

.REM x

IDENTIFICATION

PRODUCT CODE: MAINDEC-11-DVKAE-B-D
 PRODUCT NAME: DLV11 TEST
 DATE: OCTOBER 1976
 MAINTAINER: DIAGNOSTIC GROUP
 AUTHOR: R.D. FIORENTINO
 REVISION B: D.J. CASALETTO

COPYRIGHT (C) DIGITAL EQUIPMENT CORPORATION
 1975, 1976

THE MATERIAL IN THIS DOCUMENT IS FOR INFORMATION
 PURPOSES ONLY AND IS SUBJECT TO CHANGE WITHOUT NOTICE.
 DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
 FOR THE USE OF SOFTWARE ON EQUIPMENT WHICH IS NOT
 SUPPLIED BY IT.
 DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY
 FOR ANY ERRORS WHICH MAY APPEAR IN THE DOCUMENT.

1. ABSTRACT

THIS IS A LOGIC TEST OF THE DLV11 SERIAL LINE UNIT FOR
 THE LSI-11 COMPUTER. THIS TEST WILL OPERATE ON
 THE DLV11 WITHOUT ANY SPECIAL TEST DEVICES BY DEFAULT.
 HOWEVER, A SPECIAL MRAP MODULE CAN BE USED AND TESTED
 BY OPTION.

THIS IS A MULTIPLE DEVICE TEST WHICH WILL OPERATE ON THE
 CONSOLE DLV11 ADDRESSED AT 177560, AND UP TO 8 ADDITIONAL

DLV11'S WITH SPECIFIED ADDRESSES.

DEFAULT ADDRESSES FOR MULTIPLE DEVICE TESTING ARE:

17750
17650

CONSOLE

BASE ADDRESS FOR ADDITIONAL DLV11'S

FOR COMPLETE INSTRUCTIONS ON MULTIPLE DEVICE TESTING SEE
SECTION 5.5

2. REQUIREMENTS

2.1 EQUIPMENT

LSI-11 COMPUTER
DLV11 SERIAL LINE UNIT
TERMINAL FOR DLV11
TEST MODULE (BY OPTION)

2.2 STORAGE REQUIREMENTS

4K MEMORY

3. LOADING PROCEDURE

3.1 METHOD

ABSOLUTE LOADER

4. STARTING PROCEDURE

200 - NORMAL ENTRY

TO LOAD AND EXECUTE:

1. LOAD PROGRAM WITH THE ABSOLUTE LOADER.
2. IF ANY PROGRAM OPTIONS ARE REQUIRED, SET THE APPROPRIATE BIT IN THE SOFTWARE SWITCH REGISTER AT LOCATION 122. (REF. SECTION 5.1)
3. START PROGRAM AT 200.
4. PROGRAM WILL PRINT "END OF PASS" FOLLOWING EACH PASS.
TO LOAD AND EXECUTE

STARTING ADDRESS:

200 - NORMAL ENTRY

4.1 CONTROL SWITCH SETTING

THIS PROGRAM CONTAINS A SOFTWARE SWITCH REGISTER FOR OPTION SELECTION (LOC 422). FOR IT TO OPERATE THE OPERATOR MUST SELECT THE APPROPRIATE OPTION BY SETTING OR RESETTNG THE RESPECTIVE BIT IN THE WORD.

TO DO THIS , THE LSI-11 MUST BE IN ODT MODE.

LSI-11 SERIAL LINE UNIT TERMINAL FOR DLV11 TEST MODULE (BY OPTION)

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

IMPORTANT ::::: THE TEST TIMES, PASS TIME CONSTANTS WITHIN
THIS PROGRAM REFLECT OPERATION AT 9600 BAUD.
A SLOWER BAUD RATE WILL CAUSE APT TO ABORT TESTING

5.4 TESTING A DLV11 AT A UNIQUE ADDRESS

1. TO SPECIFY A CONSOLE ADDRESS OTHER THAN 177560 OR VECTOR 60, THE OPERATOR MUST SUPPLY THE PROGRAM WITH THE CORRECT ADDRESSES BY INSERTING THEM AT THE TAG LABELED "RCSR". THE ADDRESSES MUST BE IN THE FOLLOWING ORDER:

RCSR: ADDRESS OF RECEIVER CSR
RBUF: ADDRESS OF RECEIVER BUFFER
TCSR: ADDRESS OF TRANSMITTER CSR
TBUF: ADDRESS OF TRANSMITTER BUFFER
RVECT: ADDRESS OF RECEIVER VECTOR
RPSW: ADDRESS OF ASSOCIATED PSW
TVECT: ADDRESS OF TRANSMITTER VECTOR
TPSW: ADDRESS OF ASSOCIATED PSW

2. TO TEST A SINGLE DLV11 AT A UNIQUE ADDRESS, INSERT THE ADDRESS OF THE RECEIVER CSR IN LOCATION SBASE, AND THE ADDRESS OF THE VECTOR IN LOCATION SVECT1, IN THE ETABLE. THE PROGRAM WILL GENERATE THE ADDITIONAL CSR, BUFFER, AND VECTOR ADDRESSES NEEDED FOR TESTING.

5.5 TESTING MULTIPLE DLV11 MODULES

ADDITIONAL DLV11'S MUST BE ADDRESSED WITHIN A RANGE OF 8 SEQUENTIAL ADDRESSES. A BASE ADDRESS MAY BE SPECIFIED BY THE OPERATOR OR A DEFAULT ADDRESS OF 176500 AND DEFAULT VECTOR OF 300 ARE USED. THE PROGRAM GENERATES A TABLE OF 8 ADDRESSES WITH EACH CSR GIVEN IN INCREMENTS OF 10. THE PROGRAM WILL THEN SIZE FOR DEVICES PRESENT.

5.6 THE DEVICE MAP

WHEN THE PROGRAM SIZES, A DEVICE MAP IS ESTABLISHED TO REPRESENT THOSE DEVICES PRESENT AND THEIR CORRESPONDING RECEIVER CSR ADDRESSES. THE DEVICE MAP CAN ALSO BE SET BY THE OPERATOR BY SELECTING THE APPROPRIATE BIT IN THE SWITCH REGISTER (SEE SECTION 5.1) AND PROCEEDING AS INDICATED BELOW.

THE DEVICE MAP IS A 16 BIT WORD WITH BITS 0-8 ARRANGED AS FOLLOWS:



DEVICE #0 = CONSOLE DEFAULT ADDRESS = 177560; VECTOR = 60
DEVICES #1-#8 = ADDITIONAL DLV11'S
DEVICE #1 = BASE ADDRESS (SBASE); BASE VECTOR (SVECT1)
DEVICE #2 = BASE ADDRESS + 10; BASE VECTOR + 10

314
313
312
311
310
309
308
307
306
305
304
303
302
301
300
299
298
297
296
295
294
293
292
291
290
289
288
287
286
285
284
283
282
281
280
279
278
277
276
275
274
273
272
271
270
269
268
267
266
265
264
263
262
261
260
259
258
257
256
255
254
253
252
251
250
249
248
247
246
245
244
243
242
241
240
239
238
237
236
235
234
233
232
231
230
229
228
227
226
225
224
223
222
221
220
219
218
217
216
215
214
213
212
211
210
209
208
207
206
205
204
203
202
201
200
199
198
197
196
195
194
193
192
191
190
189
188
187
186
185
184
183
182
181
180
179
178
177
176
175
174
173
172
171
170
169
168
167
166
165
164
163
162
161
160
159
158
157
156
155
154
153
152
151
150
149
148
147
146
145
144
143
142
141
140
139
138
137
136
135
134
133
132
131
130
129
128
127
126
125
124
123
122
121
120
119
118
117
116
115
114
113
112
111
110
109
108
107
106
105
104
103
102
101
100
99
98
97
96
95
94
93
92
91
90
89
88
87
86
85
84
83
82
81
80
79
78
77
76
75
74
73
72
71
70
69
68
67
66
65
64
63
62
61
60
59
58
57
56
55
54
53
52
51
50
49
48
47
46
45
44
43
42
41
40
39
38
37
36
35
34
33
32
31
30
29
28
27
26
25
24
23
22
21
20
19
18
17
16
15
14
13
12
11
10
9
8
7
6
5
4
3
2
1
0

000000 000000
000002 000000
000004 000006
000006 000000
000010 000012
000012 000000
000014 000016
000016 000000
000020 000022
000022 000000
000030 000032
000032 000000
000034 000036
000036 000000
000040 000042
000042 000000
000046 000046
000050 000050
000052 000054
000054 000000
000056 000060
000060 000000
000062 000064
000064 000000
000066 000070
000070 000000
000072 000074
000074 000000
000100 000100
000102 000102
000102 000002
000200 000200
000206 012737 005426 000024
000214 012737 000340 000026
000220 005067 000166
000224 005067 000156
000230 005067 000154
000234 005067 000154
000238 005067 000152
000240 004767 005062
000244 005500
000246 000167 000726
000400 000400

005426 000024
000340 000026
000166
000156
000154
000154
000152
005062
000726

.ENABL
MC, CND, MD
RTI
NOV NPWF1 2824
NOV 8340, 2826
CLR SPASS
CLR SFATAL
CLR \$TESTN
CLR \$DEVCT
CLR \$UNIT
JSR PC, NSGPRT
WORD HI
JMP START
*400
.SBTTL APT MAILBOX-ETABLE
;*****
;EVEN

;UNASSIGNED TRAP
;TIME-OUT, BUS ERROR
;RESERVED INSTRUCTION
;TRACE TRAP
;EMT TRAP
.STYPE, .STRAP, .SAPTBL, .SAPTHDR, .SAPTYPE, STARS
;SET POWER FAIL VECTOR
;SET POWER FAIL PSM
;CLEAR PASS COUNT
;STARTING POINT AFTER POWER FAIL
;CLEAR TEST NUMBER
;CLEAR DEVICE COUNT
;SET UNIT NUMBER TO ZERO
;PRINT PROGRAM NAME
;START DIAGNOSTIC

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

000024 000460
000044 000460
000460 000000
000460 000400
000464 000010
000466 000010
000470 000000
000472 000030

.SBTTL APT PARAMETER BLOCK

```
*****  
;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT  
*****  
;*****  
;SX=      : SAVE CURRENT LOCATION  
;=24     : SET POWER FAIL TO POINT TO START OF PROGRAM  
200      : FOR APT START UP  
;=44     : POINT TO APT INDIRECT ADDRESS PNTR.  
$APTHDR  : POINT TO APT HEADER BLOCK  
;=.SX    : RESET LOCATION COUNTER  
*****  
;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-POP11 DIAGNOSTIC  
;INTERFACE SPEC.
```

```
$APTHD:  :  
SHIBTS: .WORD 0      ;; TWO HIGH BITS OF 18 BIT MAILBOX ADDR.  
$PADDR: .WORD $MAIL  ;; ADDRESS OF APT MAILBOX (BITS 0-15)  
$STMT:  .WORD 10     ;; RUN TIM OF LONGEST TEST  
$PASTH: .WORD 10     ;; RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)  
$UNITH: .WORD 0      ;; ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT  
        .WORD $ETEND-$MAIL/2 ;; LENGTH MAILBOX-ETABLE(WORDS)
```

.....
EQUATES

X=1
N=1
AVECT1=300
ABASE=176500
\$KPTR=1000
CSR=177560
BIT15=100000
BIT14=40000
BIT13=20000
BIT12=10000
BIT11=4000
BIT10=2000
BIT9=1000
BIT8=400
BIT7=200
BIT6=100
BIT5=40
BIT4=20
BIT3=10
BIT2=4
BIT1=2
BIT0=1

;; DEFAULT BASE VECTOR FOR ADDITIONAL UNITS
;; DEFAULT BASE DEVICE ADDRESS FOR ADDITIONAL UNITS
;; DEFAULT CONSOLE DEVICE ADDRESS

000001
000001
000300
176500
001000
177560
100000
040000
020000
010000
004000
002000
001000
000400
000200
000100
000040
000020
000010
000004
000002
000001

K01

.MAIN. MACY11 27(732) 04-OCT-76 14:23 PAGE 11
DVKAEB.P11 APT PARAMETER BLOCK

461 000474 000000
462 000476 000000
463 000500 000000
464 000502 000000
465 000504 000000
466
467
468

TMP0: .WORD 0
TMP1: .WORD 0
TMP2: .WORD 0
TMP3: .WORD 0
CTSTFL: .WORD 0

:LOCATION TO STORE TABLE OFFSETS
:LOCATION TO KEEP NO. OF TEST DEVICES
:LOCATION TO KEEP DEVICE MAP TEST MASK
:FLAG TO INDICATE CURRENT TEST DEVICE IS CONSOLE

```

469
470
471 000506 177560
472 000510 177562
473 000512 177564
474 000514 177566
475 000516 000060
476 000520 000062
477 000522 000064
478 000524 000066
479
480
481
482 000526 177560
483 000530 177562
484 000532 177564
485 000534 177566
486 000536 000060
487 000540 000062
488 000542 000064
489 000544 000066
490
491 000600 000000
492 000640 000020
493 000640 000010
  
```

: REGISTER AND VECTOR ADDRESSES FOR THE DLV11 UNDER TEST

```

ACSR: CSR
RBUF: CSR+2
TCSR: CSR+4
TBUF: CSR+6
RVECT: 60
RPSM: 62
TVECT: 64
TPSM: 66
  
```

: INITIAL REGISTER AND VECTOR ADDRESSES FOR THE CONSOLE DLV11

```

DRCSR: 177560 : ADDRESS OF RECEIVER STATUS REGISTER
DRBUF: 177562 : ADDRESS OF RECEIVER DATA BUFFER REGISTER
DTCSR: 177564 : ADDRESS OF TRANSMITTER STATUS REGISTER
DTBUF: 177566 : ADDRESS OF TRANSMITTER DATA BUFFER REGISTER
DRVECT: 60
DRPSM: 62
DTVECT: 64
DTPSM: 66
  
```

```

ADRTEL: .BLKH 20
VCTTEL: .BLKH 10
  
```

```

494 001200 001200 001000      START:  MOV     #1200      ; INITIALIZE STACK POINTER
495 001200 012706 001000      ;ROUTINE TO HANDLE THE TESTING OF MULTIPLE DLV11 MODULES
496
497 001204 005767 177176      TST     $PSS          ;CHECK IF ON FIRST PASS
498 001210 001161 177176      BNE     SIZED         ;IF NOT ON FIRST PASS
499 001212 132767 000001 177200  BITB    #BIT0,SENV    ;CHECK IF ON APT
500 001220 001404 177171      BEQ     MAN          ;IF NOT ON APT
501 001222 132767 000200 177171  BITB    #BIT7,SENVH   ;DID APT SIZE?
502 001230 001067 177162  MANL:   BIT     #BITB,$SREG  ;IF APT SIZED
503 001232 032767 000400 177162  BEQ     SETTBL       ;MANUAL SETTING OF $DEVH?
504 001240 001402 177162  SETTBL  SETTBL       ;NO
505 001242 000000 177162  HALT    ;INPUT DEVICE MAP
506 001244 000461 177162  BR     APTSIZ        ;CONT WHEN DEVICE MAP IS SET
507 001246 004767 000104 177222  JSR    PC,DEVADR    ;GENERATE DEVICE ADDRESS TABLE
508 001252 005067 177222  CLR    TMP2         ;CLEAR TEMP LOCATION TO COUNT DEVICES
509 001254 005067 177174  CLR    $DEVH       ;CLEAR DEVICE MAP
510 001256 012737 001316 000004  MOV    #253,R1     ;SET TIMEOUT POINTER
511 001262 016700 177160  MOV    $BASE,R0    ;LOAD BASE ADDRESS
512 001270 062700 000100 15:    ADD    #100,R0     ;POINT R0 TO UNIT#8 (UNIT#0 = CONSOLE)
513 001274 062700 000100 15:    ADD    #10,R0      ;POINT R0 TO THE NEXT UNIT ADDRESS
514 001300 162700 000010 15:    SUB    #1,R0       ;CHECK FOR DEVICE EXISTANCE
515 001304 005710 177144  TST    $DEVH       ;NO TIMEOUT - INDICATE DEVICE IN DEVICE MAP
516 001306 005267 177162  INC    TMP2        ;INCREMENT DEVICE COUNT
517 001312 005067 177134 25:    ASL    $DEVH       ;ADJUST DEVICE MAP FOR NEXT UNIT CHECK
518 001316 005067 177126  CMP    $BASE,R0    ;FINISHED SIZING?
519 001322 026700 177126  BLT    15          ;BR IF NOT
520 001326 002764 177172  MOV    DCRSR,R0    ;LOAD CONSOLE DEVICE ADDRESS
521 001330 016700 177172  MOV    #353,R1     ;SET TIMEOUT VECTOR
522 001334 012737 001354 000004  TST    (R0)        ;TEST FOR CONSOLE EXISTANCE
523 001342 005710 177106  INC    $DEVH       ;NO TIMEOUT - INDICATE CONSOLE IN DEVICE MAP
524 001344 005267 177124  INC    TMP2        ;INCREMENT DEVICE COUNT
525 001350 005267 177124 35:    INC    VCTR        ;BR TO GENERATE VECTOR ADDRESS TABLE
526 001354 000432 177124 35:    BR     VCTR
527
528 ;ROUTINE TO GENERATE DEVICE ADDRESS TABLE
529
530 001356 012702 000600  DEVRDR: MOV    #AORTBL,R2 ;POINT R2 TO TOP OF THE ADDRESS TABLE
531 001362 016700 177066  MOV    $BASE,R0    ;LOAD BASE ADDRESS IN R0
532 001366 010001 177066  MOV    R0,R1       ;CALCULATE DEVICE ADDRESS LIMIT
533 001370 062701 000100 15:    ADD    #100,R1     ;MOVE DEVICE ADDRESS TO TABLE
534 001374 010022 177066  MOV    R0,(R2)+    ;CALCULATE NEXT DEVICE ADDRESS
535 001376 062700 000010  ADD    #10,R0      ;FINISHED?
536 001402 020001 177066  CMP    R0,R1       ;BR, IF NOT
537 001404 002773 177066  BLT    15          ;RETURN
538 001406 000207 177066  RTS    PC
539
540 001410 005067 177064  APTSIZ: CLR    TMP2         ;CLEAR TEMP LOCATION TO KEEP DEVICE COUNT
541 001414 016702 177036  MOV    $DEVH,R2    ;GET DEVICE MAP
542 001420 005702 177036  TSTDVH: TST    R2     ;TEST MSB OF DEVICE MAP
543 001422 100002 177036  BPL    15          ;BR, IF CLEAR
544 001424 005267 177060  INC    TMP2        ;IF SET, INCREMENT DEVICE COUNT
545 001430 006302 177060 15:    ASL    R2          ;SHIFT NEXT BIT INTO MSB POSITION
546 001432 001401 177060  BEQ    DVADR       ;FINISHED (NO BITS LEFT SET IN MAP)?
547 001434 000771 177060  BR     TSTDVH     ;BR, IF NOT
    
```

55	001436	004767	177714		DVADT: JSR	PC,DEVAOR		:GENERATE DEVICE ADDRESS TABLE
56	001442	012702	000640		VCTADR: MOV	#VCTBL,R2		:GET LOCATION OF VECTOR TABLE
57	001444	005001			CLR	R1		:CLEAR DATA WORD
58	001445	116700	176774		MOV#B	\$VECT1,R0		:COPY BASE VECTOR
59	001454	042700	177400		BIC	#177400,R0		:CLEAR BYTE SIGN EXTENSION
60	001460	010022			IS: MOV	R0,(R2)+		:PUT VECTOR ADDRESS IN THE TABLE
61	001462	005201			INC	R1		:KEEP COUNT OF THE NO. OF ADDRESSES STORED
62	001464	062700	000010		ADD	#10,R0		:FORM NEXT SEQUENTIAL ADDRESS
63	001470	022701	000010		CHP	#10,R1		:FINISHED?
64	001474	002371			BGE	IS		:IF NO
65	001476	016700	176776		MOV	TMP2,R0		:COPY DEVICE COUNT INTO R0
66	001502	005001			CLR	R1		:CLEAR AUXILIARY REGISTER
67	001504	000300			SM#B	R0		:PUT DEVICE COUNT IN UPPER BYTE OF R0
68	001506	006300			ASL	R0		:MOVE MSB OF COUNT INTO
69	001510	006300			ASL	R0		:MSB OF R0
70	001512	006300			SHIFT: ASL	R0		:PUT MSB OF COUNT INTO CARRY
71	001514	106101			ROL#B	R1		:MOVE MSB OF COUNT INTO R1
72	001516	006300			ASL	R0		:MOVE NEXT BIT OF COUNT
73	001520	106101			ROL#B	R1		:INTO R1
74	001522	006300			ASL	R0		:MOVE LAST BIT OF DIGIT
75	001524	106101			ROL#B	R1		:INTO R1
76	001526	062701	000060		ADD	#60,R1		:CONVERT TO ASCII
77	001528	000301			SM#B	R1		:MOVE DIGIT TO UPPER BYTE
78	001534	032701	000020		BIT	#BIT4,R1		:HAVE BOTH DIGITS BEEN MAVED TOR1?
79	001540	001764			BE#	SHIFT		:BR IF NOT
80	001542	010167	003774		MOV	R1,R2		:MOVE DEVICE COUNT TO OUTPUT MESSAGE
81	001546	004767	003554		JSR	PC,MSGPRT		:PRINT MESSAGE DEFINING THE
82	001552	005542			WORD	R2		:NO. OF DEVICES SIZED
83	001554	005067	176716		SIZED: CLR	TMP1		:CLEAR TEMP LOCATION FOR TABLE OFFSETS
84	001560	012767	000002	176714	MOV	#BIT1,TMP3		:SET UP BIT MASK FOR DEVICE MAP
85	001566	032767	000001	176662	TSTTY: BIT	#BIT0,\$DEVH		:IS CONSOLE DEVICE TO BE TESTED?
86	001574	001001			BNE	TCONSL		:IF YES
87	001576	000414			BR	SETUP		:GO TEST ADDITIONAL DEVICES
88	001600	012700	000506		TCONSL: MOV	#RCSR,R0		:SET UP REGISTER ADDRESSES FOR CONSOLE
89	001604	012701	000526		MOV	#RCSR,R1		:POINT R1 TO CONSOLE ADDRESS TABLE
90	001610	005267	176670		INC	CTSTFL		:INDICATE THAT CONSOLE IS UNDER TEST
91	001614	012120			IS: MOV	(R1)+,(R0)+		:FORM CONSOLE ADDRESSES
92	001616	022700	000524		CHP	#TPSW,R0		:FINISHED?
93	001622	002374			BGE	IS		:IF NO
94	001624	000167	000116		JMP	TST1		:GO TEST CONSOLE DEVICE
95	001630	036767	176646	176620	SETUP: BIT	TMP3,\$DEVH		:TEST DEVICE BIT IN DEVICE MAP
96	001636	001010			BNE	SETADR		:BR IF SET AND TEST DEVICE
97	001640	006367	176636		ASL	TMP3		:IF CLEAR, UPDATE DEVICE MAP TEST MASK
98	001644	062767	000002	176624	ADD	#2,TMP1		:INCREMENT TABLE OFFSET
99	001652	005267	176534		INC	SUNIT		:INCREMENT UNIT NUMBER
100	001656	000764			BR	SETUP		:BR TO TEST NEXT DEVICE MAP BIT
101	001660	005267	176526		SETADR: INC	SUNIT		:INCREMENT UNIT NUMBER BEFORE TESTING DEVICE
102	001664	006367	176612		ASL	TMP3		:UPDATE DEVICE MAP TEST MASK
103	001670	016702	176602		MOV	TMP1,R2		:MOVE TABLE OFFSET TO R2
104	001674	062767	000002	176574	ADD	#2,TMP1		:UPDATE TABLE OFFSET FOR NEXT DEVICE

605	001702	016200	000600		MOV	AORTBL(R2),R0	: REGISTER ADDRESS OF NEXT DEVICE
606	001706	012701	000506		MOV	RACSR,R1	: LOCATION OF REGISTER ADDR.
607	001712	010021			ADR: MOV	R0,(R1)+	: CREATE ACTIVE DEVICE ADDRESS TABLE
608	001714	005720			TST	(R0)+	
610	001716	030027	000006		BIT	R0,#000006	: FINISHED?
611	001722	001373			BNE	ADR	: BR, IF NOT
612							
613	001724	016200	000640		MOV	VCTTBL(R2),R0	: VECTOR ADDRESS OF NEXT DEVICE
614	001730	010021			VECT: MOV	R0,(R1)+	: CREATE ACTIVE VECTOR ADDRESS TABLE
615	001732	005720			TST	(R0)+	
616	001734	030027	000006		BIT	R0,#000006	: FINISHED?
617	001740	001373			BNE	VECT	: BR, IF NOT
618	001742	000167	000000		JMP	TST1	: GO TEST THIS DEVICE
619							
620							
621							
622							
623							
624							
625							
626							
627							
628							
629							
630							
631							
632							
633							
634							
635							
636							
637							
638							
639							
640							
641							
642							
643							
644							
645							
646							
647							
648							
649							
650							
651							

TEST ABILITY TO REFERENCE RECEIVER CSR WITHOUT TRAPPING

001746	012767	000001	176430	TST1:	MOV	01,STESTN	: MOVE TEST NUMBER TO MAILBOX
001746	012737	001770	000004	LP1:	MOV	06,004	: SET TIME OUT VECTOR
001754	005777	176520			TST	RACSR	: REFERENCE THE CSR
001756	000416				BR	IS	: GO TO NEXT TEST IF NO TRAP
001770				2B:			
001770	032767	040000	176424		BIT	0BIT14,SSWREG	: CHECK FOR LOOP ON ERROR
001776	001345				BNE	LP1	: GO TO LOOP ERROR
002000	012767	000001	176374		MOV	01,0FATAL	
002006	012767	000001	176364		MOV	01,0SSTGY	: MOVE ERROR NUM TO MAILBOX
002014	005767	176402			TST	SSWREG	: CHECK FOR HALT ON ERROR
002020	100401				BMI	IS	: HALT IF SET
002022	000000				HALT		: <CANNOT ACCESS RCSR>
002024				1B:			
002024	032767	002000	176370		BIT	0BIT10,SSWREG	: CHECK FOR LOOP ON TEST
002032	001345				BNE	TST1	: GO TO LOOP ON TEST
002034	012737	000006	000004		MOV	06,004	: RESTORE TIMEOUT VECTOR

TEST ABILITY TO REFERENCE RECEIVER BUFFER WITHOUT TRAPPING

002042	012767	000002	176334	TST2:	MOV	02,STESTN	: MOVE TEST NUMBER TO MAILBOX
002042	012737	002004	000004	LP2:	MOV	02,004	: SET TIME OUT VECTOR
002050	005777	176426			TST	0RBUF	: REFERENCE THE BUFFER
002052	000416				BR	IS	: GO TO NEXT TEST
002064				2B:			
002064	032767	040000	176330		BIT	0BIT14,SSWREG	: CHECK FOR LOOP ON ERROR
002072	001345				BNE	LP2	: GO TO LOOP ERROR
002074	012767	000002	176300		MOV	02,0FATAL	
002102	012767	000001	176270		MOV	01,0SSTGY	: MOVE ERROR NUM TO MAILBOX
002110	005767	176306			TST	SSWREG	: CHECK FOR HALT ON ERROR
002114	100401				BMI	IS	: HALT IF SET
002116	000000				HALT		: <CANNOT ACCESS RBUF>
002120				1B:			
002120	032767	002000	176274		BIT	0BIT10,SSWREG	: CHECK FOR LOOP ON TEST
002128	001345				BNE	TST2	: GO TO LOOP ON TEST
002130	012737	000006	000004		MOV	06,004	: RESTORE TIMEOUT VECTOR

TEST ABILITY TO REFERENCE TRANSMITTER CSR WITHOUT TRAPPING

002136	012767	000003	176240	TST3:	MOV	03,STESTN	: MOVE TEST NUMBER TO MAILBOX
--------	--------	--------	--------	-------	-----	-----------	-------------------------------

```

662 002144 012737 002160 000004 LP3:  MOV 06,204 ;SET UP TIME OUT VECTOR
663 002151 005777 176334  TST 0TCSR ;REFERENCE THE T CSR
664 002156 000416  BR 15 ;GO TO NEXT TEST IF NO TIME-OUT
665 002160 2S:
666 002160 032767 040000 176234 BIT 0BIT14,SSMREG ; CHECK FOR LOOP ON ERROR
667 002166 001366 LP3 ; GO TO LOOP ERROR
668 002170 012767 000003 176204 MOV 04,0FATAL
669 002176 012767 000001 176174 MOV 01,0MSGTY ; MOVE ERROR NUM TO MAILBOX
670 002204 005767 176212 TST SSMREG ; CHECK FOR HALT ON ERROR
671 002210 100401 BHI 15 ; HALT IF SET
672 002212 000000 HALT ;(CANNOT ACCESS TCSR)
673 002214 1S:
674 002214 032767 002000 176200 BIT 0BIT10,SSMREG ; CHECK FOR LOOP ON TEST
675 002222 001366 TST3 ; GO TO LOOP ON TEST
676 002224 012737 000006 000004 MOV 06,204 ;RESTORE TIMEOUT VECTOR
677
678 ;TEST ABILITY TO REFERENCE TRANSMITTER BUFFER WITHOUT TRAPPING
679
680 002232 TST4:
681 002232 012767 000004 176144 MOV 04,0TESTN ; MOVE TEST NUMBER TO MAILBOX
682 002240 032767 000001 176152 BIT 0BIT0,SENV ; CHECK IF ON APT
683 002246 001406 BEQ LP4 ; DO THIS TEST IF NOT ON APT
684 002250 005767 176132 TST 0PASS ; CHECK IF FIRST PASS
685 002254 001403 BEQ LP4 ; BR IF FIRST PASS
686 002256 005767 176222 TST 0TSTFL ; IF NOT FIRST PASS - SKIP TEST
687 002262 001034 BNE TST5 ; IF CONSOLE IS UNDER TEST
688 002264 012737 002300 000004 LP4: MOV 06,204 ;SET UP TIME-OUT VECTOR
689 002272 005777 176216 TST 0TBUF ;REFERENCE THE T BUFFER
690 002276 000416 BR 15 ;GO TO NEXT TEST IF NO TIME-OUT
691 002300 2S:
692 002300 032767 040000 176114 BIT 0BIT14,SSMREG ; CHECK FOR LOOP ON ERROR
693 002306 001366 LP4 ; GO TO LOOP ERROR
694 002310 012767 000004 176064 MOV 04,0FATAL
695 002316 012767 000001 176054 MOV 01,0MSGTY ; MOVE ERROR NUM TO MAILBOX
696 002324 005767 176072 TST SSMREG ; CHECK FOR HALT ON ERROR
697 002330 100401 BHI 15 ; HALT IF SET
698 002332 000000 HALT ;(CANNOT ACCESS TBUF)
699 002334 1S:
700 002334 032767 002000 176060 BIT 0BIT10,SSMREG ; CHECK FOR LOOP ON TEST
701 002342 001366 TST4 ; GO TO LOOP ON TEST
702 002344 012737 000006 000004 MOV 06,204 ;RESTORE TIMEOUT VECTOR
703 002352 000240 NOP
704
705 ;TEST THAT "BREAK" BIT (BIT0) IN TCSR CAN BE SET AND CLEARED AND
706 ;THAT RESET CAN CLEAR IT.
707
708 TST5:
709 002354 012767 000005 176022 MOV 05,0TESTN ; MOVE TEST NUMBER TO MAILBOX
710 002362 032767 000001 176030 BIT 0BIT0,SENV ; CHECK IF ON APT
711 002370 001406 BEQ LPSA ; DO THIS TEST IF NOT ON APT
712 002372 005767 176010 TST 0PASS ; CHECK IF FIRST PASS
713 002374 001403 BEQ LPSA ; BR IF FIRST PASS
714 002400 005767 176100 TST 0TSTFL ; IF NOT FIRST PASS - SKIP TEST
715 002404 001126 BNE TST6 ; IF CONSOLE IS UNDER TEST
716 002406 005004 LPSA: CLR R4 ; SET UP 300 MS COUNT
717 002410 006204 LPSB: INC R4
    
```

.MAIN. MACY11 27(732) 04-OCT-76 14:23 PAGE 17
DVKREB.P11 ERROR 4 CANNOT ACCESS TBUF

718	002412	001376				BNE	LPSB		
719	002414	000005				RESET			
720	002416	032777	000001	176066	LPS:	BIT			: CLEAR EVERYTHING
721	002418	001416				REQ	BBITO, BTCSR		: CHECK BIT 0 (BREAK)
722	002420	032767	040000	175766		BIT	TSCON		: CONTINUE IF NOT SET
723	002422	001367				BNE	BBIT14, SSMREG		: CHECK FOR LOOP ON ERROR
724	002424	012767	000005	175736		NOV	LPS		: GO TO LOOP ERROR
725	002426	012767	000001	175726		NOV	B5, SFATAL		
726	002428	005767	175744			TST	B1, SMSGTY		: MOVE ERROR NUM TO MAILBOX
727	002430	100401				BMI	SSMREG		: CHECK FOR HALT ON ERROR
728	002432	000000				HALT	IS		: HALT IF SET
729	002434								: <BIT 0 IN TCSR SHOULD BE 0 AFTER RESET>
730	002436	032777	000001	176022	IS:	BIS	BBITO, BTCSR		: SET BIT0 (BREAK)
731	002438	032777	000001	176014	TSCON:	BIT	BBITO, BTCSR		: CHECK IF SET
732	002440	001016				BNE	TSCON1		: CONTINUE IF SET
733	002442	032767	040000	175714		BIT	BBIT14, SSMREG		: CHECK FOR LOOP ON ERROR
734	002444	001365				BNE	TSCON		: GO TO LOOP ERROR
735	002446	012767	000006	175644		NOV	B6, SFATAL		
736	002448	012767	000001	175654		NOV	B1, SMSGTY		: MOVE ERROR NUM TO MAILBOX
737	002450	005767	175672			TST	SSMREG		: CHECK FOR HALT ON ERROR
738	002452	100401				BMI	IS		: HALT IF SET
739	002454	000000				HALT			: <CANNOT SET BIT0 IN TCSR>
740	002456								
741	002458	042777	000001	175750	IS:	BIC	BBITO, BTCSR		: CLEAR BIT0
742	002460	032777	000001	175742	TSCON1:	BIT	BBITO, BTCSR		: CHECK TO SEE IF CLEAR
743	002462	001416				REQ	TSCON2		: CONTINUE IF CLEAR
744	002464	032767	040000	175642		BIT	BBIT14, SSMREG		: CHECK FOR LOOP ON ERROR
745	002466	001365				BNE	TSCON1		: GO TO LOOP ERROR
746	002468	012767	000007	175612		NOV	B7, SFATAL		
747	002470	012767	000001	175602		NOV	B1, SMSGTY		: MOVE ERROR NUM TO MAILBOX
748	002472	005767	175620			TST	SSMREG		: CHECK FOR HALT ON ERROR
749	002474	100401				BMI	IS		: HALT IF SET
750	002476	000000				HALT			: <CANNOT CLEAR BIT0 IN TCSR>
751	002478								
752	002480	012704	010000		IS:	NOV	B10000, R4		: SET UP DELAY
753	002482	005304			TSEDEC:	DEC	R4		: DELAY
754	002484	001376				BNE	TSEDEC		
755	002486	032777	000001	175666		BIS	BBITO, BTCSR		: SET IT AGAIN
756	002488	000005				RESET			: CHECK RESET CLEAR AGAIN
757	002490	032777	000001	175656		BIT	BBITO, BTCSR		: CHECK BIT 0
758	002492	001416				REQ	IS		: CONTINUE IF RESET
759	002494	032767	040000	175556		BIT	BBIT14, SSMREG		: CHECK FOR LOOP ON ERROR
760	002496	001360				BNE	TSCON2		: GO TO LOOP ERROR
761	002498	012767	000010	175526		NOV	B10, SFATAL		
762	002500	012767	000001	175516		NOV	B1, SMSGTY		: MOVE ERROR NUM TO MAILBOX
763	002502	005767	175534			TST	SSMREG		: CHECK FOR HALT ON ERROR
764	002504	100401				BMI	IS		: HALT IF SET
765	002506	000000				HALT			: <RESET DID NOT CLEAR BIT0 IN TCSR>
766	002508								
767	002510	032767	002000	175522	IS:	BIT	BBIT10, SSMREG		: CHECK FOR LOOP ON TEST
768	002700	001225				BNE	TST5		: GO TO LOOP ON TEST

769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824

```

002702
002703
002704
002710
002716
002720
002724
002725
002732
002734
002736
002744
002750
002756
002760
002766
002770
002776
003004
003010
003012
003014
003014
003022
003024
003032
003040
003044
003046
003050
003052
003054
003066
003074
003076
003104
003112
003116
003120
003122
003124
003130
003136
003146
003150
003156
003164
003170
003172
012767
032767
001406
005767
001433
006767
001152
000005
012777
105427
032777
001434
032767
001362
012767
012767
005767
100401
000000
032767
001344
012767
000012
012767
000001
005767
100401
000000
052777
000100
032777
000100
001016
032767
001365
012767
000013
012767
000001
005767
100401
000000
042777
000100
032777
000100
001416
032767
000000
012767
000014
012767
000001
005767
100401
000000
175474
175502
175462
175552
175556
175534
175434
175404
175374
175412
175400
175350
175340
175326
175276
175266
175304
175362
175354
175254
175224
175214
175232
    
```

TEST THAT TCSR BIT6 (TRANSMITTER INTERRUPT ENABLE) CAN BE SET AND RESET AND THAT RESET CLEARS IT.

```

TST6:
MOV #6,STESTH      ; MOVE TEST NUMBER TO MAILBOX
BIT #BIT0,SENV     ; CHECK IF ON APT
REQ TST6A          ; IF NOT ON APT
SPASS              ; CHECK IF ON FIRST PASS
REQ TST6A          ; BR, IF FIRST PASS
TST CTSTFL         ; IF NOT FIRST PASS - SKIP TEST
BNE TST7           ; IF CONSOLE IS UNDER TEST

TST6A:
LP6: RESET         ; SEND THE INITIAL RESET TO CLEAR
MOV #ERR12,IVECT  ; SET UP VECTOR IN CASE OF INTERRUPT
MTPS #200         ; TURN OFF INTERRUPTS
BIT #BIT6,TCSR   ; CHECK IF BIT 6 IS ON
REQ TCON1        ; CONTINUE IF NOT SET
BIT #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
BNE LP6          ; GO TO LOOP ERROR

IS:
ERR12:
BIT #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
BNE LP6          ; GO TO LOOP ERROR
MOV #12,BFATAL   ; MOVE ERROR NUM TO MAILBOX
MOV #1,MSGTY     ; CHECK FOR HALT ON ERROR
TST SSWREG       ; HALT IF SET
HALT              ; <BIT6 IN TCSR NOT CLEAR AFTER RESET>

IS:
ERR12:
BIT #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
BNE LP6          ; GO TO LOOP ERROR
MOV #12,BFATAL   ; MOVE ERROR NUM TO MAILBOX
MOV #1,MSGTY     ; CHECK FOR HALT ON ERROR
TST SSWREG       ; HALT IF SET
HALT              ; <DEVICE SHOULD NOT HAVE INTERRUPTED WITH PRIORITY BIT SET>

IS:
TCON1:
BIS #BIT6,TCSR   ; SET BIT6
BIT #BIT6,TCSR   ; CHECK IF SET
BNE TCON2       ; CONTINUE IF SET.
BIT #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
BNE TCON1       ; GO TO LOOP ERROR
MOV #13,BFATAL  ; MOVE ERROR NUM TO MAILBOX
MOV #1,MSGTY    ; CHECK FOR HALT ON ERROR
TST SSWREG      ; HALT IF SET
HALT              ; <CANNOT SET BIT6 IN TCSR>

IS:
TCON2:
BIC #BIT6,TCSR  ; CLEAR BIT6
BIT #BIT6,TCSR  ; CHECK IF CLEAR
REQ TCON3       ; CONTINUE IF CLEAR
BIT #BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
BNE TCON2       ; GO TO LOOP ERROR
MOV #14,BFATAL  ; MOVE ERROR NUM TO MAILBOX
MOV #1,MSGTY    ; CHECK FOR HALT ON ERROR
TST SSWREG      ; HALT IF SET
HALT              ; <CANNOT CLEAR BIT6 IN TCSR>
    
```

.MAIN. MACY11 27(732) 04-OCT-76 14:23 PAGE 19
 DVKREB.P11 ERROR 14 CANNOT CLEAR BIT6 IN TCSR

```

003174 052777 000100 175310 15:
003174 000005 T6CON3: BIS #BIT6,0TCSR ; SET BIT6 AGAIN
003202 000005 RESET ; TRY RESET AGAIN
003204 032777 000100 175300 BIT #BIT6,0TCSR ; CHECK TO SEE IF CLEAR
003212 001416 BEQ 15 ; CONTINUE IF CLEAR
003214 032767 040000 175200 BIT #BIT14,SSMREG ; CHECK FOR LOOP ON ERROR
003222 001364 T6CON3 ; GO TO LOOP ERROR
003224 012767 000015 175150 MOV #15,0FATAL ; MOVE ERROR NUM TO MAILBOX
003232 012767 000001 175140 MOV #1,0MSGTY ; CHECK FOR HALT ON ERROR
003240 005767 175156 TST SSMREG ; HALT IF SET
003244 100401 BHI 15 ; <RESET DID NOT CLEAR BIT6 IN TCSR>
003246 000000 HALT
003250 032767 002000 175144 15:
003256 001211 BNE #BIT10,SSMREG ; CHECK FOR LOOP ON TEST
TST6 ; GO TO LOOP ON TEST
    
```

041
042
043
044
045
046
047
048
049
050
051
052
053
054
055
056
057
058
059
060
061
062
063
064
065
066
067
068
069
070
071
072
073
074
075
076
077
078
079
080
081
082
083
084
085
086
087
088
089
090
091
092
093
094
095
096

```

003260
003260 012767 000007 175116
003266 032767 000001 175124
003274 001406
003276 005767 175104
003302 001403
003304 005767 175174
003310 001131
003312
003312 106427 000200
003316 000005
003320 032777 000100 175160
003326 001416
003330 032767 040000 175064
003336 001365
003340 012767 000016 175034
003346 012767 000001 175024
003354 005767 175042
003360 100401
003362 000000
003364
003364 052777 000100 175114
003372 032777 000100 175106
003400 001016
003402 032767 040000 175012
003410 001365
003412 012767 000017 174762
003420 012767 000001 174752
003426 005767 174770
003432 100401
003434 000000
003436
003436 042777 000100 175042
003444 032777 000100 175034
003452 001416
003454 032767 040000 174740
003462 001365
003464 012767 000020 174710
003472 012767 000001 174700
003500 005767 174716
003504 100401
003506 000000
003510
003510 052777 000100 174770
003516 000005
003520 032777 000100 174760
003526 001416
003530 032767 040000 174664
003536 001364
003540 012767 000021 174634
003546 012767 000001 174624
003554 005767 174642
  
```

```

: TEST THAT RCSR BIT6 (RCVR INTERRUPT ENABLE) CAN BE SET AND
: CLEARED, AND THAT RESET CLEARS IT.
:
TST7:
MOV 07,STESTN : MOVE TEST NUMBER TO MAILBOX
BIT 08,IT0,SENV : CHECK IF ON APT
BEQ TST7A : IF NOT ON APT
TST SPASS : CHECK IF IN FIRST PASS
BEQ TST7A : OR, IF NOT FIRST PASS
TST CTSTFL : IF NOT FIRST PASS - SKIP TEST
BNE TST10 : IF CONSOLE IS UNDER TEST

TST7A:
LPS: MTPS 0200 : DISABLE INTERRUPTS
RESET : CLEAR EVERYTHING
BIT 08,IT6,RCSR : CHECK BIT6 FOR 0
BEQ TBCON1 : CONTINUE IF RESET
BIT 08,IT14,SSMREG : CHECK FOR LOOP ON ERROR
BNE LPS : GO TO LOOP ERROR

MOV 016,SEFATAL
MOV 01,SESGTY : MOVE ERROR NUM TO MAILBOX
TST SSMREG : CHECK FOR HALT ON ERROR
BHI 15 : HALT IF SET
HALT : ;(BIT6 IN RCSR NOT CLEAR BY RESET)

IS:
TBCON1: BIS 08,IT6,RCSR : SET BIT6
BIT 08,IT6,RCSR : CHECK IF BIT SET
BNE TBCON2 : CONTINUE IF SET
BIT 08,IT14,SSMREG : CHECK FOR LOOP ON ERROR
BNE TBCON1 : GO TO LOOP ERROR

MOV 017,SEFATAL
MOV 01,SESGTY : MOVE ERROR NUM TO MAILBOX
TST SSMREG : CHECK FOR HALT ON ERROR
BHI 15 : HALT IF SET
HALT : ;(CANNOT SET BIT6 IN RCSR)

IS:
TBCON2: BIC 08,IT6,RCSR : CLEAR BIT6
BIT 08,IT6,RCSR : CHECK IF CLEAR
BEQ TBCON3 : CONTINUE IF RESET
BIT 08,IT14,SSMREG : CHECK FOR LOOP ON ERROR
BNE TBCON2 : GO TO LOOP ERROR

MOV 020,SEFATAL
MOV 01,SESGTY : MOVE ERROR NUM TO MAILBOX
TST SSMREG : CHECK FOR HALT ON ERROR
BHI 15 : HALT IF SET
HALT : ;(CANNOT CLEAR BIT6 IN RCSR)

IS:
TBCON3: BIS 08,IT6,RCSR : SET BIT6 AGAIN
RESET : ISSUE ANOTHER RESET
BIT 08,IT6,RCSR : CHECK IF RESET CLEARED BIT6.
BEQ 15 : CONTINUE IF CLEAR
BIT 08,IT14,SSMREG : CHECK FOR LOOP ON ERROR
BNE TBCON3 : GO TO LOOP ERROR

MOV 021,SEFATAL
MOV 01,SESGTY : MOVE ERROR NUM TO MAILBOX
TST SSMREG : CHECK FOR HALT ON ERROR
  
```

```

897 003560 100401      BMT    IS      ; HALT IF SET
898 003562 000000      HALT                    ; <SECOND RESET DID NOT CLEAR BIT6>
899 003564
900 003564 032767 002000 174630  IS:    BIT    #BIT10,SSMREG ; CHECK FOR LOOP ON TEST
901 003572 001232      BNE    TST7        ; GO TO LOOP ON TEST
902
903 ; TEST THAT RCSR BIT7 (RCVR DONE) IS CLEAR (BY RESET) AND
904 ; CAN BE READ RELIABLY.
905
906 †TST10:
907 003574
908 003574 012767 000010 174602  LP9:   MOV    #10,STESTN    ; MOVE TEST NUMBER TO MAILBOX
909 003602 000005      RESET                ; CLEAR EVERYTHING
910 003604 032777 030000 174674  BIT    #BIT7,RCSR    ; CHECK IF BIT7 CLEAR.
911 003612 001416      BEQ    IS            ; CONTINUE IF RESET
912 003614 032767 040000 174600  BIT    #BIT14,SSMREG ; CHECK FOR LOOP ON ERROR
913 003622 001367      BNE    LP9          ; GO TO LOOP ERROR
914 003624 012767 000022 174550  MOV    #22,SFATAL
915 003632 012767 000001 174540  MOV    #1,MSGTY
916 003640 005767 174556      TST    SSMREG        ; MOVE ERROR NUM TO MAILBOX
917 003646 100401      BMT    IS            ; CHECK FOR HALT ON ERROR
918 003650      HALT                    ; HALT IF SET
919 003650 032767 002000 174544  IS:    BIT    #BIT10,SSMREG ; CHECK FOR LOOP ON TEST
920 003656 001346      BNE    TST10        ; GO TO LOOP ON TEST
921
922 ; TEST THAT RCSR BIT15 (DATA SET STATUS) IS CLEAR (BY RESET)
923 ; AND CAN BE READ RELIABLY.
924
925 †TST11:
926 003660
927 003660 012767 000011 174516  LP10:  MOV    #11,STESTN    ; MOVE TEST NUMBER TO MAILBOX
928 003666 000005      RESET                ; CLEAR EVERYTHING
929 003670 032777 100000 174610  BIT    #BIT15,RCSR    ; CHECK IF BIT15 IS CLEAR
930 003676 001416      BEQ    IS            ; CONTINUE IF CLEAR
931 003700 032767 040000 174514  BIT    #BIT14,SSMREG ; CHECK FOR LOOP ON ERROR
932 003706 001367      BNE    LP10         ; GO TO LOOP ERROR
933 003710 012767 000023 174464  MOV    #23,SFATAL
934 003716 012767 000001 174454  MOV    #1,MSGTY
935 003724 005767 174472      TST    SSMREG        ; MOVE ERROR NUM TO MAILBOX
936 003730 100401      BMT    IS            ; CHECK FOR HALT ON ERROR
937 003732 000000      HALT                    ; HALT IF SET
938 003734      IS:    ; <BIT15 IN RCSR NOT CLEAR BY RESET>
939 003734 032767 002000 174460  BIT    #BIT10,SSMREG ; CHECK FOR LOOP ON TEST
940 003742 001346      BNE    TST11        ; GO TO LOOP ON TEST
    
```

;;TEST THAT LOADING TBUF CLEARS TCSR BIT7 (READY) AND THAT
 ;;BIT7 IS SET AFTER THE TRANSMITTER IS DONE.

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

```

003744
003744 012767 000012 174432
003752 032767 000001 174440
003760 001406
003762 005767 174420
003764 001403
003770 005767 174510
003774 001066

003776
003776 000005
004000 005077 174510
004004 032777 000200 174500
004012 001424

004014 005077 174474
004020 032777 000200 174464
004026 001416
004030 032767 040000 174364
004036 001357
004040 012767 000024 174334
004046 012767 000001 174324
004054 005767 174342
004060 100401
004062 000000
004064
004064 005001
971 004066 052701 010000
972 004072 005301
973 004074 001376
974 004076 032777 000200 174406
975 004104 001016
976 004106 032767 040000 174306
977 004114 001330
978 004116 012767 000025 174256
979 004124 012767 000001 174246
004132 005767 174264
004136 100401
004140 000000
004142
004142 032767 002000 174252
004150 001275
004152
004152 012767 000013 174224
004160 032767 000001 174232
004166 001406
004170 005767 174212
004174 001403
004176 005767 174302
    
```

```

;TST12:
MOV #12,STESTN ; MOVE TEST NUMBER TO MAILBOX
BIT #BIT0,SENV ; CHECK IF ON APT
BEQ NOAPT ; IF NOT
TST $PASS ; CHECK IF IN FIRST PASS
BEQ NOAPT ; OR IF NOT FIRST PASS
TST C1STFL ; IF NOT FIRST PASS - SKIP TEST
BNE TST13 ; IF CONSOLE IS UNDER TEST

NOAPT:
LP11: RESET ; INITIALIZE (SET TCSR BIT7)
CLR @TBUF ; MOVE ZEROS TO TBUFFER
BIT #BIT7,@TCSR ; CHECK IF BIT7 CLEAR.
BEQ T30CH1 ; CONTINUE IF CLEAR
; DO IT ONE MORE TIME IN CASE REFRESH BOMBED THE FIRST TEST
CLR @TBUF ; DROP READY WITH THIS CHAR.
BIT #BIT7,@TCSR ; CHECK FOR READY LOW
BEQ T30CH1 ; THIS IS A REAL ERROR IF NOT 0
BNE #BIT14,$SMREG ; CHECK FOR LOOP ON ERROR
LP11 ; GO TO LOOP ERROR
MOV #24,$FATAL
MOV #1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
TST $SMREG ; CHECK FOR HALT ON ERROR
BHI $ ; HALT IF SET
; <TCSR BIT DID NOT CLEAR AFTER DATA XMIT>

;T30CH1: CLR #R1 ; CLEAR COUNT REG.
BIS #10000,R1 ; SET COUNT
DECR: DEC #R1 ; DECREMENT COUNT
BNE DECR ; CONTINUE IF NOT 0
BIT #BIT7,@TCSR ; CHECK IF READY IS BACK ON.
BNE $ ; CONTINUE IF SET.
BIT #BIT14,$SMREG ; CHECK FOR LOOP ON ERROR
BNE LP11 ; GO TO LOOP ERROR
MOV #25,$FATAL
MOV #1,$MSGTY ; MOVE ERROR NUM TO MAILBOX
TST $SMREG ; CHECK FOR HALT ON ERROR
BHI $ ; HALT IF SET
; <TCSR BIT7 DID NOT SET AFTER XMIT>

;T30CH1: BIT #BIT10,$SMREG ; CHECK FOR LOOP ON TEST
BNE TST12 ; GO TO LOOP ON TEST

;TEST THAT THE TRANSMITTER CAN INTERRUPT AT THE CORRECT VECTOR
;TST13:
MOV #13,STESTN ; MOVE TEST NUMBER TO MAILBOX
BIT #BIT0,SENV ; CHECK IF ON APT
BEQ TST13A ; IF NOT ON APT
TST $PASS ; CHECK IF FIRST PASS
BEQ TST13A ; OR, IF NOT FIRST PASS
TST C1STFL ; IF NOT FIRST PASS - SKIP TEST
    
```

.MAIN. MACY11 27(732) 04-OCT-76 14:23 PAGE 23
 DVKREB.P11 ERROR 25 TCSR BIT7 DID NOT SET AFTER XMIT

```

996 004202 001043          BNE      TST14          ; IF CONSOLE IS UNDER TEST
997 004204          TST13A:
998 004204 042777 000100 174300 LP12: BIC      #BIT6,#TCSR      ; DISABLE XMITTER INTERRUPT
999 004212 012777 004304 174302      MOV      #T31CN1,#TVECT    ; SET VECTOR ADDRESS
1000 004220 005077 174300          CLR      #TPSW            ; SET PSW FOR INTERRUPT
1001 004224 106427 000000          MTPS     #0               ; ALLOW INTERRUPTS
1002 004230 052777 000100 174254      BIS      #BIT6,#TCSR      ; ENABLE INTERRUPTS
1003 004236 000240          NOP                      ; GIVE IT SOME TIME
1004 004240 032767 040000 174154      BIT      #BIT14,#SSMREG    ; CHECK FOR LOOP ON ERROR
1005 004246 001355          BNE      LP12            ; GO TO LOOP ERROR
1006 004250 012767 000026 174124      MOV      #26,#FATAL        ;
1007 004256 012767 000001 174114      MOV      #1,#MSGTY         ; MOVE ERROR NUM TO MAILBOX
1008 004264 005767 174132          TST      #SSMREG          ; CHECK FOR HALT ON ERROR
1009 004270 100401          BMI     #15              ; HALT IF SET
1010 004272 000000          HALT                    ; <NO INTERRUPT FROM DEVICE>
1011
1012 004274 032767 002000 174120      IS:      BIT      #BIT10,#SSMREG ; CHECK FOR LOOP ON TEST
1013 004302 001323          BNE      TST13          ; GO TO LOOP ON TEST
1014 004304 042777 000100 174200      T31CN1: BIC      #BIT6,#TCSR ; DISABLE INTERRUPT
1015
1016          ; TEST THAT TRANSMITTER DOES NOT INTERRUPT WHEN PSW DISABLES
1017
1018 004312          TST14:
1019 004312 012767 000014 174064      MOV      #14,#STESTN      ; MOVE TEST NUMBER TO MAILBOX
1020 004320 032767 000001 174072      BIT      #BIT0,#ENV        ; CHECK IF ON APT
1021 004326 001406          BEQ      TST14A          ; IF NOT ON APT
1022 004330 005767 174052          TST      #PASS            ; CHECK IF IN FIRST PASS
1023 004334 001403          BEQ      TST14A          ; BR, IF NOT FIRST PASS
1024 004336 005767 174142          TST      #CISTFL          ; IF NOT FIRST PASS - SKIP TEST
1025 004342 001041          BNE      TST15          ; IF CONSOLE IS UNDER TEST
1026
1027 004344          TST14A:
1028 004344 042777 000100 174140      LP13: BIC      #BIT6,#TCSR ; CLEAR INT ENABLE
1029 004352 106427 000200          MTPS     #200            ; DISABLE INTERRUPTS
1030 004356 012777 004402 174136      MOV      #26,#TVECT        ; VECTOR POINT TO ERROR
1031 004364 005077 174134          CLR      #TPSW            ; CLEAR PSW FOR INTERRUPT
1032 004370 052777 000100 174114      BIS      #BIT6,#TCSR      ; ENABLE INTERRUPTS
1033 004376 000240          NOP                      ; GIVE IT SOME TIME
1034 004400 000416          BR       #15              ; CONTINUE IF NO INTERRUPT.
1035
1036 004402 032767 040000 174012      25:      BIT      #BIT14,#SSMREG ; CHECK FOR LOOP ON ERROR
1037 004410 001355          BNE      LP13            ; GO TO LOOP ERROR
1038 004412 012767 000027 173762      MOV      #27,#FATAL        ;
1039 004420 012767 000001 173752      MOV      #1,#MSGTY         ; MOVE ERROR NUM TO MAILBOX
1040 004426 005767 173770          TST      #SSMREG          ; CHECK FOR HALT ON ERROR
1041 004432 100401          BMI     #15              ; HALT IF SET
1042 004434 000000          HALT                    ; <DEVICE GAVE INTERRUPT WITH PSW DISABLE>
1043
1044 004436 032767 002000 173756      IS:      BIT      #BIT10,#SSMREG ; CHECK FOR LOOP ON TEST
1045 004444 001322          BNE      TST14          ; GO TO LOOP ON TEST
1046
1047          ; TEST THAT XMITTER DOES NOT RE-INTERRUPT AFTER THE FIRST
1048          ; INTERRUPT IS SERVICED.
1049
1050 004446          TST15:
1051 004446 012767 000015 173730      MOV      #15,#STESTN      ; MOVE TEST NUMBER TO MAILBOX
1051 004454 032767 000001 173736      BIT      #BIT0,#ENV        ; CHECK IF ON APT
    
```

```

1052 004462 001406 BEQ TST15A ; IF NOT ON APT
1053 004464 005767 173716 TST SPASS ; CHECK IF IN FIRST PASS
1054 004470 001403 BEQ TST15A ; OR, IF FIRST PASS
1055 004472 005767 174006 TST CTSTFL ; IF NOT FIRST PASS - SKIP TEST
1056 004476 001066 BNE TST16 ; IF CONSOLE IS UNDER TEST
1057 004500
1058 004500 042777 000100 174004 TST15A:
1059 004506 012777 004570 174006 LP14: BIC #BIT6,PTCSR ; CLEAR INTERRUPT DISABLE
1060 004514 005077 174004 CLR #T33CN1,PTVECT ; SET UP VECTOR ADDRESS
1061 004520 106427 000000 #PTSM ; PSM FOR VECTOR
1062 004524 052777 000100 173760 HTPS #0 ; ALLOW INTERRUPTS
1063 004532 000240 BIS #BIT6,PTCSR ; ENABLE INTERRUPT
1064 004534 032767 040000 173660 NOP ; TIME
1065 004542 001356 BIT #BIT14,SSMREG ; CHECK FOR LOOP ON ERROR
1066 004544 012767 000030 173630 BNE LP14 ; GO TO LOOP ERROR
1067 004552 012767 000001 173620 MOV #30,SFATAL ; MOVE ERROR NUM TO MAILBOX
1068 004560 005767 173636 MOV #1,MSCTY ; CHECK FOR HALT ON ERROR
1069 004564 100401 BHI IS ; HALT IF SET
1070 004566 000000 HALT ; <DEVICE FAILED TO INTERRUPT>
1071 004570
1072 004570 012777 004610 173724 IS:
1073 004576 005016 T33CN1: MOV #ERR26,PTVECT ; SET UP VECTOR FOR RE-INTERRUPT
1074 004600 012767 004654 CLR (SP) ; POINT TO NEXT TEST FOR RTI
1075 004604 000002 MOV #TST16,-(SP) ; SERVICE INTERRUPT
1076 004606 000240 RTI
1077 004610
1078 004610 032767 040000 173604 ERR26: BIT #BIT14,SSMREG ; CHECK FOR LOOP ON ERROR
1079 004616 001356 BNE T33CN1 ; GO TO LOOP ERROR
1080 004620 012767 000031 173554 MOV #31,SFATAL ; MOVE ERROR NUM TO MAILBOX
1081 004626 012767 000001 173544 MOV #1,MSCTY ; CHECK FOR HALT ON ERROR
1082 004634 005767 173562 TST SSMREG ; HALT IF SET
1083 004640 100401 BHI IS ; <DEVICE RE-INTERRUPTED AFTER FIRST INTERRUPT SERVICE>
1084 004642 000000 HALT
1085 004644
1086 004644 032767 002000 173550 IS:
1087 004652 001275 BIT #BIT10,SSMREG ; CHECK FOR LOOP ON TEST
1088 004652 001275 BNE TST15 ; GO TO LOOP ON TEST
1089
1090
1091
1092
1093
1094
1095
1096
1097
1098
1099
1100
1101
1102
1103
1104
1105
1106
1107
004654
004654 012767 000016 173522 TST16:
004662 032767 000001 173530 MOV #16,STESTN ; MOVE TEST NUMBER TO MAILBOX
004670 001406 BEQ NOAPT1 ; CHECK IF ON APT
004672 005767 173510 TST SPASS ; IF NOT
004676 001403 BEQ NOAPT1 ; CHECK IF IN FIRST PASS
004700 005767 173600 TST CTSTFL ; OR, IF FIRST PASS
004704 001147 BNE TST999 ; IF NOT FIRST PASS - SKIP TEST
004706 NOAPT1: ; IF CONSOLE IS UNDER TEST
004706 032767 001000 173506 BIT #BIT9,SSMREG ; CHECK IF THIS TEST IS ENABLED
004714 001537 BEQ TST17 ; SKIP IF NOT
004716 000005 RESET ; CLEAR EVERYTHING
004720 106427 HTPS #200 ; DISABLE INTERRUPTS
004724 012709 MOV #20,R3 ; SET SYNC COUNTER
004730 005777 173554 TST #BUF ; CLEAR BUFFER
    
```

```

1109 004734 005004          CLR          R4          ; CLEAR DELAY COUNTER
1109 004736 005204          INC          R4          ; DELAY 300 MS
1110 004740 001376          BNE         INCR4
1111 004742 052777 000011 173536      BIS         #11,ARCSR    ; SET READER ENABLE
1112 004750 000240          NOP
1113 004752 000240          NOP
1114 004754 105777 173532      CHK1:      TSTB        @TCSR    ; CHECK FOR XMITTER DONE
1115 004760 100375          BPL        CHK1         ; WAIT TIL READY
1116 004762 012777 000125 173524      CHK2:      MOV         #125,@TBUF ; SEND DATA
1117 004770 105777 173512      CHK2:      TSTB        @ARCSR   ; WAIT FOR RCVR DONE
1118 004774 100375          BPL        CHK2
1119 004776 122777 000125 173504      CHKSYN:   CNPB        #125,@RBUF ; CHECK FOR SYNC BYTE
1120 005004 001426          BEQ        R3           ; CONTINUE IF FOUND
1121 005006 005303          DEC        R3           ; DECREMENT SYNC COUNTER
1122 005010 001367          BNE        CHK2         ; CONTINUE TIL SYNC UP
1123 005012 012777 000001 173466      MOV         @BIT0,ARCSR  ; TURN OFF WRAP MODULE
1124 005020 000240          NOP
1125 005022 000240          NOP
1126 005024 000005          RESET
1127 005026 032767 040000 173366      BIT         @BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
1128 005034 001330          BNE        LP15         ; GO TO LOOP ERROR
1129 005036 012767 000032 173336      MOV         #32,@FATAL
1130 005044 012767 000001 173326      MOV         #1,@MSGTY   ; MOVE ERROR NUM TO MAILBOX
1131 005052 005767 173344      TST        SSWREG      ; CHECK FOR HALT ON ERROR
1132 005060 100401          BHI        IS          ; HALT IF SET
1133 005062 000000          HALT                    ; <CANNOT GET TEST MODULE IN SYNC>
1134 005064 005001          IS:                   ;
1135 005066 012702 177777      T34CON:   CLR          R1          ; INITIALIZE DATA REG
1136 005068 005074 173414      MOV         @R1,R2      ; INITIALIZE R2
1137 005070 005777 173414      TST        @RBUF        ; FORCE BUFFER CLEAR
1138 005074 105777 173412      CHK1:      TSTB        @TCSR    ; CHECK XMITTER DONE
1139 005100 100375          BPL        CHK1         ; WAIT TIL DONE
1140 005102 010177 173406      SDATA:   MOV         @R1,@TBUF ; SEND DATA
1141 005106 105777 173374      CHK2:      TSTB        @ARCSR   ; CHECK FOR RCVR DONE
1142 005112 100375          BPL        CHK2         ; WAIT TIL DONE
1143 005114 017702 173370      MOV         @RBUF,R2    ; GET WRAPPED DATA
1144 005120 020102          CNP        R1,R2       ; CHECK DATA
1145 005122 001012          BNE        ZS
1146 005124 005201          INC        R1          ; INCREMENT DATA
1147 005126 105701          TSTB        R1          ; CHECK FOR END OF DATA
1148 005130 001364          BNE        SDATA       ; CONTINUE LOOP
1149 005132 012777 000001 173346      MOV         @BIT0,ARCSR  ; TURN OFF LOOP
1150 005140 000240          NOP
1151 005142 000240          NOP
1152 005144 000005          RESET
1153 005146 000422          BR
1154 005150          ZS:                   ;
1155 005152 032767 040000 173244      BIT         @BIT14,SSWREG ; CHECK FOR LOOP ON ERROR
1156 005156 001363          BNE        CHK2         ; GO TO LOOP ERROR
1157 005160 012767 000033 173214      MOV         #33,@FATAL
1158 005166 012767 000001 173204      MOV         #1,@MSGTY   ; MOVE ERROR NUM TO MAILBOX
1159 005174 005767 173222      TST        SSWREG      ; CHECK FOR HALT ON ERROR
1160 005200 100401          BHI        IS          ; HALT IF SET
1161 005202 000000          HALT                    ; <DATA DID NOT COMPARE ON SLU WRAP>
1162 005204          IS:                   ;
1163 005204 032767 002000 173210      BIT         @BIT10,SSWREG ; CHECK FOR LOOP ON TEST
    
```



```

1165 005212 001220          TST17: BNE      TST16          ; GO TO LOOP ON TEST
1166 005214 000005          CLR      RESET
1167 005216 005004          CLR      R4
1168 005218 005304          T35DEC: DEC     R4          ; DEC TIMER
1169 005220 001376          BNE      T35DEC
1170 005222 001376          TST999: BNE
1171 005224 005267 173160          INC     SDEVCT          ; INC DEVICE COUNTER
1172 005226 026767 173244 173152 T999A: CMP     THP2,SDEVCT      ; ALL DEVICES TESTED?
1173 005228 001404          BEQ     T999B          ; IF YES
1174 005230 005067 173240          CLR     CTSTFL         ; CLEAR CONSOLE UNDER TEST FLAG
1175 005232 000167 174360          JMP     SETUP          ; GO TEST NEXT DEVICE
1176 005234 005237 000406          T999B: INC     @BPASS      ; INCREMENT PASS COUNT
1177 005236 004767 000046          JSR     PC,MSGPRT      ; PRINT END OF PASS
1178 005238 005572          .WORD  R3
1179 005240 013700 000042          ACT:   NOV     @B42,RO   ; CHECK ACT
1180 005242 001406          BEQ     GORGIN        ; KEEP GOING
1181 005244 000005          RESET
1182 005246 004710          SENDA: JSR     PC,(RO)   ; ACT HOOKS
1183 005248 000240          NOP
1184 005250 000240          NOP
1185 005252 000240          NOP
1186 005254 005067 173102          GORGIN: CLR     SDEVCT   ; RESET DEVICE COUNT
1187 005256 022767 000001 173164 CMP     @1,THP2        ; IS THIS THE ONLY DEVICE UNDER TEST?
1188 005258 001026          BNE     RSTRT
1189 005316 012706 001000          NOV     @STKPTR,SP    ; INITIALIZE STACK POINTER
1190 005318 000167 174420          JMP     TST1          ; GO DO ANOTHER PASS
1191 005320 132767 000040 173065 MSGPRT: BITB   @40,SENVMH   ; WILL APT ALLOW PRINTING?
1192 005322 001403          BEQ     TYPE          ; YES
1193 005324 062716 000002          ADD     @2,(SP)       ; ADJUST RETURN
1194 005326 000207          RTS     PC            ; RETURN
1195 005328 011600          TYPE:  NOV     (SP),RO  ; GET MESSAGE ADDRESS
1196 005330 011000          NOV     (RO),RO       ; GET READY TO PRINT
1197 005332 105777 000030          WAIT:  TSTB   @TPS     ; CHECK IF TTY READY
1198 005334 100375          BPL     WAIT          ; IF NOT
1199 005336 112077 000020          MOVB   (RO),JTPB      ; PRINT THE CHARACTER
1200 005338 001372          BNE     WAIT          ; NEXT IF NOT DONE
1201 005340 062716 000002          ADD     @2,(SP)       ; ADJUST RETURN
1202 005342 000207          RTS     PC            ; RETURN
1203 005344 005067 173014          RSTRT: CLR     SUNIT    ; START OVER
1204 005346 000167 173576          JMP     START
1205 005402 177566          .WORD  177566
1206 005404 177564          TPS:   .WORD  177564
1207 005406 000000          ENDTST: 0
1208 005410          UINT:
1209 005412 012767 000034 172764          NOV     @34,SFATAL
1210 005414 012767 000001 172754          NOV     @1,MSGTY      ; MOVE ERROR NUM TO MAILBOX
1211 005416 000000          HALT   ; <UNEXPECTED INTERRUPT>
1212 005418          IS:
1213 005420 012737 005446 000024          PWRFL: NOV     @PWRUP,@24 ; SET POWER UP VECTOR
1214 005422 012737 000340 000026          NOV     @340,@26     ; SET POWER UP PSW
1215 005424 000000          HALT
1216 005426 000776          BR

```

```

1220
1221 005446 012737 005426 000024 PWRUP: MOV @PWRFL,0024 ;SET POWER FAIL VECTOR
1222 005454 005037 000026 CLR 0026 ;SET POWER UP PSW
1223 005460 005000 CLR RD ;CLEAR TTY WAIT TIMER
1224 005462 005200 TTYWT: INC RD ;INCREMENT TIMER
1225 005464 001376 BNE TTYWT ;WAIT FOR TTY POWER UP
1226 005466 C 4767 177634 JSR PC,MSGPRT ;TYPE "POWER" MESSAGE
1227 005472 005610 .WORD M4
1228 005474 000167 172520 JMP PRCONT ;START AT BEGINNING OF PRESENT PASS

005500 040515 047111 042504 M1: .ASCIZ .MAINDEC-11-DVKAEB-DLVII TEST.<15><12>
005506 026503 030461 042055
005514 045526 042501 041055
005522 020040 042011 053114
005530 044511 052040 051505
005536 006524 000012
005542 020040 042040 053105 M2: .ASCIZ . DEVICES UNDER TEST.<15><12>
005550 041511 051505 052440
005556 042116 051105 052040
005564 051505 006524 000012
005572 047105 020104 043117 M3: .ASCIZ .END OF PASS.<15><12>
005580 050040 051501 006523
005586 000012
005594 047520 042527 006522 M4: .ASCIZ .POWER.<15><12>
005602 000012
005610 000012
005616 000001 .END
  
```

ABASE = 176500
 ACTM1 = 000000
 ACTM2 = 000000
 ACTUOP = 000000
 ACT = 005522
 ROOM3 = 000001
 ROOM1 = 000000
 ROOM10 = 000000
 ROOM11 = 000000
 ROOM12 = 000000
 ROOM13 = 000000
 ROOM14 = 000000
 ROOM15 = 000000
 ROOM2 = 000000
 ROOM3 = 000000
 ROOM4 = 000000
 ROOM5 = 000000
 ROOM6 = 000000
 ROOM7 = 000000
 ROOM8 = 000000
 ROOM9 = 000000
 ADEVCT = 000000
 ADEVI = 000000
 ADR = 001712
 ADRTEL = 000600
 AENV = 000000
 AENV1 = 000000
 AFATAL = 000000
 AMOR1 = 000000
 AMOR2 = 000000
 AMOR3 = 000000
 AMOR4 = 000000
 AMAS1 = 000000
 AMAS2 = 000000
 AMAS3 = 000000
 AMAS4 = 000000
 AMSC0 = 000000
 AMSC1 = 000000
 AMSC2 = 000000
 AMSC3 = 000000
 AMSC4 = 000000
 AMSC5 = 000000
 AMSC6 = 000000
 AMSC7 = 000000
 AMSC8 = 000000
 AMSC9 = 000000
 AMSC0 = 000000
 AMSC1 = 000000
 AMSC2 = 000000
 AMSC3 = 000000
 AMSC4 = 000000
 AMSC5 = 000000
 AMSC6 = 000000
 AMSC7 = 000000
 AMSC8 = 000000
 AMSC9 = 000000
 APT10 = 001410
 ASHREC = 000000
 ATESTM = 000000
 ALMKT = 000000
 ALMMA = 000000
 AVECT1 = 000200
 AVECT2 = 000000
 BIT0 = 000001
 BIT1 = 000002
 BIT10 = 002000

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

410
 384
 66
 542
 418
 584
 991
 654
 682
 1020
 674
 710
 1051
 700
 720
 1095
 767
 730
 1123
 838
 731
 1149
 900
 741
 742
 938
 755
 757
 984
 775
 775
 1012
 1043

.P307	10	
.P308	10	
.P309	10	
.P310	10	
.P311	10	
.P312	10	
.P313	10	
.P314	10	
.P315	10	
.STRAP	10	3500
.STYB	10	
.STYD	10	
.STYF	10	3500
.STYH	10	
.STYJ	10	
.STYK	10	
.STYL	10	
.STYM	10	
.STYN	10	
.STYO	10	
.STYP	10	
.STYQ	10	
.STYR	10	
.STYS	10	
.STYT	10	
.STYU	10	
.STYV	10	
.STYW	10	
.STYX	10	
.STYY	10	
.STYZ	10	
.1170	10	

TSTB	1114	1117	1138	1141	1147	1198														
.ASCIZ	1114	1236	1240	1243																
.BYE	381	391	391	392	399	400	402	403	405	406										
.DOC	316	316																		
.EVEN																				
.IF																				
.IFF																				
.LIST																				
.MACRO																				
.MCALL																				
.NEXT																				
.NLIST																				
.PAGE																				
.REH																				
.REPT																				
.SMTL																				
.WORD																				

ERRORS DETECTED: 0
DEFAULT GLOBALS GENERATED: 0

#, DVKAEB. SEQ/SOL/CRF/PAGNUM/NL: TOC=SYSMAC.CO, DVKAEB.P11
RUN-TIME: 24 28 2 SECONDS
RUN-TIME RATIO: 217/56=3.8
CORE USED: 33K (65 PAGES)