

DQ11

INTERRUPT TESTS
MD-11-DZDQH-B

EP-DZDQH-B-DL-A
COPYRIGHT © 1976
FICHE 1 OF 1

NOV 1976
digital
MADE IN U.S.A.

IDENTIFICATION

PRODUCT CODE: MAINDOC-11-DZDOM-B-D
PRODUCT NAME: CHARACTER LENGTH AND INTERRUPT TESTS
DATE: 21 JUNE 1976
MAINTAINER: DIAGNOSTIC GROUP

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSIDERED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS DOCUMENT.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED UNDER A LICENSE AND MAY ONLY BE USED OR COPIED IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1975, 1976 BY DIGITAL EQUIPMENT CORPORATION

1. ABSTRACT

THE FUNCTION OF THE D311 DIAGNOSTICS ARE TO VERIFY THAT THE OPTION OPERATES ACCORDING TO SPECIFICATIONS.

THIS TEST WILL CHECK TRANSMITTER AND RECEIVER INTERRUPTS. ENTER T AND R TO CHECK THE AB OPTION IS INSTALLED. ALSO CHECKS VRC AND DATA TRANSFERS. THE OPTION PART CHECKS TRANSMITTER AND RECEIVER CHARACTER LENGTHS.

PRESENTLY THERE ARE SEVEN OFF LINE DIAGNOSTICS THAT ARE AVAILABLE IN SOME TO INSURE THAT IF AN ERROR SHOULD OCCUR IT WILL BE DETECTED IMMEDIATELY STRONG INSURING THAT THE COST OF ERROR WILL BE IMMEDIATE TO PROBLEM SOLVING. ADDITIONAL DIAGNOSTICS MAY BE ADDED IN THE FUTURE.

- 1. BASIC R/W TEST #1
- 2. BASIC R/W TEST #2
- 3. BASIC NR AND INTERRUPT TEST
- 4. RECEIVER TRANSMITTER EXERCISER TEST
- 5. MISC. RX AND TX TESTS. PLUS BCC TESTS.
- 6. CHARACTER DETECT TESTS.
- 7. CHARACTER LENGTH AND INTERRUPT TESTS.

THERE IS ALSO AN ONLINE TEST TO BE DISCUSSED LATER.
1. D2000 (REV) ONLINE TEST. (ITEP OVERLAY)

AND A PARAMETER INPUT PROGRAM IS AVAILABLE
1. D2003 (REV) D311 TRIAL PROGRAM (PARAMETER INPUT)

2. REQUIREMENTS

2.1 EQUIPMENT

ANY POP11 FAMILY CPU (WITH MINIMUM 4K MEMORY)-WITH OR WITHOUT A HARDWARE SWITCH REGISTER (LOC. 177570, RSR 33 (OR EQUIVALENT)
D311
SYNC MODEM (ONLY REQUIRED FOR ONLINE TEST)

2.2 STORAGE

PROGRAM WILL LOAD AND RUN IN 4K OF MEMORY. LOCATION 1400 THRU 1600 ARE ESPECIALLY TO BE NOTED AND TO BE UNTOUCHED BY OPERATOR AFTER D311 TRIAL PROGRAM HAS BEEN EXECUTED. OR AFTER THE "AUTO SIZING" HAS BEEN DONE.

3. LOADING PROCEEDURE

3.1 METHOD

A. PROGRAMS ARE IN ABSOLUTE FORMAT AND ARE LOADED USING THE ABSOLUTE LOADER.

ABSOLUTE LOADER STARTING ADDRESS +500

MEMORY
 SIZE

*	1000
*	1001
*	1002
*	1003
*	1004
*	1005
*	1006

3.1.1 LOAD THE ADDRESS OF ABS. LOADER (LOC.XXX500)

3.1.2 THEN START

4. STARTING PROCEEDURE

B. LOAD LOC. 200

B. SET SWR TO ZERO FOR "AUTO SIZING" OR LEAVE
 LEAVE SWR BIT 7=1 TO USE EXISTING PARAMETERS SET UP
 BY D011 TRAIL PROGRAM OR A PREVIOUSLY RUN D011 DIAGNOSTIC
 THAT USED THE "AUTO SIZING".
 ***REFER TO SECTION 4.1 FOR SOFTWARE SWITCH REGISTER OPERATION
 AND OPTIONS.***

NOTE: THE SOFTWARE SWITCH REGISTER IS LOCATED AT LOC.176
 SOFTWARE DISPLAY REGISTER IS LOCATED AT LOC.174

C. THEN START

THE PROGRAM WILL TYPE MAINDEC NAME AND PROGRAM NAME
 IF THIS WAS THE FIRST START UP OF THE PROGRAM) AND ALSO
 THE FOLLOWING:

"MAP OF D011 STATUS"	
1400	160010
1402	162300
1404	160020
1406	150310

THE ABOVE IS ONLY AN EXAMPLE!
 THIS WOULD INDICATE THE STATUS TABLE STARTING AT ADD.
 1400 IN THE PROGRAM. THE STATUS TABLE MUST BE VERIFIED BY THE
 USER IF AUTO SIZING IS DONE. FOR INFORMATION OF STATUS
 TABLE SEE SECTION B.4 FOR HELP.

****IF THE SOFTWARE SWITCH REGISTER IS SELECTED THEN THE FOLLOWING
 WILL BE TYPED AFTER THE PROGRAM IDENTIFIES ITSELF:
 SWR=XXXXX NEW= (REFER TO SECTION 4.1 FOR OPERATOR'S OPTION)****
 NOTE: IF USING THE SOFTWARE SWITCH REGISTER WHEN A HARDWARE
 SWITCH REGISTER IS AVAILABLE THE PROGRAM WILL NOT
 TYPE OUT THE TITLE.

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

4.1 CONTROL SWITCH SETTINGS

IF THE DIAGNOSTIC IS RUN ON A CPU WITHOUT A SWITCH
 REGISTER THEN A SOFTWARE SWITCH REGISTER IS USED WHICH ALLOWS
 THE USER THE SAME SWITCH OPTIONS AS THE HARDWARE SWITCH REGISTER.
 IF THE HARDWARE SWITCH REGISTER DOES NOT EXIST OR IF ONE DOES
 AND IT CONTAINS ALL ONES (177777) THEN THE SOFTWARE SWITCH
 REGISTER (LOC. 176) IS USED.

CONTROL:

THIS PROGRAM ALSO SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH
 REGISTER (LOC. 176) FROM THE TTY. THIS CAN BE ACCOMPLISHED BY
 DOING THE FOLLOWING:

1) TYPE CONTROL G (<G>): THIS WILL ALLOW THE TTY TO ENTER DATA INTO
 LOC. 176 AT SELECTED POINTS WITHIN THE PROGRAM.

NOTE: WHEN RUNNING THIS PROGRAM WITH A
 SWITCH REGISTERLESS PROCESSOR IT MAY BE NECESSARY
 TO DEPRESS (<G>) MORE THAN ONCE BEFORE IT IS ACCEPTED
 DUE TO THE NUMBER OF RESET INSTRUCTIONS THAT ARE EXECUTED.

2) THE MACHINE WILL THEN TYPE: SWR=XXXX.XNEW= (XXXXXX IS THE OCTAL CONTENTS
 OF THE SOFTWARE SWITCH REGISTER.)

3) AFTER THE "NEW=" HAS BEEN TYPED THEN THE OPERATOR CAN DO ONE
 OF THE FOLLOWING AT THE TTY:

A) TYPE A NUMBER TO BE LOADED INTO LOC. 176 FOLLOWED BY A (<CR>).
 (ONLY NUMBERS BETWEEN 0-7 WILL BE ACCEPTED AND ONLY 6 NUMBERS
 WILL BE ALLOWED)
 IF A (<CR>) IS THE FIRST KEY DEPRESSED THE SOFTWARE SWITCH
 REGISTER CONTENTS WILL NOT BE CHANGED.

B. IF A CONTROL U (<U>) IS DEPRESSED THEN THE PROGRAM WILL SEND YOU
 BACK TO STEP 2.

SW 15 SET: HALT ON ERROR
 SW 14 SET: LOOP ON CURRENT TEST
 SW 13 SET: INHIBIT ERROR PRINT OUT

```

SW 00: INHIBIT TYPE OUT BELL ON ERROR.
SW 01: INHIBIT ITERATIONS
SW 02: ESCAPE TO NEXT TEST
SW 03: LOOP WITH CURRENT DATA
SW 04: CATCH ERROR AND LOOP ON IT
SW 05: USE PREVIOUS STATUS TABLE. CLR-DO AUTO SIZE.
SW 06:
SW 07:
SW 08: LOCK ON SELECTED TEST
SW 09: RESTART PROGRAM AT SELECTED TEST
SW 10: RESELECT DQ11'S DESIRED ACTIVE.
    
```

4.1.2 SWITCH REGISTER RESTRICTIONS

SW 00 RESELECT DQ11'S DESIRED ACTIVE.
 PLEASE NOTE THAT A MESSAGE IS TYPED
 OUT FOR SWITCH REGISTER BEING EQUAL TO DQ11'S
 ACTIVE. THIS MEANS IF THE SYSTEM HAS
 FOUR DQ11S; BITS 00, 01, 02, 03 WILL
 BE SET IN LOC "DQACTV". USING THIS
 SWITCH ALTERS THAT LOCATION; THEREFORE
 IF FOUR DQ11S ARE IN THE SYSTEM
 DO NOT SET SWITCHS GREATER THAN
 SW 03 IN THE UP POSITION. THIS WOULD BE
 A FATAL ERROR. DO NOT SELECT MORE ACTIVE
 DQ11S THAN HAS BEEN GIVEN INFORMATION
 ABOUT IN TRIAL PROGRAM.

METHOD:
 I: LOAD ADDRESS 200
 O: START WITH SW 00=1
 O: PROGRAM WILL TYPE MESSAGE
 O: CONTINUE THE BINARY NUMBER OF DQ11S DESIRED ACTIVE
 F: EXAMPLE: 1=1 DQ11; 3=2 DQ11; 7=3 DQ11; 17=4 DQ11; 37=5 DQ11 ETC.
 F: NUMBER (IF VALID) WILL BE IN DATA LIGHTS (EXCLUDING 1: 05, 11/04, 11/34)
 F: CONTINUE WITH ANY OTHER SWITCH SETTINGS DESIRED.

SW 01 IT IS STRONGLY SUGGESTED THAT
 AT LEAST ONE PASS HAS BEEN MADE
 BEFORE TRYING TO SELECT A TEST
 THAT IS NOT IN THE ORDER OF SEQUENCE
 THE REASON BEING IS THAT THE
 PROGRAM HAS TO CLEAR AREAS AND SET
 UP PARAMETERS. ALSO WHEN A TEST IS
 SELECTED ALWAYS START AT THE VERY
 BEGINNING OF THAT TEST.

SW 09 LOOP ON CURRENT DATA:
 THIS SWITCH WILL ONLY WORK IF
 CALL "SCOPI" IS IN THAT TEST.
 THE REASON BEING THAT MOST TESTS
 DEAL WITH BLOCKS OF DIFFERENT DATA
 TO BE SENT OR RECEIVED ALL AT ONCE
 THIS IN BLOCK DATA; ONE PATTERN CAN'T BE SINGLED OUT.

4.1.3 SWITCH REGISTER PRIORITYS

ERROR SWITCHES

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

HLT (ERROR) ROUTINE SUPPORTS <IG> OPERATION

SCOPE SWITCHES

1. SW 09 (IF ENABLED BY "SCOPI")
2. SW 14
3. SW 11

SCOPE ROUTINE WILL SUPPORT <IG> OPERATION

4.2 STARTING ADDRESS

STARTING ADDRESS IS AT 000200
THERE ARE NO OTHER STARTING ADDRESSES
FOR THE DQ11 DIAGNOSTICS PREVIOUSLY MENTIONED

NOTE: IF ADDRESS 000042 IS NON-ZERO
THE PROGRAM ASSUMES IT IS UNDER
ACT11 OR DDP CONTROL AND WILL ACT ACCORDINGLY
AFTER *ALL* AVAILABLE DQ11'S ARE TESTED
THE PROGRAM WILL RETURN TO "DDP2" OR "ACT-11".

5. OPERATING PROCEDURE

WHEN PROGRAM IS INITIALLY STARTED MESSAGES AS DESCRIBED IN SECTION
FOUR WILL BE PRINTED.

AND PROGRAM WILL BEGIN RUNNING THE
DIAGNOSTIC

5.2 PROGRAM AND/OR OPERATOR ACTION

THE TYPICAL APPROACH SHOULD BE

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

THE TEST NUMBER AND PC WILL BE TYPED OUT AND
POSSIBLY AN ERROR MESSAGE (THIS DEPENDS ON THE TEST)
TO GIVE THE OPERATOR AN IDEA AS TO THE SOURCE OF THE
PROBLEM. IF IT IS NECESSARY TO KNOW MORE INFORMATION
CONCERNING THE ERROR REPORT; LOOK IN THE LISTING
FOR THAT TEST NUMBER WHICH WAS TYPED OUT

AND THEN NOTE THE PC OF THE ERROR REPORT
THIS WAY THE EXACT FUNCTIONING OF THE TEST
CAN BE INTERPEDITED

5. ERRORS

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

6.2 ERROR RECOVERY

IF FOR SOME REASON THE DQ11 SHOULD
"HANG THE BUS" (GAIN CONTROL OF BUS SO THAT
CONSOLE MANUAL FUNCTIONS ARE INHIBITED) AN INIT
OR POWER DOWN/UP IS NECESSARY FOR OPERATOR
TO REGAIN CONTROL OF CPU.
IF THIS SHOULD HAPPEN; LOOK IN LOCATION
"TSTNO" (ADDRESS 1222) FOR THE NUMBER OF THE TEST THAT
WAS RUNNING AT THE TIME OF THE CATASTROPHIC
ERROR.
IN THIS WAY THE OPERATOR WILL HAVE AN IDEA AS TO
WHAT THE DQ11 WAS DOING AT THE TIME OF THE ERROR.

6.3 ****HALT RECOVERY WHEN USING SOFTWARE SWITCH REGISTER****

IF THE SOFTWARE SWITCH REGISTER IS TO BE CHANGED AFTER A HALT
THE THE OPERATOR IS REQUIRED TO TYPE A <IG> BEFORE DEPRESSING CONTINUE.
THE FOLLOWING WILL BE TYPED:
SWR=XXXXXX NEW= (REFER TO SECTION 4.1 FOR OPERATOR OPTION)

7. RESTRICTIONS

7.1 STARTING RESTRICTIONS

SEE SECTION 4. (PLEASE)

7.2 OPERATING RESTRICTIONS

DQ11 TRIAL PROGRAM MUST BE RUN PRIOR TO THE
FIRST AND ONLY THE FIRST RUNNING OF ANY DQ11 DIAGNOSTIC
NOTE: IF NO PROGRAM OTHER THAN A
DQ11 DIAGNOSTIC WAS LOADED AFTER DQ11 TRIAL OR
IF CORE MEMORY HAS NOT BEEN CHANGED; OR IF THERE
IS NO DQ11 CONFIGURATION CHANGES; THE
DQ11 TRIAL PROGRAM NEED NEVER BE RUN AGAIN.
HOWEVER IF ANY OF THE ABOVE HAVE BEEN VIOLATED
THE DQ11 TRIAL PROGRAM MUST BE RUN AGAIN
BEFORE RUNNING THE DIAGNOSTICS
NOTE: AN ALTERNATIVE TO THE ABOVE IS ATTEMPTING
THE "AUTC SIZING" WHEN PROGRAM IS INITIALLY STARTED
WITH SW07=0.

9. MISCELLANEOUS

9.1 EXECUTION TIME

9.2 PASS COMPLETE

WHEN THE DIAGNOSTIC HAS COMPLETED
A PASS THE FOLLOWING IS AN EXAMPLE
OF THE PRINT OUT TO BE EXPECTED.

END PASS DZDQH-B CSR: 160000 VEC: 300 PASSES: 00000! ERRORS: 000000

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

9.3 TST1 (MINI MONITOR)

THE VERY FIRST "TEST" (TST1)
IS *NOT* A TEST OF THE DQ11 HARDWARE
IT IS A MINI-MONITOR USED TO CYCLE DQ11 IN THE
SYSTEM THROUGH THE DIAGNOSTIC.

REMEMBER: TST1 IS NOT A TEST OF DQ11 HARDWARE!!!!!!!

9.4 KEY LOCATIONS

RETURN (1210) CONTAINS THE ADDRESS WHERE PROGRAM WILL
RETURN WHEN ITERATION COUNT IS REACHED
OR IF LOOP ON TEST IS ASSERTED.
NEXT (1212) CONTAINS THE ADDRESS OF THE NEXT TEST
TO BE PERFORMED.
TSTNO (1222) CONTAINS THE NUMBER OF THE TEST NOW
BEING PERFORMED.
RUN (1272) THE BIT IN "RUN" ALWAYS POINTS ONE
PAST THE DQ11 CURRENTLY BEING TESTED.
EXAMPLE:
(RUN) 1272/0000000001000000
MEANS THAT DQ11 NO.05 IS THE DQ11 NOW
RUNNING.

DQCR00-DQCR17
DQST00-DQST17
(1400)-(1476)

THESE LOCATIONS CONTAIN THE INFORMATION
NEEDED TO TEST UP TO 16 (DECIMAL) DQ11S
SEQUENTIALLY. THEY CONTAIN THE CSR, VECTOR
AND STATUS CONCERNING THE CONFIGURATION
OF EACH DQ11.

DQACTV (1500) EACH BIT SET IN THIS LOCATION INDICATES
THAT THE ASSOCIATED DQ11 WILL BE TESTED
IN TURN.
EXAMPLE:
(DQACTV) 1500/0000000000011111
MEANS THAT DQ11 NO. 00,01,02,03,04
WILL BE TESTED.
EXAMPLE:

(DQACTV) 1500/000000000010001
 MEANS THAT DQ11 NO. 00,04
 WILL BE TESTED.
 DQCSR (1536) CONTAINS THE RECEIVER CSR OF THE
 CURRENT DQ11 UNDER TEST.
 DQSTAT (1510) CONTAINS THE STATUS OF THE CURRENT
 DQ11 UNDER TEST.
 BIT 15 SET: TWO SYNC CHARS/ONE SYNC CHAR
 BIT 14 SET: TEST JUMPER INSTALLED/NOT INSTALLED
 BIT 13 SET: BB OPTION INSTALLED/NOT INSTALLED
 BIT 12 SET: BA OPTION INSTALLED/NOT INSTALLED
 BIT 11 SET: ACTIVE ON FIRST NON-SYNC/ACTIVE AFTER NO. OF SYNC
 BIT 10 SET: AB OPTION INSTALLED/NOT INSTALLED
 BIT 09 SET: ODD VRC/EVEN VRC
 BIT 00-08 VECTOR "A" OF DEVICE

8.5 *** METHOD OF AUTO SIZING ***

8.5.1 FINDING THE CONTROL STATUS REGISTER.

WHEN LOOKING FOR THE CSR IT IS NECESSARY TO TAKE CARE
 THAT WHEN A CSR IS FOUND THAT IT IS INDEED A DQ11. THAT
 IS THE METHOD OF MY MADNESS FOR THIS ROUTINE.
 AN ATTEMPT TO CLEAR THE MISC. REGISTER IS TRIED
 IF A TIME-OUT TRAP OCCURES POINTERS ARE UPDATED
 AND ATTEMPTED AGAIN. IF NO TIME-OUT; THE RECEIVER "ACTIVE BIT" (BIT 12)
 IS SET AND A *COMPARE* FOR BOTH SYNC1 AND SYNC 2 IS DONE
 AT THE MISC. REGISTER. IF THEY ARE THERE THIS IS
 A DQ11. THE INFORMATION IS STORED AWAY.

8.5.2 ONE SYNC BIT OR TWO?

SINCE TOO MUCH HARDWARE MUST BE TURNED ON TO SENSE THE
 PRESENTS OF ONE SYNC OR TWO. THE PROGRAM ASSUMES TWO SYNC
 CHARS. NOTE: THIS ASSUMPTION MAY BE ALTERED AFTER AUTO SIZING
 BY ALTERING BIT 15 IN APPRIOATE DQSTXX: LOCATION.

8.5.3 "BB" OPTION INSTALLED?

TO SENSE FOR THE "BB" OPTION THE PROGRAM SELECTS THE
 CHARACTER DET. REGISTER AND THE LOADS IN ALL 1'S. IF
 ANY ONE OR COMBINATION OF BITS ARE SET THE BB OPTION
 IS ASSUMED TO EXIST.

8.5.4 "AB" OPTION INSTALLED?

TO SENSE FOR THE "AB" OPTION THE PROGRAM SELECTS THE
 POLYNOMIAL REGISTER AND WRITES ALL 1'S INTO IT. IF ANY
 ONE OR COMBINATION OF BITS ARE SET THE AB OPTION IS ASSUMED
 TO EXIST.

8.5.5 "BA" OPTION INSTALLED?

TO SENSE FOR "BA" OPTION REQUEST TO SEND AND DATA TERMINAL
 READY ARE SET; IF EITHER ONE OR BOTH ARE SET THE PROGRAM
 ASSUMES THE BA OPTION EXISTES

8.5.6 JUMPER ON END OF CABLE?

THE PROGRAM CHECKS TO SEE IF EITHER OR BOTH CLEAR TO SEND AND CARRIER ARE SET; IF SO THE PROGRAM ASSUMES THE TEST JUMPER IS ON THE END OF THE CABLE.

8.5.7 ACTIVE ON FIRST NON-SYNC?

SINCE TOO MUCH HARDWARE MUST BE TURNED ON TO SENSE FOR WHEN THE DQ11 GOES ACTIVE THE PROGRAM ASSUMES "ACTIVE ON FIRST NON-SYNC". NOTE: THIS CAN BE CHANGED BY ALTERING BIT 11 IN THE APPRIORATE DQSTXX: AFTER AUTO SIZING

8.5.8 SET FOR ODD OR EVEN PARITY?

AS ABOVE TOO MUCH HARDWARE IS NEED TO SENSE WHICH PARITY WAS SELECTED. SO THE PROGRAM ASSEMES ODD PARITY. NOTE: THIS CAN BE CHANGED BY ALTERING BIT 9 IN APPRIORATE DQSTXX: LOCATION. AFTER AUTO SIZING

8.5.9 FINDING THE VECTOR.

THE PROGRAM SETS "PRIMARY DONE" "SECONDAY DONE" AND "INTERUPT ENABLE" AND LOOKS FOR AN INTERUPT. IF IT INTERUPTS IT IS PICKED UP AND STORED AWAY. IF NO INTERUPT OCCURES THE PROGRAM ASSUMES VECTOR =300. THIS PROBLEM WILL BE FIXED IN ONE OF THE DIAGNOSTICS AND *AUTO SIZING* SHOULD BE REDONE TO GET THE CORRECT VECTOR.

9. PROGRAM DESCRIPTION

CONTAINED WITHIN LISTING

10. LISTING

FOLLOWING

L01

DZDQH MACY11 27(732) 24-SEP-76 10:11 PAGE 12
 DZDQH.B11 INTRODUCTION TO DQ11 DIAGNOSTIC

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
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
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500

```

;MAINDEC-11-DZDQH-B/⟨377⟩/CHARACTER LENGTH AND INTERRUPT TESTS
;COPYRIGHT 1975, DIGITAL EQUIPMENT CORP., MAYNARD, MASS. 01754

```

```

;REVISED 21-JUNE-76 BY S. CARPENTER

```

```

: A)SUPPORTS SOFTWARE SWITCH REGISTER
: B)SUPPORTS THE DYNAMIC LOADING OF THE SOFTWARE SWITCH REGISTER
:   BY ⟨↑G⟩.

```

```

;STARTING PROCEDURE
;LOAD PROGRAM
;LOAD ADDRESS 000200
;PRESS START
;PROGRAM WILL TYPE "MAINDEC-11-DZDQH-B/⟨377⟩/CHARACTER LENGTH AND INTERRUPT TEST
;PROGRAM WILL TYPE "R" TO INDICATE THAT TESTING HAS STARTED
;AT THE END OF A PASS, PROGRAM WILL TYPE PASS COMPLETE MESSAGE
;AND THEN RESUME TESTING

```

```

;SWITCH REGISTER OPTIONS

```

```

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

```

```

SW15=100000           :=1,HALT ON ERROR
SW14=40000            :=1,LOOP ON CURRENT TEST
SW13=20000            :=1,INHIBIT ERROR TYPEOUT
SW12=10000            :=1,DELETE TYPEOUT/BELL ON ERROR.
SW11=4000             :=1,INHIBIT ITERATIONS
SW10=2000             :=1,ESCAPE TO NEXT TEST ON ERROR
SW09=1000             :=1,LOOP WITH CURRENT DATA
SW08=400              :=1,LOOP ON ERROR
SW06=100
SW05=40
SW04=20
SW03=10
SW02=4
SW01=2
SW00=1

;LOCK ON TEST SELECT
;RESTART PROGRAM AT SELECTED TEST
;RESELECT DQ11 DESIRED ACTIVE
;NOTE: THIS MUST NOT EXCEED ORIGINAL COUNT

```


;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

177570	DSWR= 177570	: HARDWARE SWITCH REGISTER LOC.
177570	DLIGHTS=177570	: HARDWARE DISPLAY REGISTER LOC.
177776	PS=177776	: PROCESSOR STATUS WORD
001200	STACK=1200	: 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	POPPO=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

100000	SIT15=100000
040000	SIT14=40000
020000	BIT13=20000
010000	BIT12=10000
004000	BIT11=4000
002000	BIT10=2000
001000	BIT9=1000
000400	BIT8=400
000200	BIT7=200
000100	BIT6=100
000040	BIT5=40
000020	BIT4=20
000010	BIT3=10
000004	BIT2=4
000002	BIT1=2
000001	BIT0=1

;DQ11 OPTIONAL DEFINITIONS

002000	ABBIT=2000
004000	ACTBIT=4000
010000	BABIT=10000
020000	BBBIT=20000
040000	JMBIT=40000

626 J01000 ODDBIT=1000
 627 100000 SYNBIT=100000

:DQ11 SECONDARY REGISTER DEFINATIONS

630	000000	RXBA.P=0	;RECEIVER BUS ADDRESS PRIMARY.
631	000001	RXWC.P=1	;RECEIVER WORD COUNT PRIMARY.
632	000002	TXBA.P=2	;TRANSMITTER BUS ADDRESS PRIMARY.
633	000003	TXWC.P=3	;TRANSMITTER BUS ADDRESS PRIMARY.
634	000004	RXBA.S=4	;RECEIVER BUS ADDRESS SECONDARY.
635	000005	RXWC.S=5	;RECEIVER WORD COUNT SECONDARY.
636	000006	TXBA.S=6	;TRANSMITTER BUS ADDRESS SECONDARY.
637	000007	TXWC.S=7	;TRANSMITTER WORD COUNT SECONDARY.
638			
639	000010	CHARDT=10	; CHARACTER DETECT REGISTER.
640	000011	SYNC.=11	; SYNC REGISTER.
641	000012	MISC.=12	; MISCELLANEOUS REGISTER.
642	000013	TX.MUX=13	; TRANSMITTER MUX REGISTER.
643	000014	SEQ.=14	; SEQUENCE REGISTER.
644	000015	RX.BCC=15	; RECEIVER BCC REGISTER.
645	000016	TX.BCC=16	; TRANSMITTER BCC REGISTER.
646	000017	POLY.=17	; POLYNOMIAL REGISTER.
647			
648			
649			
650			

000000	000674	000676	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000676	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000700	000702	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000702	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000704	000706	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000706	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000710	000712	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000712	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000714	000716	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000716	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000720	000722	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000722	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000724	000726	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000726	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000730	000732	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000732	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000734	000736	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000736	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000740	000742	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000742	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000744	000746	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000746	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000750	000752	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000752	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000754	000756	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000756	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000760	000762	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000762	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000764	000766	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000766	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000770	000772	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000772	000000	HALT	: EXAMINE STACK TO FIND CAUSE
000000	000774	000776	.+2	: UNEXPECTED TRAP TO THIS LOCATION
000000	000776	000000	HALT	: EXAMINE STACK TO FIND CAUSE

000000
000001
000002
000003
000004
000005
000006
000007
000008
000009
000010
000011
000012
000013
000014
000015
000016
000017
000018
000019
000020
000021
000022
000023
000024
000025
000026
000027
000028
000029
000030
000031
000032
000033
000034
000035
000036
000037
000038
000039
000040
000041
000042
000043
000044
000045
000046
000047
000048
000049
000050
000051
000052
000053
000054
000055
000056
000057
000058
000059
000060
000061
000062
000063
000064
000065
000066
000067
000068
000069
000070
000071
000072
000073
000074
000075
000076
000077
000078
000079
000080
000081
000082
000083
000084
000085
000086
000087
000088
000089
000090
000091
000092
000093
000094
000095
000096
000097
000098
000099
000100
000101
000102
000103
000104
000105
000106
000107
000108
000109
000110
000111
000112
000113
000114
000115
000116
000117
000118
000119
000120
000121
000122
000123
000124
000125
000126
000127
000128
000129
000130
000131
000132
000133
000134
000135
000136
000137
000138
000139
000140
000141
000142
000143
000144
000145
000146
000147
000148
000149
000150
000151
000152
000153
000154
000155
000156
000157
000158
000159
000160
000161
000162
000163
000164
000165
000166
000167
000168
000169
000170
000171
000172
000173
000174
000175
000176
000177
000178
000179
000180
000181
000182
000183
000184
000185
000186
000187
000188
000189
000190
000191
000192
000193
000194
000195
000196
000197
000198
000199
000200

000024
015756
000340
015436
000340
015434
000340
000046
014164
000052
000000
000056
010120
012721
022021
020127
101771
012737
013737
006037
103023
005037
005722
012772
105200
001376
112712
005722
000761
051612
042712
022626
012716
000002
000207
000174
000000
000000
000200
000137

: STANDARD INTERRUPT VECTORS
.=24
.PFail ; POWER FAIL HANDLER
340 ; SERVICE AT LEVEL 7
.HLT ; ERROR HANDLER
340 ; SERVICE AT LEVEL 7
.TRPSRV ; GENERAL HANDLER DISPATCH SERVICE
340 ; SERVICE AT LEVEL 7
.=46
LOGICAL ; ACT HOOKS
.=52
.WORD 0
: THIS ROUTINE TRIES TO FORCE THE RECEIVER TO INTERRUPT
: TO ITS VECTOR WHERE IT WILL PICK UP THE STATUS LOCATION
: FOR ITS NEW PC; AND PICK UP AN IOT INSTRUCTION FOR ITS
: NEW PS. WHEN THE NEW PC IS FETCHED AN IOT INSTRUCTION IS
: EXECUTED, TRAPPING TO LOCATION 20 WHERE A ROUTINE IS EXECUTED
: TO TAKE THE PC FROM THE STACK AND USE IT AS THE VECTOR ADDRESS
.=56
VECMAP:
1\$: MOV R1, (R0)+ ; START FILLING THE VECTOR AREA
MOV #4, (R1)+ ; WITH +2; IOT (4)
CMP (R0)+, (R1)+ ; UPDATE THE POINTERS
CMP R1, #1000 ; IS ALL FLOATING VECTOR AREA DONE
BLOS 1\$; BR IF NOT ALL DONE
MOV #1\$ 2#20 ; SET FOR IOT TRAP BY DQ11
MOV DQACTV, TEMP1 ; GET THE ACTIVE DQ11 S
2\$: ROR TEMP1 ; ARE YOU ACTIVE.. DQ11
BCC 5\$; IF CARRY CLEAR.. NO MORE DQ11S
CLR PS ; CLEAR PS
TST (R2)+ ; PUT POINTER TO STATUS TABLE
MOV #340, 2-2(R2) ; TRY AND SET PRI/SEC DONE AND IE
INCB RC ; DELAY.....
BNE -2 ; DELAY
MOVB #300, (R2) ; NO INTERRUPT ASSUME 300 FIX IN TEST C
3\$: TST (R2)+ ; UPDATE POINTERS
BR 2\$; GO DO IT AGAIN
4\$: BIS (SP), (R2) ; ENTERED BY IOT TRAP BY DQ11
BIC #7, (R2) ; CLEAR UNWANTED BITS
CMP (SP)+, (SP)+ ; POP IOT JUNK OFF STACK
MOV #3\$, (SP) ; SET RETURN PC ON STACK
RTI ; GO HOME
5\$: RTS PC ; ALL SIZING IS DONE
; ***SOFTWARE SWITCH REGISTER***
.=174
DISPREG: 0 ; SOFTWARE DISPLAY REGISTER
SWREG: 0 ; SOFTWARE SWITCH REGISTER
; PROGRAM START
.=200
JMP .START ; GO TO START OF PROGRAM

965		000220		.=220					
965	000220	012702	001400	CSRMAP:	MOV	#1400,R2	:	CLEAR ALL STATUS TABLE	
965	000224	005022			CLR	(R2)+	:	DO CLEAR	
965	000226	022702	001512		CMF	#1512,R2	:	ALL TABLE DONE	
965	000232	001374			BNE	-6	:	BR IF MORE TO GO	
965	000234	005037	001504		CLR	DQNUM	:	SET NUMBER OF DQ11S TO 0	
965	000240	012702	001400		MOV	#1400,R2	:	SET TABLE POINTER	
965	000244	012701	160000		MOV	#160000,R1	:	GET FIRST FLOATING ADDRESS	
965	000250	012737	000614	000004	MOV	#55,0#4	:	SET FOR TIME OUT TRAP--NO DEVICE--	
965	000256	112761	000012	000005	1\$:	MOVB	#12,5(R1)	:	TRY AND SEL MISC REGISTER
965	000264	005061	000006		CLR	6(R1)	:	TRY AND CLEAR MISC REG	
965	000270	012711	010000		MOV	#10000,(R1)	:	TRY AND SET RX ACTIVE	
965	000274	022761	030000	000006	CMF	#30000,6(R1)	:	LOOK FOR SYNC 1 AND SYNC 2	
965	000302	001071			BNE	2\$:	THIS IS NOT A DQ11 IF I BRANCH	
965	000304	010122			MOV	R1,(R2)+	:	NOW THIS IS A DQ11 --STORE CSR	
965	000306	052712	100000		BIS	#SYNBIT,(R2)	:	SET FOR TWO SYNC CHARS	
965	000312	005011			CLR	(R1)	:	CLEAR DQ ACTIVE BIT	
965	000314	112761	000010	000005	MOVB	#10,5(R1)	:	SEL CHAR DET REGISTER	
965	000322	012761	177777	000006	MOV	#-1,6(R1)	:	WRITE INTO CHAR DET REG	
965	000330	005761	000006		TST	6(R1)	:	WAS THE REGISTER WRITTEN?	
965	000334	001402			BEQ	+6	:	APPARENTLY NO BB OPTION.	
965	000336	052712	020000		BIS	#BBBIT,(R2)	:	SET FOR BB OPTION	
965	000342	112761	000017	000005	MOVB	#17,5(R1)	:	SEL POLYNO. REGISTER	
965	000350	012761	177777	000006	MOV	#-1,6(R1)	:	WRITE POLYNO. REGISTER	
965	000356	005761	000006		TST	6(R1)	:	WAS REG WRITTEN??	
965	000362	001402			BEQ	+6	:	BR IF NO AB OPTION	
965	000364	052712	002000		BIS	#ABBIT,(R2)	:	SET FOR AB OPTION	
965	000370	012761	001400	000002	MOV	#1400,2(R1)	:	TRY TO SET .DTR. .RS.	
965	000376	032761	001400	000002	BIT	#1400,2(R1)	:	DID ANY OF THEM SET	
965	000404	001402			BEQ	+6	:	BR IF NO BA OPTION	
965	000406	052712	010000		BIS	#BASIT,(R2)	:	SET FOR BA OPTION	
965	000412	032761	030000	000002	BIT	#30000,2(R1)	:	DID .CS. .CO. SET	
965	000420	001402			BEQ	+6	:	BR IF NO JUMPER	
965	000422	052712	040000		BIS	#JUMBIT,(R2)	:	SET FOR JUMPER	
965	000426	052712	004000		BIS	#ACTBIT,(R2)	:	SET FOR ACTIVE ON FIRST NON-SYNC	
965	000432	052712	001000		BIS	#ODDBIT,(R2)	:	SET FOR ODD VRC.....	
1000	000436	005722			TST	(R2)+	:	POP POINTER	
1000	000440	005011			CLR	(R1)	:	CLEAR RCSR	
1000	000442	005061	000002		CLR	2(R1)	:	CLEAR TCSR	
1000	000446	005061	000002		CLR	2(R1)	:	CLEAR AGAIN	
1000	000452	005061	000004		CLR	4(R1)	:	CLEAR ERROR REG	
1000	000456	005061	000006		CLR	6(R1)	:	CLEAR SEC REG	
1000	000462	005237	001504		INC	DQNUM	:	UPDATE NUMBER OF DQ11S	
1000	000466	062701	000010	2\$:	ADD	#10,R1	:	UPDATE CSR POINTER BY 10 .8:	
1000	000472	022701	164000		CMF	#164000,R1	:	HAVE ALL FLOATING ADDRESSES BEEN CHECKED??	
1000	000476	001267			BNE	1\$:	BR IF NOT ALL DONE	
1000	000500	005037	001500		CLR	DQACTV	:	ZERO ACTIVE DQ11S	
1000	000504	005737	001504		TST	DQNUM	:	WERE ANY DQ11S FOUND	
1000	000510	001434			BEQ	4\$:	HEY BUDDY. NO DQ11S FOUND IN SYSTEM	
1000	000512	013701	001504		MOV	DQNUM,R1	:	SAVE NUMBER OF DQ11S	
1000	000516	010137	001276		MOV	R1,SAVNUM	:	SAVE NUMBER FOR ACT11	
1000	000522	000241		3\$:	CLC		:	CLEAR CARRY	
1000	000524	006137	001500		ROL	DQACTV	:	+++++ ACTIVE ADDRESS	
1000	000530	005237	001500		INC	DQACTV	:	SET BIT 0	
1000	000534	005301			DEC	R1	:	DEC NUMBER OF DQ11S	
1000	000536	001371			BNE	3\$:	BR IF MORE TO GO	

```

1021 000540 012737 000006 000004      MOV      #6, J#4      :RESET TIME OUT VECTOR
1022 000546 013737 001500 001502      MOV      DQACTV, SAVACT :SAVE ACTIVE
1023 000554 012737 000340 000022      MOV      #340, J#22   :SET IOT TRAP PRIO: TO 7
1024 000562 012702 001400      MOV      #1400, R2    :SET TABLE POINTER
1025 000566 012700 000300      MOV      #300, R0     :SET VECTOR START
1026 000572 012701 000302      MOV      #302, R1     :SET VECTOR+2 START
1027 000576 000137 000056      JMP      VECMAP       :GO FIND THE VECTORS
1028 000582 104402      4$:      TYPE          :TYPE MESSAGE
1029 000604 016327      MERR2     :I DIDN'T FIND ANY DQ11S. DON'T USE AUTO SIZE.
1030 000606 005000      CLR      R0          :
1031 000610 000000      HALT          :HOW CAN I TEST NO DQ11S
1032 000612 000776      BR      -2          :DON'T LET OPR HIT CCNT. SW
1033 000614 012716 000466      5$:      MOV      #2$, (SP)   :ENTERED BY TIME OUT TRAP
1034 000620 000002      RTI          :GO HOME.
1035
1036
1037
1038 001000 005377 040515 047111      .=1000      MTITLE: .ASCIZ <377>\<12>\MAINDEC-11-DZDQH-B\<377>\CHARACTER LENGTH AND INTERRUPT TESTS/
1039 001006 042504 026503 030461
1040 001014 042055 042132 044121
1041 001022 041055 041777 040510
1042 001030 040522 052103 051105
1043 001036 046040 047105 052103
1044 001044 020110 047101 020104
1045 001052 047111 042524 051122
1046 001060 050125 020124 042524
1047 001066 052123 177523      000
1048
1049      001200      .=1200
1050      :INDIRECT PCINTERS
1051
1052 001200 177570      SWR:      177570      :SWITCH REGISTER POINTER
1053 001202 177570      LIGHTS:   177570     :DISPLAY REGISTER POINTER
1054 001204 177560      TKCSR:    177560     :TELETYPE KEYBOARD CONTROL REGISTER
1055 001206 177562      TKDBR:    177562     :TELETYPE KEYBOARD DATA BUFFER
1056 001210 177564      TPCSR:    177564     :TELEPRINTER CONTROL REGISTER
1057 001212 177566      TPDBR:    177566     :TELEPRINTER DATA BUFFER
1058
1059      :PROGRAM CONTROL PARAMETERS
1060
1061 001214 000000      RETURN:   0          :SCOPE ADDRESS FOR LOOP ON TEST
1062 001216 000000      NEXT:     0          :ADDRESS OF NEXT TEST TO BE EXECUTED
1063 001220 000000      LOCK:     0          :ADDRESS FOR LOCK ON CURRENT DATA
1064 001222 000003      ICOUNT:   3          :NUMBER OF ITERATIONS THAT CURRENT TEST WILL BE EXECUTED
1065 001224 000000      LPCNT:    0          :NUMBER OF ITERATIONS COMPLETED
1066 001226 000000      TSTNO:    0          :NUMBER OF TEST IN PROGRESS
1067 001230 000000      PASCNT:   0          :NUMBER OF PASSES COMPLETED
1068 001232 000000      ERRCNT:   0          :TOTAL NUMBER OF ERRORS
1069 001234 000000      LSTERR:   0          :PC OF LAST ERROR CALL
1070
1071      :PROGRAM VARIABLES
1072
1073 001236 000000      CHAR1:    0
1074 001240 000000      CHAR2:    0
1075 001242 000000      CHAR3:    0
1076 001244 000000      TEMP1:    0          :TEMPORARY STORAGE
  
```

1001	001246	000000
1002	001250	000000
1003	001252	000000
1004	001254	000000
1005	001256	000000
1006	001260	000000
1007	001262	000000
1008	001264	000000
1009	001266	000000
1010	001270	000000
1011	001272	000000
1012	001274	000000
1013	001276	000000
1014	001300	000001
1015	001302	000000
1016	001304	000000
1017	001306	000000

TEMP2:	0
TEMP3:	000
TEMP4:	0000
TEMPS:	000000
SAVR0:	000000
SAVR1:	000000
SAVR2:	000000
SAVR3:	000000
SAVR4:	000000
SAVR5:	000000
SAVSP:	000
SAVPC:	000
SAVNUM:	0
CREAM:	.BLKW 1
RUNFLG:	0
RUN:	0
RUNCNT:	0

:	TEMPORARY STORAGE
:	TEMPORARY STORAGE
:	TEMPORARY STORAGE
:	TEMPORARY STORAGE
:	R0 STORAGE
:	R1 STORAGE
:	R2 STORAGE
:	R3 STORAGE
:	R4 STORAGE
:	R5 STORAGE
:	STACK POINTER STORAGE
:	PROGRAM COUNTER STORAGE

```

1094
1095                ;PROGRAM CONTROL FLAGS
1096
1097 001310      000      INIFLG: .BYTE 0      ;PROGRAM INITIALIZATION FLAG
1098 001311      000      STFLG:  .BYTE 0      ;TEST START FLAG
1099 001312      000      ERRFLG: .BYTE 0      ;ERROR OCCURED FLAG
1100 001313      000      LOKFLG: .BYTE 0      ;LOCK ON CURRENT TEST FLAG
1101                000000
1102
1103                ;DEFINITIONS FOR TRAP SUBROUTINE CALLS
1104                ;POINTERS TO SUBROUTINES CAN BE FOUND
1105                ;IN THE TABLE IMMEDIATLY FOLLOWING THE DEFINITIONS
1106
1107                ;*****
1108                ;*****
1109 001314      TRPTAB:
1110                SCOPE=TRAP+0      ;CALL TO SCOPE LOOP AND ITERATION HANDLER
1111 001314      014240      .SCOPE
1112                SCOPI=TRAP+1      ;CALL TO LOOP ON CURRENT DATA HANDLER
1113 001316      014352      .SCOPI
1114                TYPE=TRAP+2      ;CALL TO TELETYPE OUTPUT ROUTINE
1115 001320      014372      .TYPE
1116                INSTR=TRAP+3      ;CALL TO ASCII STRING INPUT ROUTINE
1117 001322      014500      .INSTR
1118                INSTER=TRAP+4      ;CALL TO INPUT ERROR HANDLER
1119 001324      014616      .INSTER
1120                PARAM=TRAP+5      ;CALL TO NUMERICAL DATA INPUT ROUTINE
1121 001326      014650      .PARAM
1122                SAVOS=TRAP+6      ;CALL TO REGISTER SAVE ROUTINE
1123 001330      015064      .SAVOS
1124                RESOS=TRAP+7      ;CALL TO REGISTER RESTORE ROUTINE
1125 001332      015124      .RESOS
1126                CONVRT=TRAP+10     ;CALL TO DATA OUTPUT ROUTINE
1127 001334      015156      .CONVRT
1128                CNVRT=TRAP+11     ;CALL TO DATA OUTPUT ROUTINE WITHOUT CR/LF.
1129 001336      015162      .CNVRT
1130                MSTCLR=TRAP+12    ;CALL TO ISSUE MASTER CLEAR
1131 001340      012576      .MSTCLR
1132                MEMCLR=TRAP+13    ;CALL TO CLEAR ALL SCRATCH PAD MEMORIES
1133 001342      012452      .MEMCLR
1134                CKSWR=TRAP+14     ;CALL TO ALLOW SWREG TO BE LOADED FROM TTY
1135 001344      016064      .CKSWR
1136                CNTLU=TRAP+15     ;CALL TO ALLOW LOADING OF SWREG FROM TTY.
1137 001346      016140      .CNTLU
1138
1139                ;*****
1140                ;*****
1141
1142                ;DQ11 VECTOR AND REGISTER INDIRECT POINTERS
1143
1144 001350      000000      DQ1VEC: 0      ;POINTER TO DQ11 RECEIVER INTERRUPT VECTOR
1145 001352      000000      DQ1LVL: 0     ;POINTER TO DQ11 RECEIVER INTERRUPT SERVICE PS
1146 001354      000000      DQ1TVEC: 0    ;POINTER TO DQ11 TRANSMITTER INTERRUPT VECTOR
1147 001356      000000      DQ1TLVL: 0    ;POINTER TO DQ11 TRANSMITTER INTERRUPT SERVICE PS
1148 001360      000000      DQ1RCR: 0     ;POINTER TO DQ11 RECEIVER CONTROL REGISTER
1149 001362      000000      DQ1RCSP: 0    ;POINTER TO HIGH BYTE OF DQ11 RECEIVER CONTROL REGISTER

```



```

1150 001364 000000 DQTCR: 0 ; POINTER TO DQ11 TRANSMITTER CONTROL REGISTER
1151 001366 000000 DQERR: 0 ; POINTER TO DQ11 ERROR REGISTER
1152 001370 000000 DQREG: 0 ; POINTER TO HIGH BYTE OF ERROR REGISTER
1153 001372 000000 DQSEC: 0 ; POINTER TO DQ11 SECONDARY REGISTER
1154 001374 000000 DQSECH: 0 ; POINTER TO HIGH BYTE OF DQ11 SECONDARY REGISTER
  
```

;DQ11 STATUS TABLE AND ADDRESS ASSIGNMENTS

```

1160 001400 001400 . = 1400
1161 001400 000001 DQCR00: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 00
1162 001402 000001 DQST00: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 00
1163 001404 000001 DQCR01: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 01
1164 001406 000001 DQST01: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 01
1165 001410 000001 DQCR02: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 02
1166 001412 000001 DQST02: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 02
1167 001414 000001 DQCR03: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 03
1168 001416 000001 DQST03: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 03
1169 001420 000001 DQCR04: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 04
1170 001422 000001 DQST04: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 04
1171 001424 000001 DQCR05: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 05
1172 001426 000001 DQST05: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 05
1173 001430 000001 DQCR06: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 06
1174 001432 000001 DQST06: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 06
1175 001434 000001 DQCR07: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 07
1176 001436 000001 DQST07: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 07
1177 001440 000001 DQCR10: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 10
1178 001442 000001 DQST10: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 10
1179 001444 000001 DQCR11: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 11
1180 001446 000001 DQST11: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 11
1181 001450 000001 DQCR12: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 12
1182 001452 000001 DQST12: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 12
1183 001454 000001 DQCR13: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 13
1184 001456 000001 DQST13: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 13
1185 001460 000001 DQCR14: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 14
1186 001462 000001 DQST14: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 14
1187 001464 000001 DQCR15: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 15
1188 001466 000001 DQST15: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 15
1189 001470 000001 DQCR16: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 16
1190 001472 000001 DQST16: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 16
1191 001474 000001 DQCR17: .BLKW 1 ; CONTROL STATUS REGISTER FOR DEVICE NO: 17
1192 001476 000001 DQST17: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS FOR DEVICE NO: 17
1193 001500 000001 DQACTV: .BLKW 1 ; HOLD ACTIVE BITS FOR TESTING
1194 001502 000001 SAVACT: .BLKW 1 ; SAVE NUMBER OF ACTIVE DQ11S
1195 001504 000001 DQNUM: .BLKW 1 ; OCTAL NUMBER OF TOTAL NUMBER OF DQ11S
1196 001506 000001 DQCSR: .BLKW 1 ; CSR OF DQ11 UNDER TEST
1197 001510 000001 DQSTAT: .BLKW 1 ; VECTOR AND CONFIGURATION STATUS OF DQ11 UNDER TEST
  
```

```

;PROGRAM INITIALIZATION
;LOCK OUT INTERRUPTS
;SET UP PROCESSOR STACK
;SET UP POWER FAIL VECTOR
;CLEAR PROGRAM CONTROL FLAGS AND COUNTS
;TYPE TITLE MESSAGE
  
```

1205

M02

DZDQH MACY11 27(732) 24-SEP-76 10:11 PAGE 26
 DZDQHB.P11 PROGRAM INITIALIZATION AND START UP.

1206	001512	012737	000340	177776	.START:	MOV	#340, PS	; LOCK OUT INTERRUPTS
1207	001520	012706	001200			MOV	#STACK, SP	; SET UP STACK
1208	001524	012737	015766	000024		MOV	#.PFAIL, @#24	; SET UP POWER FAIL VECTOR
1209	001532	013737	001504	001276		MOV	DQNUM, SAVNUM	
1210	001540	105037	001311			CLRB	STFLG	; CLEAR START FLAG
1211	001544	005037	001230			CLR	PASCNT	; CLEAR PASS COUNT
1212	001550	105037	001312			CLRB	ERRFLG	; CLEAR ERROR FLAG
1213	001554	005037	001302			CLR	RUNFLG	
1214	001560	012737	001400	001300		MOV	#1400, CREAM	
1215	001566	005037	001232			CLR	ERRCNT	; CLEAR ERROR COUNT
1216	001572	005037	001234			CLR	LSTERR	; CLEAR LAST ERROR POINTER
1217	001576	012737	000001	001226		MOV	#1, TSTNO	; SET UP FOR TEST 1
1218	001604	012737	001512	001214		MOV	#.START, RETURN	; SET UP FOR POWER FAIL BEFORE
1219								; TESTING STARTS
1220	001612	105737	001310			TSTB	INIFLG	; HAS INITIALIZATION BEEN PERFORMED
1221	001616	001075				BNE	12\$	
1222	001620	104402	001000			TYPE	.MTITLE	; TYPE TITLE MESSAGE
1223	001624	105137	001310			COMB	INIFLG	; IF NOT SET FLAG AND DO
1224								
1225	001630	012737	177570	001200		MOV	#DSWR, SWR	; MOV HARDWARE SWR TO SWR
1226	001636	012737	177570	001202		MOV	#DLIGHTS, LIGHTS	; MOV DISPLAY LIGHTS TO LIGHTS
1227	001644	013746	000006			MOV	@#6, -(SP)	; SAVE VECTORS
1228	001650	013746	000004			MOV	@#4, -(SP)	
1229	001654	012737	001674	000004		MOV	#64\$, @#4	; SET UP FOR TIMEOUT
1230	001662	022777	177777	177310		CMP	#-1, @SWR	; REFERENCE HARDWARE SWITCH REGISTER
1231	001670	001402				BEQ	65\$	
1232	001672	000407				BR	65\$	
1233	001674	022626			64\$:	CMP	(SP)+, (SP)+	; ADJUST STACK
1234	001676	012737	000176	001200	65\$:	MOV	#SWREG, SWR	; POINT TO SOFTWARE SWITCH REG
1235	001704	012737	000174	001202		MOV	#DISPREG, LIGHTS	; POINT TO SOFT DISPLAY REG
1236	001712	012637	000004		66\$:	MOV	(SP)+, @#4	; RESTORE VECTORS
1237	001716	012637	000006			MOV	(SP)+, @#6	
1238	001722	005737	000042			TST	@#42	; UNDER MONITOR
1239	001726	001005				BNE	67\$	
1240	001730	022737	000176	001200		CMP	#SWREG, SWR	; IS SWREG USED
1241	001736	001001				BNE	67\$	
1242	001740	104415				CNTLU		
1243	001742	105777	177232		67\$:	TSTB	@SWR	
1244	001746	100402				BMI	.+6	
1245	001750	004737	000220			JSR	PC, CSRMAP	
1246	001754	104402	016614			TYPE	.XHEAD	
1247	001760	012737	001400	001244		MOV	#1400, TEMP1	
1248	001766	017737	177252	001246		MOV	@TEMP1, TEMP2	
1249	001774	001406				BEQ	.+16	
1250	001776	104410				CONVRT		
1251	002000	016642				XSTATQ		
1252	002002	062737	000002	001244		ADD	#2, TEMP1	
1253	002010	000766				BR	.-22	
1254	002012	032777	000001	177160	12\$:	BIT	#SW00, @SWR	
1255	002020	001424				BEQ	1\$	
1256	002022	104402				TYPE		
1257	002024	016535				MNEW		
1258	002026	005000				CLR	RG	
1259	002030	000000				HALT		
1260	002032	104414				CKSWR		
1261	002034	027737	177140	001502		CMP	@SWR, SAVACT	

```

1262 002042 101404          BLOS      11$
1263 002044 104402          TYPE
1264 002046 016376          MERR3
1265 002050 000000          HALT
1266 002052 000776          BR      -2
1267 002054 017737 177120 001500 11$:  MOV      @SWR,DQACTV
1268 002062 013700 001500          MOV      DQACTV,RO
1269 002066 000000          HALT
1270 002070 104414          CKSWR
1271 002072 012700 000300 1$:  MOV      #300,RO
1272 002076 012701 000302          MOV      #302,R1
1273 002102 010120          2$:  MOV      R1,(R0)+
1274 002104 005021          CLR      (R1)+
1275 002106 022021          CMP      (RO)+,(R1)+
1276 002110 022700 001000          CMP      #1000,RO
1277 002114 001372          BNE      2$
1278
1279          :TEST START AND RESTART
1290
1281 002116 012737 000340 177776 .BEGIN: MOV      #340,PS          ;LOCK OUT INTERRUPTS
1282 002124 012706 001200          MOV      #STACK,SP      ;SET UP STACK
1283 002130 005737 000042          TST      @#42          ;IS PROGRAM UNDER MONITOR CONTROL
1284 002134 001040          BNE      3$
1285 002136 104414          CKSWR          ;CHECK FOR <↑G>
1286 002140 032777 000004 177032          BIT      #BIT2,@SWR      ;CHECK FOR LOCK ON TEST
1287 002146 001411          BEQ      1$
1288 002150 104402 016434          TYPE      .MLOCK
1289 002154 012737 000240 014250          MOV      #NOP,TTST
1290 002162 012737 000240 014252          MOV      #NOP,TTST+2      ;SET UP TO LOCK
1291 002170 000406          BR      2$
1292 002172 013737 014346 014250 1$:  MOV      BRW,TTST
1293 002200 013737 014350 014252          MOV      BRX,TTST+2      ;LOCK NOT SELECTED, SET UP FOR NORMAL SCOPE LOOP
1294 002206 032777 000002 176764 2$:  BIT      #SW01,@SWR      ;IF SW01=1, GET STARTING PC
1295 002214 001410          BEQ      3$
1296 002216 104403          INSTR
1297 002220 016422          MTSTPC
1298 002222 104405          PARAM
1299 002224 002254          TST1
1300 002226 011622          TLAST
1301 002230 000207          RETURN
1302 002232 001          .BYTE 1
1303 002233 001          .BYTE 1
1304 002234 000403          BR      4$
1305 002236 012737 002254 001214 3$:  MOV      #TST1,RETURN      ;START AT TEST 1
1306 002244 104402 016324 4$:  TYPE      MR          ;TYPE R
1307 002250 000177 176740          JMP      @RETURN          ;START TESTING
1308          ; TEST 1
1309          ;*****
1310 002254 012737 000001 001226 †TST1: MOV      #1,TSTNO
1311 002262 012737 002644 001214          MOV      #TST2,RETURN
1312 002270 012737 002644 001216          MOV      #TST2,NEXT
1313 002276 105737 001302          TSTB      RUNFLG          ;IS THIS MY FIRST TIME HERE?
1314 002302 001010          BNE      1$          ;BR IF FLAG IS SET
1315 002304 012737 000001 001304          MOV      #BIT0,RUN          ;SET RUN POINTER.
1316 002312 012737 000020 001306          MOV      #16,RUNCNT      ;SET FOR MAX OF 16 DQ11'S PER SYSTEM
1317 002320 105137 001302          COMB      RUNFLG          ;SET RUN FLAG
  
```

```

: GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
: GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
: GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
: GENERATE ADDRESS OF HIGH BYTE
: GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
: GENERATE ADDRESS OF ERROR REGISTER
: GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
: GENERATE ADDRESS OF SECONDARY REGISTER
: GENERATE ADDRESS OF HIGH BYTE

```

```

: GENERATE ADDRESS OF RECEIVER INTERRUPT SERVICE PS
: GENERATE ADDRESS OF TRANSMITTER INTERRUPT VECTOR
: GENERATE ADDRESS OF TRANSMITTER INTERRUPT SERVICE PS
: GENERATE ADDRESS OF HIGH BYTE
: GENERATE ADDRESS OF TRANSMITTER CONTROL REGISTER
: GENERATE ADDRESS OF ERROR REGISTER
: GENERATE ADDRESS OF HIGH BYTE OF ERROR REGISTER
: GENERATE ADDRESS OF SECONDARY REGISTER
: GENERATE ADDRESS OF HIGH BYTE

```

```

: DQ11 HELL RAISER!!!
: THIS TEST WILL EXERCISE:
: DQ11 RECEIVER AND TRANSMITTER INTERRUPTS
: ENTER T AND EXIT T (IF AB OPTION INSTALLED)
: VRC
: THE CABLE AND TURN AROUND (DATA ONLY)
: CHARACTER TRANSFERS.

```

```

: DQ11 HELL RAISER!!!
: THIS TEST WILL EXERCISE:
: DQ11 RECEIVER AND TRANSMITTER INTERRUPTS
: ENTER T AND EXIT T (IF AB OPTION INSTALLED)
: VRC
: THE CABLE AND TURN AROUND (DATA ONLY)
: CHARACTER TRANSFERS.

```

```

: TEST 2
*****
15: MOV #2, TSTNO
MOV #26, RETURN
MOV #30, ICOUNT
MOV #TST3, NEXT
:ADJUST SYNC CHARACTERS.

100000 001510 BIT #SYNBIT, DOSTAT :ONE SYNC CHAR OR TWO
BNE IS :BR IF TWO
012616 CLRB SYNC :SET ONE SYNC
013422 CLR XSYNC :DBL SYNC SET TO ONE.
SR :CONT.
000026 012616 15: MOVB #26, SYNC :LOAD FOR TWO SYNC
MOV #13026, XSYNC :SAME FOR DBL SYNC
25: MEMCLR :CLEAR ALL REGISTERS GIVE MSTOP
CLR GOCHAR :ZERO POINTER
CLR CHAR
CLR PS
:ZERO PRJC. PRIO.
176410 176410 SETON: CLRB #DQREG :SEL THE RX BA PRI.
MOV #RXBUF, #DQSEC :LOAD RX BA PRI.
INCB #DQREG :SEL RX WC PRI.
MOV #-200, #DQSEC :SET FOR 200 (8) CHARS
INCB #DQREG :SEL THE TX BA PRI.
MOV #SYNC, #DQSEC :LOAD WITH SYNC POINTER
INCB #DQREG :SEL THE TX WC PRI.
MOV #-202, #DQSEC :SET FOR 2 SYNC AND 200 (8) CHARS.
INCB #DQREG :SEL THE RX BA SEC
MOV #XRXBUF, #DQSEC :LOAD RX BA SEC
INCB #DQREG :SEL RX WC SEC
MOV #-200, #DQSEC :SET FOR 200(8) CHARS
INCB #DQREG :SEL THE TX BA SEC
MOV #XTXBUF, #DQSEC :LOAD IT
INCB #DQREG :SEL THE TX WC SEC
MOV #-200, #DQSEC :SET FOR 200 CHARS
MOVB #11, #DQREG :SEL THE SYNC REGISTER
MOV #SYNC, #DQSEC :LOAD SYNC
INCB #DQREG :SEL THE MISC REGISTER
MOV #104000, #DQSEC :SET 8 BITS PER CHAR AND VRC ENABLE.
BIT #JUMBIT, DOSTAT :IS JUMPER AT END OF CABLE?
BNE +10 :BR IF YES
BIS #BIT3, #DQSEC :NO CABLE SET TEST LOOP FOR DATA TURN AROUND
MOVB #17, #DQREG :SEL THE POLY REGISTER
MOV #123456, #DQSEC :SET PLOYNOMIAL.

003150 012700 012620 MOV #TXBFA, R0 :START TO FILL TX BUFFERS
003154 012703 000177 MOV #177, R3 :COUNTER
15: MOVB R3, (R0)+ :PRIMARY IS BINARY COUNT BACKWARDS.
DECB R3 :DONE?
BNE IS :NO
MOV #XTXBUF, R0 :SET SEC BUFFER
25: CLR R3
MOVB R3, (R0)- :SECONDARY IS BINARY COUNT
INCB R3 :DONE?
BPL 25 :NO
MOV #RXISR, #DQREG :SET RECEIVER INTERRUPT POINTER

```


E03

24-SEP-75 10:11 PAGE 31
 PROGRAM INITIALIZATION AND START UP.

1500	003746	105577	175616		INCB	300REG	SEL THE RX WC PRI.
1501	003747	105578	000120	175610	BISB	#BIT6+BIT4,300REG	SET WRITE EN. AND ENTER T
1502	003748	105579	177600	175604	MOV	#-200,300SEC	LOAD RX WC SEC
1503	003749	012701	012620		MOV	#TXBFA,R1	PREPARE TO CHECK DATA. SET TX POINTER
1504	003750	012702	013020		MOV	#RXBUF,R2	SET RX POINTER
1505	003751	000127	003654		JMP	35:	GO AND CHECK DATA
1506	003752	042777	000100	175550	SIC	#BIT6,300CSR	CLEAR RX SEC DONE
1507	003753	112777	000004	175552	MOVB	#4,300REG	SEL RX BA SEC
1508	003754	012777	013630	175546	MOV	#XRXBUF,300SEC	LOAD IT
1509	003755	105577	175540		INCB	300REG	SEL THE RX WC SEC
1510	003756	105578	000060	175532	BISB	#BIT5+BIT4,300REG	SET WRITE EN. AND EXIT T
1511	003757	012777	177600	175526	MOV	#-200,300SEC	WRITE RX WC SEC
1512	003758	012701	013426		MOV	#XTXBUF,R1	GET TX BUFFER POINTER
1513	003759	012702	013630		MOV	#XRXBUF,R2	GET RX POINTER
1514	003760	012700	000200		MOV	#200,R0	GET NUMBER OF CHARS
1515	003761	142711	000200		BICB	#BIT7,(R1)	CLEAR VRC
1516	003762	142712	000200		BICB	#BIT7,(R2)	CLEAR VRC
1517	003763	122122			CMPB	(R1)+,(R2)+	DATA OK?
1518	003764	001414			BEQ	75:	BR IF YES
1519	003765	112777	000012	175456	MOVB	#12,300REG	SEL MISC REG
1520	003766	052777	000002	175462	BIS	#BI+1,300SEC	STOP THE 0011 CLOCK.
1521	003767	114137	014046		MOVB	-(R1),GDCHAR	STORE GOOD CHAR
1522	003768	114237	014040		MOVB	-(R2),CHAR	STORE BAD CHAR.
1523	003769	104003			HLT	3	DATA COMPARE ERROR
1524	003770						*****STRONGLY SUGGEST SW08=1 (GOTO TOP OF TEST OF ERROR
1525	003771	122122			CMPB	(R1)+,(R2)+	POP POINTERS
1526	003772	005300			DEC	R0	ALL DATA CHECKED?
1527	003773	001354			BNE	45:	BR IF NO
1528	003774	005237	001250		DEC	TEMP3	ALL INTERRUPTS DONE?
1529	003775	001003			BNE	65:	NO KEEP INTERRUPTING
1530	003776	000005			RESET		STOP THE SHOW CLEAR THE WORLD
1531	003777	012716	003330		MOV	#ENDT92,SP	SET FOR END TEST RETURN
1532	003778	142777	000017	175416	BICB	#7,300REG	CLEAR REG POINTER
1533	003779	000002			RTI		EXIT STAGE MIDDLE

1521
1522
1523
1524
1525
1526
1527
1528
1529
1530
1531
1532
1533
1534
1535
1536
1537
1538
1539
1540
1541
1542
1543
1544
1545
1546
1547
1548
1549
1550
1551
1552
1553
1554
1555
1556
1557
1558
1559
1560
1561
1562
1563
1564
1565
1566
1567
1568
1569
1570
1571
1572
1573
1574
1575
1576

003754 012737 000003 001226
003756 012737 004314 001216
003770 104413
003772 012700 013022
003776 005001
004000 005020
004002 105201
004004 100375
004006 112777 000011 175354
004014 013737 012616 001246
004022 012737 177774 012134
004030 143737 012134 001246
004036 000241
004040 106037 001246
004044 143737 012134 001247
004052 000241
004054 106037 001247
004060 013737 001246 012136
004066 013737 001246 012140
004074 013777 001246 175270
004102 105277 175262
004106 012777 000010 175256
004114 012700 000016
004120 000300
004122 050077 175244
004126 052777 000002 175236
004134 042777 000002 175230
004142 105077 175222
004146 012777 013022 175216
004154 105277 175210
004160 012777 177734 175204
004166 105277 175176
004172 012777 012140 175172
004200 105277 175164
004204 012777 177732 175160
004212 005277 175142
004216 005277 175142
004222 005005
004224 105777 175130
004230 100404
004232 062705 000001
004236 001372
004240 104001
004242 012700 012142
004246 012701 013022
004252 012702 000044
004256

: TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
: THIS TEST WILL XMIT AND RECV CHARACTERS
: AT 2 BITS PER/CHAR.
: DATA CHECKING WILL BE PERFORMED!

: TEST 3
:*****

TST3: MOV #3, TSTNO
MOV #TST4, NEXT
MEMCLR ; CLEAR ALL THE DQ1;
MOV #RXBUFF, R0 ; LOAD THE BUFFER POINTER
CLR R1 ; SET UP TO CLEAR THE BUFFER
CLR (R0)+ ; CLEAR IT
INCB R1 ; DONE?
BPL SS ; BRANCH IF NO
MOV #11, DDQREG ; SELECT THE SYNC REG
MOV SYNC, TEMP2 ; LOAD SYNC
MOV #177774, MASK ; LOAD THE MASK
BICB MASK, TEMP2 ; SET UP A MASK TO GET THE
CLC ; CORRECT SYNC CHARACTER
RORB TEMP2 ; FOR THIS CHARACTER LENGTH
BICB MASK, TEMP2+1 ; MANIPULATE DATA TO
CLC ; COME UP WITH THE
RORB TEMP2+1 ; PROPER SYNC CHARACTER
MOV TEMP2, SYNC1 ; LOAD THE CHARACTER
MOV TEMP2, SYNC2 ; DITTO
MOV TEMP2, DDQSEC ; LOAD THE SYNC REGISTER
INCB DDQREG ; SEL THE MISC REGISTER
MOV #BIT3, DDQSEC ; SET TEST LOOP
SWAB R0 ; FLIP THE BYTES
BIS R0, DDQSEC ; SET CHARACTER LENGTH
BIS #BIT1, DDQSEC ; TURN CLOCK OFF...
BIC #BIT1, DDQSEC ; AND ON
CLRB DDQREG ; SEL RX PRIMARY ADDRESS
MOV #RXBUFF, DDQSEC ; SET ADDRESS
INCB DDQREG ; SEL RX PRIMARY CHAR COUNT
MOV #-36, DDQSEC ; SET CHAR COUNT
INCB DDQREG ; SEL TX PRIMARY ADDRESS
MOV #SYNC2, DDQSEC ; LOAD THE SYNC CHAR
INCB DDQREG ; SEL TX PRI CHAR COUNT
MOV #-38, DDQSEC ; SET CHAR COUNT
INC DDQRCR ; SET RX GO
INC DDQTCR ; SET TX GO
CLR R5 ; START TIMING
15: TSTB DDQRCR ; IS DONE UP?
BMI 25 ; BRANCH IF YES
ADD #1, R5 ; WAIT
BNE 15 ; BR IF MORE TO GO
25: MOV #TXBUFF, R0 ; ERROR--NO RX DONE
MOV #RXBUFF, R1 ; LOAD BUFFER POINTER
MOV #36, R2 ; LOAD RX BUFFER POINTER
35: ; SET UP TO COUNT CHARACTERS

```

1577 004256 112005          MOVB (R0)+,R5          :GET A CHARACTER TO COPMARE
1578 004260 005037 001246   CLR  TEMP2           :
1579 004264 112137 001246   MOVB (R1)+,TEMP2     :GET REC CHARACTER
1580 004270 013704 001246   MOV  TEMP2,R4       :MOVE TO R4
1581 004274 043705 012134   BIC  MASK,R5        :MASK OUT UNWANTED BITS
1582 004300 020504          CMP  R5,R4          :DO THE CHARACTERS MATCH?
1583 004302 001401          SEQ  43             :BR IF OK
1584 004304 104002          HLT  2             :ERROR--DATA DOESN'T MATCH
1585 004306 005302          DEC  R2             :ALL DONE?
1586 004310 001362          BNE  33            :NO--GO BACK FOR MORE
1587 004312 104400          SCOPE              :SCOPE THIS TEST

```

```

:TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
:THIS TEST WILL XMIT AND RECV CHARACTERS
:AT 3 BITS/PER/CHAR.
:DATA CHECKING WILL BE PERFORMED!

```

```

: TEST 4
*****
1597 004314 012737 000004 001226  TST4: MOV  #4,TSTNO
1598 004322 012737 004654 001216  MOV  #TST5,NEXT
1599 004330 104413          MEMCLR              :CLEAR ALL THE DQ11
1600 004332 012700 013022          MOV  #RXBUFF,R0    :LOAD THE BUFFER PCINTER
1601 004336 005001          CLR  R1            :SET UP TO CLEAR THE BUFFER
1602 004340 005020          CLR  (R0)+         :CLEAR IT
1603 004342 105201          INCB R1            :DONE?
1604 004344 100375          BPL  55            :BRANCH IF NO
1605 004346 112777 000011 175014  MOVB #11,DQREG     :SELECT THE SYNC REG
1606 004354 013737 012616 001246  MOV  SYNC,TEMP2    :LOAD SYNC
1607 004362 012737 177770 012134  MOV  #177770,MASK  :LOAD THE MASK
1608 004370 143737 012134 001246  BICB MASK,TEMP2    :SET UP A MASK TO GET THE
1609 004376 000241          CLC                :CORRECT SYNC CHARACTER
1610 004400 106037 001246          RORB TEMP2         :FOR THIS CHARACTER LENGTH
1611 004404 143737 012134 001247  BICB MASK,TEMP2+1  :MANIPULATE DATA TO
1612 004412 000241          CLC                :COME UP WITH THE
1613 004414 106037 001247          RORB TEMP2+1      :PROPER SYNC CHARACTER
1614 004420 013737 001246 012136  MOV  TEMP2,SYNCl   :LOAD THE CHARACTER
1615 004426 013737 001246 012140  MOV  TEMP2,SYNc2   :DITTO
1616 004434 013777 001246 174730  MOV  TEMP2,DQSEC   :LOAD THE SYNC REGISTER
1617 004442 105277 174722          INCB DQREG        :SEL THE MISC REGISTER
1618 004446 012777 000010 174716  MOV  #BIT3,DQSEC   :SET TEST LOOP
1619 004454 012700 000015          MOV  #15,R0
1620 004460 000300          SWAB R0           :FLIP THE BYTES
1621 004462 050077 174704          BIS  R0,DQSEC      :SET CHARACTER LENGTH
1622 004466 052777 000002 174676  BIS  #BIT1,DQSEC   :TURN CLOCK OFF...
1623 004474 042777 000002 174670  BIC  #BIT1,DQSEC   :AND ON
1624 004502 105077 174662          CLRB DQREG        :SEL RX PRIMARY ADRESS
1625 004506 012777 013022 174656  MOV  #RXBUFF,DQSEC :SET ADRESS
1626 004514 105277 174650          INCB DQREG        :SEL RX PRIMARY CHAR COUNT
1627 004520 012777 177734 174644  MOV  #-36,DQSEC    :SET CHAR COUNT
1628 004526 105277 174636          INCB DQREG        :SEL TX PRIMARY ADDRESS
1629 004532 012777 012140 174632  MOV  #SYNc2,DQSEC  :LOAD THE SYNC CHAR
1630 004540 105277 174624          INCB DQREG        :SEL TX PRI CHAR COUNT
1631 004544 012777 177732 174620  MOV  #-38,DQSEC    :SET CHAR COUNT
1632 004552 005277 174602          INC  DQRC5R       :SET RX GO

```

```

1633 004556 005277 174602      INC      DDQTCR      ;SET TX GC
1634 004562 005005              CLR      R5        ;START TIMING
1635 004564 105777 174570      1$:     TSTB      DDQRCR      ;IS DONE UP?
1636 004570 100404              BMI      2$       ;BRANCH IF YES
1637 004572 062705 000001      ADD      #1,R5    ;WAIT
1638 004576 001372              BNE      1$       ;BR IF MORE TO GO
1639 004600 104001              HLT      1        ;ERROR--NO RX DONE
1640 004602 012700 012142      2$:     MOV      #TXBUFF,R0 ;LOAD BUFFER POINTER
1641 004606 012701 013022      MOV      #RXBUFF,R1 ;LOAD RX BUFFER POINTER
1642 004612 012702 000044      MOV      #36.,R2  ;SET UP TO COUNT CHARACTERS
1643 004616              3$:
1644 004616 112005              MOVB     (R0)+,R5  ;GET A CHARACTER TO COPMARE
1645 004620 005037 001246      CLR      TEMP2    ;
1646 004624 112137 001246      MOVB     (R1)+,TEMP2 ;GET REC CHARACTER
1647 004630 013704 001246      MOV      TEMP2,R4 ;MOVE TO R4
1648 004634 043705 012134      BIC      MASK,R5  ;MASK OUT UNWANTED BITS
1649 004640 020504              CMP      R5,R4    ;DO THE CHARACTERS MATCH?
1650 004642 001401              BEQ      4$       ;BR IF OK
1651 004644 104002              HLT      2        ;ERROR--DATA DOESN'T MATCH
1652 004646 005302 4$:     DEC      R2        ;ALL DONE?
1653 004650 001362              BNE      3$       ;NO--GO BACK FOR MORE
1654 004652 104400              SCOPE           ;SCOPE THIS TEST
    
```

; TEST OF TRANSMITTER AND RECEIVER CHARATER LENGTHS
 ; THIS TEST WILL XMIT AND RECV CHARACTERS
 ; AT 4 BITS/PER/CHAR.
 ; DATA CHECKING WILL BE PERFORMED!

```

1662 ; TEST 5
1663 ; *****
1664 004654 012737 000005 001226 1$T5:  MOV      #5,TSTNO
1665 004662 012737 005214 001216      MOV      #TST6,NEXT
1666 004670 104413              MEMCLR
1667 004672 012700 013022      MOV      #RXBUFF,R0 ;CLEAR ALL THE DQ11
1668 004676 005001              CLR      R1        ;LOAD THE BUFFER POINTER
1669 004700 005020      5$:     CLR      (R0)+    ;SET UP TO CLEAR THE BUFFER
1670 004702 105201              INCB     R1        ;CLEAR IT
1671 004704 100375              BPL      5$       ;DONE?
1672 004706 112777 000011 174454      MOVB     #11,DDQREG ;BRANCH IF NO
1673 004714 013737 012616 001246      MOV      SYNC,TEMP2 ;SELECT THE SYNC REG
1674 004722 012737 177760 012134      MOV      #177760,MASK ;LOAD SYNC
1675 004730 143737 012134 001246      BICB     MASK,TEMP2 ;LOAD THE MASK
1676 004736 000241              CLC
1677 004740 106037 001246              RORB     TEMP2    ;SET UP A MASK TO GET THE
1678 004744 143737 012134 001247      BICB     MASK,TEMP2+1 ;CORRECT SYNC CHARACTER
1679 004752 000241              CLC
1680 004754 106037 001247              RCRB     TEMP2+1  ;FOR THIS CHARACTER LENGTH
1681 004760 013737 001246 012136      MOV      TEMP2,SYNC1 ;MANIPULATE DATA TO
1682 004766 013737 001246 012140      MOV      TEMP2,SYNC2 ;COME UP WITH THE
1683 004774 013777 001246 174370      MOV      TEMP2,DDQSEC ;PROPER SYNC CHARACTER
1684 005002 105277 174362              INCB     DDQREG   ;LOAD THE CHARACTER
1685 005006 012777 000010 174356      MOV      #BIT3,DDQSEC ;DITTO
1686 005014 012700 000014              MOV      #14,R0   ;LOAD THE SYNC REGISTER
1687 005020 000300              SWAB     R0        ;SEL THE MISC REGISTER
1688 005022 050077 174344      BIS      R0,DDQSEC ;SET TEST LOOP
    ; FLIP THE BYTES
    ; SET CHARACTER LENGTH
    
```

```

:689 005026 052777 000002 174336 BIS #BIT1,DDQSEC ;TURN CLOCK OFF...
:690 005034 042777 000002 174330 BIC #BIT1,DDQSEC ;AND ON
:691 005042 105077 174322 CLR DDQREG ;SEL RX PRIMARY ADDRESS
:692 005046 012777 013022 174315 MOV #RXBUFF,DDQSEC ;SET ADDRESS
:693 005054 105277 174310 INCB DDQREG ;SEL RX PRIMARY CHAR COUNT
:694 005060 012777 177734 174304 MOV #-36,DDQSEC ;SET CHAR COUNT
:695 005066 105277 174276 INCB DDQREG ;SEL TX PRIMARY ADDRESS
:696 005072 012777 012140 174272 MOV #SYNC2,DDQSEC ;LOAD THE SYNC CHAR
:697 005100 105277 174264 INCB DDQREG ;SEL TX PRI CHAR COUNT
:699 005104 012777 177732 174260 MOV #-38,DDQSEC ;SET CHAR COUNT
:699 005112 005277 174242 INC DDQRC5R ;SET RX GO
:700 005116 005277 174242 INC DDQTC5R ;SET TX GO
:701 005122 005005 CLR R5 ;START TIMING
:702 005124 005777 174230 1$: TSTB DDQRC5R ;IS DONE UP?
:703 005130 100404 BMI 2$ ;BRANCH IF YES
:704 005132 062705 000001 ADD #1,R5 ;WAIT
:705 005136 001372 BNE 1$ ;BR IF MORE TO GO
:706 005140 104001 HLT 1 ;ERROR--NO RX DONE
:707 005142 012700 012142 2$: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
:708 005146 012701 013022 MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
:709 005152 012702 000044 MOV #36,,R2 ;SET UP TO COUNT CHARACTERS
:710 005156 3$: ;
:711 005156 112005 MOV8 (R0)+,R5 ;GET A CHARACTER TO COPMARE
:712 005160 005037 001246 CLR TEMP2 ;
:713 005164 112137 001246 MOV8 (R1)+,TEMP2 ;GET REC CHARACTER
:714 005170 013704 001246 MOV TEMP2,R4 ;MOVE TO R4
:715 005174 043705 012134 BIC MASK,R5 ;MASK OUT UNWANTED BITS
:716 005200 020504 CMP R5,R4 ;DO THE CHARACTERS MATCH?
:717 005202 001401 BEQ 4$ ;BR IF OK
:718 005204 104002 HLT 2 ;ERROR--DATA DOESN'T MATCH
:719 005206 005302 4$: DEC R2 ;ALL DONE?
:720 005210 001362 BNE 3$ ;NO--GO BACK FOR MORE
:721 005212 104400 SCOPE ;SCOPE THIS TEST

```

```

:TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
:THIS TEST WILL XMIT AND RECV CHARACTERS
:AT 5 BITS/PER/CHAR.
:DATA CHECKING WILL BE PERFORMED!

```

```

: TEST 6
:*****
:731 005214 012737 000006 001226 TST6: MOV #6,TSTNO
:732 005222 012737 005554 001216 MOV #TST7,NEXT
:733 005230 104413 MEMCLR ;CLEAR ALL THE DQ11
:734 005232 012700 013022 MOV #RXBUFF,R0 ;LOAD THE BUFFER POINTER
:735 005236 005001 CLR R1 ;SET UP TO CLEAR THE BUFFER
:736 005240 005020 5$: CLR (R0)+ ;CLEAR IT
:737 005242 105201 INCB R1 ;DONE?
:738 005244 100375 BPL 5$ ;BRANCH IF NO
:739 005246 112777 000011 174114 MOV8 #11,DDQREG ;SELECT THE SYNC REG
:740 005254 013737 012616 001246 MOV SYNC,TEMP2 ;LOAD SYNC8
:741 005262 012737 177740 012134 MOV #177740,MASK ;LOAD THE MASK
:742 005270 143737 012134 001246 BIC8 MASK,TEMP2 ;SET UP A MASK TO GET THE
:743 005276 000241 CLC ;CORRECT SYNC CHARACTER
:744 005300 106037 001246 RORB TEMP2 ;FOR THIS CHARACTER LENGTH

```

```

1745 005304 143737 012134 001247 BICB MASK,TEMP2+1 ;MANIPULATE DATA TO
1746 005312 000241 CLC ;COME UP WITH THE
1747 005314 106037 001247 RORB TEMP2+1 ;PROPER SYNC CHARACTER
1748 005320 013737 001246 012136 MOV TEMP2,SYNC1 ;LOAD THE CHARACTER
1749 005326 013737 001246 012140 MOV TEMP2,SYNC2 ;DITTO
1750 005334 013777 001246 174030 MOV TEMP2,JDQSEC ;LOAD THE SYNC REGISTER
1751 005342 105277 174022 INCB JDQREG ;SEL THE MISC REGISTER
1752 005346 012777 000010 174016 MOV #BIT3,JDQSEC ;SET TEST LOOP
1753 005354 012700 000013 MOV #13,R0 ;
1754 005360 000300 SWAB R0 ;FLIP THE BYTES
1755 005362 050077 174004 BIS R0,JDQSEC ;SET CHARACTER LENGTH
1756 005366 052777 000002 173776 BIS #BIT1,JDQSEC ;TURN CLOCK OFF...
1757 005374 042777 000002 173770 BIC #BIT1,JDQSEC ;AND ON
1758 005402 105077 173762 CLRB JDQREG ;SEL RX PRIMARY ADDRESS
1759 005406 012777 013022 173756 MOV #RXBUFF,JDQSEC ;SET ADDRESS
1760 005414 105277 173750 INCB JDQREG ;SEL RX PRIMARY CHAR COUNT
1761 005420 012777 177734 173744 MOV #-36,JDQSEC ;SET CHAR COUNT
1762 005426 105277 173736 INCB JDQREG ;SEL TX PRIMARY ADDRESS
1763 005432 012777 012140 173732 MOV #SYNC2,JDQSEC ;LOAD THE SYNC CHAR
1764 005440 105277 173724 INCB JDQREG ;SEL TX PRI CHAR COUNT
1765 005444 012777 177732 173720 MOV #-38,JDQSEC ;SET CHAR COUNT
1766 005452 005277 173702 INC JDQRCR ;SET RX GO
1767 005456 005277 173702 INC JDQTCR ;SET TX GO
1768 005462 005005 CLR R5 ;START TIMING
1769 005464 105777 173670 1$: TSTB JDQRCR ;IS DONE UP?
1770 005470 100404 BMI 2$ ;BRANCH IF YES
1771 005472 062705 000001 ADD #1,R5 ;WAIT
1772 005476 001372 BNE 1$ ;BR IF MORE TO GO
1773 005500 104001 HLT 1 ;ERROR--NO RX DONE
1774 005502 012700 012142 2$: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
1775 005506 012701 013022 MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
1776 005512 012702 000044 MOV #36,R2 ;SET UP TO COUNT CHARACTERS
1777 005516 3$: ;
1778 005516 112005 MOVB (R0)+,R5 ;GET A CHARACTER TO COMPARE
1779 005520 005037 001246 CLR TEMP2 ;
1780 005524 112137 001246 MOVB (R1)+,TEMP2 ;GET REC CHARACTER
1781 005530 013704 001246 MOV TEMP2,R4 ;MOVE TO R4
1782 005534 043705 012134 BIC MASK,R5 ;MASK OUT UNWANTED BITS
1783 005540 020504 CMP R5,R4 ;DO THE CHARACTERS MATCH?
1784 005542 001401 BEQ 4$ ;BR IF OK
1785 005544 104002 HLT 2 ;ERROR--DATA DOESN'T MATCH
1786 005546 005302 4$: DEC R2 ;ALL DONE?
1787 005550 001362 BNE 3$ ;NO--GO BACK FOR MORE
1788 005552 104400 SCOPE ;SCOPE THIS TEST
1789
1790
1791 ;TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
1792 ;THIS TEST WILL XMIT AND REC/ CHARACTERS
1793 ;AT 6 BITS/PER/CHAR.
1794 ;DATA CHECKING WILL BE PERFORMED!
1795
1796 ; TEST 7
1797 ;*****
1798 005554 012737 000007 001226 TST7: MOV #7,TSTNO
1799 005562 012737 006114 001216 MOV #TSTIO,NEXT
1800 005570 104413 MEMCLR ;CLEAR ALL THE DQ11

```

1801	005572	012700	013022		MOV	#RXBUFF,R0	:LOAD THE BUFFER POINTER
1802	005576	005001			CLR	R1	:SET UP TO CLEAR THE BUFFER
1803	005600	005020		5\$:	CLR	(R0)+	:CLEAR IT
1804	005602	105201			INCB	R1	:DONE?
1805	005604	100375			BPL	5\$:BRANCH IF NO
1806	005606	112777	000011	173554	MOVB	#11,ADQREG	:SELECT THE SYNC REG
1807	005614	013737	012616	001246	MOV	SYNC,TEMP2	:LOAD SYNC
1808	005622	012737	177700	012134	MOV	#177700,MASK	:LOAD THE MASK
1809	005630	143737	012134	001246	BICB	MASK,TEMP2	:SET UP A MASK TO GET THE
1810	005636	000241			CLC		:CORRECT SYNC CHARACTER
1811	005640	106037	001246		RORB	TEMP2	:FOR THIS CHARACTER LENGTH
1812	005644	143737	012134	001247	BICB	MASK,TEMP2+1	:MANIPULATE DATA TO
1813	005652	000241			CLC		:COME UP WITH THE
1814	005654	106037	001247		RORB	TEMP2+1	:PROPER SYNC CHARACTER
1815	005660	013737	001246	012136	MOV	TEMP2,SYNC1	:LOAD THE CHARACTER
1816	005666	013737	001246	012140	MOV	TEMP2,SYNC2	:DITTO
1817	005674	013777	001246	173470	MOV	TEMP2,ADQSEC	:LOAD THE SYNC REGISTER
1818	005702	105277	173462		INCB	ADQREG	:SEL THE MISC REGISTER
1819	005706	012777	000010	173456	MOV	#BIT3,ADQSEC	:SET TEST LOOP
1820	005714	012700	000012		MOV	#12,R0	
1821	005720	000300			SWAB	R0	:FLIP THE BYTES
1822	005722	050077	173444		BIS	R0,ADQSEC	:SET CHARACTER LENGTH
1823	005726	052777	000002	173436	BIS	#BIT1,ADQSEC	:TURN CLOCK OFF...
1824	005734	042777	000002	173430	BIC	#BIT1,ADQSEC	:AND ON
1825	005742	105077	173422		CLRB	ADQREG	:SEL RX PRIMARY ADDRESS
1826	005746	012777	013022	173416	MOV	#RXBUFF,ADQSEC	:SET ADDRESS
1827	005754	105277	173410		INCB	ADQREG	:SEL RX PRIMARY CHAR COUNT
1828	005760	012777	177734	173404	MOV	#-36,ADQSEC	:SET CHAR COUNT
1829	005766	105277	173376		INCB	ADQREG	:SEL TX PRIMARY ADDRESS
1830	005772	012777	012140	173372	MOV	#SYNC2,ADQSEC	:LOAD THE SYNC CHAR
1831	006000	105277	173364		INCB	ADQREG	:SEL TX PRI CHAR COUNT
1832	006004	012777	177732	173360	MOV	#-38,ADQSEC	:SET CHAR COUNT
1833	006012	005277	173342		INC	ADQRCSR	:SET RX GO
1834	006016	005277	173342		INC	ADQTCSR	:SET TX GO
1835	006022	005005			CLR	R5	:START TIMING
1836	006024	105777	173330		TSTB	ADQRCSR	:IS DONE UP?
1837	006030	100404			BMI	2\$:BRANCH IF YES
1838	006032	062705	000001		ADD	#1,R5	:WAIT
1839	006036	001372			BNE	1\$:BR IF MORE TO GO
1840	006040	104001			HLT	1	:ERROR--NO RX DONE
1841	006042	012700	012142		MOV	#TXBUFF,R0	:LOAD BUFFER POINTER
1842	006046	012701	013022		MOV	#RXBUFF,R1	:LOAD RX BUFFER POINTER
1843	006052	012702	000044		MOV	#36.,R2	:SET UP TO COLNT CHARACTERS
1844	006056			3\$:			
1845	006056	112005			MOVB	(R0)+,R5	:GET A CHARACTER TO COMPARE
1846	006060	005037	001246		CLR	TEMP2	
1847	006064	112137	001246		MOVB	(R1)+,TEMP2	:GET REC CHARACTER
1848	006070	013704	001246		MOV	TEMP2,R4	:MOVE TO R4
1849	006074	043705	012134		BIC	MASK,R5	:MASK OUT UNWANTED BITS
1850	006100	020504			CMP	R5,R4	:DO THE CHARACTERS MATCH?
1851	006102	001401			BEG	4\$:BR IF OK
1852	006104	104002			HLT	2	:ERROR--DATA DOESN'T MATCH
1853	006106	005302		4\$:	DEC	R2	:ALL DONE?
1854	006110	001362			BNE	3\$:NO--GO BACK FOR MORE
1855	006112	104400			SCOPE		:SCOPE THIS TEST
1856							


```

1857
1858                                     : TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
1859                                     : THIS TEST WILL XMIT AND RECV CHARACTERS
1860                                     : AT 7 BITS/PER/CHAR.
1861                                     : DATA CHECKING WILL BE PERFORMED!
1862
1863                                     : TEST 10
1864                                     :*****
1865 006114 012737 000010 001226  TST10: MOV #10,TSTNO
1866 006122 012737 006454 001216  MOV #TST11,NEXT
1867 006130 104413 MEMCLR
1868 006132 012700 013022 MOV #RXBUFF,RO ;CLEAR ALL THE DQ11
1869 006136 005001 CLR R1 ;LOAD THE BUFFER POINTER
1870 006140 005020 5$: CLR (RO)+ ;SET UP TO CLEAR THE BUFFER
1871 006142 105201 INCB R1 ;CLEAR IT
1872 006144 100375 BPL 5$ ;DONE?
1873 006146 112777 000011 173214 MOVB #11,ADQREG ;BRANCH IF NO
1874 006154 013737 012616 001246 MOV SYNC,TEMP2 ;SELECT THE SYNC REG
1875 006162 012737 177600 012134 MOV #177600,MASK ;LOAD SYNC
1876 006170 143737 012134 001246 BICB MASK,TEMP2 ;LOAD THE MASK
1877 006176 000241 CLC ;SET UP A MASK TO GET THE
1878 006200 106037 001246 RORB TEMP2 ;CORRECT SYNC CHARACTER
1879 006204 143737 012134 001247 BICB MASK,TEMP2+1 ;FOR THIS CHARACTER LENGTH
1880 006212 000241 CLC ;MANIPULATE DATA TO
1881 006214 106037 001247 RORB TEMP2+1 ;COME UP WITH THE
1882 006220 013737 001246 012136 MOV TEMP2,SYNC1 ;PROPER SYNC CHARACTER
1883 006226 013737 001246 012140 MOV TEMP2,SYNC2 ;LOAD THE CHARACTER
1884 006234 013777 001246 173130 MOV TEMP2,ADQSEC ;DITTO
1885 006242 105277 173122 INCB ADQREG ;LOAD THE SYNC REGISTER
1886 006246 012777 000010 173116 MOV #BIT3,ADQSEC ;SEL THE MISC REGISTER
1887 006254 012700 000011 MOV #11,RO ;SET TEST LOOP
1888 006260 000300 SWAB RO ;FLIP THE BYTES
1889 006262 050077 173104 BIS RO,ADQSEC ;SET CHARACTER LENGTH
1890 006266 052777 000002 173076 BIS #BIT1,ADQSEC ;TURN CLOCK OFF...
1891 006274 042777 000002 173070 BIC #BIT1,ADQSEC ;AND ON
1892 C 5302 105077 173062 CLRB ADQREG ;SEL RX PRIMARY ADDRESS
1893 006306 012777 013022 173056 MOV #RXBUFF,ADQSEC ;SET ADDRESS
1894 006314 105277 173050 INCB ADQREG ;SEL RX PRIMARY CHAR COUNT
1895 006320 012777 177734 173044 MOV #-36,ADQSEC ;SET CHAR COUNT
1896 006326 105277 173036 INCF ADQREG ;SEL TX PRIMARY ADDRESS
1897 006332 012777 012140 173032 MOV #SYNC2,ADQSEC ;LOAD THE SYNC CHAR
1898 006340 105277 173024 INCB ADQREG ;SEL TX PRI CHAR COUNT
1899 006344 012777 177732 173020 MOV #-38,ADQSEC ;SET CHAR COUNT
1900 006352 005277 173002 INC ADQRC5R ;SET RX GO
1901 006356 005277 173002 INC ADQTC5R ;SET TX GO
1902 006362 005005 CLR R5 ;START TIMING
1903 006364 105777 172770 1$: TSTB ADQRC5R ;IS DONE UP?
1904 006370 100404 BMI 2$ ;BRANCH IF YES
1905 006372 062705 000001 ADD #,R5 ;WAIT
1906 006376 001372 BNE 1$ ;BR IF MORE TO GO
1907 006400 104001 HLT 1 ;ERROR--NO RX DONE
1908 006402 012700 012142 2$: MOV #TXBUFF,RO ;LOAD BUFFER POINTER
1909 006406 012701 013022 MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
1910 006412 012702 000044 MOV #36.,R2 ;SET UP TO COUNT CHARACTERS
1911 006416 3$:
1912 006416 112005 MOVB (RO)+,R5 ;GET A CHARACTER TO COMPARE

```

M03

DZDQH MACY11 27(732) 24-SEP-76 10:11 PAGE 39
 DZDQHB.P11 PROGRAM INITIALIZATION AND START UP.

1913	006420	005037	001246	CLR	TEMP2	::
1914	006424	112137	001246	MOVB	(R1)+,TEMP2	::GET REC CHARACTER
1915	005430	013704	001246	MOV	TEMP2,R4	::MOVE TO R4
1916	006434	043705	012134	BIC	MASK,R5	::MASK OUT UNWANTED BITS
1917	006440	020504		CMP	R5,R4	::DO THE CHARACTERS MATCH?
1918	006442	001401		BEQ	4\$::BR IF OK
1919	006444	104002		HLT	2	::ERROR--DATA DOESN'T MATCH
1920	006446	005302	4\$:	DEC	R2	::ALL DONE?
1921	006450	001362		BNE	3\$::NO--GO BACK FOR MORE
1922	006452	104400		SCOPE		::SCOPE THIS TEST

: TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
 : THIS TEST WILL XMIT AND RECV CHARACTERS
 : AT 9 BITS/PER/CHAR.
 : DATA CHECKING WILL BE PERFORMED!

: TEST 11
 :*****

1932	006454	012737	000011	001226	TST11: MOV	#11,TSTNO	
1933	006462	012737	007014	001216	MOV	#TST12,NEXT	
1934	006470	104413			MEMCLR		:CLEAR ALL THE DQ11
1935	006472	012700	013022		MOV	#RXBUFF,R0	:LOAD THE BUFFER POINTER
1936	006476	005001			CLR	R1	:SET UP TO CLEAR THE BUFFER
1937	006500	005020			CLR	(R0)+	:CLEAR IT
1938	006502	105201			INCB	R1	:DONE?
1939	006504	100375			BPL	5\$:BRANCH IF NO
1940	006506	112777	000011	172654	MOVB	#11,ADQREG	:SELECT THE SYNC REG
1941	006514	013737	012616	001246	MOV	SYNC,TEMP2	:LOAD SYNC
1942	006522	012737	177400	012134	MOV	#177400,MASK	:LOAD THE MASK
1943	006530	143737	012134	001246	BICB	MASK,TEMP2	:SET UP A MASK TO GET THE
1944	006536	000241			CLC		:CORRECT SYNC CHARACTER
1945	006540	106037	001246		RORB	TEMP2	:FOR THIS CHARACTER LENGTH
1946	006544	143737	012134	001247	BICB	MASK,TEMP2+1	:MANIPULATE DATA TO
1947	006552	000241			CLC		:COME UP WITH THE
1948	006554	106037	001247		RORB	TEMP2+1	:PROPER SYNC CHARACTER
1949	006560	013737	001246	012136	MOV	TEMP2,SYNC1	:LOAD THE CHARACTER
1950	006566	013737	001246	012140	MOV	TEMP2,SYNC2	:DITTO
1951	006574	013777	001246	172570	MOV	TEMP2,ADQSEC	:LOAD THE SYNC REGISTER
1952	006602	105277	172562		INCB	ADQREG	:SEL THE MISC REGISTER
1953	006606	012777	000010	172556	MOV	#BIT3,ADQSEC	:SET TEST LOOP
1954	006614	012700	000010		MOV	#10,R0	
1955	006620	000300			SWAB	R0	:FLIP THE BYTES
1956	006622	050077	172544		BIS	R0,ADQSEC	:SET CHARACTER LENGTH
1957	006626	052777	000002	172536	BIS	#BIT1,ADQSEC	:TURN CLOCK OFF...
1958	006634	042777	000002	172530	BIC	#BIT1,ADQSEC	:AND ON
1959	006642	105077	172522		CLRB	ADQREG	:SEL RX PRIMARY ADDRESS
1960	006646	012777	013022	172516	MOV	#RXBUFF,ADQSEC	:SET ADDRESS
1961	006654	105277	172510		INCB	ADQREG	:SEL RX PRIMARY CHAR COUNT
1962	006660	012777	177734	172504	MOV	#-36,ADQSEC	:SET CHAR COUNT
1963	006666	105277	172476		INCB	ADQREG	:SEL TX PRIMARY ADDRESS
1964	006672	012777	012140	172472	MOV	#SYNC2,ADQSEC	:LOAD THE SYNC CHAR
1965	006700	105277	172464		INCB	ADQREG	:SEL TX PRI CHAR COUNT
1966	006704	012777	177732	172460	MOV	#-38,ADQSEC	:SET CHAR COUNT
1967	006712	005277	172442		INC	ADQRC5R	:SET RX GO
1968	006716	005277	172442		INC	ADQTC5R	:SET TX GO

```

1969 006722 005005          CLR      R5          ; START TIMING
1970 006724 105777 172430 1$:  TSTB   @DQRCR      ; IS DONE UP?
1971 006730 100404          BMI     2$          ; BRANCH IF YES
1972 006732 062705 000001  ADD     #1,R5        ; WAIT
1973 006736 001372          BNE     1$          ; BR IF MORE TO GO
1974 006740 104001          HLT     1           ; ERROR--NO RX DONE
1975 006742 012700 012142 2$:  MOV     #TXBUFF,R0   ; LOAD BUFFER POINTER
1976 006746 012701 013022  MOV     #RXBUFF,R1   ; LOAD RX BUFFER POINTER
1977 006752 012702 000044  MOV     #36.,R2      ; SET UP TO COUNT CHARACTERS
1978 006756
1979 006756 112005          MOVB   (R0)+,R5      ; GET A CHARACTER TO COMPARE
1980 006750 005037 001246  CLR     TEMP2        ;
1981 006754 112137 001246  MOVB   (R1)+,TEMP2   ; GET REC CHARACTER
1982 006770 013704 001246  MOV     TEMP2,R4     ; MOVE TO R4
1983 006774 043705 012134  BIC    MASK,R5       ; MASK OUT UNWANTED BITS
1984 007000 020504          CMP     R5,R4        ; DO THE CHARACTERS MATCH?
1985 007002 001401          BEQ    4$          ; BR IF OK
1986 007004 104002          HLT     2           ; ERROR--DATA DOESN'T MATCH
1987 007006 005302 4$:  DEC     R2           ; ALL DONE?
1988 007010 001362          BNE    3$          ; NO--GO BACK FOR MORE
1989 007012 104400          SCOPE 3$          ; SCOPE THIS TEST

```

```

; TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
; THIS TEST WILL XMIT AND RECV CHARACTERS
; AT 9 BITS/PER/CHAR.
; DATA CHECKING WILL BE PERFORMED!

```

```

1990
1991
1992
1993
1994
1995
1996
1997
1998
1999 007014 012737 000012 001226 ; TEST 12
2000 007022 012737 007326 001216 ; *****
2001 007030 104413          MEMCLR
2002 007032 012700 013022  MOV     #RXBUFF,R0   ; CLEAR ALL THE DQ11
2003 007036 005001          CLR     R1           ; LOAD THE BUFFER POINTER
2004 007040 005020 5$:  CLR     (R0)+        ; SET UP TO CLEAR THE BUFFER
2005 007042 105201          INCB   R1           ; CLEAR IT
2006 007044 100375          BPL    5$          ; DONE?
2007 007046 112777 000011 172314 MOVB   #11,@DQREG    ; BRANCH IF NO
2008 007054 013737 012616 001246 MOV     SYNC,TEMP2   ; SELECT THE SYNC REG
2009 007062 012737 177000 012134 MOV     #177000,MASK ; LOAD SYNC
2010 007070 043737 012134 001246 BIC    MASK,TEMP2    ; LOAD THE MASK
2011 007076 000241          CLC
2012 007100 006037 001246  ROR     TEMP2        ; SET UP THE MASK FOR THE
2013 007104 013737 001246 012136 MOV     TEMP2,SYNC1  ; CORRECT SYNC CHARACTER
2014 007112 013737 001246 012140 MOV     TEMP2,SYNC2  ; SHIFT IT
2015 007120 013777 001246 172244 MOV     TEMP2,@DQSEC ; LOAD THE CHARACTER
2016 007126 105277 172236          INCB   @DQREG       ; DITTO
2017 007132 012777 000010 172232 MOV     #BIT3,@DQSEC ; LOAD THE SYNC REGISTER
2018 007140 012700 000007          MOV     #7,R0        ; SEL THE MISC REGISTER
2019 007144 000300          SWAB   R0           ; SET TEST LOOP
2020 007146 050077 172220          BIS    R0,@DQSEC    ; FLIP THE BYTES
2021 007152 052777 000002 172212 BIS    #BIT1,@DQSEC  ; SET CHARACTER LENGTH
2022 007160 042777 000002 172204 BIC    #BIT1,@DQSEC  ; TURN CLOCK OFF...
2023 007166 105077 172176          CLRB  @DQREG       ; AND ON
2024 007172 012777 013022 172172 MOV     #RXBUFF,@DQSEC ; SEL RX PRIMARY ADDRESS
                          ; SET ADDRESS

```

```

:SEL RX PRIMARY CHAR COUNT
:SET CHAR COUNT
:SEL TX PRIMARY ADDRESS
:LOAD THE SYNC CHAR
:SET CHAR COUNT
:SET RX GO
:SET TX GO
:START TIMING
:IS DONE UP?
:BRANCH IF YES
:WAIT
:BR IF MORE TO GO
:ERROR--NO RX DONE
:LOAD BUFFER POINTER
:LOAD RX BUFFER POINTER
:SET UP TO COUNT CHARACTERS
:GET ANOTHER CHAR
:GET A REC CHAR
:MASK OUT UNWANTED BITS
:DO THE CHARACTERS MATCH?
:BR IF OK
:ERROR--DATA DOESN'T MATCH
:ALL DONE?
:NO--GO BACK FOR MORE
:SCOPE THIS TEST

```

```

:TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
:THIS TEST WILL XMIT AND RECV CHARACTERS
:AT 10 BITS PER CHAR.
:DATA CHECKING WILL BE PERFORMED!

```

```

: TEST 13
*****
:ST13: MOV #13,TSTNO
MOV #TST+14,NEXT
MEMCLR
:CLR ALL THE 0011
:LOAD THE BUFFER POINTER
:SET UP TO CLEAR THE BUFFER
:CLR R1
:CLR (R0)+
:CLR IT
:DONE?
:BRANCH IF NO
:SELECT THE SYNC REG
:LOAD SYNC
:LOAD THE MASK
:SET UP THE MASK FOR THE
:CORRECT SYNC CHARACTER
:SHIFT IT
:LOAD THE CHARACTER
:BIT0
:LOAD THE SYNC REGISTER
:SEL THE MISC REGISTER
:SET TEST LOOP

```

```

000013 001226
007640 001226
013022
000011 172002
012616 001246
012727 012134
042737 001246
000041
006037 001246
012737 012136
012737 012140
012777 001246 171732
000077 171720
000077 171720

```

```

: FILL THE BYTES
: SET CHARACTER LENGTH
: TURN CLOCK OFF...
: AND ON
: SEL RX PRIMARY ADDRESS
: SET ADDRESS
: SEL RX PRIMARY CHAR COUNT
: SET CHAR COUNT
: SEL TX PRIMARY ADDRESS
: LOAD THE SYNC CHAR
: SET TX PRI CHAR COUNT
: SET CHAR COUNT
: SET RX GO
: SET TX GO
: START TIMING
: IS DONE UP?
: BRANCH IF YES
: WAIT
: BR IF MORE TO GO
: ERROR--NO RX DONE
: LOAD BUFFER POINTER
: LOAD RX BUFFER POINTER
: SET UP TO COUNT CHARACTERS

: GET ANOTHER CHAR
: GET A REC CHAR
: MASK OUT UNWANTED BITS
: DO THE CHARACTERS MATCH?
: BR IF OK
: ERROR--DATA DOESN'T MATCH
: ALL DONE?
: NO--GO BACK FOR MORE
: SCOPE THIS TEST

```

```

: TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
: THIS TEST WILL XMIT AND RECV CHARACTERS
: AT 11 BITS/PER CHAR.
: DATA CHECKING WILL BE PERFORMED!

```

: TEST 14

```

:ST14: MOV #14, TSTNO
MOV #TST15, NEXT
MEMCLR
MOV #RXBUFF, R0
CLR R1
: CLR (R0)+
CLR R1
INCB R1
BPL SS
MOV #11, @DQREG
MOV SYNC, TEMP2
MOV #17400, MASK
BIC MASK, TEMP2
CLC
ROR TEMP2
: CLEAR ALL THE DQ11
: LOAD THE BUFFER POINTER
: SET UP TO CLEAR THE BUFFER
: CLEAR IT
: DONE?
: BRANCH IF NO
: SELECT THE SYNC REG
: LOAD SYNC
: LOAD THE MASK
: SET UP THE MASK FOR THE
: CORRECT SYNC CHARACTER
: SHIFT IT

```

```

001246 012135 MOV TEMP2, SYNC1 ;LOAD THE CHARACTER
001247 012140 MOV TEMP2, SYNC2 ;DITTO
001248 171420 MOV TEMP2, JDDQSEC ;LOAD THE SYNC REGISTER
001249 171420 INCB JDDREG ;SEL THE MISC REGISTER
001250 171426 MOV #BIT3, JDDQSEC ;SET TEST LOOP
001251 000000 MOV #5, R0 ;
001252 000000 SWAB R0 ;FLIP THE BYTES
001253 000000 MOV #0, R0 ;SET CHARACTER LENGTH
001254 171366 BIT3, JDDQSEC ;TURN CLOCK OFF...
001255 171360 BIT1, JDDQSEC ;AND ON
001256 171360 CLR JDDREG ;SEL RX PRIMARY ADDRESS
001257 171346 MOV #RXBUFF, JDDQSEC ;SET ADDRESS
001258 171346 INCB JDDREG ;SEL RX PRIMARY CHAR COUNT
001259 171334 MOV #35, JDDQSEC ;SET CHAR COUNT
001260 171326 INCB JDDREG ;SEL TX PRIMARY ADDRESS
001261 012136 MOV #SYNC1, JDDQSEC ;LOAD THE SYNC CHAR
001262 171314 INCB JDDREG ;SEL TX PRI CHAR COUNT
001263 171302 MOV #38, JDDQSEC ;SET CHAR COUNT
001264 171272 INC JDDRCR ;SET RX GO
001265 171272 INC JDDTCSR ;SET TX GO
001266 171260 CLR R5 ;START TIMING
15: 15: TSTB JDDRCR ;IS DONE UP?
000001 15: BMI 25 ;BRANCH IF YES
000001 15: ADD #1, R5 ;WAIT
000001 15: BNE 15 ;BR IF MORE TO GO
001270 012142 HLT 1 ;ERROR--NO RX DONE
25: 25: MOV #TXBUFF, R0 ;LOAD BUFFER POINTER
001271 013022 MOV #RXBUFF, R1 ;LOAD RX BUFFER POINTER
000001 25: MOV #36, R2 ;SET UP TO COUNT CHARACTERS
35: 35: MOV (R0)+, R5 ;GET ANOTHER CHAR
001272 012134 MOV (R1)+, R4 ;GET A REC CHAR
001273 004304 BIC MASK, R5 ;MASK OUT UNWANTED BITS
001274 001000 CMP R5, R4 ;DO THE CHARACTERS MATCH?
001275 001000 BEQ 45 ;BR IF OK
001276 001000 HLT 2 ;ERROR--DATA DOESN'T MATCH
45: 45: DEC R2 ;ALL DONE?
001277 001000 BNE 35 ;NO--GO BACK FOR MORE
001278 104400 SCOPE ;SCOPE THIS TEST

```

```

;TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
;THIS TEST WILL XMIT AND RECVD CHARACTERS
;AT 12 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED!

```

```

: TEST 15
*****
55: ST15: MOV #15, TSTNO
MOV #TST16, NEXT
MEMCLR ;CLEAR ALL THE D311
MOV #RXBUFF, R0 ;LOAD THE BUFFER POINTER
CLR R1 ;SET UP TO CLEAR THE BUFFER
55: CLR (R0)+ ;CLEAR IT
INCB R1 ;DONE?
BPL 55 ;BRANCH IF NO

```

```

010152 012737 000015 001226
010160 012737 010464 001216
010166 104413
010170 012700 013022
010174 000001
010176 000020
010180 105201
010182 100278

```

```

010204 112777 000011 171156 MOVB #11,JDQREG ;SELECT THE SYNC REG
010206 013737 012616 001246 MOV SYNC,TEMP2 ;LOAD SYNC
010208 013737 170000 012134 MOV #170000,MASK ;LOAD THE MASK
010210 043737 012134 001246 BIC MASK,TEMP2 ;SET UP THE MASK FOR THE
010212 000241 001246 CLC ;CORREC; SYNC CHARACTER
010214 006037 001246 ROR TEMP2 ;SHIFT IT
010216 013737 001246 012136 MOV TEMP2,SYNC1 ;LOAD THE CHARACTER
010218 013737 001246 012140 MOV TEMP2,SYNC2 ;DITTO
010220 013737 001246 171106 MOV TEMP2,JDQSEC ;LOAD THE SYNC REGISTER
010222 105277 171100 INCB JDQREG ;SEL THE MISC REGISTER
010224 012700 000010 171074 MOV #BIT3,JDQSEC ;SET TEST LOOP
010226 012700 000004 MOV #4,R0 ;
010302 000300 SWAB R0 ;FLIP THE BYTES
010304 050077 171062 BIS R0,JDQSEC ;SET CHARACTER LENGTH
010306 052777 000002 171054 BIS #BIT1,JDQSEC ;TURN CLOCK OFF...
010308 042777 000002 171046 BIC #BIT1,JDQSEC ;AND ON
010310 105077 171040 CLRB JDQREG ;SEL RX PRIMARY ADDRESS
010312 012777 013022 171034 MOV #RXBUFF,JDQSEC ;SET ADDRESS
010314 105277 171026 INCB JDQREG ;SEL RX PRIMARY CHAR COUNT
010316 012777 177734 171022 MOV #-36,JDQSEC ;SET CHAR COUNT
010318 105277 171014 INCB JDQREG ;SEL TX PRIMARY ADDRESS
010320 012777 012136 171010 MOV #SYNC1,JDQSEC ;LOAD THE SYNC CHAR
010322 105277 171002 INCB JDQREG ;SEL TX PRI CHAR COUNT
010324 012777 177732 170776 MOV #-38,JDQSEC ;SET CHAR COUNT
010326 005277 170760 INC JDQRCR ;SET RX GO
010328 005277 170760 INC JDQTCR ;SET TX GO
010330 005005 CLR R5 ;START TIMING
010332 010404 170746 15: TSTB JDQRCR ;IS DONE UP?
010334 100404 BMI 25 ;BRANCH IF YES
010336 062705 000001 ADD #1,R5 ;WAIT
010338 001372 BNE 15 ;BR IF MORE TO GO
010340 040001 HLT 1 ;ERROR--NO RX DONE
010342 012700 012142 25: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
010344 012701 013022 MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
010346 012702 000044 MOV #36,R2 ;SET UP TO COUNT CHARACTERS
010348 012005 35: MOV (R0)+,R5 ;GET ANOTHER CHAR
010350 012104 MOV (R1)+,R4 ;GET A REC CHAR
010352 043705 012134 BIC MASK,R5 ;MASK OUT UNWANTED BITS
010354 020504 CMP R5,R4 ;DO THE CHARACTERS MATCH?
010356 001401 BEQ 45 ;BR IF OK
010358 104002 HLT 2 ;ERROR--DATA DOESN'T MATCH
010360 005302 43: DEC R2 ;ALL DONE?
010362 001367 BNE 35 ;NO--GO BACK FOR MORE
010364 104400 SCOPE ;SCOPE THIS TEST

```

;TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 13 BITS PER CHAR.
;DATA CHECKING WILL BE PERFORMED!

```

; TEST 16
;*****
TST16: MOV #16,TSTNO
MOV #TST17,NEXT

```

```

010464 012737 000016 001226
010472 012737 010776 001216

```


22949	010500	104413		MEMCLR					:CLEAR ALL THE DQ11
22950	010502	012700	013022	MOV	#RXBUFF,RO				:LOAD THE BUFFER POINTER
22951	010506	005001		CLR	R1				:SET UP TO CLEAR THE BUFFER
22952	010510	005020		5\$: CLR	(R0)+				:CLEAR IT
22953	010512	105201		INCB	R1				:DONE?
22954	010514	100375		BPL	5\$:BRANCH IF NO
22955	010516	112777	000011	MOV	#11,ADQREG	170644			:SELECT THE SYNC REG
22956	010524	013737	012616	MOV	SYNC,TEMP2	001246			:LOAD SYNC
22957	010532	012737	160000	MOV	#160000,MASK	012134			:LOAD THE MASK
22958	010540	043737	012134	BIC	MASK,TEMP2	001246			:SET UP THE MASK FOR THE
22959	010546	000241		CLC					:CORRECT SYNC CHARACTER
22960	010550	006037	001246	ROR	TEMP2				:SHIFT IT
22961	010554	013737	001246	MOV	TEMP2,SYNC1	012136			:LOAD THE CHARACTER
22962	010562	013737	001246	MOV	TEMP2,SYNC2	012140			:DITTO
22963	010570	013777	001246	MOV	TEMP2,ADQSEC	170574			:LOAD THE SYNC REGISTER
22964	010576	105277	170566	INCB	ADQREG				:SEL THE MISC REGISTER
22965	010602	012777	000010	MOV	#BIT3,ADQSEC	170562			:SET TEST LOOP
22966	010610	012700	000003	MOV	#3,RO				
22967	010614	000300		SWAB	RO				:FLIP THE BYTES
22968	010616	050077	170550	BIS	RO,ADQSEC				:SET CHARACTER LENGTH
22969	010622	052777	000002	BIS	#BIT1,ADQSEC	170542			:TURN CLOCK OFF...
22970	010630	042777	000002	BIC	#BIT1,ADQSEC	170534			:AND ON
22971	010636	105077	170526	CLRB	ADQREG				:SEL RX PRIMARY ADDRESS
22972	010642	012777	013022	MOV	#RXBUFF,ADQSEC	170522			:SET ADDRESS
22973	010650	105277	170514	INCB	ADQREG				:SEL RX PRIMARY CHAR COUNT
22974	010654	012777	177734	MOV	#-36,ADQSEC	170510			:SET CHAR COUNT
22975	010662	105277	170502	INCB	ADQREG				:SEL TX PRIMARY ADDRESS
22976	010666	012777	012136	MOV	#SYNC1,ADQSEC	170476			:LOAD THE SYNC CHAR
22977	010674	105277	170470	INCB	ADQREG				:SEL TX PRI CHAR COUNT
22978	010700	012777	177732	MOV	#-38,ADQSEC	170464			:SET CHAR COUNT
22979	010706	005277	170446	INC	ADQRCSR				:SET RX GO
22980	010712	005277	170446	INC	ADQTCSR				:SET TX GO
22981	010716	005005		CLR	R5				:START TIMING
22982	010720	105777	170434	1\$: TSTB	ADQRCSR				:IS DONE UP?
22983	010724	100404		BMI	2\$:BRANCH IF YES
22984	010726	062705	000001	ADD	#1,R5				:WAIT
22985	010732	001372		BNE	1\$:BR IF MORE TO GO
22986	010734	104001		HLT	1				:ERROR--NO RX DONE
22987	010736	012700	012142	2\$: MOV	#TXBUFF,RO				:LOAD BUFFER POINTER
22988	010742	012701	013022	MOV	#RXBUFF,R1				:LOAD RX BUFFER POINTER
22989	010746	012702	000044	MOV	#36,R2				:SET UP TO COUNT CHARACTERS
22990	010752			3\$:					
22991	010752	012005		MOV	(R0)+,R5				:GET ANOTHER CHAR
22992	010754	012104		MOV	(R1)+,R4				:GET A REC CHAR
22993	010756	043705	012134	BIC	MASK,R5				:MASK OUT UNWANTED BITS
22994	010762	020504		CMP	R5,R4				:DO THE CHARACTERS MATCH?
22995	010764	001401		BEQ	4\$:BR IF OK
22996	010766	104002		HLT	2				:ERROR--DATA DOESN'T MATCH
22997	010770	005302		4\$: DEC	R2				:ALL DONE?
22998	010772	001367		BNE	3\$:NO--GO BACK FOR MORE
22999	010774	104400		SCOPE					:SCOPE THIS TEST
23000									
23001									
23002									
23003									
23004									

:TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
 :THIS TEST WILL XMIT AND RECV CHARACTERS
 :AT 14 BITS/PER/CHAR.

:DATA CHECKING WILL BE PERFORMED!

```

000005                                     : TEST 17
000006                                     :*****
000007 010776 012737 000017 00:226 TST17: MOV #17,TSTNO
000008 011004 012737 011310 001216      MOV #TST20,NEXT
000009 011012 104413      MEMCLR
000010 011014 012700 013022      MOV #RXBUFF,RC ;CLEAR ALL THE DQ11
000011 011020 005001      CLR R1 ;LOAD THE BUFFER POINTER
000012 011022 005020      CLR (R0)+ ;SET UP TO CLEAR THE BUFFER
000013 011024 105201      INCB R1 ;CLEAR IT
000014 011026 100375      BPL SS ;DONE?
000015 011030 112777 000011 170332      MOVB #11,ADQREG ;BRANCH IF NO
000016 011036 012737 012616 001246      MOV SYNC,TEMP2 ;SELECT THE SYNC REG
000017 011044 012737 140000 012134      MOV #140000,MASK ;LOAD SYNC
000018 011052 043737 012134 001246      BIC MASK,TEMP2 ;LOAD THE MASK
000019 011060 000241      CLC ;SET UP THE MASK FOR THE
000020 011062 006037 001246      ROR TEMP2 ;CORRECT SYNC CHARACTER
000021 011066 013737 001246 012136      MOV TEMP2,SYNC1 ;SHIFT IT
000022 011074 013737 001246 012140      MOV TEMP2,SYNC2 ;LOAD THE CHARACTER
000023 011102 013777 001246 170262      MOV TEMP2,ADQSEC ;DITTO
000024 011110 105277 170254      INCB ADQREG ;LOAD THE SYNC REGISTER
000025 011114 012777 000010 170250      MOV #BIT3,ADQSEC ;SEL THE MISC REGISTER
000026 011122 012700 000002      MOV #2,R0 ;SET TEST LOOP
000027 011126 000300      SWAB R0 ;FLIP THE BYTES
000028 011130 050077 170236      BIS R0,ADQSEC ;SET CHARACTER LENGTH
000029 011134 052777 000002 170230      BIS #BIT1,ADQSEC ;TURN CLOCK OFF...
000030 011142 042777 000002 170222      BIC #BIT1,ADQSEC ;AND ON
000031 011150 105077 170214      CLRB ADQREG ;SEL RX PRIMARY ADDRESS
000032 011154 012777 013022 170210      MOV #RXBUFF,ADQSEC ;SET ADDRESS
000033 011162 105277 170202      INCB ADQREG ;SEL RX PRIMARY CHAR COUNT
000034 011166 012777 177734 170176      MOV #-36,ADQSEC ;SET CHAR COUNT
000035 011174 105277 170170      INCB ADQREG ;SEL TX PRIMARY ADDRESS
000036 011200 012777 012136 170164      MOV #SYNC1,ADQSEC ;LOAD THE SYNC CHAR
000037 011206 105277 170156      INCB ADQREG ;SEL TX PRI CHAR COUNT
000038 011212 012777 177732 170152      MOV #-38,ADQSEC ;SET CHAR COUNT
000039 011220 005277 170134      INC ADQRCR ;SET RX GO
000040 011224 005277 170134      INC ADQTCR ;SET TX GO
000041 011230 005005      CLR R5 ;START TIMING
000042 011232 105777 170122 1S: TSTB ADQRCR ;IS DONE UP?
000043 011236 100404      BMI 2S ;BRANCH IF YES
000044 011240 052705 000001      ADD #1,R5 ;WAIT
000045 011244 001372      BNE 1S ;BR IF MORE TO GO
000046 011246 104001      HLT ;ERROR--NO RX DONE
000047 011250 012700 012142 2S: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
000048 011254 012701 013022      MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
000049 011260 012702 000044      MOV #36.,R2 ;SET UP TO COUNT CHARACTERS
000050 011264 3S:
000051 011264 012005      MOV (R0)+,R5 ;GET ANOTHER CHAR
000052 011266 012104      MOV (R1)+,R4 ;GET A REC CHAR
000053 011270 043705 012134      BIC MASK,R5 ;MASK OUT UNWANTED BITS
000054 011274 020504      CMP R5,R4 ;DO THE CHARACTERS MATCH?
000055 011276 001401      BEQ 4S ;BR IF OK
000056 011300 104002      HLT ;ERROR--DATA DOESN'T MATCH
000057 011302 005302 4S: DEC R2 ;ALL DONE?
000058 011304 001367      BNE 3S ;NO--GO BACK FOR MORE

```

```

2361 011306 104400
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371 011310 012737 000020 001226
2372 011316 012737 011622 001216
2373 011324 104413
2374 011326 012700 013022
2375 011332 005001
2376 011334 005020
2377 011336 105201
2378 011340 100375
2379 011342 112777 000011 170020
2380 011350 013737 012616 001246
2381 011356 012737 100000 012134
2382 011364 043737 012134 001246
2383 011372 000241
2384 011374 006037 001246
2385 011400 013737 001246 012136
2386 011406 013737 001246 012140
2387 011414 013777 001246 167750
2388 011422 105277 167742
2389 011426 012777 000010 167736
2390 011434 012700 000001
2391 011440 000300
2392 011442 050077 167724
2393 011446 052777 000002 167716
2394 011454 042777 000002 167710
2395 011462 105077 167702
2396 011466 012777 013022 167676
2397 011474 105277 167670
2398 011500 012777 177734 167664
2399 011506 105277 167656
2400 011512 012777 012136 167652
2401 011520 105277 167644
2402 011524 012777 177732 167640
2403 011532 005277 167622
2404 011536 005277 167622
2405 011542 005005
2406 011544 105777 167610
2407 011550 100404
2408 011552 062705 000001
2409 011556 001372
2410 011560 104001
2411 011562 012700 012142
2412 011566 012701 013022
2413 011572 012702 000044
2414 011576
2415 011576 012005
2416 011600 012104

```

SCOPE ;SCOPE THIS TEST

```

:TEST OF TRANSMITTER AND RECEIVER CHARACTER LENGTHS
:THIS TEST WILL XMIT AND RECV CHARACTERS
:AT 15 BITS/PER CHAR.
:DATA CHECKING WILL BE PERFORMED!

```

```

: TEST 20
:*****
TST20: MOV #20,TSTNO
MOV #TST21,NEXT
MEMCLR ;CLEAR ALL THE DQ11
MOV #RXBUFF,R0 ;LOAD THE BUFFER POINTER
CLR R1 ;SET UP TO CLEAR THE BUFFER
SS: CLR (R0)+ ;CLEAR IT
INCB R1 ;DONE?
BPL SS ;BRANCH IF NO
MOVB #11,ADQREG ;SELECT THE SYNC REG
MOV SYN,TEMP2 ;LOAD SYNC
MOV #10,JO00,MASK ;LOAD THE MASK
BIC MASK,TEMP2 ;SET UP THE MASK FOR THE
CLC ;CORRECT SYNC CHARACTER
ROR TEMP2 ;SHIFT IT
MOV TEMP2,SYNC1 ;LOAD THE CHARACTER
MOV TEMP2,SYNC2 ;DITTO
MOV TEMP2,ADQSEC ;LOAD THE SYNC REGISTER
INCB ADQREG ;SEL THE MISC REGISTER
MOV #BIT3,ADQSEC ;SET TEST LOOP
MOV #1,R0
SHAB R0 ;FLIP THE BYTES
BIS R0,ADQSEC ;SET CHARACTER LENGTH
BIS #BIT1,ADQSEC ;TURN CLOCK OFF...
BIC #BIT1,ADQSEC ;AND ON
CLRB ADQREG ;SEL RX PRIMARY ADDRESS
MOV #RXBUFF,ADQSEC ;SET ADDRESS
INCB ADQREG ;SEL RX PRIMARY CHAR COUNT
MOV #-36,ADQSEC ;SET CHAR COUNT
INCB ADQREG ;SEL TX PRIMARY ADDRESS
MOV #SYNC1,ADQSEC ;LOAD THE SYNC CHAR
INCB ADQREG ;SEL TX PRI CHAR COUNT
MOV #-38,ADQSEC ;SET CHAR COUNT
INC ADQRC5R ;SET RX GO
INC ADQTC5R ;SET TX GO
CLR R5 ;START TIMING
1$: TSTB ADQRC5R ;IS DONE UP?
BMI 2$ ;BRANCH IF YES
ADD #1,R5 ;WAIT
BNE 1$ ;BR IF MORE TO GO
HLT 1 ;ERROR--NO RX DONE
2$: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
MOV #36,R2 ;SET UP TO COUNT CHARACTERS
3$: MOV (R0)+,R5 ;GET ANOTHER CHAR
MOV (R1)+,R4 ;GET A REC CHAR

```

```

2417 011602 043705 012134      BIC    MASK,R5      ;MASK OUT UNWANTED BITS
2418 011606 020504              CMP    R5,R4       ;DO THE CHARACTERS MATCH?
2419 011610 001401              BEQ    4$          ;BR IF OK
2420 011612 104002              HLT   2            ;ERROR--DATA DOESN'T MATCH
2421 011614 005302              4$: DEC    R2       ;ALL DONE?
2422 011616 001367              BNE   3$          ;NO--GO BACK FOR MORE
2423 011620 104400              SCOPE                ;SCOPE THIS TEST

```

```

;TEST OF TRANSMITTER AND RECEIVER CHARATER LENGHTHS
;THIS TEST WILL XMIT AND RECV CHARACTERS
;AT 16 BITS/PER/CHAR.
;DATA CHECKING WILL BE PERFORMED!

```

```

; TEST 21
;*****
2424 011622 012737 000021 001226 4$: TST21: MOV    #21,TSTNO
2425 011630 012737 014052 001216      MOV    #.EOP,NEXT
2426 011636 104413              MEMCLR                ;CLEAR ALL THE DQ11
2427 011640 012700 013022              MOV    #RXBUFF,RO    ;LOAD THE BUFFER POINTER
2428 011644 005001              CLR    R1             ;SET UP TO CLEAR THE BUFFER
2429 011646 005020              5$: CLR    (R0)+      ;CLEAR IT
2430 011650 105201              INCB   R1             ;DONE?
2431 011652 100375              BPL   5$             ;BRANCH IF NO
2432 011654 112777 000011 167506      MOVB   #11,ADQREG    ;SELECT THE SYNC REG
2433 011662 013737 012616 001246      MOV    SYNC,TEMP2   ;LOAD SYNC
2434 011670 012737 000000 012134      MOV    #000000,MASK ;LOAD THE MASK
2435 011676 043737 012134 001246      BIC    MASK,TEMP2   ;SET UP THE MASK FOR THE
2436 011704 000241              CLC                    ;CORRECT SYNC CHARACTER
2437 011706 006037 001246              ROR    TEMP2        ;SHIFT IT
2438 011712 013737 001246 012136      MOV    TEMP2,SYNC1  ;LOAD THE CHARACTER
2439 011720 013737 001246 012140      MOV    TEMP2,SYNC2  ;DITTO
2440 011726 013777 001246 167436      MOV    TEMP2,ADQSEC ;LOAD THE SYNC REGISTER
2441 011734 105277 167430              INCB   ADQREG        ;SEL THE MISC REGISTER
2442 011740 012777 000010 167424      MOV    #BIT3,ADQSEC ;SET TEST LOOP
2443 011746 012700 000000              MOV    #C,RO
2444 011752 000300              SWAB   RO            ;FLIP THE BYTES
2445 011754 050077 167412              BIS    RO,ADQSEC    ;SET CHARACTER LENGTH
2446 011760 052777 000002 167404      BIS    #BIT1,ADQSEC ;TURN CLOCK OFF...
2447 011766 042777 000002 167376      BIC    #BIT1,ADQSEC ;AND ON
2448 011774 105077 167370              CLRB   ADQREG        ;SEL RX PRIMARY ADRESS
2449 012000 012777 013022 167364      MOV    #RXBUFF,ADQSEC ;SET ADDRESS
2450 012006 105277 167356              INCB   ADQREG        ;SEL RX PRIMARY CHAR COUNT
2451 012012 012777 177734 167352      MOV    #-36,ADQSEC  ;SET CHAR COUNT
2452 012020 105277 167344              INCB   ADQREG        ;SEL TX PRIMARY ADDRESS
2453 012024 105277 012136 167340      MOV    #SYNC1,ADQSEC ;LOAD THE SYNC CHAR
2454 012032 105277 167332              INCB   ADQREG        ;SEL TX PRI CHAR COUNT
2455 012036 012777 177732 167326      MOV    #-38,ADQSEC  ;SET CHAR COUNT
2456 012044 005277 167310              INC    ADQRCR        ;SET RX GO
2457 012050 005277 167310              INC    ADQTCSR       ;SET TX GO
2458 012054 005005              CLR    R5            ;START TIMING
2459 012056 105777 167276              1$: TSTB   ADQRCR    ;IS DONE UP?
2460 012062 100404              BMI   2$            ;BRANCH IF YES
2461 012064 062705 000001              ADD    #1,R5        ;WAIT
2462 012070 001372              BNE   1$           ;BR IF MORE TO GO
2463 012072 104001              HLT   1            ;ERROR--NO RX DCNE

```

```

2473 012074 012700 012142 2$: MOV #TXBUFF,R0 ;LOAD BUFFER POINTER
2474 012100 012701 013022 MOV #RXBUFF,R1 ;LOAD RX BUFFER POINTER
2475 012104 012702 000044 MOV #35.,R2 ;SET UP TO COUNT CHARACTERS
2476 012110 3$:
2477 012110 0120C5 MOV (R0)+,R5 ;GET ANOTHER CHAR
2478 012112 012104 MOV (R1)+,R4 ;GET A REC CHAR
2479 012114 043705 012134 BIC MASK,R5 ;MASK OUT UNWANTED BITS
2480 012120 020504 CMP R5,R4 ;DO THE CHARACTERS MATCH?
2481 012122 001401 BEQ 4$ ;BR IF OK
2482 012124 104002 HLT 2 ;ERROR--DATA DOESN'T MATCH
2483 012126 005302 4$: DEC R2 ;ALL DONE?
2484 012130 001367 SNE 3$ ;NO--GO BACK FOR MORE
2485 012132 104400 SCOPE ;SCOPE THIS TEST

```

```

2486 012134 000000
2487 012136 026 026 MASK: 0
2488 012140 026 026 SYNC1: .BYTE 26,26
2489 012142 TXBUFF: .BYTE 26,26
2490 012142 177777 †B<11111111111111111111>
2491 012144 000000 †B<00000000000000000000>
2492 012146 125252 †B<10101010101010101010>
2493 012150 052525 †B<01010101010101010101>
2494 012152 000001 †B<00000000000000000001>
2495 012154 000002 †B<00000000000000000010>
2496 012156 000004 †B<00000000000000000100>
2497 012160 000010 †B<00000000000000001000>
2498 012162 000020 †B<0000000000000010000>
2499 012164 000040 †B<000000000000100000>
2500 012166 000100 †B<000000000001000000>
2501 012168 000200 †B<000000000010000000>
2502 012170 000400 †B<000000001000000000>
2503 012172 001000 †B<000000100000000000>
2504 012174 002000 †B<000001000000000000>
2505 012176 004000 †B<000010000000000000>
2506 012200 010000 †B<000100000000000000>
2507 012202 020000 †B<001000000000000000>
2508 012204 040000 †B<001000000000000000>
2509 012206 040000 †B<010000000000000000>
2510 012210 100000 †B<100000000000000000>
2511 012212 077777 †B<01111111111111111111>
2512 012214 137777 †B<10111111111111111111>
2513 012216 157777 †B<11011111111111111111>
2514 012220 167777 †B<11101111111111111111>
2515 012222 173777 †B<11110111111111111111>
2516 012224 175777 †B<11111011111111111111>
2517 012226 176777 †B<11111101111111111111>
2518 012230 177377 †B<11111110111111111111>
2519 012232 177577 †B<11111111011111111111>
2520 012234 177677 †B<11111111101111111111>
2521 012236 177737 †B<11111111110111111111>
2522 012240 177757 †B<11111111111011111111>
2523 012242 177767 †B<11111111111101111111>
2524 012244 177773 †B<11111111111110111111>
2525 012246 177775 †B<11111111111111011111>
2526 012250 177776 †B<11111111111111101111>

```

```

2529 012252 000100          .BLKW 100
2530 012452          .MEMCLR:
2531 012452 005077 166702    CLR 3DJRCR
2532 012456 005077 166702    CLR 3DQTCR
2533 012462 005077 166700    CLR 3DJERR
2534 012466 012705 000020    MOV #16.,R5
2535 012472 152777 000020 166670 1$: BISB #BIT4,3DQREG
2536 012500 142777 000140 166662 BICB #140,3DQREG
2537 012506 005077 166660    CLR 3DQSEC
2538 012512 105277 166652    INCB 3DQREG
2539 012516 005305          DEC R5
2540 012520 001364          BNE 1$
2541 012522 105077 166642    CLRB 3DQREG
2542 012526 105077 166630    CLRB 3DJRCRSH
2543 012532 012705 000020    MOV #16.,R5
2544 012536 112777 000010 166624 2$: MOVB #10,3DQREG
2545 012544 005077 166622    CLR 3DQSEC
2546 012550 112777 000014 166612 MOVB #14,3DQREG
2547 012556 005077 166610    CLR 3DQSEC
2548 012562 105277 166574    INCB 3DJRCRSH
2549 012566 005305          DEC R5
2550 012570 001362          BNE 2$
2551 012572 105077 166564    CLRB 3DJRCRSH
2552 012576          .MSTCLR:
2553 012576 112777 000012 166564 MOVB #MISC.,3DQREG
2554 012604 012777 000040 166560 MOV #BITS,3DQSEC
2555 012612 000002          RTI
2556 012614 026 026      .SYNC: .BYTE 26,26
2557 012616 026 026      SYNC: .BYTE 26,26
2558 012620 000000      TXBFA: 0
2559 013022 013022          .=. +200
2560 013022          RXBUFF:
2561 013022 000200          .BLKW 200
2562 013422 026 026      XSYNC: .BYTE 26,26
2563 013424 026 026      XSYNC2: .BYTE 26,26
2564 013426 000000      XTXBUF: 0
2565 013630          .=. +200
2566 013630 000000      XRXBUF: 0
2567 014032          .=. +200
2568 014032 000000      ERR: 0
2569 014034 000000      FOLY: 0
2570 014036 000000      XPOLY: 0
2571 014040 000000      CHAR: 0
2572 014042 000000      COUNT: 0
2573 014044 000000      ADDR: 0
2574 014046 000000      GDCHAR: 0
2575 014050 000000      DETCAR: 0
2576
2577          ;END OF PASS
2578          ;TYPE NAME OF TEST
2579          ;UPDATE PASS COUNT
2580          ;CHECK FOR EXIT TO ACT-11
2581          ;RESTART TEST
2582
2583 014052 005037 001234      .EOP: CLR LSTERR          ;CLEAR LAST ERROR PC
2584 014056 005037 001312      CLR ERRFLG          ;CLEAR ERROR FLAG

```

2585	014062	005237	001230		INC	PASCNT		;UPDATE PASS COUNT
2586	014066	104402			TYPE			
2587	014070	016302			MEPASS			
2589	014072	104402			TYPE			
2589	014074	016453			MCSRX			
2590	014076	104411			CNVRT			
2591	014100	014210			XCSR			
2592	014102	104402			TYPE			
2593	014104	016471			MVECX			
2594	014106	104411			CNVRT			
2595	014110	014216			XVEC			
2596	014112	104402			TYPE			
2597	014114	016477			MPASSX			
2598	014116	104411			CNVRT			
2599	014120	014224			XPASS			
2600	014122	104402			TYPE			
2601	014124	016510			MERRX			
2602	014126	104411			CNVRT			
2603	014130	014232			XERR			
2604	014132	013777	001230	165042	MOV	PASCNT, @LIGHTS		;DISPLAY PASS COUNT
2605	014140	005337	001276		DEC	SAVNUM		
2606	014144	001013			BNE	RESTR		
2607	014146	013737	001504	001276	MOV	DQNUM, SAVNUM		
2608	014154	013701	000042		MOV	@#42, R1		;CHECK FOR ACT-11 OR DDP
2609	014160	001405			BEQ	RESTR		;IF NOT, CONTINUE TESTING
2610	014162	000005			RESET			
2611	014164				LOGICAL:			
2612	014164	004711			JSR	PC, (R1)		
2613	014166	000240			NOP			
2614	014170	000240			NOP			
2615	014172	000240			NOP			
2616	014174	104414			RESTR:	CKSWR		
2617	014176	012737	002254	001214	MOV	#TST1, RETURN		
2618	014204	000137	002254		JMP	TST1		
2619	014210	000001			XCSR:	1		
2620	014212	006	002		.BYTE	6,2		
2621	014214	001360			DQRCSR			
2622	014216	000001			XVEC:	1		
2623	014220	003	002		.BYTE	3,2		
2624	014222	001350			DQRVEC			
2625	014224	000001			XPASS:	1		
2626	014226	006	002		.BYTE	6,2		
2627	014230	001230			PASCNT			
2628	014232	000001			XERR:	1		
2629	014234	006	002		.BYTE	6,2		
2630	014236	001232			ERRCNT			
2631								
2632								;SCOPE LOOP AND ITERATION HANDLER
2633								
2634	014240	104414			.SCOPE:	CKSWR		
2635	014242	032777	040000	164730	BIT	#BIT14, @SWR		
2636	014250	001407			TTST:	BEQ	1\$	
2637	014252	000432			BR	3\$		
2638	014254	105777	164724		TSTB	@TKCSR		
2639	014260	100027			BPL	3\$		
2640	014262	017700	164720		MOV	@TKDBR, R0		

2641	014266	000412				BR	2\$
2642	014270	032777	004000	164702	1\$:	BIT	#SW11,@SWR
2643	014276	001006				BNE	2\$
2644	014300	005237	001224			INC	LPCNT
2645	014204	023737	001224	001222		CMF	LPCNT,ICOUNT
2646	014312	001012				BNE	3\$
2647	014314	105037	001312		2\$:	CLRB	ERRFLG
2648	014320	005037	001224			CLR	LPCNT
2649	014324	012737	000017	001222		MOV	#15.,ICOUNT
2650	014332	013737	001216	001214		MOV	NEXT,RETURN
2651	014340	013716	001214		3\$:	MOV	RETURN,(SP)
2652	014344	000002				RTI	
2653	014346	001407			BRW:	1407	
2654	014350	000432			BRX:	432	
2655							
2656							:CHECK FOR FREEZE ON CURRENT DATA
2657							
2658	014352	104414			.SCOPI:	CKSWR	
2659	014354	032777	001000	164616		BIT	#SW09,@SWR
2660	014362	001402				BEQ	1\$
2661	014364	013716	001220			MOV	LOCK,(SP)
2662	014370	000002			1\$:	RTI	
2663							
2664							:TELETYPE OUTPUT ROUTINE
2665							
2666	014372	010546			.TYPE:	MOV	R5,-(SP)
2667	014374	017605	000002			MOV	@2(SP),R5
2668	014400	062766	000002	000002		ADD	#2,2(SP)
2669	014406	005737	016062		1\$:	TST	@#RDSW
2670	014412	001004				BNE	300\$
2671	014414	032777	010000	164556		BIT	#SW12,@SWR
2672	014422	001024				BNE	3\$
2673	014424	105715			300\$:	TSTB	(R5)
2674	014426	100014				BPL	2\$
2675	014430	105777	164554			TSTB	@TPCSR
2676	014434	100375				BPL	.-4
2677	014436	012777	000015	164546		MOV	#15,@TPDBR
2678	014444	105777	164540			TSTB	@TPCSR
2679	014450	100375				BPL	.-4
2680	014452	012777	000012	164532		MOV	#12,@TPDBR
2681	014460	105777	164524		2\$:	TSTB	@TPCSR
2682	014464	100375				BPL	2\$
2683	014466	112577	164520			MOVB	(R5)+,@TPDBR
2684	014472	001345				BNE	1\$
2685	014474	012605			3\$:	MOV	(SP)+,R5
2686	014476	000002				RTI	
2687							
2688							:ASCII STRING INPUT ROUTINE
2689							
2690	014500	010346			.INSTR:	MOV	R3,-(SP)
2691	014502	010446				MOV	R4,-(SP)
2692	014504	017637	000004	014522		MOV	@4(SP),MSG
2693	014512	062766	000002	000004		ADD	#2,4(SP)
2694	014520	104402			.INST1:	TYPE	
2695	014522	000000			.MSG:	0	
2696	014524	012704	016654			MOV	#INBUF,R4

2697	014530	012703	000007			MOV	#7,R3	
2698	014534	105777	164444			1\$: TSTB	@TKCSR	
2699	014540	100375				BPL	1\$	
2700	014542	117714	164440			MOVB	@TKDBR,(R4)	
2701	014546	142714	000200			BICB	#200,(R4)	
2702	014552	121427	000025			CMPB	(R4),#25	;IS IT <IG>
2703	014556	001003				BNE	200\$	
2704	014560	104402	016242			TYPE,MCRLF		
2705	014564	000755				BR	.INST1	
2706	014566	122427	000015		200\$:	CMPB	(R4)+,#15	
2707	014572	001423				BEQ	INSTR2	
2708	014574	117777	164406	164410		MOVB	@TKDBR,@TPDBR	
2709	014602	105777	164402		2\$:	TSTB	@TPCSR	
2710	014606	100375				BPL	2\$	
2711	014610	005303				DEC	R3	
2712	014612	001350				BNE	1\$	
2713	014614	000402				BR	.INSTG	
2714	014616	010346			.INSTE:	MOV	R3,-(SP)	
2715	014620	010446				MOV	R4,-(SP)	
2716	014622	104402			.INSTG:	TYPE		
2717	014624	016236				MQM		
2718	014626	005737	016062			TST	@RDSW	
2719	014632	001402				BEQ	400\$	
2720	014634	104402	016242			TYPE,MCRLF		
2721	014640	000727			400\$:	BR	.INST1	
2722	014642	012604			INSTR2:	MOV	(SP)+,R4	
2723	014644	012603				MOV	(SP)+,R3	
2724	014646	000002				RTI		
2725								
2726								;CONVERT ASCII STRING TO OCTAL
2727								
2728	014650	010546			.PARAM:	MOV	R5,-(SP)	
2729	014652	010446				MOV	R4,-(SP)	
2730	014654	016605	000004			MOV	4(SP),R5	
2731	014660	012537	015054			MOV	(R5)+,LOLIM	
2732	014664	012537	015056			MOV	(R5)+,HILIM	
2733	014670	012537	015060			MOV	(R5)+,DEVADR	
2734	014674	112537	015062			MOVB	(R5)+,LOBITS	
2735	014700	112537	015063			MOVB	(R5)+,ADRCNT	
2736	014704	010566	000004			MOV	R5,4(SP)	
2737	014710	005005			PARAM1:	CLR	R5	
2738	014712	012704	016654			MOV	#INBUF,R4	
2739	014716	122714	000015			CMPB	#15,(R4)	
2740	014722	001420				BEQ	PARERR	
2741	014724	121427	000060		1\$:	CMPB	(R4),#60	
2742	014730	002415				BLT	PARERR	
2743	014732	121427	000067			CMPB	(R4),#67	
2744	014736	003012				BGT	PARERR	
2745	014740	142714	000060			BICB	#60,(R4)	
2746	014744	152405				BISB	(R4)+,R5	
2747	014746	122714	000015			CMPB	#15,(R4)	
2748	014752	001414				BEQ	LIMITS	
2749	014754	006305				ASL	R5	
2750	014756	006305				ASL	R5	
2751	014760	006305				ASL	R5	
2752	014762	000760				BR	1\$	

```

000015 PRERR: CNPB      015, R4)      :IS FIRST CHARACTER A 'CA'
016062      BNE      1208          :IS CKSWR ROUTINE BEING USED
1208:      INSTER      PARAM1

:TEST TO SEE IF NUMBER IS WITHIN LIMITS

015056 LIMITS: CNP      RS, HILIM
015054      BHI      PRERR
015062      CNP      RS, LOLIM
      BLO      PRERR
      BBTB     LOBITS, RS
      BNE      PRERR

:STORE NUMBER AT SPECIFIED ADDRESS

015062 IS:      MOV      DEVAR, R4
015063      MOV      R5, (R4)+
015064      ADD     #2, R5
015065      DECB   RORCNT
PARTI:     MOV      (SP)+, R4
      MOV      (SP)+, R5
LOLIM:
HILIM:
DEVAR:
LOBITS:
RORCNT: LOBITS+1

:SAVE PC OF TEST THAT FAILED AND R0-R5
016067 000004 001274 .SAV05: MOV      4(SP), SAVPC
      :SAVE R0-R5
SAV05:    MOV      R5, SAVR5
      MOV      R4, SAVR4
      MOV      R3, SAVR3
      MOV      R2, SAVR2
      MOV      R1, SAVR1
      MOV      R0, SAVR0
      RTI

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



```

015563 001402 BEQ WRKO.FM
015564 104402 TYPE
015565 000000 ERRMSG: 0
015566 005737 015546 WRKO.FM: TST DATAHD
015567 001402 BEQ TYPDAT
015568 104402 TYPE
015569 000000 DATAHD: 0
015570 005737 015560 TYPDAT: TST DATABP
015571 001402 BEQ RESREG
015572 104410 CONVRT
015573 000000 DATABP: 0
015574 104407 RESREG: RESOS
015575 005777 163310 HALTS: TST 2SWR
015576 100005 BPL EXITER
015577 010046 PUSHRO
015578 016600 000002 MOV 2(SP),RO
015579 000000 HALT
015580 012600 POPRO
015581 104414 EXITER: CKSWR
015582 005237 001232 INC ERRCNT
015583 032777 000400 163260 BIT #SW08,2SWR
015584 001007 BNE 15
015585 032777 002000 163250 BIT #SW10,2SWR
015586 001407 BEQ 25
015587 013737 001216 001214 MOV NEXT,RETURN
015588 012706 001200 15: MOV #STACK,SP
015589 000177 163244 25: JMP 2RETURN
015590 000002 RTI
015591 000001 ERTAB0: 1
015592 006 002 .BYTE 6.2
015593 001274 SAVPC
015594 000001 XTSTN: 1
015595 003 002 .BYTE 3.2
015596 001226 TSTNO
:ENTER HERE ON POWER FAILURE

015756 012737 016000 000024 .PFAIL: MOV #RESTART,24 ;SET UP FOR POWER UP TRAP
015757 000000 HALT ;HALT ON POWER DOWN NORMAL
015758 000777 BR .
:PROCESSOR WILL TRAP HERE WHEN POWER IS RESTORED

016000 012737 015766 000024 RESTAR: MOV #.PFAIL,24 ;SET UP FOR POWER FAILURE
016001 012706 001200 MOV #STACK,SP
016002 005037 016716 CLR TEMP
016003 005237 016716 INC TEMP
016004 001375 BNE .-4
016005 104402 TYPE
016006 016244 MPFAIL
016007 104411 CNVRT
016008 016054 PFTAB
016009 005037 001312 CLR ERRFLG

```

```

016040 005037 001234
016044 104412
016046 104413
016050 000177 163140
016054 000001
016056 003 002
016050 001226

016062 000000

016064 005737 000042
016070 001042
016072 022737 000176 001200
016100 001036
016102 105777 163076
016105 100033
016110 017737 163072 014522
016116 042737 177600 014522
016124 122737 000007 014522
016132 001021
016134 104402 016212
016140 005137 016062
016144 104402 016216
016150 104411 016204
016154 104403 016225
016160 104405
016162 000000
016164 177777
016166 000176
016170 000 001
016172 104402 016242
016176 005037 016062
016202 000002
016204 000001
016206 006 002
016210 000176
016212 057377 000107
016216 051777 051127 020075
016224 000
016225 040 047040 053505
016232 020075 000
016236
016236 020040 000077
016242 000377
016244 050377 051127 043040
016252 044501 042514 027104
016260 051040 051505 040524
016266 052122 040440 020124
016274 042524 052123 000040
016302 042777 042116 050040
016310 051501 020123 055104

```

```

CLR LSTERR
MSTCLR
MEMCLR
JMP QRETURN
PFTAB: 1
.BYTE 3,2
TSTNO

```

```

;CHECK SWITCH REGISTER ROUTINE. CHECKS FOR 16 TO ALLOW CHANGING
;OF LOC.176.
;LOCATIONS USED:
RDSW: .WORD 0

```

```

.CKSWR: TST 2*42
BNE OUT
CMP #SWREG, SWR ;SOFTWARE SWITCH REGISTER PRESENT
BNE OUT ;NO GET OUT
TSTB #TKCSR ;YES WAIT FOR
BPL OUT ;READY GET CHARACTER
MOV #TKDBR, .MSG ;AND STRIP OFF
BIC #177600, .MSG ;THE GARBAGE
CMPB #7, .MSG ;IS IT A 16?
BNE OUT

```

```

.CNTLU: COM 2*RDSW
TYPE, SMSWR
CNVRT, SWREGC
INSTR, SMNEW
PARAM
0
177777
SWREG

```

```

.BYTE 0,1
TYPE, MCRLF
OUT: CLR 2*RDSW
RTI

```

```

SWREGC: 1
.BYTE 6,2
SWREG
$CNTG: .ASCIZ <377>/16/
$MSWR: .ASCIZ <377>/SWR= /
$MNEW: .ASCIZ / NEW= /

```

```

.EVEN
MQM: .ASCIZ / ?/
MCRLF: .ASCIZ <377>
MPFAIL: .ASCIZ <377>/PWR FAILED. RESTART AT TEST

```

```

MEPASS: .ASCIZ <377>/END PASS DZDQH /

```


G05

027724 MACY11 27 (732) 24-SEP-76 10:11 PAGE 59
 027740B.P11 GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

3033	016316	050504	020110	000040		
3034	016324	051377	000		MR: .ASCIZ <377>/R/	
3035	016327	377	051120	043517	MERR2: .ASCIZ <377>/PROGRAM INDICATES NO DEVICES PRESENT./	
3036	016334	040522	020115	047111		
3037	016342	044504	040503	042524		
3038	016350	020123	047516	042040		
3039	016356	053105	041511	051505		
3040	016364	050040	042522	042523		
3041	016372	052116	000056			
3042	016376	044777	051516	043125	MERR3: .ASCIZ <377>/INSUFFICIENT DATA! /	
3043	016404	044506	044503	047105		
3044	016412	020124	040504	040524		
3045	016420	000041				
3046	016422	052377	051505	020124	MTSTPC: .ASCIZ <377>/TEST PC- /	
3047	016430	041520	000055			
3048	016434	046377	041517	020113	MLOCK: .ASCIZ <377>/LOCK ON SELECTED TEST /	
3049	016442	047117	051440	046105		
3050	016450	041505	042524	020104		
3051	016456	042524	052123	000		
3052	016463	103	051123	020072	MCSRX: .ASCIZ /CSR: /	
3053	016470	000				
3054	016471	126	041505	020072	MVECX: .ASCIZ /VEC: /	
3055	016476	000				
3056	016477	120	051501	042523	MPASSX: .ASCIZ /PASSES: /	
3057	015504	035123	000040			
3058	016510	051105	047522	051522	MERRX: .ASCIZ /ERRORS: /	
3059	016516	020072	000			
3060	016521	377	052377	051505	MTSTN: .ASCIZ <377><377> /TEST NO: /	
3061	016526	020124	047516	020072		
3062	016534	000				
3063	016535	377	042523	020124	MNEW: .ASCIZ <377>/SET SWITCH REG TO DQ11'S DESIRED ACTIVE. /	
3064	016542	053523	052111	044103		
3065	016550	051040	043505	052040		
3066	016556	020117	050504	030461		
3067	016564	051447	042040	051505		
3068	016572	051111	042105	040440		
3069	016600	052103	053111	027105		
3070	016606	000				
3071	016607	120	035103	000040	MERRPC: .ASCIZ /PC: /	
3072	016614	046777	050101	047440	XHEAD: .ASCIZ <377>/MAP OF DQ11 STATUS/<377>	
3073	016622	020106	050504	030461		
3074	016630	051440	040524	052524		
3075	016636	177523	000			
3076		016642			.EVEN	
3077	016642	000002			XSTATQ: 2	
3078	016644	006	003		.BYTE 6,3	
3079	016646	001244			TEMP1	
3080	016650	006	002		.BYTE 6,2	
3081	016652	001246			TEMP2	
3082					.EVEN	
3083						
3084					;BUFFERS FOR INPUT-OUTPUT	
3085						
3086	016654	000000			INBUF: 0	
3087		016716			. = +40	
3088	016716	000000			TEMP: 0	

H05

000048.P11 MACY11 27(1732) 24-SEP-76 10:11 PAGE 60
GENERAL UTILITIES (TYPE OUT, ERROR, SCOPE, ETC.)

```
016760 000000 .=. +40
017022 005015 MDATA: 0
017063 015 042522 042503 .=. +40
017116 005015 042012 052101 EM0: .ASCIZ <15><12>/RECEIVER DONE PRIMARY NOT SET!/
017144 005015 050504 042440 EM1: .ASCIZ <15><12>/DATA COMPARISON ERROR.../
017202 005015 047516 051040 EM2: .ASCIZ <15><12>/DQ ERROR FLAG SET. /
017257 015 054105 042520 EM3: .ASCIZ <15><12>/NO RECEIVER INTERRUPTS!!!! /
017305 015 042412 050130 DHO: .ASCIZ <15><12>/EXPECTED FOUND RX ADDR. TX ADDR. MASK /
017322 015 042012 042521 DH1: .ASCIZ <15><12>/EXPECTED RECEIVED /
017322 000005 DH2: .ASCIZ <15><12>/DQERR /
017324 006 004 .EVEN
017326 001270 DTO: 5
017330 006 001 .BYTE 6.4
017332 001266 SAVR5
017334 006 004 .BYTE 6.1
017336 001260 SAVR4
017340 006 004 .BYTE 6.4
017342 001256 SAVR1
017344 006 002 .BYTE 6.2
017346 012134 MASK
017350 000002 DT1: 2
017352 003 006 .BYTE 3.6
017354 014046 GDCHAR
017356 003 002 .BYTE 3.2
017358 014040 CHAR
017362 000001 DT2: 1
017364 006 002 .BYTE 6.2
017366 014032 ERR
017370 .ERRTAB:
017370 000000 0
017372 000000 0
017374 000000 0
017376 017022 EMO
017400 000000 ;HALT 1
017402 000000 0
017404 017063 EMI
017406 017202 DHO ;HALT 2
017410 017322 DTO
017412 017063 EMI
017414 017257 DH1 ;HALT 3
017416 017350 DT1
017420 017116 EM2
017422 017305 DH2 ;HALT 4
017424 017362 DT2
017426 017144 EM3
017430 000000 ;HALT 5
017432 000000 0
000001 .ENC
```


MERRPC	016607	2914	3071*																	
MERRX	016510	2601	3058*																	
MERR2	016327	1029	1332	3035*																
MERR3	016376	1264	3042*																	
MISC. =	000012	643*	2553																	
MLOCK	016434	1288	3048*																	
MNEW	016535	1257	3063*																	
MPASSX	016477	2597	3056*																	
MPFAIL	016244	2973	3026*																	
MQM	016236	2717	3024*																	
MR	016324	1306	3034*																	
MSTCLR=	104412	1130*	2978																	
MTITLE	001000	1038*	1222																	
MTSTN	016521	2910	3060*																	
MTSTPC	016422	1297	3046*																	
MVECX	016471	2593	3054*																	
NEXT	001216	1062*	1312*	1379*	1531*	1598*	1665*	1732*	1799*	1866*	1933*	2000*	2062*	2124*						
		2186*	2248*	2310*	2372*	2434*	2650	2946												
ODDBIT=	001000	626*	1300																	
OUT	016176	2993	2995	2997	3001	3013*														
PARAM =	104405	1120*	1298	3007																
PARAM1	014710	2737*	2758																	
PARERR	014764	2740	2742	2744	2750*	2763	2765	2767												
PARTI	015046	2756	2776*																	
PASCNT	001230	1067*	1211*	2585*	2604	2627														
PC =%	000007	581*	953*	1245*	2612*															
PFTAB	016054	2975	2981*																	
POLY	014034	2569*																		
POLY. =	000017	648*																		
POPRO =	012600	595*	2939																	
POP1SP=	005726	593*																		
POP2SP=	022626	597*																		
PS =	177776	587*	940*	1205*	1281*	1392*														
PUSHRO=	010046	594*	2936																	
PUSH1S=	005746	592*																		
PUSH2S=	024646	596*																		
RDSW	016062	2669	2718	2755	2989*	3003*	3013*													
RESREG	015662	2930	2933*																	
RESTAR	016000	2960	2966*																	
RESTRT	014174	2606	2609	2616*																
RESOS =	104407	1124*	2933																	
RETURN	001214	1061*	1218*	1305*	1307	1311*	1377*	2617*	2650*	2651	2946*	2948	2980							
RUN	001304	1092*	1315*	1318	1323*	1329*	1336*													
RUNCNT	001306	1093*	1316*	1325*	1327*															
RUNFLG	001302	1091*	1213*	1313	1317*															
RXBA.P=	000000	632*																		
RXBA.S=	000004	636*																		
RXBUFF	013022	1394	1485	1490	1533	1558	1574	1600	1625	1641	1667	1692	1709	1734						
		1759	1775	1801	1826	1842	1868	1893	1909	1935	1960	1976	2002	2024						
		2040	2064	2086	2102	2126	2148	2164	2198	2210	2226	2250	2272	2298						
		2312	2334	2350	2374	2396	2412	2436	2458	2474	2560*									
		1429	1475*																	
RXISR	003500	633*																		
RXWC.P=	000001	637*																		
RXWC.S=	000005	646*																		
RX.BCC=	000015	646*																		
RO =%	000000	574*	931*	933	943*	1025*	1030*	1258*	1268*	1271*	1273*	1275	1276	1419*						

SYMBOLS
69
U.S. AIR FORCE

01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00
01	02	03	04	05	06	07	08	09	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96	97	98	99	00

01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64 65 66 67 68 69 70 71 72 73 74 75 76 77 78 79 80 81 82 83 84 85 86 87 88 89 90 91 92 93 94 95 96 97 98 99 00

2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399
2300	2301	2302	2303	2304	2305	2306	2307	2308	2309	2310	2311	2312	2313	2314	2315	2316	2317	2318	2319	2320	2321	2322	2323	2324	2325	2326	2327	2328	2329	2330	2331	2332	2333	2334	2335	2336	2337	2338	2339	2340	2341	2342	2343	2344	2345	2346	2347	2348	2349	2350	2351	2352	2353	2354	2355	2356	2357	2358	2359	2360	2361	2362	2363	2364	2365	2366	2367	2368	2369	2370	2371	2372	2373	2374	2375	2376	2377	2378	2379	2380	2381	2382	2383	2384	2385	2386	2387	2388	2389	2390	2391	2392	2393	2394	2395	2396	2397	2398	2399

MACY11 27.732 24-SEP-76 10:11 PAGE 76
DZDQHB.P11 CROSS REFERENCE TABLE -- PERMANENT SYMBOLS

	1934	2001	2063	2125	2187	2249	2311	2373	2435	2577	2632	3092			
..MACY11	1	1363													
..MACY11	1	534	551	570	651	909	1036	1112	1114	1116	1118	1120	1122	1124	1126
..MACY11	1130	1130	1132	1134	1136	1138	1198	1313	1390	1532	1599	1656	1733	1800	1957
..MACY11	1934	2001	2063	2125	2187	2249	2311	2373	2435	2577	2632	3092			
..MACY11	1	43	534	570	651	909	1094								
..MACY11	1	43	50	58	101	186	192								
..MACY11	6														
..MACY11	5034	570	651	909	1036	1198	2577	2632							
..MACY11	921	2989													

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

* DZDQHB.SEG/SOL/CRF/PAGNUM/NL:000-UNI:V.P:1.DZDQHB.P11
RUN-TIME: 20 31 4 SECONDS
RUN-TIME: RAT:0: 125 58=2.1
CORE USED: 20K (39 PAGES)

