010202	010454			12\$		;TRANSMITTER
1427	010204	340	340		.BYTE	340,340	;LEVEL
1428	010206	052777	001000	171174	BIS	#CRCEN,@PARCSR	;TURN OFF CRC
1429	010214	052777	000020	171162	BIS	#RCVEN,@RXCSR	;TURN ON THE RECEIVER
1430	010222	052777	000100	171154	BIS	#RINTEN,@RXCSR	;TURN ON REC INTERRUPT ENABLE
1431	010230	105777	171156		1\$:	TSTB @TXCSR	;TEST FOR TX DONE
1432	010234	100375			BPL	1\$	;BR IF NOT SET
1433	010236	052777	000020	171146	2\$:	BIS #SEND,@TXCSR	;TURN ON SEND
1434	010244	012777	000400	171142	MOV	#TSOM,@TXDBUF	;TURN ON START OF MESSAGE
1435	010252	012737	000005	010302	MOV	#5,68\$	;LOAD THE NUMBER
1436	010260	032777	004000	171126	66\$:	BIT #TIMER,@TXDBUF	;CHECK THE TIMER BIT
1437	010266	001374			BNE	66\$	;BR IF SET
1438	010270	032777	004000	171116	67\$:	BIT #TIMER,@TXDBUF	;CHECK THE BIT
1439	010276	001774			BEQ	67\$	;BR IF CLEAR
1440	010300	005327			DEC	(PC)+	;DECREMENT THE NUMBER
1441	010302	000005			5		;OF TIMES TO REPEAT
1442	010304	001365			BNE	66\$	;BR IF MORE TO GO
1443	010306	105777	171100		3\$:	TSTB @TXCSR	;WAIT FOR DONE
1444	010312	100401			BMI	4\$	;BR IF SET
1445	010314	104000			HLT		;EXTERNAL CLOCKING STOPPED
1446	010316	005077	171072		4\$:	CLR @TXDBUF	;PUSH OUT DATA
1447	010322	052777	000100	171062	BIS	#TXINTE,@TXCSR	;TURN ON TRANSMITTER INT ENABLE
1448	010330	005037	177776		CLR	PS	;LOWER PROCESOR STATUS
1449	010334				5\$:		
1450	010334	012737	000040	010364	MOV	#32,73\$	;LOAD THE NUMBER
1451	010342	032777	004000	171044	71\$:	BIT #TIMER,@TXDBUF	;CHECK THE TIMER BIT
1452	010350	001374			BNE	71\$	;BR IF SET
1453	010352	032777	004000	171034	72\$:	BIT #TIMER,@TXDBUF	;CHECK THE BIT
1454	010360	001774			BEQ	72\$	;BR IF CLEAR
1455	010362	005327			DEC	(PC)+	;DECREMENT THE NUMBER
1456	010364	000040			73\$:	32	;OF TIMES TO REPEAT
1457	010366	001365			BNE	71\$	;BR IF MORE TO GO
1458	010370	104001			HLT	1	;FAILED TO INTERRUPT IN TIME
1459	010372	104400			6\$:	SCOPE	;SCOPE THIS TEST
1460							
1461							
1462							
1463							
1464							
1465							
1466	010374	017737	171006	001324	11\$:	MOV @RXDBUF,DATA	;GET THE REGISTER AND DATA
1467	010402	123737	001240	001324	CMPB	TEMP2,DATA	;CHECK IT
1468	010410	001401			BEQ	+4	;BR IF OK
1469	010412	104002			HLT	2	;COMPARISON ERROR
1470	010414	105237	001240		INCB	TEMP2	;COUNT UP EXPECTED
1471	010420	105737	001240		TSTB	TEMP2	;CHECK TO SEE IF DONE
1472	010424	001012			BNE	7\$	;BR IF NO
1473	010426	105777	170752		10\$:	TSTB @RXCSR	;CHECK FOR DONE
1474	010432	100375			BPL	10\$	;BR IF NOT YET
1475	010434	032777	001000	170744	BIT	#REOM,@RXDBUF	;CHECK FOR END OF MSG

; INTERRUPT SERVICE ROUTINES

; RECEIVER:

```

1476 010442 001001          BNE      .+4          ;BR IF SET
1477 010444 104003          HLT      3            ;END OF MSG FAILED TO SET
1478 010446 012716 010372  MOV     #6$, (SP)    ;CRUNCH STACK
1479
1480 010452 000002          7$:     RTI           ;RETURN
1481
1482
1483          ;TRANSMITTER:
1484 010454 113777 001236 170732 12$:     MOV     TEMP1, @TXDBUF ;LOAD THE TRANSMITTER BUFFER
1485 010462 105237 001236          INCB    TEMP1        ;UP THE COUNT
1486 010466 122737 000377 001236  CMP     #377, TEMP1  ;ARE WE DONE
1487 010474 001026          BNE     13$          ;BR IF NO
1488 010476 012777 010506 170674  MOV     #21$, @DUPTVC ;SETUP FOR NEXT PART
1489 010504 000422          BR      13$          ;LEAVE
1490 010506 012777 000377 170700 21$:     MOV     #377, @TXDBUF ;LOAD BUFFER
1491 010514 012777 010524 170656  MOV     #22$, @DUPTVC ;SETUP NEXT PART
1492 010522 000413          BR      13$          ;LEAVE
1493 010524 012777 001000 170662 22$:     MOV     #TEOM, @TXDBUF ;SET END OF MSG
1494 010532 000240          NOP                    ;STALL
1495 010534 000240          NOP                    ;DITTO
1496 010536 042777 000120 170646  BIC     #SEND!TXINTE, @TXCSR ;TURN OFF TRANSMITTER
1497 010544 012777 006722 170626  MOV     #NO.BTRAP, @DUPTVC ;LOAD VECTOR
1498 010552 012716 010334 13$:     MOV     #5$, (SP)    ;CRUNCH STACK
1499 010556 000002          RTI           ;RETURNS
1500
1501
1502
1503
1504
1505
1506
1507
1508
1509
1510
1511
1512

```

```

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

```

```

;*****
; TEST 7
;*****
;*****
TST7: 1513 010560 012737 000007 001226  MOV     #7, @TSTNO
1514 010566 012737 011360 001216  MOV     #TST10, NEXT
1515 010574 052777 000400 170610  BIS     #MRESET, @TXCSR ;RESET THE DEVICE
1516 010602 004737 004772          JSR     PC, SMALL    ;WAIT FOR RESET TO FINISH
1517 010606 005001          CLR     R1           ;CLEAR OUT DATA
1518 010610 012737 102010 007100  MOV     #CRC.CCITT, XPOLY ;SET UP THE POLYNOMIAL
1519 010616 012737 177777 007104  MOV     #-1, CALBCC ;SETUP FOR THE FIRST TIME
1520 010624 013737 007104 010646 16$:     MOV     CALBCC, 20$  ;ALLOW FOR THE NEXT CHARACTER
1521 010632 010137 010644          MOV     R1, 17$     ;LOAD DATA
1522 010636 004537 006726          JSR     R5, SIMBCC   ;GO CALCULATE SOFTWARE BCC
1523 010642 000010          B.      8           ;BASED ON THESE PARAMETERS
1524 010644 000001          17$:     .BLKW 1          ;DATA
1525 010646 000001          20$:     .BLKW 1          ;PREVIOUS BCC
1526 010650 105201          INCB    R1           ;INCREMENT DATA
1527 010652 001364          BNE     16$          ;BR IF MORE TO GO
1528 010654 012737 000001 001236  MOV     #1, TEMP1    ;LOAD DATA
1529 010662 005037 001240          CLR     TEMP2        ;CLEAR EXPECTED
1530 010666 012737 000340 177776  MOV     #340, PS     ;PS = 7
1531 010674 052777 004000 170510  BIS     #SYSTST, @TXCSR ;ENTER SYSTEM TEST MODE

```



1532	010702	004537	006674		JSR	R5, SETVEC	: LOAD INTERRUPT VECTORS
1533	010706	011074			11\$		: RECEIVER
1534	010710	011142			12\$		: TRANSMITTER
1535	010712	340	340		.BYTE	340, 340	: LEVEL
1536	010714	052777	000020	170462	BIS	#RCVEN, @RXCSR	: TURN ON THE RECEIVER
1537	010722	052777	000100	170454	BIS	#RINTEN, @RXCSR	: TURN ON REC INTERRUPT ENABLE
1538	010730	105777	170456		1\$: TSTB	@TXCSR	: TEST FOR TX DONE
1539	010734	100375			BPL	1\$	: BR IF NOT SET
1540	010736	052777	000020	170446	2\$: BIS	#SEND, @TXCSR	: TURN ON SEND
1541	010744	012777	000400	170442	MOV	#TSOM, @TXDBUF	: TURN ON START OF MESSAGE
1542	010752	012737	000005	011002	MOV	#5, 68\$	: LOAD THE NUMBER
1543	010760	032777	004000	170426	66\$: BIT	#TIMER, @TXDBUF	: CHECK THE TIMER BIT
1544	010766	001374			BNE	66\$	: BR IF SET
1545	010770	032777	004000	170416	67\$: BIT	#TIMER, @TXDBUF	: CHECK THE BIT
1546	010776	001774			BEQ	67\$	: BR IF CLEAR
1547	011000	005327			DEC	(PC)+	: DECREMENT THE NUMBER
1548	011002	000005			5		: OF TIMES TO REPEAT
1549	011004	001365			BNE	66\$	: BR IF MORE TO GO
1550	011006	105777	170400		3\$: TSTB	@TXCSR	: WAIT FOR DONE
1551	011012	100401			BMI	4\$	: BR IF SET
1552	011014	104000			HLT		: EXTERNAL CLOCKING STOPPED
1553	011016	005077	170372		4\$: CLR	@TXDBUF	: PUSH OUT DATA
1554	011022	052777	000100	170362	BIS	#TXINTE, @TXCSR	: TURN ON TRANSMITTER INT ENABLE
1555	011030	005037	177776		CLR	PS	: LOWER PROCESOR STATUS
1556	011034				5\$:		
1557	011034	012737	000040	011064	MOV	#32, 73\$	: LOAD THE NUMBER
1558	011042	032777	004000	170344	71\$: BIT	#TIMER, @TXDBUF	: CHECK THE TIMER BIT
1559	011050	001374			BNE	71\$	: BR IF SET
1560	011052	032777	004000	170334	72\$: BIT	#TIMER, @TXDBUF	: CHECK THE BIT
1561	011060	001774			BEQ	72\$	: BR IF CLEAR
1562	011062	005327			DEC	(PC)+	: DECREMENT THE NUMBER
1563	011064	000040			73\$: 32.		: OF TIMES TO REPEAT
1564	011066	001365			BNE	71\$	: BR IF MORE TO GO
1565	011070	104001			HLT	1	: FAILED TO INTERRUPT IN TIME
1566	011072	104400			6\$: SCOPE		: SCOPE THIS TEST

: INTERRUPT SERVICE ROUTINES

: RECEIVER:

1573	011074	017737	170306	001324	11\$: MOV	@RXDBUF, DATA	: GET THE REGISTER AND DATA
1574	011102	123737	001240	001324	CMPB	TEMP2, DATA	: CHECK IT
1575	011110	001401			BEQ	.+4	: BR IF OK
1576	011112	104002			HLT	2	: COMPARISON ERROR
1577	011114	105237	001240		INCB	TEMP2	: COUNT UP EXPECTED
1578	011120	105737	001240		TSTB	TEMP2	: CHECK TO SEE IF DONE
1579	011124	001005			BNE	7\$	: BR IF NO
1580	011126	004537	006674		JSR	R5, SETVEC	: YES--RESET THE VECTORS
1581	011132	011246			14\$		: RECEIVER
1582	011134	011142			12\$		: TRANSMITTER
1583	011136	340	340		.BYTE	340, 340	: LEVEL
1584							
1585	011140	000002			7\$: RTI		: RETURN
1586							
1587							

```

1588                                     : TRANSMITTER:
:589 011142 113777 001236 170244 12$: MOVB TEMP1, @TXDBUF ;LOAD THE TRANSMITTER BUFFER
1590 011150 105237 001236          INCB TEMP1 ;UP THE COUNT
1591 011154 122737 000377 001236  CMPB #377, TEMP1 ;ARE WE DONE
1592 011162 001026          BNE 13$ ;BR IF NO
1593 011164 012777 011174 170206  MOV #21$, @DUPTVC ;SETUP FOR NEXT PART
1594 011172 000422          BR 13$ ;LEAVE
1595 011174 012777 000377 170212 21$: MOV #377, @TXDBUF ;LOAD BUFFER
1596 011202 012777 011212 170170  MOV #22$, @DUPTVC ;SETUP NEXT PART
1597 011210 000413          BR 13$ ;LEAVE
1598 011212 012777 001000 170174 22$: MOV #TEOM, @TXDBUF ;SET END OF MSG
1599 011220 000240          NOP ;STALL
1600 011222 000240          NOP ;DITTO
1601 011224 042777 000120 170160  BIC #SEND!TXINTE, @TXCSR ;TURN OFF TRANSMITTER
1602 011232 012777 006722 170140  MOV #NO.BTRAP, @DUPTVC ;LOAD VECTOR
1603 011240 012716 011034          13$: MOV #5$, (SP) ;CRUNCH STACK
1604 011244 000002          RTI ;RETURNS
1605
1606 011246 117737 170134 001324 14$: MOVB @RXDBUF, DATA ;GET FIRST PART OF CRC
1607 011254 105777 170124          TSTB @RXCSR ;WAIT FOR SECOND PART
1608 011260 100375          BPL -4 ;DITTO
1609 011262 017737 170120 001242  MOV @RXDBUF, TEMP3 ;GET THE REST OF THE CRC
1610 011270 113737 001242 001325  MOVB TEMP3, DATA+1 ;SET UP CRC CHARACTER
1611 011276 012716 011304          MOV #15$, (SP) ;SETUP FOR RETURN
1612 011302 000002          RTI ;RETURN
1613 011304 012737 000340 177776 15$: MOV #340, PS ;RAISE PS
1614 011312 005137 007104          COM CALBCC ;INVERT BCC
1615 011316 023737 007104 001324  CMP CALBCC, DATA ;COMPARE SOFTWARE AND HARDWARE BCC
1616 011324 001401          BEQ +4 ;BR IF OK
1617 011326 104004          HLT 4 ;BCC COMPARISON ERROR
1618 011330 032737 010000 001242  BIT #CRCERR, TEMP3 ;CHECK THE ERROR BIT
1619 011336 001401          BEQ +4 ;BR IF NO ERROR
1620 011340 104004          HLT 4 ;BCC ERROR--RECEIVER DOESN'T
1621                                     ;AGREE WITH WHAT TX SENT
1622 011342 052777 000400 170042  BIS #MRESET, @TXCSR ;RESET THE DEVICE
1623 011350 004737 004772          JSR PC, SMALL ;WAIT FOR RESET TO FINISH
1624 011354 000137 011072          JMP 6$ ;LEAVE
1625
1626                                     ;***** TEST 10 *****
1627                                     ;*TEST OF THE DUP RUNNING A BINARY COUNT
1628                                     ;*PATTERN WITH A CRC CALCULATION
1629                                     ;:*****
1630
1631                                     ;:*****
1632                                     ;: TEST 10 *
1633                                     ;:
1634                                     ;:*****
1635                                     ;:*****
1636
1637 011360 012737 000010 001226  TST10: MOV #10, @TSTNO
1638 011366 012737 012166 001216  MOV #TST11 NEXT
1639 011374 052777 000400 170010  BIS #MRESET, @TXCSR ;RESET THE DEVICE
1640 011402 004737 004772          JSR PC, SMALL ;WAIT FOR RESET TO FINISH
1641 011406 105737 001322          TSTB TCNFLAG
1642 011412 001532          BEQ 6$
1643 011414 005001          CLR R1 ;CLEAR OUT DATA

```



```

1644 011416 012737 102010 007100      MOV      #CRC.CCITT,XPOLY      ;SET UP THE POLYNOMIAL
1645 011424 012737 177777 007104      MOV      #-1,CALBCC           ;SETUP FOR THE FIRST TIME
1646 011432 013737 007104 011454 16$:    MOV      CALBCC,20$          ;ALLOW FOR THE NEXT CHARACTER
1647 011440 010137 011452          MOV      R1,17$             ;LOAD DATA
1648 011444 004537 006726          JSR      R5,SIMBCC          ;GO CALCULATE SOFTWARE BCC
1649 011450 000010          B.                          ;BASED ON THESE PARAMETERS
1650 011452 000001          17$:    .BLKW      1              ;DATA
1651 011454 000001          20$:    .BLKW      1              ;PREVIOUS BCC
1652 011456 105201          INCB     R1                 ;INCREMENT DATA
1653 011460 001364          BNE     16$                ;BR IF MORE TO GO
1654 011462 012737 000001 001236    MOV      #1,TEMP1          ;LOAD DATA
1655 011470 005037 001240          CLR     TEMP2              ;CLEAR EXPECTED
1656 011474 012737 000340 177776    MOV      #340,PS           ;PS = 7
1657 011502 052777 010000 167702    BIS     #MEXT,@TXCSR
1658 011510 004537 006674          JSR      R5,SETVEC         ;LOAD INTERRUPT VECTORS
1659 011514 011702          11$:    ;RECEIVER
1660 011516 011750          12$:    ;TRANSMITTER
1661 011520          340     340                ;LEVEL
1662 011522 052777 000020 167654    BIS     #RCVEN,@RXCSR     ;TURN ON THE RECEIVER
1663 011530 052777 000100 167646    BIS     #RINTEN,@RXCSR   ;TURN ON REC INTERRUPT ENABLE
1664 011536 105777 167650          1$:    TSTB     @TXCSR           ;TEST FOR TX DONE
1665 011542 100375          BPL     1$                 ;BR IF NOT SET
1666 011544 052777 000020 167640 2$:    BIS     #SEND,@TXCSR      ;TURN ON SEND
1667 011552 012777 000400 167634    MOV     #TSM,@TXDBUF     ;TURN ON START OF MESSAGE
1668 011560 012737 000005 011610    MOV     #5,68$           ;LOAD THE NUMBER
1669 011566 032777 004000 167620 66$:    BIT     #TIMER,@TXDBUF   ;CHECK THE TIMER BIT
1670 011574 001374          BNE     66$               ;BR IF SET
1671 011576 032777 004000 167610 67$:    BIT     #TIMER,@TXDBUF   ;CHECK THE BIT
1672 011604 001774          BEQ     67$               ;BR IF CLEAR
1673 011606 005327          DEC     (PC)+             ;DECREMENT THE NUMBER
1674 011610 000005          68$:    5                      ;OF TIMES TO REPEAT
1675 011612 001365          BNE     66$               ;BR IF MORE TO GO
1676 011614 105777 167572          3$:    TSTB     @TXCSR       ;WAIT FOR DONE
1677 011620 100401          BMI     4$                 ;BR IF SET
1678 011622 104000          HLT
1679 011624 005077 167564          4$:    CLR     @TXDBUF        ;EXTERNAL CLOCKING STOPPED
1680 011630 052777 000100 167554    BIS     #TXINTE,@TXCSR   ;PUSH OUT DATA
1681 011636 005037 177776          CLR     PS                 ;TURN ON TRANSMITTER INT ENABLE
1682 011642          5$:    ;LOWER PROCESOR STATUS
1683 011642 012737 000040 011672    MOV     #32,73$          ;LOAD THE NUMBER
1684 011650 032777 004000 167536 71$:    BIT     #TIMER,@TXDBUF   ;CHECK THE TIMER BIT
1685 011656 001374          BNE     71$               ;BR IF SET
1686 011660 032777 004000 167526 72$:    BIT     #TIMER,@TXDBUF   ;CHECK THE BIT
1687 011666 001774          BEQ     72$               ;BR IF CLEAR
1688 011670 005327          DEC     (PC)+             ;DECREMENT THE NUMBER
1689 011672 000040          73$:    32                  ;OF TIMES TO REPEAT
1690 011674 001365          BNE     71$               ;BR IF MORE TO GO
1691 011676 104001          HLT     1                  ;FAILED TO INTERRUPT IN TIME
1692 011700 104400          6$:    SCOPE              ;SCOPE THIS TEST
1693
1694
1695          ;INTERRUPT SERVICE ROUTINES
1696          ;-----
1697
1698          ;RECEIVER:
1699 011702 017737 167500 001324 11$:    MOV     @RXDBUF,DATA     ;GET THE REGISTER AND DATA

```

DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 37  
 DZDPDA.P11 BINARY PATTERN TEST WITH BCC IN EXTERNAL MODE

```

1700 011710 123737 001240 001324      CMPB   TEMP2,DATA      ;CHECK IT
1701 011716 001401                    BEQ    .+4             ;BR IF OK
1702 011720 104002                    HLT    2              ;COMPARISON ERROR
1703 011722 105237 001240      INCB   TEMP2          ;COUNT UP EXPECTED
1704 011726 105737 001240      TSTB  TEMP2          ;CHECK TO SEE IF DONE
1705 011732 001005                    BNE    7$            ;BR IF NO
1706 011734 004537 006674      JSR    R5,SETVEC     ;YES--RESET THE VECTORS
1707 011740 012054                    14$
1708 011742 011750                    12$
1709 011744      340      340      .BYTE  340,340      ;RECEIVER
;TRANSMITTER
;LEVEL
1710
1711 011746 000002                    7$:   RTI             ;RETURN
1712
1713
1714
1715 011750 113777 001236 167436 12$:   MOVB  TEMP1,@TXDBUF  ;LOAD THE TRANSMITTER BUFFER
1716 011756 105237 001236      INCB  TEMP1          ;UP THE COUNT
1717 011762 122737 000377 001236      CMPB  #377,TEMP1    ;ARE WE DONE
1718 011770 001026                    BNE    13$          ;BR IF NO
1719 011772 012777 012002 167400      MOV   #21$,@DUPTVC  ;SETUP FOR NEXT PART
1720 012000 000422                    BR     13$          ;LEAVE
1721 012002 012777 000377 167404 21$:   MOV   #377,@TXDBUF  ;LOAD BUFFER
1722 012010 012777 012020 167362      MOV   #22$,@DUPTVC  ;SETUP NEXT PART
1723 012016 000413                    BR     13$          ;LEAVE
1724 012020 012777 001000 167366 22$:   MOV   #TEOM,@TXDBUF ;SET END OF MSG
1725 012026 000240                    NOP
1726 012030 000240                    NOP
1727 012032 042777 000120 167352      BIC   #SEND!TXINTE,@TXCSR ;TURN OFF TRANSMITTER
1728 012040 012777 006722 167332      MOV   #NO.BTRAP,@DUPTVC ;LOAD VECTOR
1729 012046 012716 011642 13$:   MOV   #5$, (SP)    ;CRUNCH STACK
1730 012052 000002                    RTI             ;RETURNS
1731
1732 012054 117737 167326 001324 14$:   MOVB  @RXDBUF,DATA  ;GET FIRST PART OF CRC
1733 012062 105777 167316      TSTB  @RXCSR        ;WAIT FOR SECOND PART
1734 012066 100375                    BPL   .-4           ;DITTO
1735 012070 017737 167312 001242      MOV   @RXDBUF,TEMP3 ;GET THE REST OF THE CRC
1736 012076 113737 001242 001325      MOVB  TEMP3,DATA+1  ;SET UP CRC CHARACTER
1737 012104 012716 012112      MOV   #15$, (SP)    ;SETUP FOR RETURN
1738 012110 000002                    RTI             ;RETURN
1739 012112 012737 000340 177776 15$:   MOV   #340,PS      ;RAISE PS
1740 012120 005137 007104      COM   CALBCC        ;INVERT BCC
1741 012124 023737 007104 001324      CMP   CALBCC,DATA  ;COMPARE SOFTWARE AND HARDWARE BCC
1742 012132 001401                    BEQ   .+4           ;BR IF OK
1743 012134 104004                    HLT   4             ;BCC COMPARISON ERROR
1744 012136 032737 010000 001242      BIT   #CRCERR,TEMP3 ;CHECK THE ERROR BIT
1745 012144 001401                    BEQ   .+4           ;BR IF NO ERROR
1746 012146 104004                    HLT   4             ;BCC ERROR--RECEIVER DOESN'T
1747
1748 012150 052777 000400 167234      BIS   #MRESET,@TXCSR ;AGREE WITH WHAT TX SENT
1749 012156 004737 004772      JSR   PC,SMALL     ;RESET THE DEVICE
1750 012162 000137 011700      JMP   6$           ;WAIT FOR RESET TO FINISH
1751
1752
1753
1754
1755
;***** TEST 11 *****
;THIS TEST WILL CHECK FOR ABORT SEQUENCE
;OF THE DUP IN A DATA STREAM
;*****

```





# M03

DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 39  
 DZDPDA.P11 DATA STREAM ABORT SEQUENCE TEST

1812	012446	017702	166734		MOV	@RXDBUF,R2	;READ THE BUFFER
1813	012452	032702	002000		BIT	#RABORT,R2	;TEST ABORT
1814	012456	001001			BNE	7\$	;BR IF SET
1815	012460	104010			HLT	10	;FAILED TO RECEIVE ABORT
1816	012462	012716	012372	7\$:	MOV	#11\$, (SP)	;SET UP FOR RETURN
1817	012466	052777	000400	166716	BIS	#MRESET,@TXCSR	;RESET THE DEVICE
1818	012474	004737	004772		JSR	PC,SMALL	;WAIT FOR RESET TO FINISH
1819	012500	000002		10\$:	RTI		;RETURN

```

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

```

```

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

```

1835	012502	012737	000012	001226	TST12:	MOV	#12,@TSTNO	
1836	012510	012737	013272	001216		MOV	#TST13,NEXT	
1837	012516	052777	000400	166666		BIS	#MRESET,@TXCSR	;RESET THE DEVICE
1838	012524	004737	004772			JSR	PC,SMALL	;WAIT FOR RESET TO FINISH
1839	012530	005000				CLR	R0	;CLEAR FOR SOFTWARE
1840	012532	004537	006674			JSR	R5,SETVEC	;SET UP THE VECTORS
1841	012536	012762				3\$		;RECEIVER
1842	012540	013162				14\$		;TRANSMITTER
1843	012542	340	340			.BYTE	340,340	;LEVEL
1844	012544	012737	000340	177776		MOV	#340,PS	;PROC STATUS=7
1845	012552	052777	001000	166630		BIS	#CRCEN,@PARCSR	
1846	012560	052777	000120	166616		BIS	#RCVEN!RINTEN,@RXCSR	;TURN ON RECEIVER
1847	012566	052777	004020	166616		BIS	#SEND!SYSTST,@TXCSR	;START TRANSMITTER
1848	012574	005037	177776			CLR	PS	;LOWER PS
1849	012600	105777	166606		1\$:	TSTB	@TXCSR	;CHECK FOR DONE
1850	012604	100375				BPL	1\$	;BR IF NOT YET
1851	012606	052777	000400	166600		BIS	#TSU,1,@TXDBUF	;TURN ON START OF MSG
1852	012614	052777	000100	166570		BIS	#TXINTE,@TXCSR	;TURN ON INT. ENABLE
1853	012622	012737	000764	012652		MOV	#500,68\$	;LOAD THE NUMBER
1854	012630	032777	004000	166556	66\$:	BIT	#TIMER,@TXDBUF	;CHECK THE TIMER BIT
1855	012636	001374				BNE	66\$	;BR IF SET
1856	012640	032777	004000	166546	67\$:	BIT	#TIMER,@TXDBUF	;CHECK THE BIT
1857	012646	001774				BEQ	67\$	;BR IF CLEAR
1858	012650	005327				DEC	(PC)+	;DECREMENT THE NUMBER
1859	012652	000764			68\$:	500.		;OF TIMES TO REPEAT
1860	012654	001365				BNE	66\$	;BR IF MORE TO GO
1861	012656	104001				HLT	1	;DEVICE FAILED TO INTERRUPT IN TIME
1862	012660	012706	001150		2\$:	MOV	#STACK,SP	;RESET THE STACK
1863	012664	104400				SCOPE		;SCOPE THIS TEST
1864	012666	004537	006674		20\$:	JSR	R5,SETVEC	;SET UP VECTORS
1865	012672	013112				23\$		;RECEIVER
1866	012674	006722				NO.BTRAP		;TRANSMITTER
1867	012676	340	340			.BYTE	340,340	;LEVEL



```

1868 012700 052777 000020 166476      BIS      #RCVEN, @RXCSR
1869 012706 105777 166500      21$:    TSTB    @TXCSR      ; TEST DONE
1870 012712 100375      BPL      21$      ; BR IF NOT SET
1871 012714 012777 000070 166472      MOV      #70, @TXDBUF ; PUSH OUT DATA CHARACTER
1872 012722 012737 000062 012752      MOV      #50, 73$ ; LOAD THE NUMBER
1873 012730 032777 004000 166456      71$:    BIT      #TIMER, @TXDBUF ; CHECK THE TIMER BIT
1874 012736 001374      BNE      71$      ; BR IF SET
1875 012740 032777 004000 166446      72$:    BIT      #TIMER, @TXDBUF ; CHECK THE BIT
1876 012746 001774      BEQ      72$      ; BR IF CLEAR
1877 012750 005327      DEC      (PC)+ ; DECREMENT THE NUMBER
1878 012752 000062      73$:    50. ; OF TIMES TO REPEAT
1879 012754 001365      BNE      71$      ; BR IF MORE TO GO
1880 012756 104001      HLT      1 ; FAILED TO INTERRUPT IN TIME
1881 012760 000737      BR       2$      ; FINISH
1882
1883      ; INTERRUPT SVC ROUTINES
1884
1885      ; RECEIVER
1885 012762 017704 166416      3$:    MOV      @RXCSR, R4 ; GET THE CONTROL REGISTER
1886 012766 017705 166414      MOV      @RXDBUF, R5 ; GET THE BUFFER
1887 012772 032705 000400      BIT      #RSOM, R5 ; CHECK FOR START OF MSG
1888 012776 001001      BNE      4$      ; BR IF SET
1889 013000 104011      HLT      11 ; FAILED TO RECEIVE SOM
1890 013002 032704 000200      4$:    BIT      #RXDONE, R4 ; CHECK FOR DONE
1891 013006 001001      BNE      5$      ; BR IF SET
1892 013010 104007      HLT      7 ; FALSE INTERRUPT
1893 013012 122705 000377      5$:    CMPB    #377, R5 ; CHECK DATA
1894 013016 001401      BEQ      6$      ; BR IF A MATCH
1895 013020 104002      HLT      2 ; DATA ERROR
1896 013022 012777 013032 166344      6$:    MOV      #10$, @DUPRVC ; RELOAD THE VECTOR
1897 013030 000002      7$:    RTI ; RETURN
1898 013032 017705 166350      10$:   MOV      @RXDBUF, R5 ; GET THE BUFFER
1899 013036 122705 000377      CMPB    #377, R5 ; CHECK THE CHARACTER
1900 013042 001401      BEQ      11$     ; BR IF A MATCH
1901 013044 104002      HLT      2 ; DATA ERROR
1902 013046 042777 000020 166330      11$:   BIC      #RCVEN, @RXCSR ; TURN OFF THE RECEIVER
1903 013054 012777 013064 166312      MOV      #12$, @DUPRVC ; RELOAD THE VECTOR
1904 013062 000762      BR       7$     ; RETURN
1905 013064 017704 166314      12$:   MOV      @RXCSR, R4 ; GET THE CONTROL REGISTER
1906 013070 012705 001406      MOV      #RXDBUF, R5 ; GET THE BUFFER
1907 013074 122705 000252      CMPB    #252, R5 ; CHECK THE CHARACTER
1908 013100 001402      BEQ      13$     ; BR IF A MATCH
1909 013102 104007      HLT      7 ; FALSE INTERRUPT
1910 013104 000751      BR       7$     ; DEVICE INTERRUPTED AFTER RX ENABLE
1911 013106 104007      13$:   HLT      7 ; WAS CLEARED
1912 013110 000747      BR       7$     ; GET THE CONTROL REG
1913 013112 017704 166266      23$:   MOV      @RXCSR, R4 ; GET THE BUFFER
1914 013116 017705 166264      MOV      @RXDBUF, R5 ; CHECK START OF MSG
1915 013122 032715 000400      BIT      #RSOM, (R5) ; BR IF SET
1916 013126 001001      BNE      24$     ; SOM FAILED TO SET
1917 013130 104011      HLT      11 ; CHECK DATA
1918 013132 122705 000070      24$:   CMPB    #70, R5 ; BR IF A MATCH
1919 013136 001401      BEQ      25$     ; DATA FAILED TO MATCH AFTER
1920 013140 104002      HLT      2 ; RESTARTING RECEIVER
1921
1922 013142      25$:
1923 013142 052777 000400 166242      BIS      #MRESET, @TXCSR ; RESET THE DEVICE
  
```

```

1924 013150 004737 004772 JSR PC, SMALL ;WAIT FOR RESET TO FINISH
1925 013154 012716 012660 MOV #2$, (SP) ;CRUNCH STACK
1926 013160 000002 RTI ;RETURN
1927 ;TRANSMITTER
1928 013162 105777 166224 14$: TSTB @TXCSR ;CHECK DONE
1929 013166 100401 BMI 30$ ;BR IF SET
1930 013170 104007 HLT 7 ;FALSE INTERRUPT
1931 013172 012777 000377 166214 30$: MOV #377, @TXDBUF ;LOAD A CHARACTER
1932 013200 005200 INC R0 ;IN THE # TO DO
1933 013202 022700 000002 CMP #2, R0 ;CHECK TO SEE IF ALL ARE SENT
1934 013206 001030 BNE 15$ ;BR IF MORE TO GO
1935 013210 012777 013222 166162 MOV #16$, @DUPTVC ;RELOAD THE VECT
1936 013216 005000 CLR R0 ;CLEAR CHAR COUNT
1937 013220 000423 BR 15$
1938 013222 105777 166164 16$: TSTB @TXCSR ;TEST DONE
1939 013226 100401 BMI 17$ ;BR IF SET
1940 013230 104007 HLT 7 ;FALSE INTERRUPT
1941 013232 012777 000252 166154 17$: MOV #252, @TXDBUF ;LOAD A DATA CHARACTER
1942 013240 005200 INC R0 ;INC THE # TO DO
1943 013242 022700 000003 CMP #3, R0 ;CHECK FOR ALL DONE
1944 013246 001010 BNE 15$ ;BR IF MORE TO GO
1945 013250 012777 001400 166136 MOV #TEOM!TSON, @TXDBUF ;END MSG
1946 013256 042777 000100 166126 BIC @TXINTE, @TXCSR
1947 013264 012716 012666 MOV #20$, (SP) ;CRUNCH STACK
1948 013270 000002 15$: RTI

```

```

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

```

```

;*****
;TEST 13
;*****

```

```

1959 *****
1960 013272 012737 000013 001226 †ST13: MOV #13, @TSTNO
1961 013300 012737 013732 001216 MOV @TST14, NEXT
1962 013306 052777 000400 166076 BIS @MRESET, @TXCSR ;RESET THE DEVICE
1963 013314 004737 004772 JSR PC, SMALL ;WAIT FOR RESET TO FINISH
1964 013320 012737 000340 177776 MOV #340, PS ;SET STATUS=7
1965 013326 005000 CLR R0
1966 013330 005002 CLR R2 ;SETUP FOR SOFTWARE
1967 013332 012701 000377 MOV #377, R1 ;CALCULATION OF BCC
1968 013336 012737 102010 007100 MOV #CRC, CCITT, XPOLY ;LOAD THE POLYNOMIAL
1969 013344 012737 177777 007104 MOV #-1, CALBCC ;SETUP FOR FIRST TIME
1970 013352 013737 007104 013374 1$: MOV CALBCC, 3$ ;ALLOW FOR THE NEXT CHARACTER
1971 013360 010137 013372 MOV R1, 2$ ;LOAD DATA
1972 013364 004537 006726 JSR R5, SIMBCC ;GO CALCULATE SOFTWARE BCC
1973 013370 000010 B. ;BASED ON THOSE PARAMETERS
1974 013372 000001 2$: .BLKW 1 ;DATA
1975 013374 000001 3$: .BLKW 1 ;PREVIOUS BCC
1976 013376 005200 INC R0 ;INC THE # OF CHARS TO DO
1977 013400 022700 000005 CMP #5, R0 ;ARE WE DONE?
1978 013404 001362 BNE 1$ ;BR IF NO
1979 013406 005000 CLR R0 ;CLEAR OUT HOLD

```



```

1980 013410 004537 006674 JSR RS,SETVEC ;LOAD INTERRUPT VECTORS
1981 013414 013544 6$ ;RECEIVER
1982 013416 013650 11$ ;TRANSMITTER
1983 013420 340 340 .BYTE ;LEVEL
1984 013422 052777 010377 165760 BIS #PR1SEC!377,@PARCSR ;ENTER SECONDARY MODE
1985 013430 052777 000120 165746 BIS #RCVEN!RINTEN,@RXCSR ;TURN ON RECEIVER AND INTERRUPTS
1986 013436 052777 004020 165746 BIS #SEND!SYSTST,@TXCSR ;TURN ON TRANSMITTER
1987 013444 105777 165742 20$: TSTB @TXCSR
1988 013450 100375 BPL 20$
1989 013452 012777 000400 165734 MOV #TSM,@TXDBUF ;START MESSAGE
1990 013460 052777 000100 165724 BIS #TXINTE,@TXCSR ;TURN ON INTERRUPT ENABLE
1991 013466 005037 177776 CLR PS ;LOWER PS
1992 013472 4$:
1993 013472 012737 000040 013522 MOV #32,68$ ;LOAD THE NUMBER
1994 013500 032777 004000 165706 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
1995 013506 001374 BNE 66$ ;BR IF SET
1996 013510 032777 004000 165676 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
1997 013516 001774 BEQ 67$ ;BR IF CLEAR
1998 013520 005327 DEC (PC)+ ;DECREMENT THE NUMBER
1999 013522 000040 68$: 32. ;OF TIMES TO REPEAT
2000 013524 001365 BNE 66$ ;BR IF MORE TO GO
2001 013526 104001 HLT 1 ;FAILED TO INTERRUPT IN TIME
2002 013530 5$:
2003 013530 052777 000400 165654 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2004 013536 004737 004772 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2005 013542 104400 SCOPE ;SCOPE THIS TEST
2006
2007 ;INTERRUPT SERVICE ROUTINES.
2008
2009 ;RECEIVER
2010 013544 017737 165636 001324 6$: MOV @RXDBUF,DATA ;GET THE DATA
2011 013552 120137 001324 CMPB R1,DATA ;CHECK IT
2012 013556 001401 BEQ +4 ;BR IF A MATCH
2013 013560 104002 HLT 2 ;DATA ERROR
2014 013562 005200 INC R0 ;UPDATE THE # OF CHARS TO DO
2015 013564 022700 000004 CMP #4,R0 ;CHECK FOR ALL CHARS DONE
2016 013570 001003 BNE 7$ ;BR IF MORE TO GO
2017 013572 012777 013602 165574 MOV #10$,@DUPRVC ;SETUP TO GET BCC
2018 013600 000002 7$: RTI ;RETURN
2019
2020 013602 117737 165600 001324 10$: MOVB @RXDBUF,DATA ;GET THE FIRST HALF OF BCC
2021 013610 105777 165570 TSTB @RXCSR ;WAIT FOR
2022 013614 100375 BPL -4 ;THE SECOND HALF
2023 013616 117737 165564 001325 MOVB @RXDBUF,DATA+1 ;GET THE SECOND HALF
2024 013624 005137 007104 COM CALBCC ;INVERT BCC
2025 013630 023737 007104 001324 CMP CALBCC,DATA ;CHECK IT
2026 013636 001401 BEQ +4 ;BR IF OK
2027 013640 104004 HLT 4 ;BCC COMPARE ERROR
2028 013642 012716 013530 MOV #5$,(SP) ;FINISH TEST
2029 013646 000002 RTI ;RETURN
2030
2031 ;TRANSMITTER
2032 013650 012777 000377 165536 11$: MOV #377,@TXDBUF ;LOAD A DATA CHARACTER
2033 013656 005202 INC R2 ;INC THE # OF CHARS TO DO
2034 013660 022702 000005 CMP #5,R2 ;CHECK TO SEE OF DONE
2035 013664 001017 BNE 13$ ;BR IF MORE TO GO

```

```

2036 013666 012777 013676 165504 MOV #12$, @DUPTVC ; SETUP NEXT VECTOR
2037 013674 000413 BR 13$ ; RETURN
2038 013676 012777 001000 165510 12$: MOV #TEOM, @TXDBUF ; END MSG
2039 013704 000240 NOP ; WAIT
2040 013706 000240 NOP ; DITTO
2041 013710 042777 000120 165474 BIC #SEND!TXINTE, @TXCSR ; TURN OFF TRANSMITTER
2042 013716 012777 006722 165454 MOV #NO.BTRAP, @DUPTVC ; RESET THE VECTOR
2043 013724 012716 013472 13$: MOV #4$, (SP) ; GO BACK TO WAIT LOOP
2044 013730 000002 RTI ; RETURN

```

```

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

```

```

:*****
: TEST 14
:*****

```

```

2051 :*****
2052 :
2053 : TEST 14
2054 :
2055 :*****
2056 :*****
2057 013732 012737 000014 001226 TST14: MOV #14, @TSTNO
2058 013740 012737 014104 001216 MOV #TST15, NEXT
2059 013746 052777 000400 165436 BIS #MRESET, @TXCSR ; RESET THE DEVICE
2060 013754 004737 004772 JSR PC, SMALL ; WAIT FOR RESET TO FINISH
2061 013760 012777 101026 165422 MOV #DECMOD!26!CRCEN, @PARCSR
2062 013766 052777 004000 165416 BIS #SYSTST, @TXCSR ; ENTER SYSTEM TEST MODE
2063 013774 052777 000020 165410 BIS #SEND, @TXCSR ; TURN ON TRANSMITTER
2064 014002 012777 000426 165404 MOV #TSOM!26, @TXDBUF ; OUTPUT A SYNC CHAR
2065 014010 012737 000005 014040 MOV #5, 68$ ; LOAD THE NUMBER
2066 014016 032777 004000 165370 66$: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
2067 014024 001374 BNE 66$ ; BR IF SET
2068 014026 032777 004000 165360 67$: BIT #TIMER, @TXDBUF ; CHECK THE BIT
2069 014034 001774 BEQ 67$ ; BR IF CLEAR
2070 014036 005327 DEC (PC)+ ; DECREMENT THE NUMBER
2071 014040 000005 68$: S ; OF TIMES TO REPEAT
2072 014042 001365 BNE 66$ ; BR IF MORE TO GO
2073 014044 017704 165342 MOV @TXCSR, R4 ; GET THE CSR
2074 014050 032704 000200 BIT #TXDONE, R4 ; CHECK TRANSMITTER DONE
2075 014054 001001 BNE 15$ ; BR IF SET
2076 014056 104016 HLT 16 ; TXDONE FAILED TO SET
2077 014060 032704 001000 15$: BIT #TXACT, R4 ; TEST ACTIVE
2078 014064 001001 BNE 25$ ; BR IF SET
2079 014066 104017 HLT 17 ; ACTIVE FAILED TO SET
2080 014070 25$:
2081 014070 052777 000400 165314 BIS #MRESET, @TXCSR ; RESET THE DEVICE
2082 014076 004737 004772 JSR PC, SMALL ; WAIT FOR RESET TO FINISH
2083 014102 104400 SCOPE ; SCOPE THIS TEST

```

```

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

```

```

:*****

```



```

2092
2093
2094
2095
2096
2097 014104 012737 000015 001226 TST15: MOV #15,@TSTNO
2098 014112 012737 014300 001216 MOV #TST16,NEXT
2099 014120 052777 000400 165264 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2100 014126 004737 004772 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2101 014132 012777 101026 165250 MOV #DECMOD!26!CRCEN,@PARCSR
2102 014140 052777 004000 165244 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2103 014146 052777 000020 165236 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
2104 014154 012777 000426 165232 MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2105 014162 105777 165224 1$: TSTB @TXCSR ;CHECK FOR DONE
2106 014166 100375 BPL 1$ ;BR IF NOT YET
2107 014170 012777 000426 165216 MOV #TSOM!26,@TXDBUF ;LOAD A SECOND SYNC
2108 014176 105777 165210 2$: TSTB @TXCSR ;AND NOW WAIT
2109 014202 100375 BPL 2$ ;FOR DONE AGAIN
2110 014204 012777 001000 165202 MOV #TEOM,@TXDBUF ;SET END OF MSG
2111 014212 042777 000020 165172 BIC #SEND,@TXCSR ;TURN OFF TRANSMITTER
2112 014220 012737 000025 014250 MOV #25,66$ ;LOAD THE NUMBER
2113 014226 032777 004000 165160 66$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2114 014234 001374 BNE 66$ ;BR IF SET
2115 014236 032777 004000 165150 67$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2116 014244 001774 BEQ 67$ ;BR IF CLEAR
2117 014246 005327 DEC (PC)+ ;DECREMENT THE NUMBER
2118 014250 000025 68$: 25 ;OF TIMES TO REPEAT
2119 014252 001365 BNE 66$ ;BR IF MORE TO GO
2120 014254 105777 165132 TSTB @TXCSR ;CHECK DONE
2121 014260 100401 BMI 3$ ;BR IF SET
2122 014262 104016 HLT 16 ;DONE FAILED TO SET AFTER TURNING OFF TX.
2123 014264 032777 001000 165120 3$: BIT #TXACT,@TXCSR ;CHECK ACTIVE
2124 014272 001401 BEQ 4$ ;BR IF OFF
2125 014274 104020 HLT 20 ;ACTIVE IS STILL SET-SHOULD BE RESET
2126 014276 104400 4$: SCOPE ;SCOPE FOR THIS TEST.
2127
2128
2129
2130
2131
2132
2133
2134
2135
2136
2137
2138
2139
2140 014300 012737 000016 001226 TST16: MOV #16,@TSTNO
2141 014306 012737 014646 001216 MOV #TST17,NEXT
2142 014314 052777 000400 165070 BIS #MRESET,@TXCSR ;RESET THE DEVICE
2143 014322 004737 004772 JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2144 014326 012777 101026 165054 MOV #DECMOD!26!CRCEN,@PARCSR
2145 014334 052777 004000 165050 BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2146 014342 052777 000020 165034 BIS #RCVEN,@RXCSR ;LOAD RCVEN
2147 014350 052777 000020 165034 BIS #SEND,@TXCSR ;TURN ON TRANSMITTER

```

```

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

```

```

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

```

```

2148 014356 012777 000426 165030      MOV      #TSM!26, @TXDBUF      ;OUTPUT A SYNC CHAR
2149 014364 105777 165022      1$:      TSTB     @TXCSR           ;CHECK TRANSMITTER DONE
2150 014370 100375                BPL      1$                  ;WAIT TILL SET
2151 014372 012777 000125 165014      MOV      #125, @TXDBUF       ;LOAD DATA
2152 014400 012737 000005 014430      MOV      #5, 68$            ;LOAD THE NUMBER
2153 014406 032777 004000 165000 66$:      BIT      #TIMER, @TXDBUF     ;CHECK THE TIMER BIT
2154 014414 001374                BNE      66$                 ;BR IF SET
2155 014416 032777 004000 164770 67$:      BIT      #TIMER, @TXDBUF     ;CHECK THE BIT
2156 014424 001774                BEQ      67$                 ;BR IF CLEAR
2157 014426 005327                DEC      (PC)+               ;DECREMENT THE NUMBER
2158 014430 000005                5                          ;OF TIMES TO REPEAT
2159 014432 001365                BNE      66$                 ;BR IF MORE TO GO
2160 014434 105777 164744      TSTB     @RXCSR           ;CHECK FOR RECEIVER DONE
2161 014440 100002                BPL      2$                  ;BR IF NOT SET
2162 014442 104021                HLT      21                  ;DEVICE SYNC'S UP IN LESS THAN 2 SYNC'S!!
2163 014444 000472                BR       5$                  ;LEAVE
2164 014446                2$:
2165 014446 052777 000400 164736      BIS      #MRESET, @TXCSR     ;RESET THE DEVICE
2166 014454 004737 004772      JSR      PC, SMALL          ;WAIT FOR RESET TO FINISH
2167 014460 012777 101026 164722      MOV      #RCEN!DECMOD!26, @PARCSR ;LOAD THE MODE AND SYNC CHAR
2168 014466 052777 000020 164710      BIS      #RCVEN, @RXCSR      ;TURN ON RECEIVER
2169 014474 052777 004000 164710      BIS      #SYSTST, @TXCSR     ;ENTER SYSTEM TEST MODE
2170 014502 052777 000020 164702      BIS      #SEND, @TXCSR       ;TURN ON TRANSMITTER
2171 014510 012777 000426 164676      MOV      #TSM!26, @TXDBUF     ;OUTPUT A SYNC CHAR
2172 014516 105777 164670      69$:      TSTB     @TXCSR           ;CHECK DONE
2173 014522 100375                BPL      69$                 ;BR IF NOT SET
2174 014524 012777 000426 164662      MOV      #TSM!26, @TXDBUF     ;SEND SYNC
2175 014532 105777 164654      3$:      TSTB     @TXCSR           ;CHECK DONE
2176 014536 100375                BPL      3$                  ;WAIT
2177 014540 012777 000125 164646      MOV      #125, @TXDBUF       ;LOAD DATA
2178 014546 012737 000020 014576      MOV      #20, 74$           ;LOAD THE NUMBER
2179 014554 032777 004000 164632 72$:      BIT      #TIMER, @TXDBUF     ;CHECK THE TIMER BIT
2180 014562 001374                BNE      72$                 ;BR IF SET
2181 014564 032777 004000 164622 73$:      BIT      #TIMER, @TXDBUF     ;CHECK THE BIT
2182 014572 001774                BEQ      73$                 ;BR IF CLEAR
2183 014574 005327                DEC      (PC)+               ;DECREMENT THE NUMBER
2184 014576 000020                74$:      20                          ;OF TIMES TO REPEAT
2185 014600 001365                BNE      72$                 ;BR IF MORE TO GO
2186 014602 105777 164576      TSTB     @RXCSR           ;CHECK FOR DONE
2187 014606 100401                BMI      4$                  ;BR IF SET
2188 014610 104022                HLT      22                  ;FAILED TO RECEIVE DATA
2189 014612 017737 164570 001236 4$:      MOV      @RXDBUF, TEMP1      ;READ DATA
2190 014620 122737 000125 001236      CMPB     #125, TEMP1         ;CHECK IT
2191 014626 001401                BEQ      5$                  ;BR IF MATCH
2192 014630 104022                HLT      22                  ;DATA COMPARE ERROR
2193 014632                5$:
2194 014632 052777 000400 164552      BIS      #MRESET, @TXCSR     ;RESET THE DEVICE
2195 014640 004737 004772      JSR      PC, SMALL          ;WAIT FOR RESET TO FINISH
2196 014644 104400                SCOPE                        ;SCOPE THIS TEST
2197
2198
2199
2200
2201
2202
2203
;***** TEST 17 *****
;THIS TEST PROVES THE RECEIVER WILL STRIP THE FIRST
;TWO SYNC'S AND WILL PRESENT ALL SUBSEQUENT SYNC'S.
;*****

```



```

2204      :*****
2205      :*
2206      :TEST 17
2207      :*
2208      :*****
2209      :*****
2210 014646 012737 000017 001226 TST17: MOV #17,@TSTNO
2211 014654 012737 015070 001216      MOV #TST20,NEXT
2212 014662 052777 000400 164522      BIS #MRESET,@TXCSR ;RESET THE DEVICE
2213 014670 004737 004772      JSR PC,SMALL ;WAIT FOR RESET TO FINISH
2214 014674 012777 101026 164506      MOV #DECMOD!26!CRCN,@PARCSR
2215 014702 052777 004000 164502      BIS #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2216 014710 052777 000020 164466      BIS #RCVEN,@RXCSR ;LOAD RCVEN
2217 014716 052777 000020 164466      BIS #SEND,@TXCSR ;TURN ON TRANSMITTER
2218 014724 012777 000426 164462      MOV #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2219 014732 032777 004000 164454 64$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2220 014740 001374      BNE 64$ ;BR IF SET
2221 014742 032777 004000 164444 65$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2222 014750 001774      BEQ 65$ ;BR IF CLEAR
2223 014752 105777 164434 69$: TSTB @TXCSR ;CHECK DONE
2224 014756 100375      BPL 69$ ;BR IF NOT SET
2225 014760 012777 000426 164426      MOV #TSOM!26,@TXDBUF ;SEND SYNC
2226 014766 105777 164420 70$: TSTB @TXCSR ;CHECK DONE
2227 014772 100375      BPL 70$ ;BR IF NOT SET
2228 014774 012777 000426 164412      MOV #TSOM!26,@TXDBUF ;SEND SYNC
2229 015002 012737 000020 015032      MOV #20,75$ ;LOAD THE NUMBER
2230 015010 032777 004000 164376 73$: BIT #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2231 015016 001374      BNE 73$ ;BR IF SET
2232 015020 032777 004000 164366 74$: BIT #TIMER,@TXDBUF ;CHECK THE BIT
2233 015026 001774      BEQ 74$ ;BR IF CLEAR
2234 015030 005327      DEC (PC)+ ;DECREMENT THE NUMBER
2235 015032 000020 75$: 20 ;OF TIMES TO REPEAT
2236 015034 001365      BNE 73$ ;BR IF MORE TO GO
2237 015036 105777 164342      TSTB @RXCSR ;CHECK FOR DONE
2238 015042 100401      BMI 1$ ;BR IF SET
2239 015044 104021      HLT 21 ;DONE NOT SET-DEVICE FAILED TO SYNC UP
2240 015046 117737 164334 001236 1$: MOVB @RXDBUF,TEMP1 ;READ BUFFER
2241 015054 122737 000026 001236      CMPB #26,TEMP1 ;CHECK FOR SYNC
2242 015062 001401      BEQ 2$ ;BR IF OK
2243 015064 104022      HLT 22 ;DATA ERROR
2244 015066 104400 2$: SCOPE ;SCOPE THIS TEST

```

```

***** TEST 20 *****
*THIS TEST PROVES THE DUP11 WILL
*IDLE SYNC. IDLE 64. SYNC
*****

```

```

2245      :*****
2246      :*
2247      :TEST 20
2248      :*
2249      :*****
2250      :*****
2251      :*****
2252      :*****
2253      :*
2254      :TEST 20
2255      :*
2256      :*****
2257      :*****
2258 015070 012737 000020 001226 TST20: MOV #20,@TSTNO
2259 015076 012737 015334 001216      MOV #TST21,NEXT

```

```

2260 015104 052777 000400 164300      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2261 015112 004737 004772      JSR      PC_SMALL ;WAIT FOR RESET TO FINISH
2262 015116 012777 101026 164264      MOV      #DECMOD!26!CRLEN,@PARCSR
2263 015124 052777 004000 164260      BIS      #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2264 015132 052777 000020 164244      BIS      #RCVEN,@RXCSR ;LOAD RCVEN
2265 015140 052777 000020 164244      BIS      #SEND,@TXCSR ;TURN ON TRANSMITTER
2266 015146 012777 000426 164240      MOV      #TSON!26,@TXDBUF ;OUTPUT A SYNC CHAR
2267 015154 105777 164232      64$:    TSTB   @TXCSR ;CHECK DONE
2268 015160 100375      BPL      64$ ;BR IF NOT SET
2269 015162 012777 000426 164224      MOV      #TSON!26,@TXDBUF ;SEND SYNC
2270 015170 105777 164216      65$:    TSTB   @TXCSR ;CHECK DONE
2271 015174 100375      BPL      65$ ;BR IF NOT SET
2272 015176 012777 000426 164210      MOV      #TSON!26,@TXDBUF ;SEND SYNC
2273 015204 005037 001236      CLR      TEMP1
2274 015210 005037 001240      CLR      TEMP2
2275 015214 012737 000100 001236      MOV      #64,TEMP1 ;LOAD # OF SYNC
2276 015222 012737 000010 015252      MOV      #10,70$ ;LOAD THE NUMBER
2277 015230 032777 004000 164156      68$:    BIT     #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2278 015236 001374      BNE      68$ ;BR IF SET
2279 015240 032777 004000 164146      69$:    BIT     #TIMER,@TXDBUF ;CHECK THE BIT
2280 015246 001774      BEQ      69$ ;BR IF CLEAR
2281 015250 005327      DEC      (PC)+ ;DECREMENT THE NUMBER
2282 015252 000010      70$:    10 ;OF TIMES TO REPEAT
2283 015254 001365      BNE      68$ ;BR IF MORE TO GO
2284 015256 105777 164130      1$:    TSTB   @TXCSR ;CHECK DONE
2285 015262 100401      BMI      2$ ;BR IF SET
2286 015264 104016      HLT      16 ;DONE FAILED TO SET
2287 015266 012777 000426 164120      2$:    MOV      #TSON!26,@TXDBUF ;LOAD A SYNC
2288 015274 005337 001236      DEC      TEMP1 ;LOWER THE # OF SYNC TO DO
2289 015300 001001      BNE      4$ ;BR IF MORE TO GO
2290 015302 104400      3$:    SCOPE ;SCOPE THIS TEST
2291
2292 015304 105777 164074      4$:    TSTB   @RXCSR ;CHECK RECEIVER DONE
2293 015310 100375      BPL      4$ ;WAIT TILL SET
2294 015312 017737 164070 001240      MOV      @RXDBUF,TEMP2 ;GET THE BUFFER
2295 015320 122737 000026 001240      CMPB   #26,TEMP2 ;CHECK IT FOR SYNC
2296 015326 001753      BEQ      1$ ;BR IF OK
2297 015330 104021      HLT      21 ;CHARACTER IS TEMP2 NOT A SYNC!
2298 015332 000763      BR      3$ ;LEAVE TEST
2299
2300
2301
2302
2303
2304
2305
2306
2307
2308
2309
2310
2311
2312
2313
2314 015334 012737 000021 001226      TST21: MOV     #21,@TSTNO
2315 015342 012737 016054 001216      MOV     #TST22,NEXT

```

```

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

```

```

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

```



```

2316 015350 012737 000340 177776      MOV      #340,PS          ;RAISE STATUS
2317 015356 004537 006674          JSR      R5,SETVEC      ;SET UP VECTORS
2318 015362 015642          5$          ;BASED ON
2319 015364 006722          NO.BTRAP ;THESE
2320 015366      340      340      .BYTE 340,340 ;PARAMETERS
2321
2322 015370 052777 000400 164014      BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2323 015376 004737 004772          JSR      PC,SMALL      ;WAIT FOR RESET TO FINISH
2324 015402 012777 101026 164000      MOV      #DECMOD!26!CRCEN,@PARCSR
2325 015410 052777 004000 163774      BIS      #SYSTST,@TXCSR ;ENTER SYSTEM TEST MODE
2326 015416 052777 000420 163760      BIS      #RCVEN!STPSYN,@RXCSR ;LOAD RCVEN!STPSYN
2327 015424 052777 000020 163760      BIS      #SEND,@TXCSR   ;TURN ON TRANSMITTER
2328 015432 012777 000426 163754      MOV      #TSOM!26,@TXDBUF ;OUTPUT A SYNC CHAR
2329 015440 105777 163746      64$: TSTB @TXCSR ;CHECK DONE
2330 015444 100375          BPL      64$ ;BR IF NOT SET
2331 015446 012777 000426 163740      MOV      #TSOM!26,@TXDBUF ;SEND SYNC
2332 015454 105777 163732      65$: TSTB @TXCSR ;CHECK DONE
2333 015460 100375          BPL      65$ ;BR IF NOT SET
2334 015462 012777 000426 163724      MOV      #TSOM!26,@TXDBUF ;SEND SYNC
2335 015470 105777 163716      66$: TSTB @TXCSR ;CHECK DONE
2336 015474 100375          BPL      66$ ;BR IF NOT SET
2337 015476 012777 000426 163710      MOV      #TSOM!26,@TXDBUF ;SEND SYNC
2338 015504 005037 177776          CLR      PS ;LOWER PS
2339 015510 052777 000100 163666      BIS      #RINTEN,@RXCSR ;TURN ON INTERRUPTS
2340 015516 105777 163670      1$: TSTB @TXCSR ;CHECK TX DONE
2341 015522 100375          BPL      1$ ;WAIT FOR SET
2342 015524 012777 000252 163662      MOV      #252,@TXDBUF ;LOAD A CHARACTER
2343 015532 105777 163654      2$: TSTB @TXCSR ;CHECK TX DONE
2344 015536 100375          BPL      2$ ;WAIT TO BE SET
2345 015540 012777 000026 163646      MOV      #26,@TXDBUF ;LOAD THE SYNC CHAR
2346 015546 105777 163640      3$: TSTB @TXCSR ;CHECK DONE AGAIN
2347 015552 100375          BPL      3$ ;WAIT
2348 015554 012777 000125 163632      MOV      #125,@TXDBUF ;LOAD ANOTHER CHARACTER
2349 015562 105777 163624      4$: TSTB @TXCSR ;CHECK DONE
2350 015566 100375          BPL      4$ ;WAIT
2351 015570 012777 001000 163616      MOV      #TEOM,@TXDBUF ;SET END OF MESSAGE
2352 015576 042777 000020 163606      BIC      #SEND,@TXCSR ;TURN OFF TRANSMITTER
2353 015604 012737 000050 015634      MOV      #40,71$ ;LOAD THE NUMBER
2354 015612 032777 004000 163574      69$: BIT @TIMER,@TXDBUF ;CHECK THE TIMER BIT
2355 015620 001374          BNE      69$ ;BR IF SET
2356 015622 032777 004000 163564      70$: BIT @TIMER,@TXDBUF ;CHECK THE BIT
2357 015630 001774          BEQ      70$ ;BR IF CLEAR
2358 015632 005327          DEC      (PC)+ ;DECREMENT THE NUMBER
2359 015634 000050          71$: 40. ;OF TIMES TO REPEAT
2360 015636 001365          BNE      69$ ;BR IF MORE TO GO
2361 015640 104023          HLT      23 ;FAILED TO TAKE A RECEIVER INTERRUPT
2362
2363 ;RECEIVER INTERRUPT SERVICE ROUTINE
2364 015642 017700 163536      5$: MOV @RXCSR,R0 ;READ CSR
2365 015646 017701 163534      MOV @RXDBUF,R1 ;READ BUFFER
2366 015652 032700 000200      BIT @RXDONE,R0 ;CHECK FOR DONE
2367 015656 001001          BNE      6$ ;BR IF SET
2368 015660 104024          HLT      24 ;RX DONE FAILED TO SET-ERRONEOUS INTERRUPT
2369 015662 032700 004000      6$: BIT #REACT,R0 ;CHECK FOR ACTIVE
2370 015666 001001          BNE      7$ ;BR IF SET
2371 015670 104025          HLT      25 ;RX ACTIVE FAILED TO SET

```

2372	015672			7\$:						
2373	015672	005701			TST	R1				;CHECK FOR ERROR
2374	015674	100001			BPL	10\$				;BR IF NO ERROR
2375	015676	104026			HLT	26				;RECEIVER ERROR
2376	015700	122701	000252	10\$:	CMPB	#252,R1				;CHECK FOR CORRECT DATA
2377	015704	001401			BEQ	11\$				;BR IF OK
2378	015706	104022			HLT	22				;DATA FAILED TO MATCH
2379	015710	012777	015720	11\$:	MOV	#12\$,@DUPRVC				;LOAD VECTOR
2380	015716	000455			BR	20\$				;CONTINUE
2381	015720	017700	163460	12\$:	MOV	@RXCSR,R0				;READ CSR
2382	015724	017701	163456		MOV	@RXDBUF,R1				;READ BUFFER
2383	015730	032700	000200		BIT	#RXDONE,R0				;CHECK FOR DONE
2384	015734	001001			BNE	13\$				;BR IF OK
2385	015736	104024			HLT	24				;RX DONE FAILED TO SET-ERRONEOUS INTERRUPT
2386	015740	005701		13\$:	TST	R1				;TEST FOR ERROR
2387	015742	100001			BPL	14\$				;BR IF NO ERROR
2388	015744	104026			HLT	26				;ERROR SET
2389	015746	122701	000026	14\$:	CMPB	#26,R1				;CHECK CHARACTER
2390	015752	001422			BEQ	16\$				;BR IF OK-IF NOT, THEN
2391	015754	122701	000125		CMPB	#125,R1				;CHECK FOR CLEARING SYNC
2392	015760	001402			BEQ	15\$				;BR IF A NEXT CHARACTER
2393	015762	104022			HLT	22				;ERRONEOUS CHARACTER
2394	015764	000415			BR	16\$				;BR TO END OF TEST
2395	015766	104021		15\$:	HLT	21				;STRIPPED OUT THE SYNC CHAR!!
2396	015770	012777	016000	16\$:	MOV	#21\$,@DUPRVC				;SET UP VECTOR
2397	015776	000425			BR	20\$				;LEAVE
2398	016000	017700	163400	21\$:	MOV	@RXCSR,R0				;GET CSR
2399	016004	017701	163376		MOV	@RXDBUF,R1				;GET BUFFER
2400	016010	122701	000125		CMPB	#125,R1				;CHECK DATA
2401	016014	001401			BEQ	16\$				;BR IF A MATCH
2402	016016	104022			HLT	22				;DATA COMPARE ERROR
2403	016020	032777	004000	16\$:	BIT	#REACT,@RXCSR				;TEST ACTIVE
2404	016026	001001			BNE	17\$				;BR IF ON
2405	016030	104025			HLT	25				;ACTIVE SHOULD BE ON
2406	016032			17\$:						
2407	016032	052777	000400	18\$:	BIS	#MRESET,@TXCSR				;RESET THE DEVICE
2408	016040	004737	004772		JSR	PC,SMALL				;WAIT FOR RESET TO FINISH
2409	016044	012706	001150		MOV	#STACK,SP				;RESET STACK
2410	016050	104400			SCOPE					;SCOPE THIS TEST
2411	016052	000002		20\$:	RTI					;RETURN

```

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

```

```

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

```

2425	016054	012737	000022	001226	TST2:	MOV	#22,@TSTNO		
2426	016062	012737	016536	001216		MOV	#TST23,NEXT		
2427	016070	012737	000340	177776		MOV	#340,PS		



2428	016076	005000			CLR	RO		; CLR OUT DATA POINTER
2429	016100	005001			CLR	R1		; DITTO
2430	016102	004537	006674		JSR	R5, SETVEC		; SET UP INTERRUPTS
2431	016106	016316			4\$			; RECEIVER
2432	016110	016470			17\$			; TRANSMITTER
2433	016112	340	340		.BYTE	340, 340		; LEVEL
2434								
2435	016114	052777	000400	163270	BIS	#MRESET, @TXCSR		; RESET THE DEVICE
2436	016122	004737	004772		JSR	PC, SMALL		; WAIT FOR RESET TO FINISH
2437	016126	012777	101026	163254	MOV	#DECMOD!26!CRCEN, @PARCSR		
2438	016134	052777	004000	163250	BIS	#SYSTST, @TXCSR		; ENTER SYSTEM TEST MODE
2439	016142	052777	000020	163234	BIS	#RCVEN, @RXCSR		; LOAD RCVEN
2440	016150	052777	000020	163234	BIS	#SEND, @TXCSR		; TURN ON TRANSMITTER
2441	016156	012777	000426	163230	MOV	#TSOM!26, @TXDBUF		; OUTPUT A SYNC CHAR
2442	016164	032777	004000	163222	64\$: BIT	#TIMER, @TXDBUF		; CHECK THE TIMER BIT
2443	016172	001374			BNE	64\$		; BR IF SET
2444	016174	032777	004000	163212	65\$: BIT	#TIMER, @TXDBUF		; CHECK THE TIMER BIT
2445	016202	001774			BEQ	65\$		; BR IF CLEAR
2446	016204	105777	163202		69\$: TSTB	@TXCSR		; CHECK DONE
2447	016210	100375			BPL	69\$		; BR IF NOT SET
2448	016212	012777	000426	163174	MOV	#TSOM!26, @TXDBUF		; SEND SYNC
2449	016220	005037	177776		CLR	PS		
2450	016224	052777	000100	163152	BIS	#RINTEN, @RXCSR		; TURN ON INT ENABLES
2451	016232	052777	000100	163152	BIS	#TXINTE, @TXCSR		; DITTO
2452	016240				30\$:			
2453	016240	012737	000310	016270	MOV	#200, 74\$		; LOAD THE NUMBER
2454	016246	032777	004000	163140	72\$: BIT	#TIMER, @TXDBUF		; CHECK THE TIMER BIT
2455	016254	001374			BNE	72\$		; BR IF SET
2456	016256	032777	004000	163130	73\$: BIT	#TIMER, @TXDBUF		; CHECK THE BIT
2457	016264	001774			BEQ	73\$		; BR IF CLEAR
2458	016266	005327			DEC	(PC)+		; DECREMENT THE NUMBER
2459	016270	000310			74\$: 200.			; OF TIMES TO REPEAT
2460	016272	001365			BNE	72\$		; BR IF MORE TO GO
2461	016274	104023			HLT	23		; FAILED TO FINISH TEST
2462	016276				3\$:			
2463	016276	052777	000400	163106	BIS	#MRESET, @TXCSR		; RESET THE DEVICE
2464	016304	004737	004772		JSR	PC, SMALL		; WAIT FOR RESET TO FINISH
2465	016310	012706	001150		MOV	#STACK, SP		; RESET THE STACK
2466	016314	104400			SCOPE			; SCOPE THIS TEST
2467								
2468								
2469								
2470	016316	017702	163062		4\$: MOV	@RXCSR, R2		; SAVE CSR
2471	016322	017703	163060		MOV	@RXDBUF, R3		; SAVE BUFFER
2472	016326	032702	004000		BIT	#REACT, R2		; TEST RX ACTIVE
2473	016332	001004			BNE	5\$		; BR IF OK
2474	016334	104025			HLT	25		; ACTIVE NOT SET
2475	016336	012716	016276		MOV	#3\$, (SP)		; SETUP FOR RETURN
2476	016342	000432			BR	12\$		
2477	016344	032702	000200		5\$: BIT	#RXDONE, R2		; TEST DONE
2478	016350	001004			BNE	6\$		; BR IF OK
2479	016352	104024			HLT	24		; FALSE INTERRUPT
2480	016354	012716	016276		MOV	#3\$, (SP)		; SETUP FOR RETURN
2481	016360	000423			BR	12\$		
2482	016362	005703			6\$: TST	R3		; CHECK FOR ERROR
2483	016364	100004			BPL	7\$		; BR IF NO ERROR

```

2484 016366 104026 HLT 26 ;DATA ERROR
2485 016370 012716 016276 MOV #3$, (SP) ;SET UP RETURN
2486 016374 000415 BR 12$
2487 016376 120103 7$: CMPB R1,R3 ;CHECK DATA
2488 016400 001404 BEQ 10$ ;BR IF OK
2489 016402 104022 HLT 22 ;BAD DATA
2490 016404 012716 016276 MOV #3$, (SP) ;SETUP RETURN
2491 016410 000407 BR 12$
2492 016412 005201 10$: INC R1 ;UPDATE DATA
2493 016414 001002 BNE 11$ ;BR IF MORE TO GO
2494 016416 012716 016276 MOV #3$, (SP) ;SETUP RETURN
2495 016422 012777 016432 162744 11$: MOV #22$, @DUPRVC ;SETUP NEW RETURN FOR INTERRUPT
2496 016430 000002 12$: RTI ;RETURN
2497 016432 017702 162746 22$: MOV @RXCSR, R2
2498 016436 017703 162744 MOV @RXDBUF, R3
2499 016442 005703 TST R3
2500 016444 100001 BPL 23$
2501 016446 104026 HLT 26 ;ERROR
2502 016450 120103 23$: CMPB R1,R3
2503 016452 001401 BEQ 24$
2504 016454 104022 HLT 22 ;DATA COMPARE ERROR
2505 016456 105201 24$: INCB R1
2506 016460 001363 BNE 12$
2507 016462 012716 016276 MOV #3$, (SP)
2508 016466 000760 BR 12$
2509
2510 ;TRANSMITTER
2511 016470 010077 162720 17$: MOV R0, @TXDBUF ;PUSH OUT DATA
2512 016474 105200 INCB R0 ;UPDATE IT
2513 016476 001014 BNE 21$ ;BR IF MORE
2514 016500 105777 162706 20$: TSTB @TXCSR ;CHECK FOR NEXT DONE
2515 016504 100375 BPL 20$ ;WAIT
2516 016506 052777 001000 162700 BIS #TEOM, @TXDBUF ;END MSG
2517 016514 042777 000120 162670 BIC #SEND!TXINTE, @TXCSR ;SHUT OF TRANSMITTER
2518 016522 012777 006722 162650 MOV #NO.BTRAP, @DUPTVC ;RESET VECTOR ADRS
2519 016530 012716 016240 21$: MOV #30$, (SP)
2520 016534 000002 RTI ;RETURN
2521
2522
2523
2524 ;***** TEST 23 *****
2525 ;*THIS TEST PROVES THAT A BINARY COUNT
2526 ;*PATTERN CAN BE RUN IN DEC MODE
2527 ;*WITH A BCC CALCULATION USING
2528 ;*THE CRC16 POLYNOMIAL
2529 ;*****
2530 ;*****
2531 ;*
2532 ;* TEST 23
2533 ;*
2534 ;*****
2535 ;*****
2536 016536 012737 000023 001226 TST23: MOV #23, @TSTNO
2537 016544 012737 017360 001216 MOV #TST24, NEXT
2538 016552 012737 000340 177776 MOV #340, PS
2539 016560 005000 CLR R0

```



2540	016562	012737	120001	007100	MOV	#CRC16,XPOLY	;SET THE POLYNOMIAL
2541	016570	005037	007104		CLR	CALBCC	;CLEAR OUT OLD BCC
2542	016574	013737	007104	016616	MOV	CALBCC,35\$	;LOAD BCC
2543	016602	010037	016614		MOV	R0,2\$	;LOAD DATA
2544	016606	004537	006726		JSR	R5,SIMBCC	;CALCULATE A SOFTWARE BCC
2545	016612	000010			B.		;BASED
2546	016614	000000			.WORD	0	;ON THESE
2547	016616	000000		2\$:	.WORD	0	;PARAMETERS
2548	016620	105200		35\$:	INCB	R0	;UPDATE DATA
2549	016622	001364			BNE	1\$	;BR IF MORE TO GO
2550	016624	005000			CLR	R0	;CLR OUT DATA POINTER
2551	016626	005001			CLR	R1	;DITTO
2552	016630	004537	006674		JSR	R5,SETVEC	;SET UP INTERRUPTS
2553	016634	017044			4\$		;RECEIVER
2554	016636	017312			17\$		;TRANSMITTER
2555	016640	340	340		.BYTE	340,340	;LEVEL
2556							
2557	016642	052777	000400	162542	BIS	#MRESET,@TXCSR	;RESET THE DEVICE
2558	016650	004737	004772		JSR	PC.SMALL	;WAIT FOR RESET TO FINISH
2559	016654	012777	100026	162526	MOV	#DECMOD!26,@PARCSR	;LOAD THE MODE AND SYNC CHARACTER
2560	016662	052777	004000	162522	BIS	#SYSTST,@TXCSR	;ENTER SYSTEM TEST MODE
2561	016670	052777	000020	162506	BIS	#RCVEN,@RXCSR	;LOAD RCVEN
2562	016676	052777	000020	162506	BIS	#SEND,@TXCSR	;TURN ON TRANSMITTER
2563	016704	012777	000426	162502	MOV	#TSOM!26,@TXDBUF	;OUTPUT A SYNC CHAR
2564	016712	032777	004000	162474	64\$:	BIT	#TIMER,@TXDBUF ;CHECK THE TIMER BIT
2565	016720	001374			BNE	64\$	;BR IF SET
2566	016722	032777	004000	162464	65\$:	BIT	#TIMER,@TXDBUF ;CHECK THE TIMER BIT
2567	016730	001774			BEQ	65\$	;BR IF CLEAR
2568	016732	105777	162454		69\$:	TSTB	@TXCSR ;CHECK DONE
2569	016736	100375			BPL	69\$	;BR IF NOT SET
2570	016740	012777	000426	162446	MOV	#TSOM!26,@TXDBUF	;SEND SYNC
2571	016746	005037	177776		CLR	PS	
2572	016752	052777	000100	162424	BIS	#RINTEN,@RXCSR	;TURN ON INT ENABLES
2573	016760	052777	000100	162424	BIS	#TXINTE,@TXCSR	;DITTO
2574	016766				30\$:		
2575	016766	012737	000310	017016	MOV	#200,74\$	;LOAD THE NUMBER
2576	016774	032777	004000	162412	72\$:	BIT	#TIMER,@TXDBUF ;CHECK THE TIMER BIT
2577	017002	001374			BNE	72\$	;BR IF SET
2578	017004	032777	004000	162402	73\$:	BIT	#TIMER,@TXDBUF ;CHECK THE BIT
2579	017012	001774			BEQ	73\$	;BR IF CLEAR
2580	017014	005327			DEC	(PC)+	;DECREMENT THE NUMBER
2581	017016	000310			200.		;OF TIMES TO REPEAT
2582	017020	001365			74\$:	BNE	72\$ ;BR IF MORE TO GO
2583	017022	104023			HLT	23	;FAILED TO FINISH TEST
2584	017024				3\$:		
2585	017024	052777	000400	162360	BIS	#MRESET,@TXCSR	;RESET THE DEVICE
2586	017032	004737	004772		JSR	PC.SMALL	;WAIT FOR RESET TO FINISH
2587	017036	012706	001150		MOV	#STACK,SP	;RESET THE STACK
2588	017042	104400			SCOPE		;SCOPE THIS TEST
2589							
2590							
2591							
2592	017044	017702	162334		4\$:	MOV	@RXCSR,R2 ;SAVE CSR
2593	017050	017703	162332		MOV	@RXDBUF,R3	;SAVE BUFFER
2594	017054	032702	004000		BIT	#REACT,R2	;TEST RX ACTIVE
2595	017060	001004			BNE	5\$	;BR IF OK





```

2652 017276 105201 24S: INCB R1
2653 017300 001327 BNE 12S
2654 017302 012777 017162 162064 MOV #13S, @DUPRVC
2655 017310 000723 BR 12S
2656
2657 : TRANSMITTER
2658 017312 010077 162076 17S: MOV RO, @TXDBUF ; PUSH OUT DATA
2659 017316 105200 INCB RO ; UPDATE IT
2660 017320 001014 BNE 21S ; BR IF MORE
2661 017322 105777 162064 20S: TSTB @TXCSR ; CHECK FOR NEXT DONE
2662 017326 100375 BPL 20S ; WAIT
2663 017330 052777 001000 162056 BIS #TEOM, @TXDBUF ; END MSG
2664 017336 042777 000120 162046 BIC #SEND, TXINTE, @TXCSR ; SHUT OF TRANSMITTER
2665 017344 012777 006722 162026 MOV #NO. BTRAP, @DUPTVC ; RESET VECTOR ADRS
2666 017352 012716 016766 21S: MOV #30S, (SP)
2667 017356 000002 RTI ; RETURN
2668
2669
2670
2671
2672 :***** TEST 24 *****
2673 :*TEST TO PROVE THE DEVICE IDLES SYNCs AND
2674 :*WILL SHIFT OUT DATA AT THE APPROPRIATE TIME
2675 :*****
2676
2677 :*****
2678 :*
2679 :* TEST 24
2680 :*
2681 :*****
2682 :*****
2683 017360 012737 000024 001226 TST24: MOV #24, @TSTNO
2684 017366 012737 017556 001216 MOV #TST25, NEXT
2685 017374 052777 000400 162010 BIS #MRESET, @TXCSR ; RESET THE DEVICE
2686 017402 004737 004772 JSR PC, SMALL ; WAIT FOR RESET TO FINISH
2687 017406 052777 014000 161776 BIS #MMODE, @TXCSR ; ENTER MAINT MODE
2688 017414 012777 000020 161762 MOV #RCVEN, @RXCSR ; TURN ON RECEIVER
2689 017422 012777 100026 161760 MOV #DECMOD!26, @PARCSR ; ENTER DECMODE AND SYNC CHAR
2690 017430 052777 000020 161754 BIS #SEND, @TXCSR ; TURN ON TRANSMITTER
2691 017436 012777 000426 161750 MOV #TSOM!26, @TXDBUF ; PUSH OUT SYNCs
2692 017444 104412 000044 PKCLK 36.
2693 017450 012777 000252 161736 MOV #252, @TXDBUF ; LOAD DATA
2694 017456 104412 000024 PKCLK 20. ; PUSH OUT ANOTHER SYNC
2695 017462 105777 161716 TSTB @RXCSR ; CHECK TO SEE IF SYNC ARRIVED
2696 017466 100401 BMI 15 ; BR IF YES
2697 017470 104021 HLT 21
2698 017472 017737 161710 001324 1S: MOV @RXDBUF, DATA ; GET THE REC CHAR
2699 017500 122737 000026 001324 CMPB #26, DATA ; CHECK FOR SYNC
2700 017506 001401 BEQ 2S ; BR IF MATCH
2701 017510 104021 HLT 21 ; FAILED TO RECEIVE THIRD SYNC
2702 017512 042777 000020 161672 2S: BIC #SEND, @TXCSR ; TURN OFF TRANSMITTER
2703 017520 104412 000016 PKCLK 14. ; PUSH OUT DATA
2704 017524 105777 161654 TSTB @RXCSR ; CHECK FOR REC DATA
2705 017530 100401 BMI 3S ; BR IF YES
2706 017532 104026 HLT 26 ; FAILED TO GET A DATA DONE
2707 017534 017737 161646 001324 3S: MOV @RXDBUF, DATA ; GET THE DATA

```





```

2764 020006 001401          BEQ      6$          ;BR IF OK
2765 020010 104022          HLT      22          ;SYNC CHAR DOES NOT MATCH SENT
2766 020012          6$:
2767 020012 012702 000003      MOV      #3,R2      ;SET UP FOR NEXT SYNC
2768 020016 005000          CLR      R0          ;DITTO
2769 020020 105201          INCB    R1          ;DITTO
2770 020022 110137 001236      MOVB    R1,TEMP1
2771 020026 110137 001240      MOVB    R1,TEMP2
2772 020032 001300          BNE     1$          ;BR IF MORE TO GO
2773 020034 104400          SCOPE          ;SCOPE THIS TEST
2774
2775
2776
2777

```

```

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

```

```

*****
*
* TEST 26
*
*****

```

```

2786
2787 020036 012737 000026 001226  TST26: MOV      #26,@TSTNO
2788 020044 012737 020376 001216      MOV      #TST27,NEXT
2789 020052 012737 000340 177776      MOV      #340,PS          ;RAISE PROCESSOR STATUS
2790 020060 004537 006674          JSR      RS,SETVEC        ;SETUP VECTORS
2791 020064 020340          6$          ;RECEIVER
2792 020066 020272          3$          ;TRANSMITTER
2793 020070          340          ;LEVEL
2794 020072 005001          CLR      R1          ;CLEAR CHAR COUNT
2795
2796 020074 052777 000400 161310      BIS      #MRESET,@TXCSR   ;RESET THE DEVICE
2797 020102 004737 004772          JSR      PC,SMALL        ;WAIT FOR RESET TO FINISH
2798 020106 012777 100026 161274      MOV      #DECMOD!26,@PARCSR ;LOAD THE MODE AND SYNC CHARACTER
2799 020114 052777 004000 161270      BIS      #SYSTST,@TXCSR   ;ENTER SYSTEM TEST MODE
2800 020122 052777 000020 161254      BIS      #RCVEN,@RXCSR    ;LOAD RCVEN
2801 020130 052777 000020 161254      BIS      #SEND,@TXCSR     ;TURN ON TRANSMITTER
2802 020136 012777 000426 161250      MOV      #TSON!26,@TXDBUF ;OUTPUT A SYNC CHAR
2803 020144 105777 161242          64$: TSTB    @TXCSR          ;CHECK DONE
2804 020150 100375          BPL     64$          ;BR IF NOT SET
2805 020152 012777 000426 161234      MOV      #TSON!26,@TXDBUF ;SEND SYNC
2806 020160 105777 161226          65$: TSTB    @TXCSR          ;CHECK DONE
2807 020164 100375          BPL     65$          ;BR IF NOT SET
2808 020166 012777 000426 161220      MOV      #TSON!26,@TXDBUF ;SEND SYNC
2809 020174 005037 177776          CLR      PS          ;LOWER PROCESSOR STSTATUS
2810 020200 052777 000100 161176      BIS      #RINTEN,@RXCSR   ;TURN ON INTERRUPT ENABLES
2811 020206 052777 000100 161176      BIS      #TXINTE,@TXCSR   ;DITTO
2812 020214
2813 020214 012737 000040 020244          1$: MOV      #32,70$        ;LOAD THE NUMBER
2814 020222 032777 004000 161164          68$: BIT      #TIMER,@TXDBUF ;CHECK THE TIMER BIT
2815 020230 001374          BNE     68$          ;BR IF SET
2816 020232 032777 004000 161154          69$: BIT      #TIMER,@TXDBUF ;CHECK THE BIT
2817 020240 001774          BEQ     69$          ;BR IF CLEAR
2818 020242 005327          DEC     (PC)+        ;DECREMENT THE NUMBER
2819 020244 000040          70$: 32.          ;OF TIMES TO REPEAT

```

```

2820 020246 001365      BNE      68$      ;BR IF MORE TO GO
2821 020250 104023      HLT      23      ;FAILED TO FINISH TEST
2822 020252                2$:
2823 020252 052777 000400 161132  BIS      #MRESET,@TXCSR ;RESET THE DEVICE
2824 020260 004737 004772                JSR      PC,SMALL    ;WAIT FOR RESET TO FINISH
2825 020264 012706 001150                MOV      #STACK,SP  ;RESET THE STACK
2826 020270 104400                SCOPE

```

: INTERRUPT SERVICE ROUTINES

```

2830                : TRANSMITTER
2831 020272 005000      3$: CLR      R0      ;CLEAR DATA
2832 020274 010077 161114  MOV      R0,@TXDBUF ;LOAD DATA TO BUFFER
2833 020300 012777 020310 161072  MOV      #4$,@DUPTVC ;SETUP FOR NEXT INTERRUPT
2834 020306 000411                BR       5$      ;LEAVE
2835 020310 012777 001000 161076  4$: MOV      #TEOM,@TXDBUF ;END OF MSG--OUTPUT CRC
2836 020316 042777 000120 161066  BIC      #SEND!TXINTE,@TXCSR ;TRUN OFF THE
2837 020324 012777 006722 161046  MOV      #NO.BTRAP,@DUPTVC ;TRANSMITTER AND TXINTEN
2838 020332 012716 020214      5$: MOV      #1$, (SP) ;SETUP TO RETURN
2839 020336 000002                RTI      ;RETURN

```

: RECEIVER

```

2841                : RECEIVER
2842 020340 017737 161042 001324  6$: MOV      @RXDBUF,DATA ;GET THE DATA
2843 020346 005201                INC      R1      ;CHECK FOR LAST CHAR
2844 020350 022701 000004                CMP      #4,R1    ;AND BRANCH IF
2845 020354 001007                BNE     10$      ;NOT YET
2846 020356 032737 010000 001324  BIT      #CRCERR,DATA ;CHECK FOR CRC ERROR
2847 020364 001401                BEQ     7$      ;BR IF CRC ERROR SEEN
2848 020366 104014                HLT     14      ;FAILED TO CATCH CRC ERROR!!!!
2849 020370 012716 020252      7$: MOV      #2$, (SP) ;FINISH TEST
2850 020374 000002      10$: RTI     ;RETURN

```

```

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

```

```

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

```

```

2865 020376 012737 000027 001226  TST27: MOV      #27,@TSTNO
2866 020404 012737 021402 001216  MOV      #TST30,NEXT
2867 020412 012737 000340 177776  MOV      #340,PS    ;RAISE PROCESSOR STATUS
2868 020420 004537 006674                JSR      R5,SETVEC  ;SET UP VECTORS
2869 020424 021032                10$:                ;BASED ON
2870 020426 020654                2$:                ;THESE
2871 020430                340 340            ;PARAMETERS
2872 020432 005037 001236  CLR      TEMP1
2873 020436 005037 001240  CLR      TEMP2
2874 020442 005037 001242  CLR      TEMP3
2875 020446 005037 001244  CLR      TEMP4

```



```

2876 020452 005037 001246 CLR TEMP5
2877
2878 020456 052777 000400 160726 BIS #MRESET, @TXCSR ; RESET THE DEVICE
2879 020464 004737 004772 JSR PC_SMALL ; WAIT FOR RESET TO FINISH
2880 020470 012777 100026 160712 MOV #DECMOD!26, @PARCSR ; LOAD THE MODE AND SYNC CHARACTER
2881 020476 052777 004000 160706 BIS #SYSTST, @TXCSR ; ENTER SYSTEM TEST MODE
2882 020504 052777 000420 160672 BIS #RCVEN!STPSYN, @RXCSR ; LOAD RCVEN!STPSYN
2883 020512 052777 000020 160672 BIS #SEND, @TXCSR ; TURN ON TRANSMITTER
2884 020520 012777 000426 160666 MOV #TSOM!26, @TXDBUF ; OUTPUT A SYNC CHAR
2885 020526 105777 160660 64$: TSTB @TXCSR ; CHECK DONE
2886 020532 100375 BPL 64$ ; BR IF NOT SET
2887 020534 012777 000426 160652 MOV #TSOM!26, @TXDBUF ; SEND SYNC
2888 020542 105777 160644 65$: TSTB @TXCSR ; CHECK DONE
2889 020546 100375 BPL 65$ ; BR IF NOT SET
2890 020550 012777 000426 160636 MOV #TSOM!26, @TXDBUF ; SEND SYNC
2891 020556 052777 000100 160620 BIS #RINTEN, @RXCSR ; TURN ON INTERRUPTS
2892 020564 052777 000100 160620 BIS #TXINTE, @TXCSR ; DITTO
2893 020572 005037 177776 CLR PS ; LOWER PROCESSOR STATUS
2894 020576 100$:
2895 020576 012737 000144 020626 MOV #100, 70$ ; LOAD THE NUMBER
2896 020604 032777 004000 160602 68$: BIT #TIMER, @TXDBUF ; CHECK THE TIMER BIT
2897 020612 001374 BNE 68$ ; BR IF SET
2898 020614 032777 004000 160572 69$: BIT #TIMER, @TXDBUF ; CHECK THE BIT
2899 020622 001774 BEQ 69$ ; BR IF CLEAR
2900 020624 005327 DEC (PC)+ ; DECREMENT THE NUMBER
2901 020626 000144 70$: 100. ; OF TIMES TO REPEAT
2902 020630 001365 BNE 68$ ; BR IF MORE TO GO
2903 020632 104023 HLT 23 ; FAILED TO FINISH TEST
2904 020634 1$:
2905 020634 052777 000400 160550 BIS #MRESET, @TXCSR ; RESET THE DEVICE
2906 020642 004737 004772 JSR PC_SMALL ; WAIT FOR RESET TO FINISH
2907 020646 012706 001150 MOV #STACK, SP ; RESET THE STACK
2908 020652 104400 SCOPE ; SCOPE THIS TEST
2909
2910 ; INTERRUPT SERVICE ROUTINES
2911 ; TRANSMITTER
2912
2913 020654 012777 000252 160532 2$: MOV #252, @TXDBUF ; LOAD FIRST DATA CHAR
2914 020662 012737 000026 001236 MOV #26, TEMP1 ; LOAD DATA
2915 020670 012777 020700 160502 MOV #3$, @DUPTVC ; RELOAD VECTOR
2916 020676 000452 BR 7$ ; LEAVE
2917 020700 013777 001236 160506 3$: MOV TEMP1, @TXDBUF ; MOV DATA TO BUFFER
2918 020706 105237 001236 INCB TEMP1 ; UPDATE DATA
2919 020712 122737 000032 001236 CMPB #32, TEMP1 ; CHECK FOR DONE
2920 020720 001041 BNE 7$ ; BR IF MORE TO SEND
2921 020722 012777 020732 160450 MOV #4$, @DUPTVC ; RELOAD VECTOR
2922 020730 000435 BR 7$ ; RETURN
2923 020732 012777 001000 160454 4$: MOV #TEOM, @TXDBUF ; PUT OUT BCC
2924 020740 012777 020750 160432 MOV #5$, @DUPTVC ; RELOAD VECTOR
2925 020746 000426 BR 7$ ; RETURN
2926 020750 013777 001240 160436 5$: MOV TEMP2, @TXDBUF ; LOAD DATA
2927 020756 105237 001240 INCB TEMP2 ; UPDATE DATA
2928 020762 122737 000100 001240 CMPB #100, TEMP2 ; CHECK FOR FINISH
2929 020770 001015 BNE 7$ ; BR IF MORE TO GO
2930 020772 012777 021002 160400 MOV #6$, @DUPTVC ; RELOAD VECTOR
2931 021000 000411 BR 7$ ; RETURN

```

```

2932 021002 012777 001000 160404 6$: MOV #TEOM, @TXDBUF ; PUSH OUT DATA BCC
2933 021010 042777 000120 160374 BIC #SEND!TXINTE, @TXCSR ; SHUT DOWN TRANSMITTER
2934 021016 012777 006722 160354 MOV #NO.BTRAP, @DUPTVC ; RESET VECTOR
2935 021024 012716 020576 7$: MOV #100$, (SP) ; SETUP RETURN
2936 021030 000002 RTI ; RETURN
2937
2938 ; RECEIVER
2939
2940 021032 017737 160346 001242 10$: MOV @RXCSR, TEMP3 ; SAVE CSR
2941 021040 017737 160342 001244 MOV @RXDBUF, TEMP4 ; SAVE BUFFER
2942 021046 105737 001242 TSTB TEMP3 ; CHECK FOR DONE
2943 021052 100401 BMI 11$ ; BR IF SET
2944 021054 104024 HLT 24 ; FALSE INTERRUPT
2945 021056 005737 001244 11$: TST TEMP4 ; CHECK FOR ERROR
2946 021062 100001 BPL 12$ ; BR IF NO ERROR
2947 021064 104026 HLT 26 ; RECEIVER ERROR
2948 021066 122737 000252 001244 12$: CMPB #252, TEMP4 ; CHECK DATA
2949 021074 001401 BEQ 13$ ; BR IF A MATCH
2950 021076 104022 HLT 22 ; DATA COMPARE ERROR
2951 021100 012737 000026 001246 13$: MOV #26, TEMP5 ; LOAD NEXT EXPECTED
2952 021106 012777 021116 160260 MOV #14$, @DUPRVC ; RELOAD VECTOR
2953 021114 000531 BR 26$ ; LEAVE
2954 021116 017737 160264 001244 14$: MOV @RXDBUF, TEMP4 ; GET DATA
2955 021124 005737 001244 TST TEMP4 ; CHECK FOR ERROR
2956 021130 100001 BPL 15$ ; BR IF NO ERROR
2957 021132 104026 HLT 26 ; DATA ERROR
2958 021134 123737 001246 001244 15$: CMPB TEMP5, TEMP4 ; CHECK DATA
2959 021142 001401 BEQ 16$ ; BR IF A MATCH
2960 021144 104022 HLT 22 ; DATA COMPARE ERROR
2961 021146 105237 001246 16$: INCB TEMP5 ; UPDATE DATA
2962 021152 122737 000032 001246 CMPB #32, TEMP5 ; CHECK FOR FIRST PART FINISH
2963 021160 001107 BNE 26$ ; BR IF MORE TO GO
2964 021162 012777 021172 160204 MOV #17$, @DUPRVC ; SET UP NEXT VECTOR
2965 021170 000503 BR 26$ ; LEAVE
2966 021172 017737 160210 001244 17$: MOV @RXDBUF, TEMP4 ; GET THE BUFFER
2967 021200 005737 001244 TST TEMP4 ; TEST FOR ERROR
2968 021204 100001 BPL .+4 ; BR IF OK
2969 021206 104026 HLT 26 ; RECEIVER ERROR
2970 021210 012777 021220 160156 MOV #18$, @DUPRVC ; RELOAD THE VECTOR
2971 021216 000470 BR 26$ ; LEAVE
2972 021220 017737 160162 001324 18$: MOV @RXDBUF, DATA ; GET DATA
2973 021226 032737 010000 001324 BIT #CRCERR, DATA ; CHECK FOR CRC ERROR
2974 021234 001001 BNE 19$ ; BR IF OK
2975 021236 104014 HLT 14 ; CRC ERROR!!!!!!
2976 021240 012777 021254 160126 19$: MOV #20$, @DUPRVC ; SET UP VECTOR
2977 021246 005037 001330 CLR MIND ; SETUP FOR NEXT DATA
2978 021252 000452 BR 26$ ; LEAVE
2979 021254 017737 160126 001244 20$: MOV @RXDBUF, TEMP4 ; GET DATA
2980 021262 005737 001244 TST TEMP4 ; CHECK FOR ERROR
2981 021266 100001 BPL 21$ ; BR IF NO ERROR
2982 021270 104026 HLT 26 ; RECEIVER ERROR
2983 021272 123737 001330 001244 21$: CMPB MIND, TEMP4 ; CHECK DATA
2984 021300 001401 BEQ 22$ ; BR IF A MATCH
2985 021302 104022 HLT 22 ; DATA ERROR
2986 021304 105237 001330 22$: INCB MIND ; UPDATE SOFTWARE DATA
2987 021310 122737 000100 001330 CMPB #100, MIND ; CHECK FOR FINISH

```



```

2988 021316 001030          BNE      26$          ;BR IF MORE TO GO
2989 021320 012777 021330 160046  MOV     #23$,@DUPRVC ;RELOAD FINAL VECTOR
2990 021326 000424          BR       26$          ;LEAVE
2991 021330 017737 160052 001244 23$:  MOV     @RXDBUF,TEMP4 ;GET DATA
2992 021336 005737 001244          TST     TEMP4         ;CHECK FOR ERROR
2993 021342 100001          BPL     24$          ;BR IF OK
2994 021344 104026          HLT     26           ;RECEIVER ERROR ON FIRST OCTET
2995                                ;OF SECOND BCC
2996 021346 105777 160032          24$:  TSTB   @RXCSR         ;TEST DONE
2997 021352 100375          BPL     24$          ;BR IF NOT SET
2998 021354 017737 160026 001324  MOV     @RXDBUF,DATA ;GET SECOND BCC OCTET
2999 021362 032737 010000 001324  BIT     #CRCERR,DATA ;CHECK FOR BCC ERROR
3000 021370 001001          BNE     25$          ;BR IF OK
3001 021372 104014          HLT     14          ;BCC ERROR ON SECOND PART OF MSG
3002 021374 012716 020634          25$:  MOV     #1$, (SP)   ;SETUP TO FINISH TEST
3003 021400 000002          26$:  RTI              ;RETURN
3004
3005
3006
3007
3008
3009
3010
3011
3012
3013
3014
3015
3016
3017
3018
3019
3020
3021
3022

```

```

;***** TEST 30 *****
;THIS TEST IS AN AID FOR DEBUGGING CRC
;ERRORS. A CHARACTER IS LOADED INTO THE
;DUP AND PUSHED OUT BIT BY BIT WHILE
;ALLOWING THE OPERATOR TO MONITOR THE CRC
;CHARACTER AS IT IS GENERATED. THE DATA CHARACTER
;CAN ALSO BE CHANGED BY THE OPERATOR.
;PUT SW09=1 TO LOCK ON BITS. TO CONTINUE HIT
;ANY KEY ON THE TTY. AFTER 16 TIMES PUT DOWN SW09 TO LEAVE
;*****

```

```

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

```

```

3023 021402 012737 000030 001226  TST30: MOV     #30,@TSTNO
3024 021410 012737 002764 001216  MOV     #.EOP,NEXT
3025 021416 052777 000400 157766  BIS     #MRESET,@TXCSR ;RESET THE DEVICE
3026 021424 004737 004772          JSR     PC,SMALL      ;WAIT FOR RESET TO FINISH
3027 021430 012737 120001 007100  MOV     #CRC16,XPOLY  ;LOAD THE POLYNOMIAL
3028 021436 012737 000125 021604  MOV     #125,3$       ;LOAD DATA TO SOFTWARE BCC-CHANGE CHARACTER HERE
3029 021444 013737 021604 001252  MOV     3$,SAVR1
3030 021452 005037 007104          CLR     CALBCC        ;CLEAR FOR SOFTWARE BCC
3031 021456 013737 007104 021606  MOV     CALBCC,4$
3032 021464 005037 001242          CLR     TEMP3
3033 021470 005037 001244          CLR     TEMP4        ;CLEAR BIT COUNTER
3034 021474 005037 001246          CLR     TEMP5
3035 021500 012777 100026 157702  MOV     #DECMOD!26,@PARCSR ;LOAD MODE AND SYNC CHARACTER
3036 021506 052777 014000 157676  BIS     #MMODE,@TXCSR  ;ENTER MAINT MODE-PROGRAM CLOCKING
3037 021514 052777 000420 157662  BIS     #RCVEN!STPSYN,@RXCSR ;TURN ON RECEIVER
3038 021522 052777 000020 157662  BIS     #SEND,@TXCSR  ;TURN ON TRANSMITTER
3039 021530 012777 000426 157656  MOV     #TSOM!26,@TXDBUF ;LOAD A SYNC
3040 021536 104412 000044          PKCLK  ,36           ;PUSH OUT 2 SYNC
3041 021542 013777 021604 157644  MOV     3$,@TXDBUF   ;LOAD DATA
3042 021550 104412 000020          PKCLK  ,16           ;PUSH OUT ANOTHER SYNC
3043 021554 104412 000002          1$:  PKCLK  ,2           ;PUSH OUT A BIT

```

DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 61  
 DZDPDA.P11 CRC DEBUGGING AID TEST

```

3044 021560 013737 001244 001254      MOV    TEMP4,SAVR2      ;SET UP TO TYPE
3045 021566 005237 001242                INC    TEMP3
3046 021572 005237 001244                INC    TEMP4            ;UPDATE BIT COUNTER
3047 021576 004537 006726      2$:   JSR    R5,SIMBCC      ;CALCULATE SOFTWARE BCC BASED ON THESE PARAMETERS
3048 021602 000001                1      ;SHIFTS
3049 021604 000000      3$:   .WORD 0          ;DATA
3050 021606 000000      4$:   .WORD 0          ;PREVIOUS BCC
3051 021610 004737 021706      JSR    PC,5$           ;CHECK TO SEE IF WE SHOULD WAIT FOR SCOPING
3052 021614 000241                CLC
3053 021616 106037 021604      RORB   3$             ;CLEAR FOR NEXT ROTATE
3054 021622 013737 007104 021606      MOV    CALBCC,4$      ;SET UP THE NEXT BIT
3055 021630 022737 000006 001244      CMP    #6,TEMP4       ;FOR THE SOFTWARE BCC
3056 021636 001002                BNE    .+6
3057 021640 005077 157550      CLR    @TXDBUF
3058 021644 022737 000014 001242      CMP    #12.,TEMP3
3059 021652 001003                BNE    12$
3060 021654 012777 001000 157532      MOV    #TEOM,@TXDBUF
3061 021662 022737 000020 001244      12$:  CMP    #16.,TEMP4     ;ALL DONE WITH THE CHARACTER?
3062                                ;INCREASE COMPARE # TO FORCE
3063                                ;CRC OUT OF THE GENERATOR
3064 021670 001331                BNE    1$
3065 021672 052777 000400 157512      BIS    #MRESET,@TXCSR ;BR IF MORE TO GO
3066 021700 004737 004772      JSR    PC,SMALL       ;RESET THE DEVICE
3067 021704 104400                SCOPE                  ;WAIT FOR RESET TO FINISH
3068                                ;SCOPE THIS TEST
3069 021706 032777 001000 157266      5$:   BIT    #SW09,@SWR   ;SW09=1?
3070 021714 001432                BEQ    6$             ;BR IF NO
3071 021716 013704 007104                MOV    CALBCC,R4     ;THE DATA CHARACTER IS ALWAYS
3072 021722 012737 000001 001256      MOV    #1,SAVR3      ;FOLLOWED BY A ZERO. THE DATA IN
3073 021730 000241                CLC                    ;CRC SHOWS WHICH BIT OF THE 2 CHARS
3074 021732 006004                ROR    R4             ;IS BEING GENERATED
3075 021734 006137 001256      ROL    SAVR3
3076 021740 103374                BCC    11$
3077 021742 105737 001246      TSTB   TEMPS
3078 021746 001006                BNE    10$
3079 021750 104402 023100      TYPE   ,EM17         ;TYPE MSG
3080 021754 104402 023127      TYPE   ,MH1          ;TYPE HEADER
3081 021760 105137 001246      COMB   TEMPS
3082 021764 104410                10$:  CONVRT
3083 021766 023450                DT1
3084 021770 105777 157210      7$:   TSTB   @TKCSR      ;CHECK TTY DONE--GO SCOPE THE CRC GENERATOR
3085 021774 100375                BPL    7$            ;BR IF NOT YET
3086 021776 017701 157204      MOV    @TKDBR,R1     ;READ THE BUFFER
3087 022002 000207      6$:   RTS    PC        ;RETURN
3088
3089

```



3090

(1)	022004	042377	050125	043040	EM1:	.ASCIZ	<377>/DUP FAILED TO INTERRUPT IN TIME /	
(1)	022046	042377	052101	020101	EM2:	.ASCIZ	<377>/DATA COMPARE ERROR /	
(1)	022073	377	047105	020104	EM3:	.ASCIZ	<377>/END OF MESSAGE /	
(1)	022114	041377	041503	042440	EM4:	.ASCIZ	<377>/BCC ERROR /	
(1)	022130	052377	040522	051516	EM5:	.ASCIZ	<377>/TRANSMITTER DONE /	
(1)	022153	106	044501	042514	EM6:	.ASCIZ	/FAILED TO SET /	
(1)	022172	051377	041505	044505	EM7:	.ASCIZ	<377>/RECEIVER INTERRUPT IN HALF-DUPLEX /	
(1)	022236	043377	046101	042523	EM10:	.ASCIZ	<377>/FALSE INTERRUPT /	
(1)	022260	040777	047502	052122	EM11:	.ASCIZ	<377>/ABORT SEQUENCE ERROR /	
(1)	022307	377	052123	051101	EM12:	.ASCIZ	<377>/START OF MESSAGE /	
(1)	022332	052777	042516	050130	EM13:	.ASCIZ	<377>/UNEXPECTED RECEIVER INTERRUPT /	
(1)	022372	052777	042516	050130	EM14:	.ASCIZ	<377>/UNEXPECTED TRANSMITTER INTERRUPT /	
(1)	022435	377	051124	047101	EM20:	.ASCIZ	<377>/TRANSMITTER DONE /	
(1)	022460	051377	041505	044505	EM21:	.ASCIZ	<377>/RECEIVER DONE /	
(1)	022500	052377	040522	051516	EM22:	.ASCIZ	<377>/TRANSMITTER ACTIVE /	
(1)	022525	377	042522	042503	EM23:	.ASCIZ	<377>/RECEIVER ACTIVE /	
(1)	022547	106	044501	042514	EM24:	.ASCIZ	/FAILED TO SET. /	
(1)	022567	106	044501	042514	EM25:	.ASCIZ	/FAILED TO CLEAR. /	
(1)	022611	377	054523	041516	EM26:	.ASCIZ	<377>/SYNC ERROR /	
(1)	022626	042377	052101	020101	EM27:	.ASCIZ	<377>/DATA ERROR /	
(1)	022643	377	042504	044526	EM30:	.ASCIZ	<377>/DEVICE FAILED TO INTERRUPT IN TIME /	
(1)	022710	043377	046101	042523	EM31:	.ASCIZ	<377>/FALSE INTERRUPT /	
(1)	022732	052377	040522	051516	EM32:	.ASCIZ	<377>/TRANSMITTER BCC ERROR IN DEC MODE /	
(1)	022777	377	042522	042503	EM33:	.ASCIZ	<377>/RECEIVER BCC ERROR IN DEC MODE /	
(1)	023041	377	042522	042503	EM15:	.ASCIZ	<377>/RECEIVER ERROR /	
(1)	023062	041377	041503	042440	EM16:	.ASCIZ	<377>/BCC ERROR!! /	
(1)	023100	041777	041522	043440	EM17:	.ASCIZ	<377>/CRC GENERATOR STATUS /	
(1)	023127	377	040504	040524	MH1:	.ASCIZ	<377>/DATA CHAR DATA BIT IN CRC GEN. CRC FOR THIS BIT /	
(1)								
(1)								
(1)								

(1)	023222							
(1)	023222	000000				0		
(1)	023224	000000				0		
(1)	023226	000000				0		
(1)	023230	022004				EM1		
(1)	023232	000000				0		;HALT 1
(1)	023234	000000				0		
(1)								
(1)	023236	022046				EM2		
(1)	023240	000000				0		;HALT 2
(1)	023242	000000				0		
(1)								
(1)	023244	022073				EM3		
(1)	023246	022153				EM6		;HALT 3
(1)	023250	000000				0		
(1)								
(1)	023252	022114				EM4		
(1)	023254	000000				0		;HALT 4
(1)	023256	000000				0		
(1)								
(1)	023260	022130				EM5		
(1)	023262	022153				EM6		;HALT 5
(1)	023264	000000				0		
(1)								
(1)	023266	022172				EM7		

# K05

(1)	023270	000000	0		;HALT 6
(1)	023272	000000	0		
(1)					
(1)	023274	022236	EM10		
(1)	023276	000000	0		;HALT 7
(1)	023300	000000	0		
(1)					
(1)	023302	022260	EM11		
(1)	023304	000000	0		;HALT 10
(1)	023306	000000	0		
(1)					
(1)	023310	022307	EM12		
(1)	023312	022153	EM6		;HALT 11
(1)	023314	000000	0		
(1)					
(1)	023316	022332	EM13		
(1)	023320	000000	0		;HALT 12
(1)	023322	000000	0		
(1)					
(1)	023324	022372	EM14		
(1)	023326	000000	0		;HALT 13
(1)	023330	000000	0		
(1)					
(1)	023332	023062	EM16		
(1)	023334	000000	0	;HALT14	
(1)	023336	000000	0		
(1)					
(1)	023340	022460	EM21		
(1)	023342	022547	EM24	;HALT15	
(1)	023344	000000	0		
(1)					
(1)					
(1)	023346	022435	EM20		
(1)	023350	022547	EM24	;HALT16	
(1)	023352	000000	0		
(1)					
(1)	023354	022500	EM22		
(1)	023356	022547	EM24	;HALT17	
(1)	023360	000000	0		
(1)					
(1)	023362	022500	EM22		
(1)	023364	022567	EM25	;HALT20	
(1)	023366	000000	0		
(1)					
(1)	023370	022611	EM26		
(1)	023372	000000	0	;HALT21	
(1)	023374	000000	0		
(1)					
(1)	023376	022626	EM27		
(1)	023400	000000	0	;HALT22	
(1)	023402	000000	0		
(1)					
(1)	023404	022643	EM30		
(1)	023406	000000	0	;HALT23	
(1)	023410	000000	0		
(1)					



(1)	023412	022710		EM31	
(1)	023414	000000		0	;HALT24
(1)	023416	000000		0	
(1)					
(1)	023420	022525		EM23	
(1)	023422	022547		EM24	;HALT25
(1)	023424	000000		0	
(1)					
(1)	023426	023041		EM15	
(1)	023430	000000		0	;HALT26
(1)	023432	000000		0	
(1)					
(1)	023434	022732		EM32	
(1)	023436	000000		0	;HALT 27
(1)	023440	000000		0	
(1)					
(1)	023442	022777		EM33	
(1)	023444	000000		0	;HALT 30
(1)	023446	000000		0	
(1)	023450	000003		3	
(1)	023452	006	021	.BYTE	6,17.
(1)	023454	001252		SAVR1	
(1)	023456	006	017	.BYTE	6,15.
(1)	023460	001254		SAVR2	
(1)	023462	006	002	.BYTE	6,2
(1)	023464	001256		SAVR3	
(1)					
(1)	023466				
3091		000001			

DT1:

CORMAX:  
.END













RCRC7T= 100000	235#													
RCVEN = 000020	198#	1355	1429	1536	1662	1774	1846	1868	1902	1985	2146	2168	2216	
	2264	2326	2439	2561	2688	2737	2800	2882	3037					
REACT= 004000	191#	2369	2403	2472	2594	2757								
REOM = 001000	208#	1475												
RESREG 004614	956	959#												
RESTRY 003116	619	623	631#											
RESOS = 104407	291#	959												
RETURN 001214	140#	414*	596*	598	631*	668*	671	974*	976	1009	1125*	1135*	1136	
RING = 040000	188#													
RINTEN= 000100	196#	1356	1430	1537	1663	1774	1846	1985	2339	2450	2572	2810	2891	
RSCM = 000400	209#	1887	1915											
RTS = 000004	200#													
RUN 001314	180#	410*	1052	1055*	1056*	1063*	1064*							
RXCSR 001404	310#	635	1071*	1075	1355*	1356*	1429*							
	1663*	1733	1774*	1799	1810	1846*	1868*	1885	1902*	1905	1913	1985*	2021	
	2146*	2160	2168*	2186	2216*	2237	2264*	2292	2326*	2339*	2364	2381	2398	
	2403	2439*	2450*	2470	2497	2561*	2572*	2592	2620	2623	2644	2688*	2695	
	2704	2737*	2750	2800*	2810*	2882*	2891*	2940	2996	3037*				
RXDBUF 001406	311#	1077*	1078*	1079	1080	1081	1466	1475	1573	1606	1609	1699	1732	
	1735	1800	1812	1886	1898	1906	1914	2010	2020	2023	2189	2240	2294	
	2365	2382	2399	2471	2498	2593	2622	2625	2645	2698	2707	2756	2842	
	2941	2954	2966	2972	2979	2991	2998							
RXDERR= 100000	204#													
RXDONE= 000200	195#	1801	1890	2366	2383	2477	2599							
RD =%000000	40#	442*	443*	444	515*	520	522	524	526*	527*	528	530*	531*	
	532*	533*	535*	536	537	538	539	540	541	562*	570*	571*	573*	
	575*	577	578	649	655*	669*	802	807*	820	833*	837*	847	863*	
	965*	984*	988*	991*	1065*	1071	1072	1073	1074*	1094	1096	1098	1101*	
	1118*	1119	1121	1123	1125	1129	1130	1138	1156	1163*	1165*	1166	1169*	
	1170*	1171	1174*	1177*	1179*	1181	1187*	1770*	1783*	1784	1839*	1932*	1933	
	1936*	1942*	1943	1965*	1976*	1977	1979*	2014*	2015	2364*	2366	2369	2381*	
	2383	2398*	2428*	2511	2512*	2539*	2543	2548*	2550*	2658	2659*	2730*	2752*	
	2753	2768*	2831*	2832										
R1 =%000001	41#	516*	520*	521*	522*	523*	524*	574*	575	576*	577	622*	626	
	801	808*	821	825*	827	828	829	830	862*	1013*	1014	1016	1017	
	1102*	1157	1175*	1176	1177	1186*	1517*	1521	1526*	1643*	1647	1652*	1799*	
	1801	1967*	1971	2011	2365*	2373	2376	2382*	2386	2389	2391	2399*	2400	
	2429*	2487	2492*	2502	2505*	2551*	2609	2614*	2649	2652*	2731*	2769*	2770	
	2771	2794*	2843*	2844	3086*									
R2 =%000002	42#	800	809*	1012*	1014*	1015*	1103*	1158	1176*	1178*	1179	1185*	1800*	
	1804	1812*	1813	1966*	2033*	2034	2470*	2472	2477	2497*	2592*	2594	2599	
	2644*	2727*	2743*	2767*										
R3 =%000003	43#	704	711*	721*	724*	725	730*	799	810*	822	834*	835*	836*	
	837	846*	847*	852*	855*	861*	1138*	1771*	1807*	1808	2471*	2482	2487	
	2498*	2499	2502	2593*	2604	2609	2645*	2646	2649					
R4 =%000004	44#	705	710*	714*	715*	716	723*	726	729*	737	746*	747	749	
	751	753*	754	755	776*	777*	781*	798	811*	823	831*	834	839*	
	841*	843*	860*	922*	923*	924*	925*	926*	927*	928	929	930	1885*	
	1890	1905*	1913*	2073*	2074	2077	3071*	3074*						
R5 =%000005	45#	688	689*	693	698	700*	736	738*	739	740	741	742	743	
	744	745*	754*	757*	758*	759*	767	769	771	777	778*	782*	797	
	812*	824	832*	844*	859*	876*	880*	884*	920*	921*	922	924	1143	
	1144	1145	1146	1147*	1159	1160	1161	1188*	1213*	1247*	1281*	1315*	1350*	
	1424*	1522*	1532*	1580*	1648*	1658*	1706*	1766*	1840*	1864*	1886*	1887	1893	
	1898*	1899	1906*	1907	1914*	1915	1918	1972*	1980*	2317*	2430*	2544*	2552*	















DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 75  
DZDPDA.P11 CROSS REFERENCE TABLE -- USER SYMBOLS

.SCOPI	003312	280	678#	
.SETFL	004242	300	876#	888
.START	001562	117	402#	414
.TRPSR	004316	103	896#	
.TRPTA	001344	276#	901	
.TYPE	003336	282	688#	







ADC	1165															
ADCB	1056	1064														
ADD	521	523	549	690	707	778	826	836	885	901	924	927	985	1057	1066	
ASL	1094	1096	1098													
BCC	757	758	759	899	923	925										
BEQ	519	3076														
	427	430	488	498	501	507	546	552	560	589	623	651	660	679	681	
	717	748	756	851	909	916	932	938	947	952	956	961	973	989	1108	
	1167	1345	1362	1374	1391	1439	1454	1468	1546	1561	1575	1616	1619	1642	1672	
	1687	1701	1742	1745	1791	1805	1857	1876	1894	1900	1908	1919	1997	2012	2026	
	2069	2116	2124	2156	2182	2191	2222	2233	2242	2280	2296	2357	2377	2390	2392	
	2401	2445	2457	2488	2503	2567	2579	2610	2628	2650	2700	2709	2749	2764	2817	
	2847	2899	2949	2959	2984	3070										
BGT	752															
BHI	768															
BIC	571	835	877	900	926	966	990	1171	1177	1178	1180	1496	1601	1727	1902	
	1946	2041	2111	2352	2517	2664	2702	2836	2933							
BICB	715	753														
BIS	499	502	987	1179	1181	1211	1245	1279	1313	1347	1349	1355	1356	1357	1397	
	1418	1423	1428	1429	1430	1433	1447	1515	1531	1536	1537	1540	1554	1622	1639	
	1657	1662	1663	1666	1680	1748	1764	1773	1774	1775	1779	1786	1817	1837	1845	
	1846	1847	1851	1852	1868	1923	1962	1984	1985	1986	1990	2003	2059	2062	2063	
	2081	2099	2102	2103	2142	2145	2146	2147	2165	2168	2169	2170	2194	2212	2215	
	2216	2217	2260	2263	2264	2265	2322	2325	2326	2327	2339	2407	2435	2438	2439	
	2440	2450	2451	2463	2516	2557	2560	2561	2562	2572	2573	2585	2663	2685	2687	
	2690	2732	2733	2735	2737	2739	2796	2799	2800	2801	2810	2811	2823	2878	2881	
	2882	2883	2891	2892	2905	3025	3036	3037	3038	3065						
BISB	754															
BIT	437	551	559	588	650	657	678	691	908	913	970	972	1107	1166	1359	
	1361	1371	1373	1388	1390	1436	1438	1451	1453	1475	1543	1545	1558	1560	1618	
	1669	1671	1684	1686	1744	1788	1790	1801	1813	1854	1856	1873	1875	1887	1890	
	1915	1994	1996	2066	2068	2074	2077	2113	2115	2123	2153	2155	2179	2181	2219	
	2221	2230	2232	2277	2279	2354	2356	2366	2369	2383	2403	2442	2444	2454	2456	
	2472	2477	2564	2566	2576	2578	2594	2599	2635	2746	2748	2757	2814	2816	2846	
	2896	2898	2973	2999	3069											
BITB	771	1052														
BLO	770															
BLOS	565															
BLT	750															
BMI	1367	1379	1444	1551	1677	1929	1939	2121	2187	2238	2285	2696	2705	2751	2943	
BNE	438	445	529	554	558	579	587	619	658	663	692	699	722	772	780	
	845	849	854	858	879	883	914	934	971	992	1018	1048	1053	1059	1069	
	1106	1120	1122	1124	1131	1183	1243	1277	1311	1360	1365	1372	1377	1389	1394	
	1437	1442	1452	1457	1472	1476	1487	1527	1544	1549	1559	1564	1579	1592	1653	
	1670	1675	1685	1690	1705	1718	1785	1789	1794	1802	1809	1814	1855	1860	1874	
	1879	1888	1891	1916	1934	1944	1978	1995	2000	2016	2035	2067	2072	2075	2078	
	2114	2119	2154	2159	2180	2185	2220	2231	2236	2278	2283	2289	2355	2360	2367	
	2370	2384	2404	2443	2455	2460	2473	2478	2493	2506	2513	2549	2565	2577	2582	
	2595	2600	2615	2636	2653	2660	2744	2747	2754	2758	2772	2815	2820	2845	2897	
	2902	2920	2929	2963	2974	2988	3000	3056	3059	3064	3078					
BPL	434	654	694	697	713	719	911	963	1385	1432	1474	1539	1608	1665	1734	
	1778	1781	1811	1850	1870	1988	2022	2106	2109	2150	2161	2173	2176	2224	2227	
	2268	2271	2293	2330	2333	2336	2341	2344	2347	2350	2374	2387	2447	2483	2500	
	2515	2569	2605	2621	2624	2647	2662	2741	2761	2804	2807	2886	2889	2946	2956	
	2968	2981	2993	2997	3085											
BR	420	428	510	525	550	568	593	652	656	728	760	762	881	888	1051	



	1061	1128	1133	1489	1492	1594	1597	1720	1723	1881	1904	1910	1912	1937	2037
	2163	2298	2380	2394	2397	2476	2481	2486	2491	2508	2598	2603	2608	2613	2617
CLC	2643	2655	2834	2916	2922	2925	2931	2953	2965	2971	2978	2990			
CLR	513	517	838	840	842	1054	1062	1172	3052	3073					
	406	411	412	443	527	556	562	576	605	648	665	666	745	1015	1162
	1222	1256	1290	1325	1354	1383	1421	1446	1448	1517	1529	1553	1555	1643	1655
	1679	1681	1770	1771	1776	1839	1848	1936	1965	1966	1979	1991	2273	2274	2338
	2428	2429	2449	2539	2541	2550	2551	2571	2728	2729	2730	2731	2768	2794	2809
CLRB	2831	2872	2873	2874	2875	2876	2893	2977	3030	3032	3033	3034	3057		
CMP	407	408	606	664	855	880	918								
	423	429	444	528	577	578	662	767	769	915	960	1016	1017	1058	1067
	1119	1121	1123	1130	1615	1741	1784	1808	1933	1943	1977	2015	2025	2034	2627
CMPB	2753	2844	3055	3058	3061										
	564	716	747	749	751	755	878	882	1242	1276	1310	1467	1486	1574	1591
	1700	1717	1804	1893	1899	1907	1918	2011	2190	2241	2295	2376	2389	2391	2400
COM	2487	2502	2609	2649	2699	2708	2763	2919	2928	2948	2958	2962	2983	2987	
COMB	1168	1170	1614	1740	2024										
DEC	432	440	3081												
	506	721	844	857	988	991	1182	1363	1375	1392	1440	1455	1547	1562	1673
	1688	1792	1858	1877	1998	2070	2117	2157	2183	2234	2281	2288	2358	2458	2580
DECB	2743	2818	2900												
EMT	618	779	848	853											
HALT	64														
INC	95	563	567	572	967	1003	1050								
	607	661	969	1076	1078	1082	1085	1087	1089	1091	1783	1807	1932	1942	1976
INCB	2014	2033	2492	2614	2752	2843	3045	3046							
	1470	1485	1526	1577	1590	1652	1703	1716	2505	2512	2548	2652	2659	2769	2918
JMP	2927	2961	2986												
JSR	117	435	439	555	598	632	671	903	976	1009	1136	1624	1750		
	626	1104	1212	1213	1246	1247	1280	1281	1314	1315	1348	1350	1398	1419	1424
	1516	1522	1532	1580	1623	1640	1648	1658	1706	1749	1765	1766	1818	1838	1840
	1864	1924	1963	1972	1980	2004	2060	2082	2100	2143	2166	2195	2213	2261	2317
	2323	2408	2430	2436	2464	2544	2552	2558	2586	2686	2736	2790	2797	2824	2868
MOV	2879	2906	3026	3047	3051	3066									
	402	403	404	409	413	414	416	417	418	421	422	424	425	442	470
	515	516	520	522	524	526	530	531	532	533	535	544	545	573	574
	575	584	585	591	592	594	595	596	608	622	631	649	655	667	668
	669	682	688	689	700	704	705	706	710	711	720	723	724	725	726
	729	730	736	737	738	739	740	741	744	746	776	777	781	782	793
	797	798	799	800	801	802	807	808	809	810	811	812	820	821	822
	823	824	825	827	830	831	833	834	846	859	860	861	862	863	876
	896	898	902	917	920	922	928	929	930	965	974	975	984	1002	1005
	1006	1012	1013	1014	1060	1065	1070	1071	1072	1073	1074	1075	1077	1079	1080
	1081	1083	1084	1086	1088	1090	1093	1095	1097	1101	1102	1103	1118	1125	1135
	1138	1143	1144	1156	1157	1158	1159	1160	1161	1163	1169	1174	1175	1176	1184
	1185	1186	1187	1208	1209	1210	1217	1218	1224	1240	1241	1244	1251	1252	1258
	1274	1275	1278	1285	1286	1292	1308	1309	1312	1319	1320	1327	1342	1343	1346
	1358	1369	1370	1386	1387	1395	1396	1402	1416	1417	1420	1422	1434	1435	1450
	1465	1478	1488	1490	1491	1493	1497	1498	1513	1514	1518	1519	1520	1521	1528
	1530	1541	1542	1557	1573	1593	1595	1596	1598	1602	1603	1609	1611	1613	1637
	1638	1644	1645	1646	1647	1654	1656	1667	1668	1683	1699	1719	1721	1722	1724
	1728	1729	1735	1737	1739	1762	1763	1772	1782	1787	1796	1799	1800	1812	1816
	1835	1836	1844	1853	1862	1871	1872	1885	1886	1896	1898	1903	1905	1906	1913
	1914	1925	1931	1935	1941	1945	1947	1960	1961	1964	1967	1968	1969	1970	1971
	1989	1993	2010	2017	2028	2032	2036	2038	2042	2043	2057	2058	2061	2064	2065
	2073	2097	2098	2101	2104	2107	2110	2112	2140	2141	2144	2148	2151	2152	2167

	2171	2174	2177	2178	2189	2210	2211	2214	2218	2225	2228	2229	2258	2259	2262
	2266	2269	2272	2275	2276	2287	2294	2314	2315	2316	2324	2328	2331	2334	2337
	2342	2345	2348	2351	2353	2364	2365	2379	2381	2382	2396	2398	2399	2409	2425
	2426	2427	2437	2441	2448	2453	2465	2470	2471	2475	2480	2485	2490	2494	2495
	2497	2498	2507	2511	2518	2519	2536	2537	2538	2540	2542	2543	2559	2563	2570
	2575	2587	2592	2593	2597	2602	2607	2612	2616	2618	2622	2625	2642	2644	2645
	2654	2658	2665	2666	2683	2684	2688	2689	2691	2693	2698	2707	2724	2725	2726
	2727	2738	2742	2756	2767	2787	2788	2789	2798	2802	2805	2808	2813	2825	2832
	2833	2835	2837	2838	2842	2849	2865	2866	2867	2880	2884	2887	2890	2895	2907
	2913	2914	2915	2917	2921	2923	2924	2926	2930	2932	2934	2935	2940	2941	2951
	2952	2954	2964	2966	2970	2972	2976	2979	2989	2991	2998	3002	3023	3024	3027
	3028	3029	3031	3035	3039	3041	3044	3054	3060	3071	3072	3086			
MOV8	405	410	441	503	504	505	511	512	534	536	537	538	539	540	541
	569	570	620	621	698	714	742	743	828	829	832	837	847	852	884
	912	945	1145	1146	1484	1589	1606	1610	1715	1732	1736	2020	2023	2240	2626
	2770	2771													
NOP	591	592	627	628	629	630	997	1219	1220	1221	1253	1254	1255	1287	1288
	1289	1321	1322	1323	1494	1495	1599	1600	1725	1726	2039	2040			
RESET	624	1004													
ROL	3075														
ROLB	509	1055	1063												
ROR	839	841	843	1164	1173	3074									
RORB	514	518	3053												
RTI	683	701	731	783	803	813	864	886	977	993	1150	1154	1225	1259	1293
	1328	1403	1480	1499	1585	1604	1612	1711	1730	1738	1819	1897	1926	1948	2018
	2029	2044	2411	2496	2520	2619	2667	2839	2850	2936	3003				
RTS	998	1019	1139	1147	1188	3087									
SEC	508														
SUB	897	921													
TRAP	277	279	281	283	285	287	289	291	293	295	297	299			
TST	419	553	557	586	680	933	937	946	951	955	962	1105	1129	2373	2386
	2482	2499	2604	2646	2760	2945	2955	2967	2980	2992					
TSTB	426	433	487	497	500	653	659	693	696	712	718	850	910	931	1047
	1344	1366	1378	1384	1431	1443	1471	1473	1538	1550	1578	1607	1641	1664	1676
	1704	1733	1777	1780	1810	1849	1869	1928	1938	1987	2021	2105	2108	2120	2149
	2160	2172	2175	2186	2223	2226	2237	2267	2270	2284	2292	2329	2332	2335	2340
	2343	2346	2349	2446	2514	2568	2620	2623	2661	2695	2704	2740	2750	2803	2806
	2885	2888	2942	2996	3077	3084									
.ASCIZ	121	1023	3090												
.BLKB	176	177	178	179	180	247	248								
.BLKW	253	254	337	338	339	341	342	343	345	346	347	349	350	351	353
	354	355	357	358	359	361	362	363	365	366	367	1036	1524	1525	1650
.BYTE	1651	1974	1975												
	261	262	263	264	326	328	452	453	460	461	468	469	477	478	634
	637	640	643	979	982	1024	1026	1116	1117	1216	1250	1284	1318	1353	1427
	1535	1583	1661	1709	1769	1843	1867	1983	2320	2433	2555	2793	2871	3090	
.ENABL	1														
.END	3091														
.ENDC	1197	1200	1201	1210	1229	1232	1233	1242	1244	1263	1266	1267	1276	1278	1297
	1300	1301	1310	1312	1331	1334	1335	1344	1366	1378	1395	1406	1408	1409	1418
	1420	1443	1458	1473	1479	1501	1503	1505	1506	1515	1517	1528	1550	1565	1584
	1625	1627	1629	1630	1639	1643	1654	1676	1691	1710	1751	1753	1755	1756	1764
	1795	1823	1827	1828	1837	1861	1880	1951	1953	1954	1962	2001	2047	2049	2050
	2059	2073	2087	2089	2090	2099	2120	2130	2132	2133	2142	2160	2186	2200	2202
	2203	2212	2223	2237	2248	2250	2251	2260	2284	2302	2307	2308	2316	2361	2415
	2418	2419	2427	2428	2442	2446	2461	2497	2509	2525	2529	2530	2538	2550	2564



	2568	2583	2644	2656	2673	2675	2676	2685	2715	2717	2718	2726	2750	2778	2780
.EQUIV	2781	2789	2821	2854	2857	2858	2867	2903	3008	3016	3017	3025	3091		
.EVEN	64														
.IF	181	267	1023	1028	3090										
	1196	1197	1200	1209	1210	1228	1229	1232	1241	1242	1262	1263	1266	1275	1276
	1296	1297	1300	1309	1310	1330	1331	1334	1343	1358	1370	1397	1405	1406	1408
	1417	1420	1435	1450	1473	1501	1502	1503	1505	1514	1517	1542	1557	1580	1584
	1606	1626	1627	1629	1638	1641	1643	1668	1683	1706	1710	1732	1752	1753	1755
	1763	1787	1822	1823	1827	1836	1853	1872	1950	1951	1953	1961	1993	2046	2047
	2049	2058	2065	2086	2087	2089	2098	2112	2129	2130	2132	2141	2152	2178	2199
	2200	2202	2211	2219	2229	2247	2248	2250	2259	2276	2301	2302	2307	2315	2353
	2414	2415	2417	2418	2426	2427	2428	2434	2442	2453	2497	2507	2524	2525	2527
	2529	2537	2538	2539	2556	2564	2575	2620	2654	2672	2673	2675	2684	2714	2715
	2717	2725	2746	2777	2778	2780	2788	2813	2853	2854	2857	2866	2895	3007	3008
.IFF	3016	3024	3091												
	1209	1210	1241	1242	1275	1276	1309	1310	1343	1344	1358	1370	1387	1417	1418
	1435	1450	1514	1515	1542	1557	1638	1639	1668	1683	1763	1764	1787	1836	1837
	1853	1872	1961	1962	1993	2058	2059	2065	2098	2099	2112	2141	2142	2152	2178
	2211	2212	2223	2229	2259	2260	2276	2315	2316	2353	2418	2426	2427	2442	2446
	2453	2507	2527	2537	2538	2556	2568	2575	2656	2684	2685	2725	2726	2750	2788
.IFT	2789	2813	2866	2867	2895	3024									
	1358	1370	1387	1435	1450	1542	1557	1668	1683	1787	1853	1872	1993	2065	2112
	2152	2178	2219	2229	2276	2353	2417	2428	2434	2442	2453	2497	2507	2527	2539
.IIF	2556	2564	2575	2620	2654	2746	2813	2895							
	1209	1210	1222	1241	1242	1256	1275	1276	1290	1309	1310	1324	1343	1344	1417
	1418	1423	1424	1428	1514	1515	1531	1532	1536	1638	1639	1657	1662	1763	1764
	1836	1837	1961	1962	2051	2058	2059	2061	2062	2063	2091	2098	2099	2101	2102
	2103	2134	2141	2142	2144	2145	2146	2174	2204	2211	2212	2214	2215	2216	2225
	2228	2252	2259	2260	2262	2263	2264	2269	2272	2315	2316	2322	2324	2325	2326
	2331	2334	2337	2426	2427	2434	2435	2437	2438	2439	2442	2448	2494	2495	2537
	2538	2556	2557	2559	2560	2561	2564	2570	2616	2617	2684	2685	2725	2726	2788
	2789	2796	2798	2799	2800	2805	2808	2866	2867	2878	2880	2881	2882	2887	2890
.IRP	3024	3025													
	277	279	281	283	285	287	289	291	293	295	297	299	337	1196	1202
	1209	1228	1234	1241	1262	1268	1275	1296	1302	1309	1330	1336	1343	1405	1410
	1417	1502	1507	1514	1626	1631	1638	1752	1756	1763	1822	1829	1836	1950	1954
	1961	2046	2051	2058	2086	2091	2098	2129	2134	2141	2199	2204	2211	2247	2252
	2259	2301	2308	2315	2414	2419	2426	2524	2530	2537	2672	2677	2684	2714	2718
	2725	2777	2781	2788	2853	2859	2866	3007	3017	3024	3090				
.LIST	1	14	35	83	85	95	119	121	279	281	283	285	287	289	291
	293	295	297	299	301	375	394	543	599	1023	1140	1195	1196	1200	1210
	1227	1228	1232	1242	1261	1262	1266	1276	1295	1296	1300	1310	1330	1334	1344
	1404	1405	1408	1418	1501	1502	1505	1515	1625	1626	1629	1639	1751	1752	1755
	1764	1821	1822	1827	1837	1949	1950	1953	1962	2046	2049	2059	2086	2089	2099
	2129	2132	2142	2199	2202	2212	2247	2250	2260	2301	2307	2316	2414	2418	2427
	2524	2529	2538	2671	2672	2675	2685	2714	2717	2726	2776	2777	2780	2789	2852
	2853	2857	2867	3006	3007	3016	3025	3090							
.MACRO	1	3090													
.NLIST	1	14	35	83	85	95	119	121	279	281	283	285	287	289	291
	293	295	297	299	301	375	394	543	599	1023	1140	1195	1196	1200	1210
	1227	1228	1232	1242	1261	1262	1266	1276	1295	1296	1300	1310	1330	1334	1344
	1404	1405	1408	1418	1501	1502	1505	1515	1625	1626	1629	1639	1751	1752	1755
	1764	1821	1822	1827	1837	1949	1950	1953	1962	2046	2049	2059	2086	2089	2099
	2129	2132	2142	2199	2202	2212	2247	2250	2260	2301	2307	2316	2414	2418	2427
	2524	2529	2538	2671	2672	2675	2685	2714	2717	2726	2776	2777	2780	2789	2852
	2853	2857	2867	3006	3007	3016	3025	3090							

C07

DZDPD-A MACY11 27(732) 21-OCT-76 15:54 PAGE 84  
DZDPDA.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

.PAGE	1	35	85	183	257	304	332	375	394	599	1038	1140	3090		
.REM	375														
.REPT	95														
.SBTTL	1	35	85	119	394	599	1195	1227	1261	1295	1330	1404	1501	1625	1751
.TITLE	1821	1949	2046	2086	2129	2199	2247	2301	2414	2524	2671	2714	2776	2852	3006
.WORD	14	249	250	251	252	2546	2547	3049	3050						

ERRORS DETECTED: 0  
DEFAULT GLOBALS GENERATED: 0

\*,DZDPDA.SEQ/SOL/CRF/PAGNUM/NL:TOC=DZDPXX.MAC,DZDPDA.P11  
RUN-TIME: 28 41 6 SECONDS  
RUN-TIME RATIO: 193/77=2.5  
CORE USED: 26K (51 PAGES)



